2016

An Analysis and Validation of an Online Photographic Identity Exposure Evaluation System

Elliott V. Iannello

An Analysis and Validation of an Online Photographic Identity Exposure Evaluation System

Elliott V. Iannello

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Roy S. Nutter, Ph.D., Chair
Bojan Cukic, Ph.D.
Frances L. Vanscoy, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2016

Keywords: Web Crawling, Facial Recognition, Cyber Security, Social Media Privacy

Abstract

An Analysis and Validation of an Online Photographic Identity Exposure Evaluation System

Elliott V. Iannello

The rapid growth in volume over the last decade of personal photos placed online due to the advent of social media has made users highly susceptible to malicious forms of attack. A system was proposed and constructed using Open Source technologies capable of acquiring the necessary data to conduct a measurement of online photographic exposure to aid in assessing a user's digital privacy. The system's effectiveness at providing feedback on the level of exposure was tested by using a controlled set of three subjects. Each subject provided three training photos each that simulated what would be easily ascertainable from social media profiles, online professional portfolios, or public photography. The system was able to successfully biometrically identify 23 images out of 14,000 that related to one of the respective candidates. This validates the system as an automated threat and vetting tool for online photographic privacy. VeriLook 5.4 one-to-many matching grossly underperformed on the images gathered with a mere 21% at best true acceptance rate. The scoring algorithm used herein to evaluate each candidate's online photographic exposure was proven to be effective. The system developed was able to show that a candidate's assumption of their digital footprint size is not always correct. Additional testing of the scoring algorithm is recommended before a conclusion can be made with about its universal accuracy.

Acknowledgments

I would like to thank my adviser Dr. Nutter for his steadfast guidance and patience with me during the entirety of my graduate study. I am also in great appreciation of Dr. Cukic for affording me the opportunity to work on this project and for supporting me in the initial research efforts. Last but not least, Dr. Vanscoy for taking time out of her busy schedule to help revise my thesis work down the home stretch and for always providing interesting conversations.

I would also like to acknowledge the team of students spanning two Universities and two countries who helped contribute to the construction of the system detailed in this document. My colleagues from WVU Jacob Wolen, Jacob Tyo, and Domenick Poster for their substantial contributions to this project and for their continued support and friendship. Steven Samoil, Kenneth Lai, Travis Manderson, and their adviser professor Dr. Svetlana Yanushkevich from the University of Calgary for their contributions in aiding in the Biometric functionality of the system. This would not have been possible without a great team.

Last but not least, I thank my parents for their lifelong support and love and I would like to dedicate this to my Grandfather who taught me the value and true meaning of education.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation For Conducting Study

Over the last two decades, the Internet has grown into one of the most disruptive and pervasive technologies mankind has ever developed. Today it encompasses several billion different devices and spans the globe. It has truly shrunk the world – not just in terms of communication but information catalog as well. While there are innumerable benefits to this technology there are also great threats and risks – one of top being Online Identity Privacy.

With the advent of such web services as Blogging, online Shopping, and most of all online Social Media, users are creating virtual identities on an unprecedented scale – qualitatively, the majority of internet users place more than one aspect of personal identity information (PII) into some form of an online profile. Personal photos – referring to any photos that showcase a feature of the user that can be biometrically measured, in particular the face – are now considered a form of PII that are very frequently a part of an online profile. The short of it is that the volume of PII we place out on the Internet makes us highly susceptible to attacks like Phishing – which is where an online profile is hijacked an impersonated to gain leverage in finding out additional PII– Credit Card Theft, and even Total Identity Theft (where a person's entire persona is used fraudulently to achieve some sort gain; common on dating sites). There is also inherent danger to individuals in certain professions where unknown and/or uncontrolled online photographic exposure can harm the integrity of their work; but they still have a right to use beneficial online services (i.e. Amazon) safely [1].

There has been a plethora of research already conducted to help develop methods of deterring online privacy attacks. Looking at such things as attribute similarity and similarity of friend networks [2] we can develop algorithms to help identify profiles that have been phished/cloned and stolen. Advances in Cryptography have also allowed us to provide technologies to users that allow them to authenticate to sites that house a considerable amount of PII more securely – we can even provide a means of whistle blowing on dishonest profiles and even public key servers. "Crypto-Book is an extension to existing digital identity infrastructures that offers privacy-preserving, digital identities through the use of public key cryptography and ring signatures" [3]. Using the any-trust model, Crypto-Book uses existing cross-site authentication protocols to construct a private key by combining a number of public keys from Key Servers. If a user has private key constructed and a list of public keys that correspond to other existing profiles associated with the social site, they can construct a ring signature to be used as a means of privacy preservation; as a ring signature is unique in the sense that it can be verified by a third party while still maintain anonymity since it was constructed from one member of a set (you never know which member) [3]. There are many more methods of online authentication being developed specifically for preserving user identification.

The system and experiment presented throughout the rest of this document attempts to construct and validate a methodology for measuring an individual's Online Photographic Exposure – focusing solely on the biometric measurement of the face – and then prove or disprove its effectiveness. To perform this experiment a system was built to generate the data needed for comparison that utilizes two key technologies: Facial Recognition and Web Crawling/Scraping. The motivation is to provide a degree of awareness via a rating on a photo of its potential to lead to successful identification – essentially vetting a photo for use online. This metric of visibility coupled with other aforementioned technologies could provide a means of safely using popular features of OSN's (like photo "tagging"). Using the rating on a photo to place it into a risk range (i.e. low, medium, high) is another important outcome of this research. Having a clear idea of where a photo falls so a pragmatic series of steps and precautions can be assigned for each risk range will give users a level of transparency and knowledge for mitigating potential attacks on their identity. The motive is to impart data

that helps a user make an informed decision on how they are sharing their Photographic PII.

1.2 Key Terms

Web Crawling - A Web Crawler (also known as a Web Robot, Web Indexer, Spider, and Bot) is a piece of software that automates the process of scanning, or "crawling", web pages in a methodical manner in order to index them. The result is a directed structure of how different web pages are linked together. All modern day Search Engines (i.e. Google, Bing) are dependent on this type of technology to provide up-to-date data to users and keep a current topography of the Internet.

Web Scraping - Web Scraping is a technique (also referred to as Web Harvesting or Web Data Extraction) where a piece of software is written to emulate human-like online interaction with a website via a browser in order to extract information from the website. This is normally done by using some form of parser to sort through the raw data received from a server hosting the website after an HTTP request is made to it.

Boolean Search - Boolean Search is the technique of using Boolean Logic to refine web searches. The Boolean operators AND, OR, and NOT are used in combination or on their own to define relationships and limits between sets of ideas/data. For example, if one were to search for Peanut Butter AND Jelly the Search Engine would look for specific instances where both terms are relevant versus the search for Peanut Butter OR Jelly where the results could contain instances of each term relevant to itself or relevant to each other. The OR operator versus the AND would broaden the scope of the search.

Privacy Exposure - Within the context of this report, Privacy Exposure refers to the measure of how readily available and accessible an individual's personal information is on the Internet. A high-level of Privacy Exposure would therefore mean the subject has easily ascertainable Personal Identification Information (PII) online; conversely low-level Privacy Exposure would infer that details related to the individual are very difficult to find online.

Digital Footprint - A Digital Footprint (also known as cyber shadow and Internet footprint) is the trail/traces of evidence that a user leaves online when performing any action be it passive or active. An active action is the purposeful release of personal data by

means of a website or social media. A passive action is when personal data or usage data is collected without the user knowing. Every website is hosted on a server and a server logs all of the clients connecting to it – this is even a form of leaving a passive digital footprint. A good real world analogy for this digital phenomenon is the fingerprints we leave on the things we touch due to the oils in our skin. Digital footprints are not used to constitute a digital identity or online passport – rather they are a form of meta-data. Being aware of your digital footprint and the actions you are performing to create it will have an impact upon personal online privacy.

Face Detection - Face detection is a form of biometric measurement where an algorithmic process is applied to find a human face in an image. It is not concerned with determining an identity or profile to which to associate the face.

Face Recognition - Face recognition is a form of biometric measurement where an algorithmic process is applied to determine whether two or more faces are of the same individual. If a face cannot be automatically detected in a photograph, then automated recognition is often impossible.

Online Social Network - An Online Social Network (OSN) is a web-based platform that builds a social network or social relation among users who share similar interests, activities, backgrounds or real-life connections.

1.3 Problem Statement

There are two hypotheses to be tested: first, that it is possible to construct a cohesive system with Open Source and readily available technologies that can be used to move from quick and seamless photographic PII data collection into biometric analysis of that data; second, that a candidate's disclosed frequency of social media usage directly correlates with how many photos the system will be able to positively match using Open Source Face Recognition, with higher levels of disclosed OSN usage meaning a higher number of photos successfully matched.

1.4 Organization of Thesis

A literature review of relevant academic works on Web-Crawling, Social Networks, Online Privacy, and Face Detection/Recognition follow in Chapter 2 to provide an overview of the fields that influenced the development of the describe system and experiment. Chapter 3 details the design and architecture of the system implemented to act as a proof of concept and to facilitate the experiment. Chapter 4 discusses the setup and methodology of the experiment, with Chapter 5 showcasing the results. Chapter 6 is the conclusion of the study and suggests avenues of future work and research.

Chapter 2

Literature Review

There are many different aspects that comprise the system detailed herein. It was constructed by two teams over a span of two years with each team focusing on a separate discipline - those being biometrics/computer vision for performing facial recognition and web crawling for efficient and discrete ascertainment of data. Another layer to the project was incorporating an understanding of Online Social Media Networks so that the tools built to analyze and process the data captured were geared to fit certain behaviors that are prominently seen on social media websites; this hopefully gives an analyst greater and clearer context for the information the system generates.

Due to the breadth of scope, each aspect of the system was not fully explored in detail. Web Crawling, Facial Recognition, Online Privacy, and Social Media are all extremely broad and growing fields of research that are continually being explored by the academic community. With this in mind, it was of a higher precedence to get common and widely available methods for performing each necessary step to work together in a fluid application. The following sections give an overview of fields of study related to the components built to make up the system as well as topics that informed the trade-craft and operation of the system.

2.1 Web Crawlers

A Web Crawler (also known as a Web Robot, Web Indexer, Spider, and Bot) is a piece of software that automates the process of scanning, or "crawling", web pages in a methodical manner in order to index them. The result is a directed structure of how different web pages are linked together. All modern day Search Engines (i.e. Google, Bing) are dependent on this type of technology to provide up-to-date data to users and keep a current topography of the Internet. There are many different ways of going about crawling the web, most of which end up producing a biased and partial view of the web. The bias is a result of the decision making process that is built into the crawler to tell it where to go next. But the fundamental practice of finding all the possible links to the current page is universal – that is to generate a list of hyperlinks or references to other webpages and repeat this process recursively, thus indefinitely cycling links into one of three states: Visited, Unvisited, and Erroneous [4].

There are four main strategies by which unvisited links can be generally managed:

- DFS (depth-first search) – Uses a Stack for a data structure
- BFS (breadth-first search) – Uses a Queue for a data structure
- DEG (higher degree) – Uses a Priority Queue (Heap) for a data structure
- RND (random) – Picks a URL at Random from a Uniform List

DFS takes the simplest and least hardware dependent approach to solving the management of domains to crawl and is the most widely used. The most sophisticated and advanced web crawlers however run on distributed systems and are not hard pinned to one strategy – although it is usually an adapted algorithm somewhere between BFS and RND [4].

A web crawl can be represented as a directed graph of nodes and edges, creating topography of the portion of the web that the crawler is able to reach. Nodes are pages and edges links between pages (either one or multi directional). The number of links/edges being drawn to a respective node can measure a node's awareness within the structure; a high awareness means a lot of other webpages reference that particular node. What tends to happen is that a higher concentration of traffic begins to build around web pages that are the most popular

in regards to the subject matter they entail. What ends up happening, intuitively one could say, is described by [4] et al. as clustering or cores of web pages: "Cores are densely directed bipartite sub-graphs, consisting of many hub pages pointing to many authorities."

This idea is at the heart of Google's now infamous PageRank algorithm that is at the center of their very popular search engine. It is essentially calculating the degree of a web page's accessibility. The key idea is that the most frequented and trusted sites having higher accessibility. The study conducted by [4] et al. concludes that by demonstrating that PageRank distribution follows a form of the linguistic probability Zipf's Law such that:

$$log(Prob(X = d) = -\lambda log(d) \tag{2.1}$$

"If the in-degree (respectively out-degree) distribution of a graph follows a Zipf law, P rob(X = d) is the probability for a vertex to have in- (resp. out-) degree d. In other words, the number of vertices with degree d is k.d (k depends on the number of vertices n). A graph class such that the degree of almost all graphs follow a Zipf law is called scale-free because some parameters like are scale invariant. Scale-free graphs have been extensively studied. Many graphs modeling social networks, interaction between objects (proteins, peoples, neurons...) or other network properties seem to have the scale-free property. For Web crawls, a measure from Broder al. on a 200 000 000 pages crawl show that the in and out-degrees follow Zipf law. The exponents are in = 2.1 for in-degree and out = 2.72 for out-degree." [4]

Zipf's law can be defined generally as the observation that "probability of occurrence of words or other items starts high and tapers off. Thus, a few [words] occur very often while many others occur rarely." As concluded by Bennouas et al., anything that follows this law can be crawled and will show cores of clustering in the graph that can be created as the output to any crawl [4].

2.1.1 With Regards to Image Crawling

When going after images with web crawlers it becomes prudent to introduce behaviors and practices into the crawling process that are specific to this domain. Image Crawling presents a unique problem in that the keywords and links found on a webpage that drive

the crawler forward are generally text-based, as opposed to content-based (an image). Since images are the primary concern, it becomes necessary for the crawler to generate a Keyword-to-Image database (some form of mapper will do if a lighter-weight solution is necessary) to keep track of the progression/topography of the crawl [5].

It's relatively easy to pull down all images on a web-page; it's another matter entirely to do it with bias to a key term. Without the human eye, it becomes difficult to discern if the crawler is finding content that is accurately reflecting what the user wants. There are some tricks to bettering results – for instance curating images based on a certain resolution range (most of the time images below 70x70 pixels are thumbnails) and examining anchor text on images for keywords relative to initial search terms. Generally some form of ranking is required of any crawler. Employing some machine learning techniques to train a classifier to automate the process of discernment between navigational and content links helps to narrow the search space for relevant content (image other otherwise) [6]. For image crawling, it is best to measure distances between where a keyword is found and an image by counting the number of edges on the directed graph created by the crawl - this technique assumes too that images and keywords are Nodes rather than entire webpages. The greater the distance the less of an association; combining this with a scoring algorithm where a base-score is given to an image based on it's resolution, frequency of occurrence on a page, and whether or not it is hyperlinked, it becomes possible to have an image crawler that is highly efficient running on hardware found in most personal computers. As Fujimoto et al. demonstrates, they were able to employ these techniques on a machine with a 2.4GHz Intel Pentium 4 processor, 512MB of RAM, and 4.28Mbps of bandwidth to achieve the following results: on average 349 images of 2205 were chosen out of 192 web pages crawled in 33.38 seconds [5].

2.2 Overview of Facial Identification Technologies

An extensive and rapidly growing field in its own right, facial identification is a technology that is by definition a sub-genre of the Computer Vision field. Although there are many different Open Source and proprietary algorithms that have been developed, each offering up their own pros and cons, the algorithmic process of performing the task of taking some form of visual data representing a human face (image or video) and providing identification and validation of that data can be broken down into three fundamental steps: Facial Detection, Feature Extraction, and Facial Recognition [7].

Within the context of this work the definitions for Facial Detection and Recognition are provided as key terms in Chapter 1. Feature Extraction however acts as a bridge and can be best described by the popularly established taxonomy of Zhao et al, defines three types of algorithmic process to the overall endeavor of facial recognition. They are feature extraction via "generic methods based on lines, curves, and edge, feature-template-based methods that are used to detect facial features such as eyes, and structural matching methods that take into consideration geometrical constraints on the features" [8].

As far as Open Source technologies go - or any in the field for that matter - OpenCV has arguably been the most popular option given its been estimated to have over 7 million downloads and has been adopted for use by Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota. The community of users is around 47 thousand which is pertinent since it indicates there is a lot of support and documentation around the product. Most major programming languages are supported by their libraries meaning integration into pre-existing or individually developed platforms should be easy [9].

Facial detection libraries in OpenCV utilize the widely accepted Haar feature-based cascade classifiers developed in 2001 by Paul Viola and Michael Jones. Another name for this algorithm for performing object detection is Viola/Jones. The idea is that once a classifier is trained on a large set of data, it will be able repetitiously detect similar objects in an image that have a unified set of identified features with those defined during the training process. Viola/Jones initial findings yielded a 95% accuracy rate of detection with just 200 features to train off. Larger training sets gave subsequently higher accuracy [10]. For performing

facial recognition, OpenCV offers three algorithms out of the box that have been widely adopted and varied upon within the community: Eigenfaces - Principal Component Analysis, Fisherfaces - Linear Discriminant Analysis, and Local Binary Pattern Histograms (LBP). Eigenfaces and Fisherfaces are more comprehensive in approach - image data is treated as a high-dimensional vector. This can bring about issues when there is a lot of external variation produced by an external source within the image data being processed (this is worse with Eigenfaces as Fisherfaces provides measures for dealing with some variance). LBP takes an opposite approach by basically focusing on local areas of pixels within an image and comparing them against their neighbors. This switches the emphasis of constraint from having great posture of the subject, lighting, and obstruction of the face to the resolution of the image - which are all potential hindrances in their own right to any facial identification algorithm. [11].

Any given way, OpenCV offers the most robust package for the free price point with the best community for supporting it. There really isn't much else out there that remotely comes close to its scope. SimpleCV is another popular framework but it contains OpenCV. PyVision and VXL are computer vision library options that are ligher weight for Python and C++ respectively, but each of these languages are encompassed in OpenCV with no compromise to the algorithms available for use.

It is a lot more difficult to provide an overview of the proprietary options for Facial Identification suites due to the cost of purchasing licenses and the intellectual property protection of the algorithms they use. However, VeriLook by NeuroTechnology is an affordable option (339 Euros or 361.91 USD) with a very high adoption rate (over one million algorithm deployments worldwide), customer support, and accolades. Although the precise algorithms it uses are undisclosed to the public, it is touted as being able to perform one-to-one (verification) and one-to-many (identification) face matching with relatively good tolerance to posture of the face within compared images. This is more ideal perhaps than some of the Open Source options that require a high volume of photographic data for training the algorithms to be done before any reliable results can be produced [12].

2.3 Overview of Threats to Online Data Privacy

The advent of the prolific use of the Internet in the mid 1990's through the present day has truly changed the way we do everything. It has redefined our commerce all the way to how we socialize. It could easily be said to be the most disruptive technology man has ever created. Along with its many gifts, the Internet has brought many problems with it as well and none more immediately pressing to users than their online privacy.

There are several key types of data that are the driving force behind the most popularly exploited and maliciously attacked online services. Each of these distinct genres of data however can are linked in that the compromise of one can cascade into the compromise of the rest.

First is geo-location data. It is now common place to see the option to check-in to locations on OSN's upon arrival and this gets recorded and potentially displayed (depending on the user's account settings) on the user's profile. Exercise apps and devices have this type of data as an integral component as well (i.e. FitBit). This geo-location data in the wrong hands can be used to exploit patterns found in a user's behavior or demographic of users. A few scenarios include staging robbery while a user is out on their morning run or at the gym, or if the malicious intent is predatory in nature an abuser of this data can use it to formulate times when an user is alone or susceptible [13].

The second type of data is financial data. White-collar crime has been a growing problem now that the majority of markets from around the world are globally linked. Threats present themselves on a global scale now and the common practice of online shopping popularized by such websites as Amazon and E-bay have expanded this threat to the everyday consumer. Part of the problem is the model in which these exchanges are made - merchant-centric is not conducive to limiting the exposure of the user. Moving to adapt e-commerce technology toward a model that is peer to peer in nature and limits the unnecessary exposure of data sharing to only trusted peers would begin to help remedy the rampant amount of identity theft and man-in-middle style attacks on the financial data of the everyday consumer [14].

The third and final type of data is Personal Identification Information (PII). This is any information - biometric or textual - that discloses aspects of an individual user's identity.

This type of data is what is compromised when identity theft occurs. While user's are being more cautious with respect to keeping PII that is in a textual form private, photographic PII has become a greater problem given advancements in the field of computer vision. Most major OSN's employ the use of some kind of feature detection and facial recognition technology to provide "tagging" features to photos posted as media on the OSN. Although seemingly benign in initial intent, biometric technology in OSN's has opened up new avenues for conducting socially engineered attacks on user accounts (i.e. phishing), the ability for malicious users to quickly cross-reference non -identified users with identified users to figure out who you are through process of elimination, and in the case of marketing where and how your likeness is being used to tailor information to an assumed demographic. Potentially using this data to covertly profile for insurance risks, criminal activity, and/or access or denial to product all can be done with OSN biometric data [13].

In addition to just following the data to identify risks, there is also the topography of the web itself with special consideration given to OSN's. Websites such as StackOverflow.com construct relational graphs from the interactions between users who supply questions and answers. Any site that has a heavy emphasis on content creation through interaction runs the risk of exposure through topography as the popular posts, threads, and answers - all linked to a user - have a greater number of neighbors within the graph structure. This naturally creates a high awareness of certain nodes and thus exposure. The "Like" or "Favorite" feature that is a staple of most OSN posts contributes to this same phenomenon [15].

An insipid threat, one philosophical in nature, is perhaps the most dangerous of all - reduction of identification to a digital footprint. "If people's identity must be invariably verified by the biometric scars to be valid and accepted - if we aren't ourselves without mechanical confirmation that our codified bodies match some previously recorded information - then repeated validation of our bodies might become an increasingly important activity in social life." Increase the influence and dependency on the technology and society inadvertently dehumanizes the user behind the screen by reduction to mere data. Remedies to the threats imposing on online privacy have to encompass the legal, technological, and business policies governing use of online technologies, but above all else our most immediate and apparent weapon is prudent restraint [13].

2.4 Online Social Media Networks

Online Social Networks use directed structures to form a number of links between users. For example, Facebook suggests "people you may know" by looking for friends of friends. It's a simple algorithm – odds are good I know my neighbors and the neighbors of my neighbors. In the case of websites like StackOverflow, a directed social network is constructed a means to perform search based queries with experts (computer geeks) being used to create the answers. People are now the database – coining the phrase "crowd sourcing" that is synonymous with the paradigm. But as with most anything there are risks; in the case of Social Search what we see is a node in the network having a wide reaching level of awareness if it is an active and accurate source. Experts who are frequent suppliers of highly praised data get weighted (by a user vote) and are now passed along chains of acquaintances giving them a distributed awareness with respect to certain subject matter. If uninhibited, what will occur is a directed graph where one node is a neighbor with all other nodes – a single node to "rule them all" if you will. This poses a major privacy threat to that individual and all other who seem to become local maxima within the directed structure [2].

The same outcome could happen on social sites that encourage photo sharing and tagging. A highly photogenic and outgoing individual could become tagged in potentially thousands of photographs and gain a high level of distributed awareness. The solution is to devise a privacy model that limits awareness – finding an optimal awareness distribution so that neighbors only get information on an "as needed" basis and information that user is allowing them to have. This solution has been shown to yield a Nash Equilibrium [2].

Wang et al. provided a method for trying to measure the awareness of a user's profile within an OSN. This proposed Privacy Index is a numerical value from 100-0 where 100 is full exposure. The Privacy Index is mathematically defined as:

$$P_{idx}(i,j) = 100 \times \frac{\omega(i,j))}{\omega(j)} \tag{2.2}$$

where $\omega(i,j)$ represents the privacy weight of user j's visible attributes to user i. The normalized score shows the relational awareness between two users by examining factors with varying weight to exposure such as how many pieces of PII are listed on a user's profile,

their privacy settings, and privacy impact factors inherent of the OSN [16].

Attention has rarely been given to how data is controlled and modeled for disclosure within a social network as the favored approach has always placed an emphasis on the propagation and composition privacy preservation. It has always been more about building a mote around the castle than making the castle itself secure. But rather than vet the account, why not vet what's inside it? The User. As the adage goes: locks are there only to keep honest people honest. By analyzing two key factors an evaluation can be formulated:

1. Peer to Peer Interaction and Recommendation
2. Time/History

It then becomes possible to aggregate a score that places a weighted value of trust on a user in a social networking system [17].

The idea is to look at the frequency of interactions between two users, look at the type of interactions, and then allow them a way to express a vote of confidence in one another. This could be done directly (i.e. StackOverflow) or indirectly/subliminally (i.e. Facebook likes). This is a context-based way of empirically representing trust. Once an aggregate score has been determined, a categorization is assigned based on ranges [3]. A social networking site can use these categories behind the scenes to suggest friends that are trustworthy, or answers, or even publically disclose them allowing for transparency to bring about a level of healthy ethic of social astigmatism – aka no body wants to be the "untrusted node" in the system. It makes users accountable for their behavior [17].

In short, Online Social Sites need to implement modern day data mining techniques to build a level of intelligence that allows a system to vet a node's trust while simultaneously giving a node/user control over distributed awareness within the directed structure. Even more simply: switch the paradigm to focus on inner stability vs. outer shell – not harder locks but smarter locks.

2.4.1 The Relevance of Facebook

Facebook is the largest Online Social Network with 1.18 billion active users as of August 2015. As of Q3 2015, Facebook had a market cap of 300 billion dollars. Its size combined

with its global reach and impact make it a prime target for individuals looking to exploit its user in a number of ways using a number of different techniques. Facebook's counter measures to phishing style attacks on user account and privacy policy are important due to its dominance of the social media market – they have the resources and man-power to trend set.

A prominent feature of Facebook is photo sharing. The company stores over 300 petabytes of user data, the vast majority being photos. Additionally, 91 percent of the Millennials (15-34 years of age) are active Facebook Users. This has become a biometric goldmine, making their facial detection and recognition software used in photo tagging one of the best in the business due to sheer training data volume. Facebook also invests resources into facial detection and recognition research and development. According to a 2014 article published by ExtremeTech.com, DeepFace (Facebook's recognition algorithm) can view two irrespective photos in terms of angle or lighting and deliver 97.25 percent accuracy of recognizing the same face in both respective photos [18].

This level of performance has spawned privacy issues that effect both general users of the website and special cases of usage for law enforcement and security/intelligence operations. Furthermore, the technology could seep into other industries as well, if Facebook begins selling their tech or data to marketing agencies that wish to aggregate data to predict individual consumer shopping habits for the creation of custom and tailored advertisements [13].

Since its inception in 2004, Facebook has shifted the nature of our digital footprint and online interactions and it is still influencing how the Internet is growing and developing greatly. Nothing is more controversial about the social media giant than its privacy policies. Over the years there have been numerous cases and court hearings in regards to Facebook's aqueous user-privacy policy/agreement. At the epicenter of the issue is its skew in business plan to serve up large amounts of personal user data to advertising agencies for data mining efforts to predict and place customized ads. It all happens behind the scenes every time you click a button on their website – creating a slow drip of a new type of finger print: preference. With the reason discussion to lift the "Safe Harbor" policy, Facebook could see yet another law suit – this time from EU – due to its behind the scenes data collection.

Facebook in short is too large to ignore; a leading influence in personal privacy for better or worse, and controls an absurd amount of photographic data that they have used to hone a biometric algorithm for picking humans out of photos like a needle from a hay bail. They are the leader of the Social Media Waltz, and right now we have no choice but to follow their rhythm [18].

Chapter 3

System Architecture and Design

The following details the requirements, design, and tools for implementation as well as serves as a guide for use of the UCAN Assessment System. This system provides a unified tool-set that fluidly facilitates the process of data acquisition of a search candidate, constructing an online domain of interest for the candidate, crawling the domain for images of the candidate, performing facial recognition and detection on the body of images found, and finally giving users of the system tools for analyzing, storing, modifying images of the candidate to yield a report of exposure and mitigation strategy.

This system was the result of several student contributors developing it over the course of several years. I owe a debt of gratitude to Jacob Wolen, Jacob Tyo, and Domenick Poster for their work in making this system a reality - it's cohesiveness and completeness would not have been possible without them.

3.1 Requirements Specification

The UCAN Assessment Systems is a suite of applications designed to act as a vetting tool for Online Photographic Exposure. It is comprised of six main operations: Automated Search, Image Web-Crawling, Cloud-Based Data Storage, Management of Data Stored, Facial Detection/Recognition, and Image Modification Tools. UCAN is designed to provide a unified environment for all of these components to better provide quick, stealthy, and secure feedback and mitigation strategies to users. A modular design allows each part to be

updated and interchanged as needed to stay current with modern web technologies. This is achieved by decoupling the graphical user interface from the scripts and executable files that run the underlying processes necessary to achieve each piece of functionality. It was also a requirement to choose technology that would allow the system run efficiently across popular modern operating systems (Windows, OS-X, Linux).

There are three areas where different pieces of the overall UCAN system operate. They are the local machine the application is running on, the Internet, and a private secure web server, which houses the Cloud Database. Any personal data entered into the Boolean Search Form is stored on a secure private database housed by the project server. Any information that is entered to be searched does get entered into public search engines. Images retrieved by the scraper (SmallScrape.py) and the web crawler are stored on the local machine in addition to being uploaded to the Cloud Database. Local log files (.txt file type) are also kept and disclose the specific searches and results run on that particular machine. User accounts help to provide persistent monitoring and privileges of use across multiple machines as well as give each user access secure access to the system. Cloud processes are monitored separately by services setup on the server upon which they run.

3.1.1 Scope and Constraints

The UCAN Assessment System was created in principle as an experiment with the hypothesis being that it was possible to create a cohesive system to perform the automated process of gathering online image data for facial recognition software to perform identification on a candidate using only readily available open source technologies. This approach confined the scope of the project to only creating a system that operated using the most popular and widely available open source frameworks and techniques at a very fundamental level of implementation - this honed the scope to general practices for performing each necessary component of the overall assessment process. In depth analysis and comparison of more advanced techniques used to perform web crawling, facial detection and recognition, cloud storage, and interface design are beyond the scope of this study.

The primary constraint on the system is born out of the focus scoped - there is not much

room for depth of exploration of each sub-system due to a need to form a cohesive product in a reasonable amount of time so results can be produced. This is compounded by the fact the study is concerned with Open Source/readily available technologies only - the idea being to show it is relatively easy to gain access to and setup these technologies and get them into a unified environment. Compatibility issues arising from the interactions of subsystems as well as getting each subsystem to run properly and consistently on each operating system (Windows, OS-X, Linux) was also a limiting factor when it came to choosing development languages, tools, frameworks, and dependency libraries.

3.1.2 Development Solutions

Java was chosen as the primary development language due the fact that the Java Virtual Machine allows the majority of the software developed to be agnostic to the operating system upon which it runs. Java is a very well supported language with many native libraries that support the development of a graphical user interface. Crawler4J is an Open Source Java web crawling project that can be found on GitHub. It was an easy fit for the study due to its popularity, easy integration - being written in Java, and it is an actively contributed to open source project.

Python was chosen as the language of choice for developing the systems that pertain to the field of computer vision and OpenCV was chosen as the Open Source framework for providing the necessary utility to perform facial detection, recognition, and modification. It is an interpreted language and hence no compiler is needed to run programs - this may be slower at run time but less installation and setup configuration is required. Python also works really well with OpenCV and BeautifulSoup - a dependency designed to facilitate web scraping and analytics of the data ascertained. Lastly, it has a concise and expressive nature which helps in reducing development time on more complex aspects of systems.

3.2 Features

The following section is a break-down and description of all the core features included in the UCAN Assessment System. They are disclosed by module and each module is listed in successive order of intended use if processing a search candidate for the first time.

3.2.1 User Accounts

User accounts are an integral aspect of keeping the system secure and data persistent across platforms. Creation of an account was kept simple and no customization options are stored with association to a user. The highlights of this feature are as follows:

- Admin Level Users
  - Have access to full feature set
  - Access to Facial Recognition, Database, and OpenCV Settings
  - Can create users
- User Associated Logging and Access
  - Secure access via password
  - Secure storage of password in database
  - Local and server logs have associated users to actions performed for cross reference of activity

3.2.2 Search Request Form

The Search Request Form (SRF) is used to quickly catalog a search candidate's personal information so that it can be used to construct a boolean search term to be sent to an Internet search engine's API so that it may execute a search on said term. The results are logged, as well as the profile information entered and each search term created with respect to the results they generated. This feature is generally used first by a user to create a list of potential domains to crawl to find data online associated to the search candidate. The highlights of this feature are as follows:

- Intuitive User Interface
  - Tabbed Data Entry for Searches makes for organized and quick manual Data input
- Cloud-Based Storage
  - Populated fields are uploaded to Secure Database
  - Creates a unique and persistent ID associated to each Person Profile for reuse and consistent data accumulation on an Individual
  - Automatically creates associations to Images associated with the Profile
  - Able to handle multiple instances of client side operations across multiple machines
  - Easily Scalable
- Search Request Form File Parsing
  - Will Load in Data automatically from the Search Request Form specific to the data aggregation of candidates for to be run through this system
  - The form can only be of the file type .doc
- Boolean Search Construction
  - Easy to use graphical user interface for quick construction of Boolean Searches (commonly seen under any Internet Search Engine's advanced options)
  - Ensures proper syntax for passing the search to a Search Engine
  - Logs all searches run for ease of reuse
  - Free text entry area still allows users the dynamic option of manual entry
- Automated Search String Permutations
  - Will automatically run every permutation of a submitted Boolean Search String
  - User can toggle on or off as needed

- Java Client
  - Will automatically run every permutation of a submitted Boolean Search String
  - User can toggle on or off as needed

3.2.3 Scraper (SmallScrape.py)

A low impact and subversive web scraper written in Python, aptly named SmallScrape.py, is run upon a user requesting a search. The term generated by the Search Request Form is used by SmallScrape to scrape images off a results page of an image search run by an online search engine. Each image scraped is stored as a result in the database for later assessment and the respective source URL of each image is added to the list of domains for the web crawler to use as seeds. This also is a technique the utilizes the already expansive capabilities and web comprehension behind modern day search engines, such as Google which is what SmallScrape.py scrapes. Search engines are web crawlers, so SmallScrape.py offers a quick way up front to locate domains and images that are likely relative to the candidate. Highlights of this feature are as follows:

- User-Agent Spoofing
  - Appears to a search engine to be a modern day browser making a GET HTTP request
- Data Slurping
  - Scrapes only 100 images at a time before a delay is triggered to reduce the amount of load placed on an online search engine's servers.
- Decoupled from System
  - Can be easily edited,added to, or even replaced based on current necessity (as long as file name remains the same)
- Optimized for Speed

  - Uses as few external libraries as possible
  - Avoids any unnecessary iterative or recursive processes
  - Runs on separate threads from Java Client

- Local Storage
  - Files are stored in a local designated directory and then are uploaded separately

3.2.4 Web Crawler

The web crawler provides the bulk of the image data to be processed for evaluation of a candidate's online exposure/digital footprint. It can be run indefinitely if given a robust set of initial seeds - also known as domains of interests - to crawl over and expand from. Even with the multi-threaded approach taken in its design, crawls can be slow - this is due to no discrimination of images downloaded from a web-page and a forced delay on each HTTP request performed, crawls can take 24-48 hours before yielding substantial results. Everything is stored in the Cloud Database as well as locally, and the crawler will still run even if it cannot connect to the cloud storage. It was written in Java and packaged as separate module to be used independently of graphical user interface. Although the Java Client of the UCAN Assessment System provides a graphical interface for controlling the crawler, a command line interface is still provided if users run the crawler without the GUI. The crawler follows each Robots Exclusion Standards for all URL's encountered that are allowed to be crawled. The highlights of this feature are as follows:

- Exhaustive and Aggregate Crawling
  - Will continue to add domains to be crawled as new pages are found (dependent on the robustness of initial seeds)
  - Will dynamically add root domain of a website to ensure an entire website is crawled
  - No Discrimination in selection of Image Files Downloaded, all images on a seeded web-page are downloaded

  - Exhausts a domain by iteratively truncating the URL down to the root or index page

- User-Agent Spoofing
  - Appears to a search engine to be a modern day browser making a GET HTTP request
  - Randomly picks from a predetermined, hard-coded, list of User-Agent tokens of Firefox and Chrome browser versions

- HTTP Request Delay
  - Randomly chooses a delay between 1-7 seconds to look more like authenticate user behavior
  - Each HTTP Request has a delay to prevent the crawler from placing too heavy a load on a server
  - Prevents the crawler from looking like a DDoS attack

- Multiple Threads
  - Number of threads can be easily increased or decreased by users (default is four)
  - Each thread is an instance of the crawler
  - Seeded domains are evenly distributed amongst threads

- Decoupled from System
  - Separate Java executable (.jar) file called as an imported library from Java Client
  - Command Line Interface provided if run without GUI

- Local Storage
  - Users can specify a location for local files to be stored, a default folder is created otherwise
  - Can be disabled entirely

- Built on Open Source Technology
  - Crawler4J (open source project available on GitHub) serves as base and was modified

3.2.5 Cloud Database

The advantages of using a cloud based solution for our data storage vastly outweighs any cons. Having all the images and their associated meta-data aggregate in one place allows the system to maintain a holistic and persistent profile of candidates across multiple web crawlers. It also provides a more controlled and secure environment to store data - having only a single point that needs secured inherently makes it easier to provide robust security of data. This reduces the administrative burden and overall cost of storing and sustaining the data generated by the system. Even though the UCAN Assessment System's web crawler module creates a local copy of images to be stored in a local file directory upon the machine the client it running, the cloud database still provides the best way to ensure the long-term integrity of the image data collected. This allowed for the creation of a simple relational model to permanently profile a candidate and all the information necessary data needed to perform analysis on them. Images needed to train the facial recognition software, their textual personal information used to perform boolean searches, and the subsequent results are all pieced together to form the candidate's profile. The highlights of this feature are as follows:

- Hosted on Private Secure Server
- Image Files stored as Byte Arrays
- PostgreSQL Database
- Three Tiered Security Model
  - Cloud Server Firewall
  - Operating System - IP Address Discrimination and Non-conventional Ports for SSH and SCP

  - Separate Authentication required to access Database

- Data integrity preserved through regular backups of database
- Scalable Resources
- No load placed on client machine, utilizes server
- Allows for running crawlers across multiple machines and still having results all in one place
- Allows for persistent candidate profiling over time

3.2.6 Face Recognition and Detection

The system has an entire module dedicated to the biometric practice of facial detection and recognition. It allows users to easily provide photos for training, run detection and recognition on a selected candidate, and see ranked results through a graphical user interface that is once again designed to be agnostic to the technology running underneath it to perform the necessary biometric operations. OpenCV is used as the default framework to implement fundamental algorithms of facial detection (feature extraction) and facial recognition (identification). The module uses Haar Feature-based Cascade Classifiers for the detection process and Eigenfaces for recognition. The user provides photos for training via the Search Request Form and those photos are held in the database and are a part of the candidate's profile. The image results gathered and stored by the scraper and web crawler modules are the search/test spaces. Results are presented to the user by rank - going from the highest confidence rating given by the algorithm to the lowest. The highlights of this feature are as follows:

- Full Automation of training, detection, and recognition
- Uses candidate's profile images to train on for facial recognition
- Manual override of automation

  - Users can enter paths to local directories of images for training and testing spaces respectively

- Decoupled from System
  - OpenCV libraries used, but can be swapped out for other biometric solutions for facial detection and recognition

- Results Table
  - Provides clear and easy viewing of results
  - Can be ordered by column in ascending or descending order
  - Scores are highlighted by color (green, yellow, red) to provide easy visual representation of confidence in a respective image

- One button access to File Viewer

3.2.7 Image Modification

This feature provides a tool to use as a potential mitigation strategy in a scenario where an image is found by the crawler that is both rated high in confidence by facial recognition and determined to be on an undesirable or high risk website by a user of the system. Rather than be removed from the website (if possible), it can be modified slightly and replaced so that any other instances of facial recognition - be it for malicious purposes or otherwise - will not provide as high a confidence rating on the image. The tool provided allows users to alter two key parts of the face - the shape of the nose and width between the eyes - and then save a new image file with the modifications reflected. An easy to use graphical interface is provided to users allowing images to be modified simply by moving a series of sliders. The underlying algorithms for performing the facial feature modification are Python scripts developed by Wolen [19] and Poster [7] being called from the Java Client providing the interface. The highlights of this feature are as follows:

- Preview of Modifications on Example Image

- Decoupled from System
- Slider or Manual Entry of Adjustment Percentages
- One Button Access to Image Downloader tool for easily pulling down images desired for modification

3.2.8 Image Downloader

This feature is a tool for providing easy and secure access to downloading image files from the cloud storage database. It provides several options for doing this and allows users to choose between downloading from the images retrieved by the scraper, web crawler, or both. The highlights of this feature are as follows:

- Image Download by Numerical Range in Database Table
- Image Download by Name (for specific image retrieval)
- Image Download of Crawler and Scraper Results by Candidate Profile
- Image Download of Candidate Profile Training Images

3.2.9 Log File and Image File Viewing

This feature is a tool for viewing files, both images and system logs, without having to leave the system or search file directories via a local machines operating system. This is merely a convenience tool to help expedite the analysis process. The highlights of this feature are as follows:

- Can open and list files in a local directory on Client's machine
- Previous and Next File buttons for easy browsing of directory files
- One Button access to viewing an image on its source website
- One Button access to the Image Modification tool

3.3 System Architecture

Considering that one of the major initial constraints of the study was having to construct a system comprised of many large parts that could each be explored in depth in their own right, it became immediately evident that the design of the system should favor modularity as much as possible. This "Lego Block" approach provided a number of advantages beyond meeting the defined constraints and scope - it provided a natural way to break up development and testing and allowed for easy identification of areas where additional automation may be necessary. Additionally, it allowed for separate pieces to be improved by either iteration or replacement as better ideas and algorithms were formulated. Figure 3.1 is a diagram of the overall system and gives a general sense of how information flows through the system.

Each yellow block represents a self contained application/sub-system that can be used independently of the system as a whole. Each grey circle represents a data store. Although data can be stored both locally and on the cloud, this feature can be toggled on our off and the diagram represents the data store that provides the most long-term integrity of the information, which is the intended place for it to be contained. Green diamonds represent information that must be provided by a user of the system. The Search Request Form (SRF), which can be automatically uploaded and populated into the system or manually entered, provides PII information on a candidate and the Default Sites represent a list of website URL's that users provide - they are generally websites with a high volume of images or they could be contextually specific to a candidate. Applying mitigation strategies are optional and are selective to only images that are deemed to have a high exposure risk by users; for this reason they are recommended to be applied in a secure environment.

All of the components that make up the system are easily managed and accessed through a Graphical User Interface Client Application that serves as the unifying agent. The data flow of the system and the order in which sub-systems need to be used to complete a candidate's assessment informed the layout and organization of the client's interface. This is shown in more detail in Section 3.4.

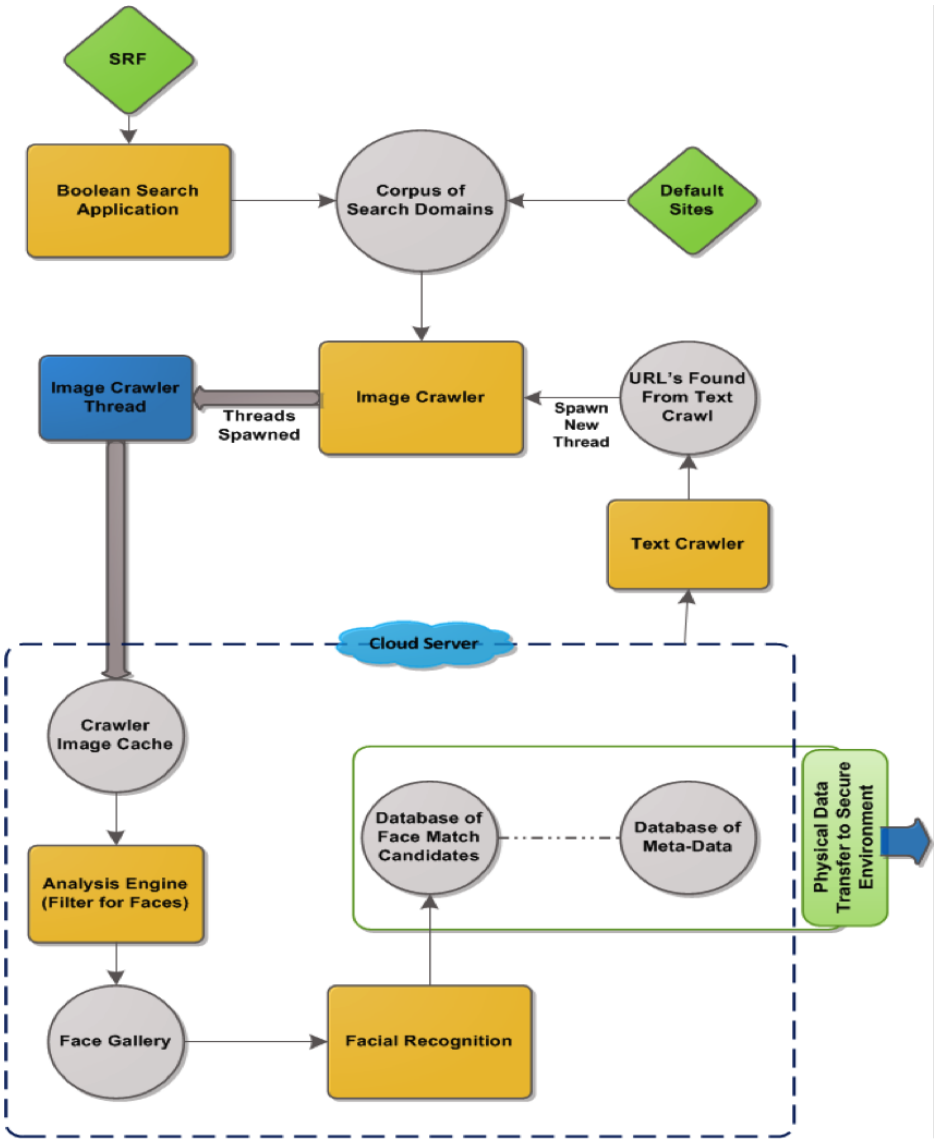

Figure 3.1: Overall System Architecture Diagram

The upper half of the system, outside of the Cloud Server bounding box, is interacted with via the Internet Search module accessible from the applications main menu. The items encapsulated in the Cloud Server bounding box are interacted with through the Facial Recognition module. Options to configure and manage more granular aspects of each component are available via the general Settings menu available from the Main Menu.

3.3.1 Data Flow

The following Figure 3.2 showcases the entire process in sequence of acquiring the necessary PII from a candidate (this includes the profile photos for training the facial recognition algorithm) to using it to generate a corpus of domains (URL's of websites) to serve as seeds for a web crawl to storing the information found and finally to analyzing it for an exposure evaluation. The data in the first four steps which comprise the top of the diagram is primarily textual and represents the space on the Internet upon which the candidate has a potential digital footprint. The bottom row's data is primarily photographic and constitutes affirmation of a digital footprint. This nine step process effectively encompasses the use of every aspect that makes up the system.

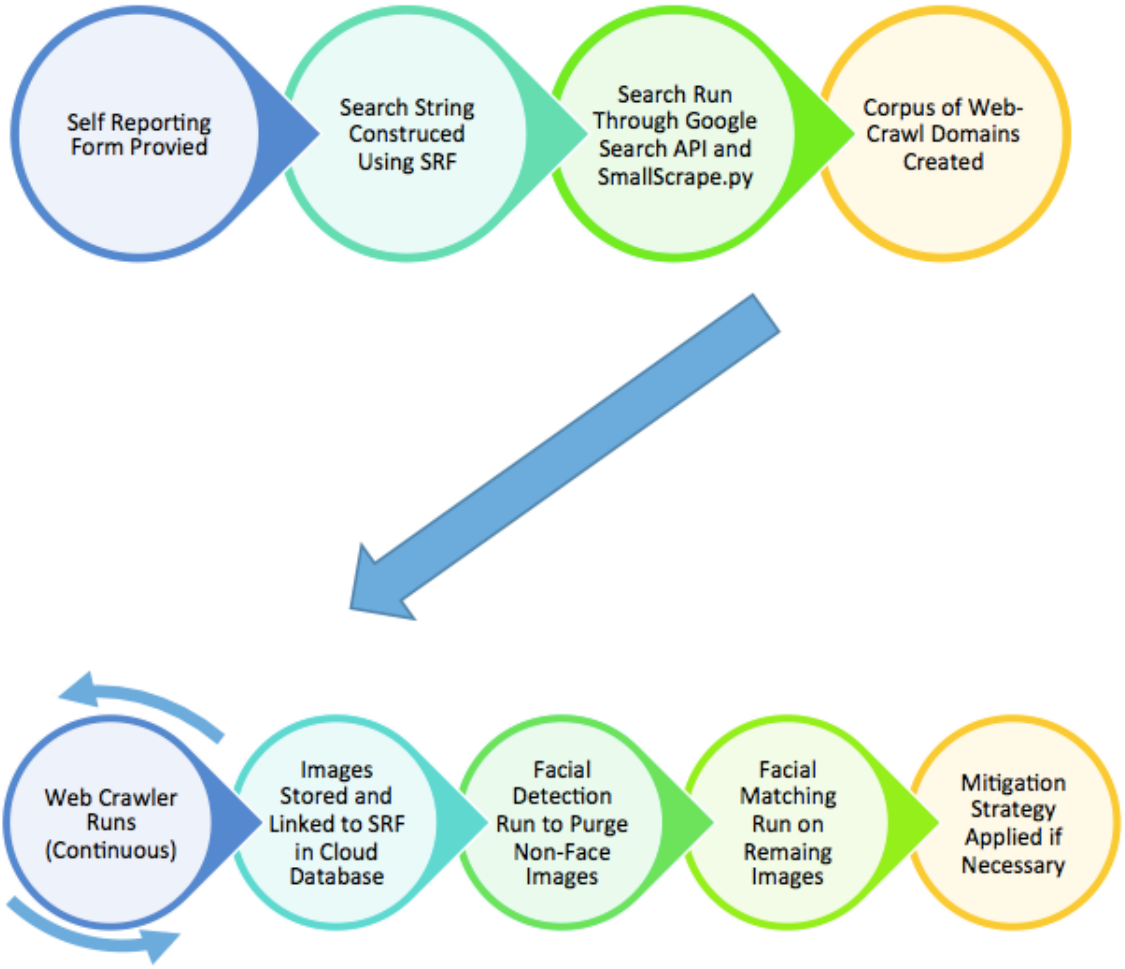

Figure 3.2: Overall System Data Flow Diagram

Search Domain Formulation Data Flow

Figure 3.3 is a more granular view of the top half (first four steps) of Figure 3.2 which details the progression of how users take a candidate's profile data and have the system create a body of domains to be used as seeds for the web crawler. This all starts with the Search Request Form (SRF) being filled out by a candidate and then a user uploading, or manually entering, the contents of the form into the appropriate fields to construct a boolean search. Once this is done, any number of search engine API's can be sent the search term and the resulting list of website URL's back from each respective API are then combined with any website URL's entered manually by the user (these are normally websites that have a large amount of aggregated photographic data or they could have been indicated as frequently used by a candidate).

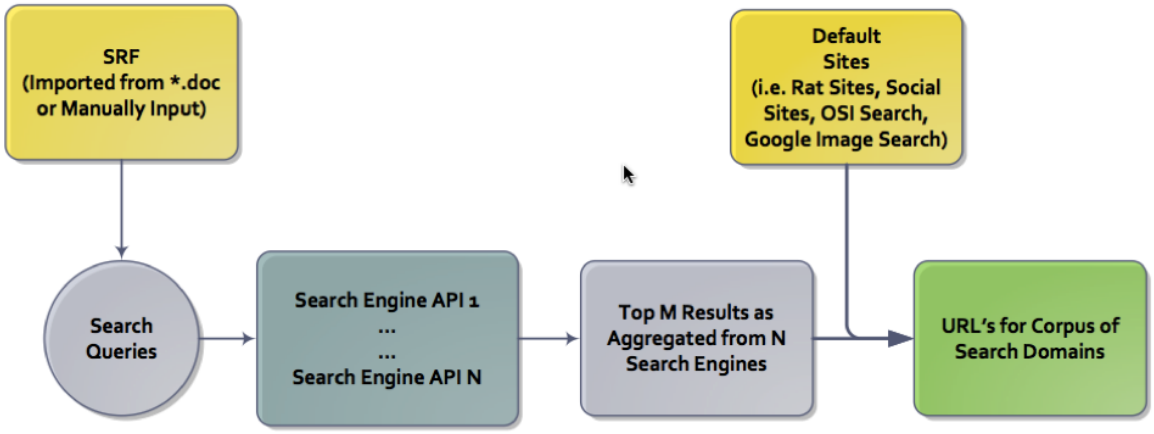

Figure 3.3: Search Domain Formulation Data Flow Diagram

Web Crawling and Candidate Identification Data Flow

Figure 3.4 is a more granular view of the bottom half (last 5 steps) of Figure 3.2 which details the progression of how users take the corpus of search domains, perform a web crawl, and analyze the resulting image data. It is dependent upon the data generated by the operations shown in Figure 3.3 but the process it details can be run asymmetrically and independently of what occurs in the previous figure - manual entry of a corpus of search domains or use of preexisting candidate profile data can be used to skip the sequence of

actions seen in Figure 3.3. Please make note that this is an iterative process being directed and monitored by a user. If an image is deemed to be in need of modification, the user can pull down the respective image to a secure environment to use the Image Modification tools available through the UCAN Assessment System. The crawler and facial recognition subsystems need no be halted to do this either, and the longer this iterative process runs the more comprehensive a candidate's exposure evaluation should become in theory.

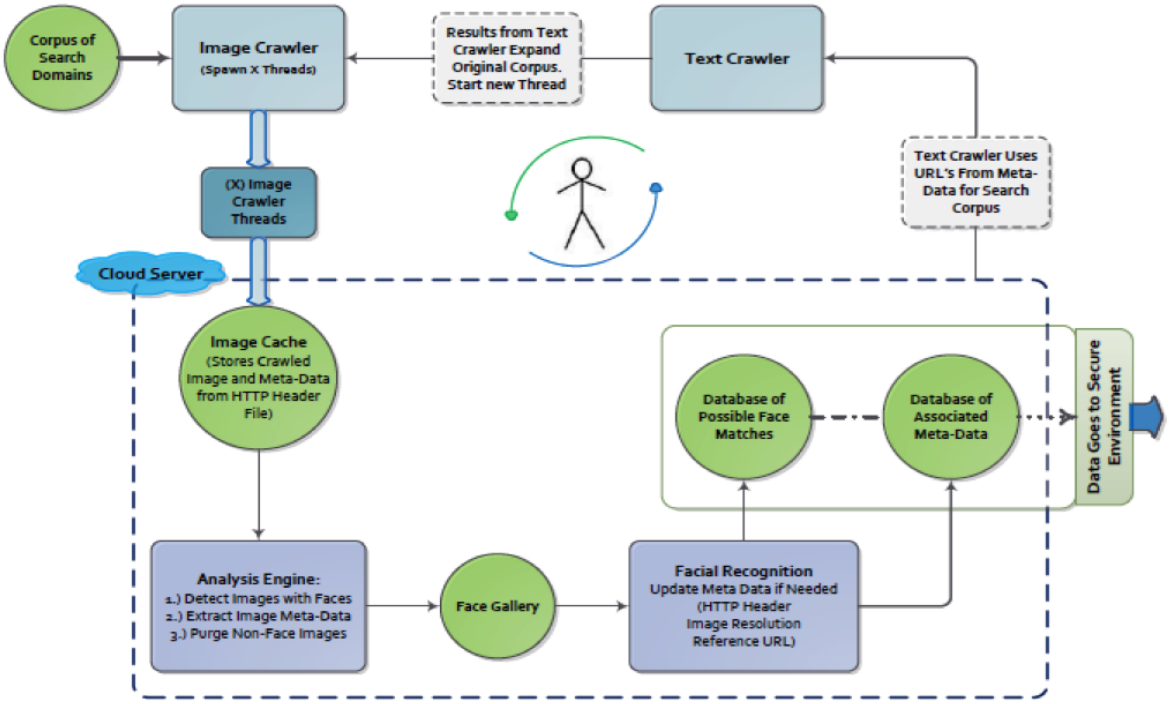

Figure 3.4: Web Crawling and Candidate Identification Data Flow Diagram

3.3.2 Database Schema

Figure 3.5 is a UML Database Schema diagram that showcases the relational model between each table and its respective fields in the database. It serves to show the primary organization of the data storage solution in place for all data captured by the UCAN Assessment System. The schema is in 3rd Normal Form and PostgreSQL best practices for indexing of primary fields was used to help cut down the amount of time it takes to query for data.

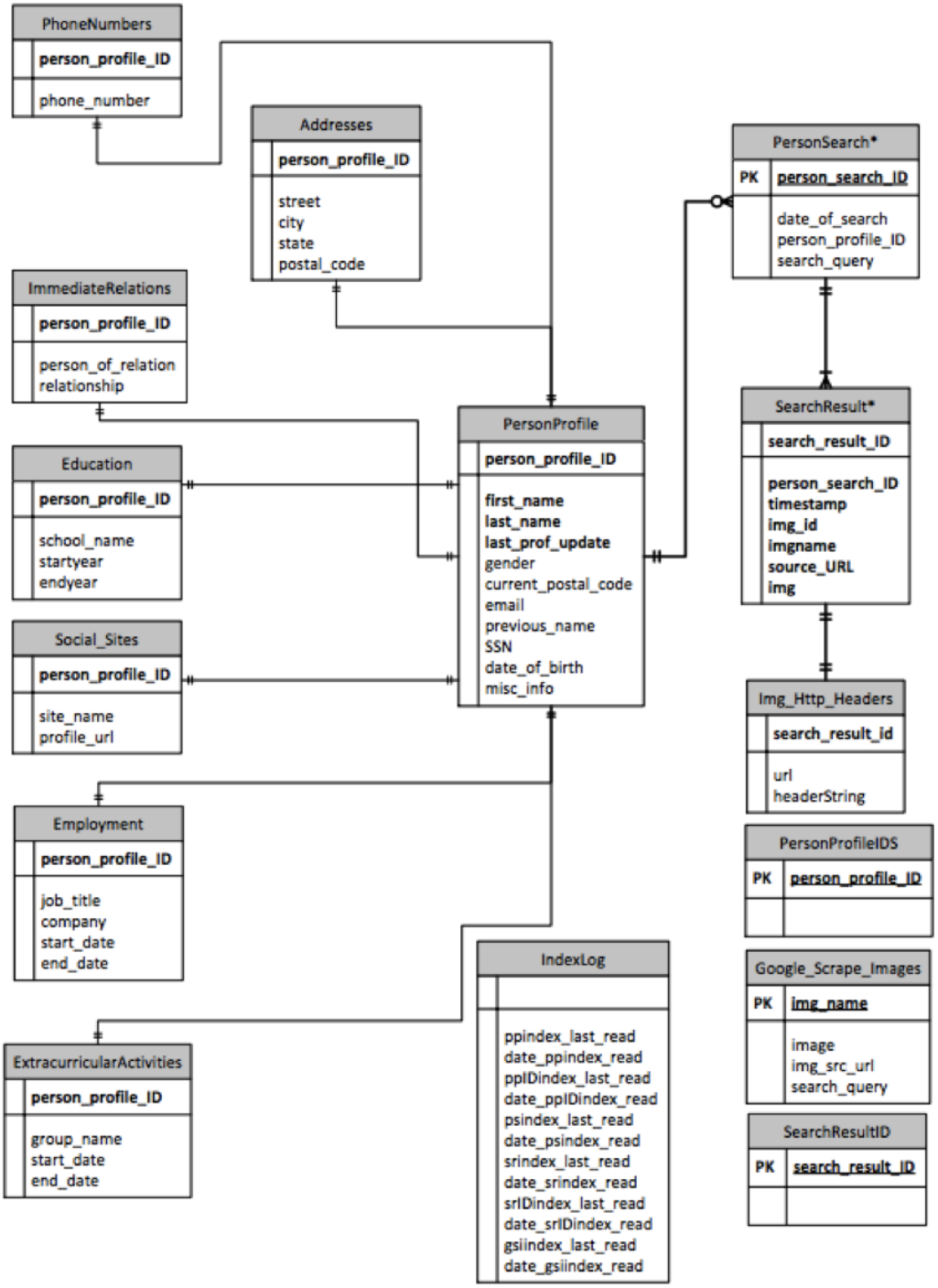

Figure 3.5: Database Schema

3.4 Interface Design

This section showcases the user interface of the system. It is ordered in such a manner as to reflect the way users process data through the system to obtain and analyze results. Screen shots with descriptions providing explanation of use give a visual representation of the full UCAN Assessment system.

3.4.1 Login and Main Menu

First thing that appears upon loading the client application. An account must already be created by an administrator for a user to login. A simple status connection status code is displayed to inform users if authentication of their account with the database is possible. An initial admin account must be created directly in the database by a designated server administrator.

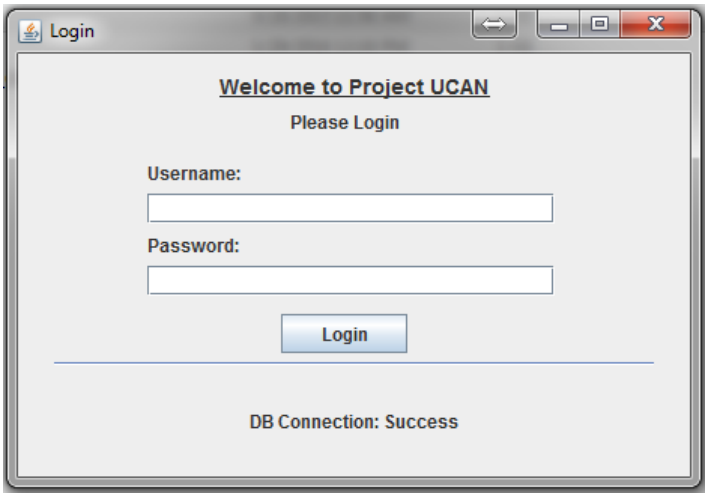

Figure 3.6: Login Interface

There are two different menu interfaces that can be presented to a user upon a successful login. Figure **??** shows what an administrator of the system would see and Figure **??** respectively shows what a general user woulds see. The Settings menu is all that is currently restricted from users, but additional features can be restricted rather easily by only modifying code that governs the menu.

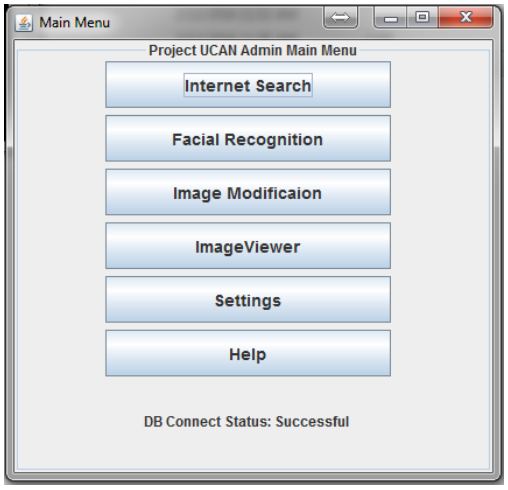

Figure 3.7: Admin Main Menu

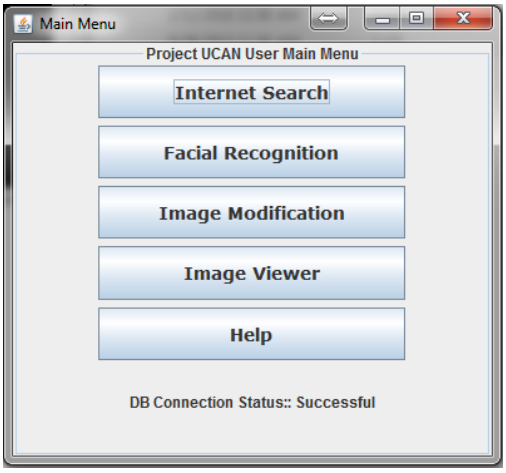

Figure 3.8: User Main Menu

3.4.2 Internet Search

Search Request Form and Boolean Search Constructor

This interface is the first presented if the Internet Search option is clicked from the main menu (Figures 3.7 and 3.8). The Search Reqeust Form Tab contains fields that are considered Personal Identification Information (PII). Each field can be either manually populated or the Load File button can be used to automatically import a profile to fill out the form. Using the checkboxes on the left side of the field-name will place the contents of the field into the Query Constructor area. All populated fields are uploaded to the cloud database once the Search button is pressed. A Log File Name is required for each profile created – the system discerns between profiles by generating a unique ID number using the name entered into the Name field. The same name will always generate the same ID number, hence permanently linking to two. To change this, check the Set File Number field and enter a custom ID number. Checking the Search all Permutations button on the right will auto-generate all Boolean Search (only will change Boolean Operators) permutations of the string in the query-constructor.

Constructing a Query can be done by either manually typing in the Query Constructor or by using the checkboxes on data fields in combination with the Boolean Operator buttons on the right side of the UI. Boolean Operator Buttons are persistent across all tabs – operators

always have access to them. Clicking one of them will insert the respective Boolean Token in the Query Constructor at the place of cursor. Syntax for formatting Boolean search strings are based of Google's Advanced Search Standards.

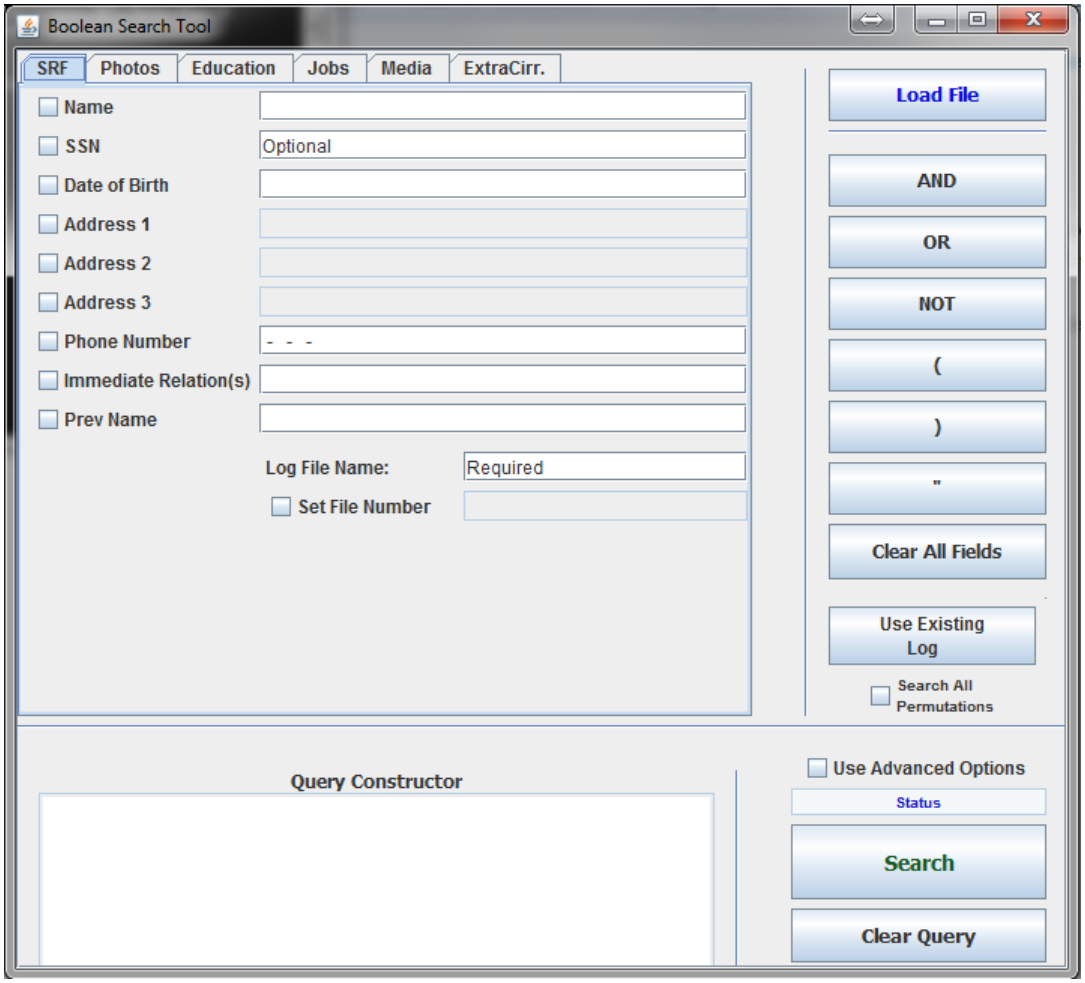

Figure 3.9: Search Request Form and Boolean Search Constructor

The photos tab is for uploading photographic information provided by a candidate or acquired by a user and is an integral part of the candidate's profile as these images are used to train the facial detection and recognition software used in other modules for performing biometric analysis of images gathered by the web crawler. The Select Photos button in the upper right portion of the tab will bring up a file browser where a local file(s) can be selected to be uploaded and stored as part of the candidate's profile.

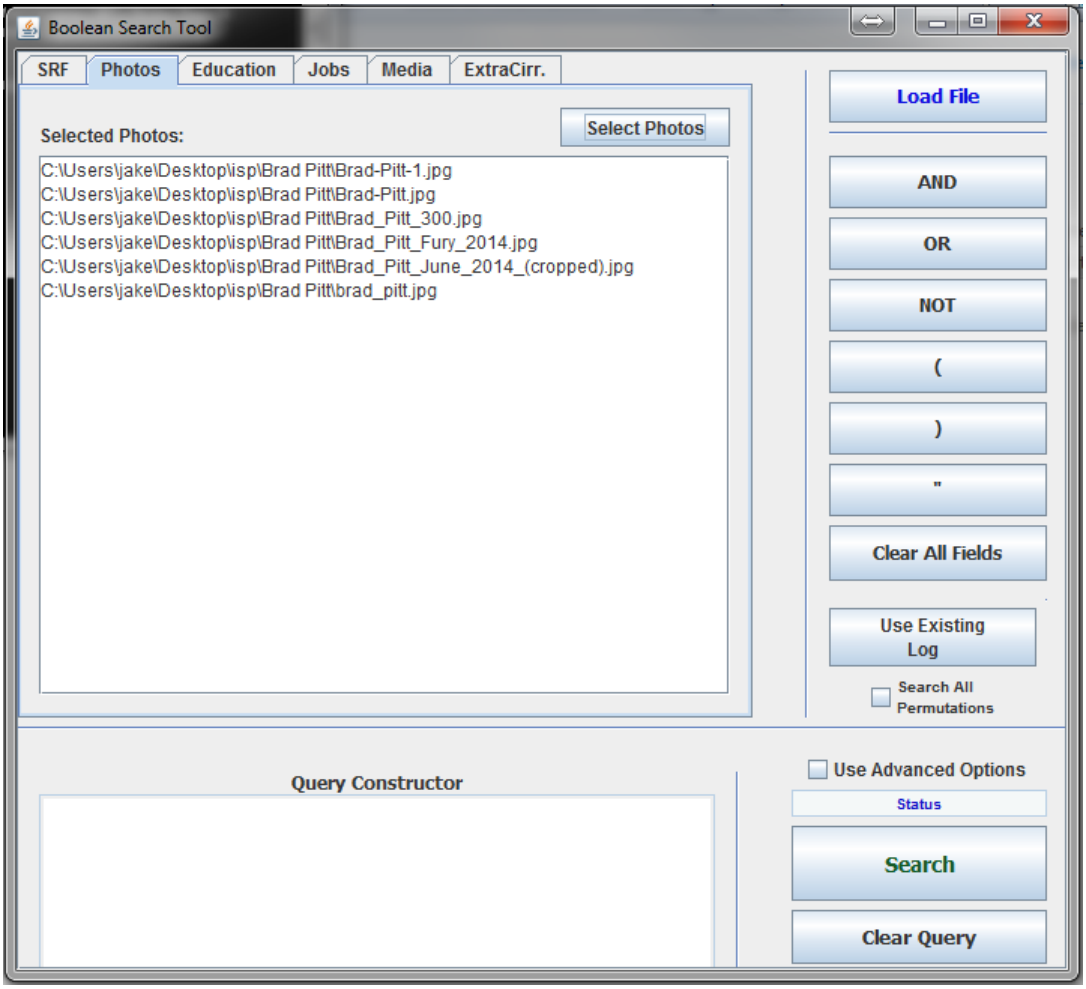

Figure 3.10: Search Candidate Profile Photos Upload Manager Tab

The Education Tab of the Boolean Search Tool (Figure 3.11) allows operators to enter up to five schools/institutions associated with an individual profile. Fields that are populated are uploaded to the cloud database. Fields on this tab require specific string syntax – each field's syntax can be found in the tool-tip associated with the field; simply hovering the mouse cursor over the field can access this.

The Jobs Tab of the Boolean Search Tool (Figure 3.12) allows operators to enter up to five places of employment associated with an individual profile. Fields that are populated are uploaded to the cloud database. Fields on this tab require specific string syntax – each field's syntax can be found in the tool-tip associated with the field; simply hovering the mouse cursor over the field can access this.

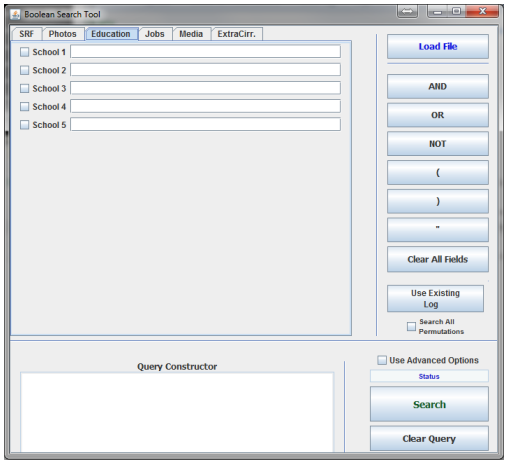

Figure 3.11: Education Information Tab

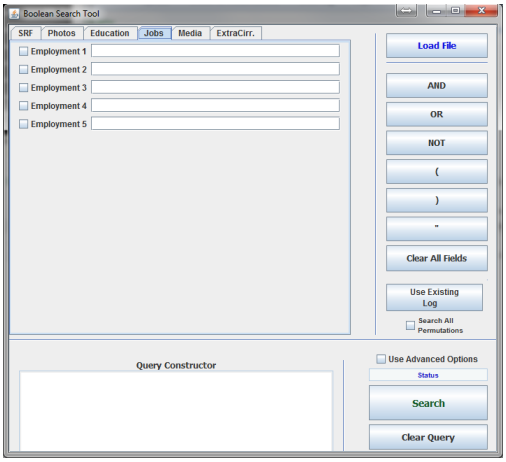

Figure 3.12: Employment Information Tab

The Media Tab of the Boolean Search Tool (Figure 3.13) allows operators to enter up to five Social Sites (URL's to profiles) associated with an individual profile. Fields that are populated are uploaded to the cloud database. Fields on this tab require specific string syntax – each field's syntax can be found in the tool-tip associated with the field; simply hovering the mouse cursor over the field can access this.

The ExtraCurr. Tab of the Boolean Search Tool (Figure 3.14) allows operators to enter up to five extra-curricular activities associated with and individual profile. Fields that are populated are uploaded to the cloud database. Fields on this tab require specific string syntax – each field's syntax can be found in the tool-tip associated with the field; simply hovering the mouse cursor over the field can access this.

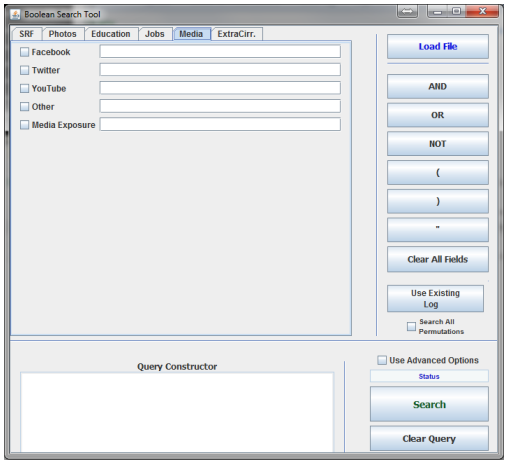

Figure 3.13: Social Media Information Tab

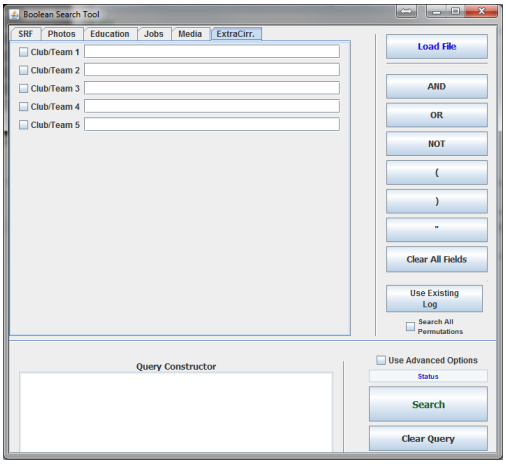

Figure 3.14: Extra Curricular Activities Tab

Web Crawler

The Web Crawler Manager interface automatically pops-up once a Search started from the Search Request Form is completed. The manager depicted in Figure 3.15 allows an operator to setup and launch an Image Web Crawler. From here an Image Folder can be explicitly defined or the default used for storing images locally. The Crawl Seeds File can be changed manually to utilize a different set of domains; and finally the number of threads the crawler will spawn can be defined with the default always being four. The Boolean Search Results section shows a list of websites ascertained from the Boolean Search Tool. Operators can manually add in any valid website URL's here to be crawled by the crawler. The View Crawl Logs button will display the log file for the domains to be crawled (essentially the contents of the Boolean Search Results but with more details). View Data Storage will launch the previously reviewed Manage Storage interface. Crawler Launch Info displays a quick and easy window detailing the settings of this instance of the crawler. Hitting the Start Crawler button will launch the crawler. As many instances of a web crawler can be running at any particular time.

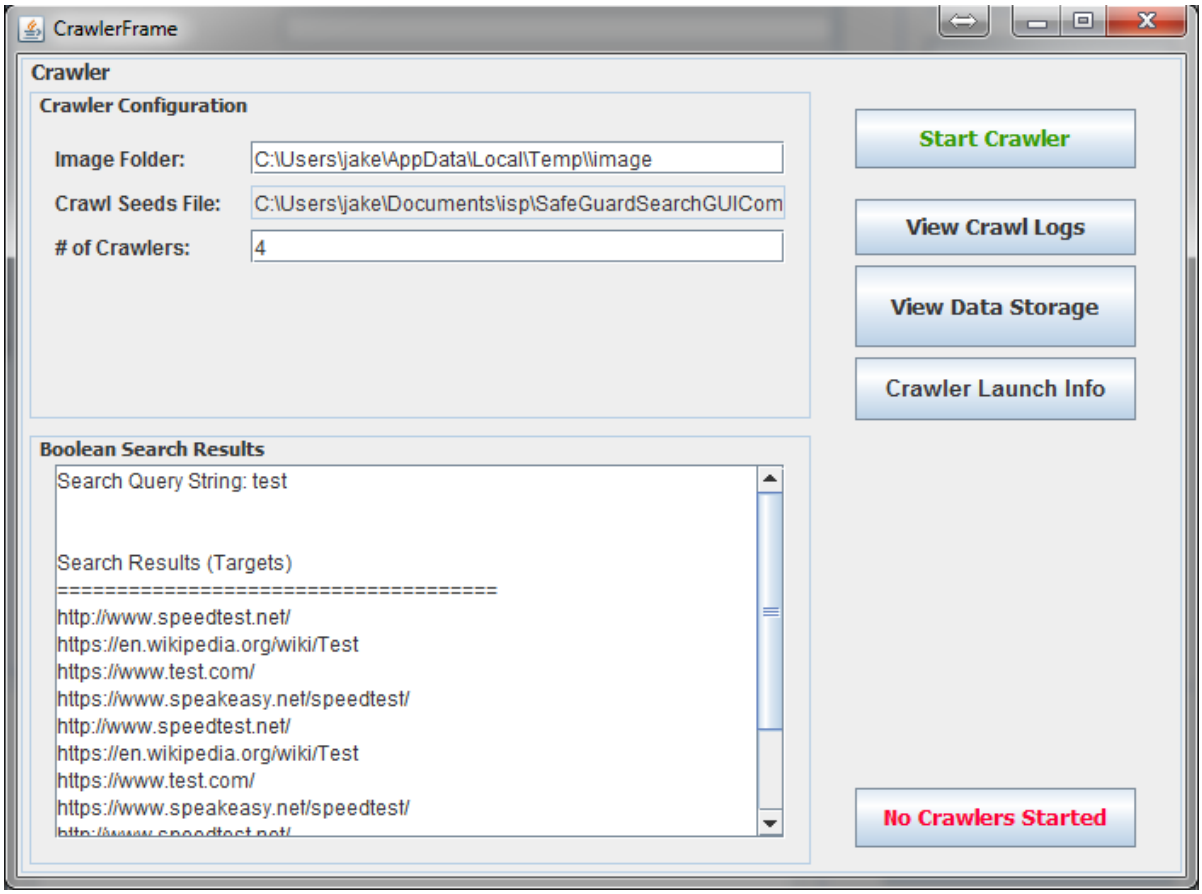

Figure 3.15: Web Crawler Manager

While a Crawler is running the Start Crawler button changes its text and is disabled. Now that a crawler is running, all of its properties are final. If something needs changed, the crawler must halt and restarted. Stopping a crawler that is running is as simple – press the Stop Crawlers button. This does not pause, but stops the instance of crawler (terminating each of it's threads); there is no current way to pause a crawler.

3.4.3 Facial Recognition

Accessible from the main menu, the Facial Recognition tool (Figure **??**) was built to give users an easy way to analyze the data Internet Search functionality of the system gathers in an asymmetric way - meaning users do not have to wait for a web crawl to be finished before facial recognition can be performed on the images collected. There are just a few simple steps that the interface guides you through to perform the underlying biometric operations. Select the user profile you wish process - by doing so the system automatically sets the training and testing image spaces to what users provided on the Search Request Form and the images already gathered by the web scraper and crawler respectively. This can be overridden via the checking the "Use Advanced Options" box and entering directories/folders that contain the desired images for each set. Result files are still always recorded and associated to the profile subject. Lastly, users must select an output path for the results - this is once again just choosing a file directory. The "Run Recognition" button begins the operation and an active status can be seen directly to the button's left. It should also be noted that in the upper left of the UI a link to the "Settings" menu can be found which gives the option to change aspects of the computer vision framework running underneath the UI (if permitted). In the upper right a "Help/Example" link gives an informative briefing on how to use the tool.

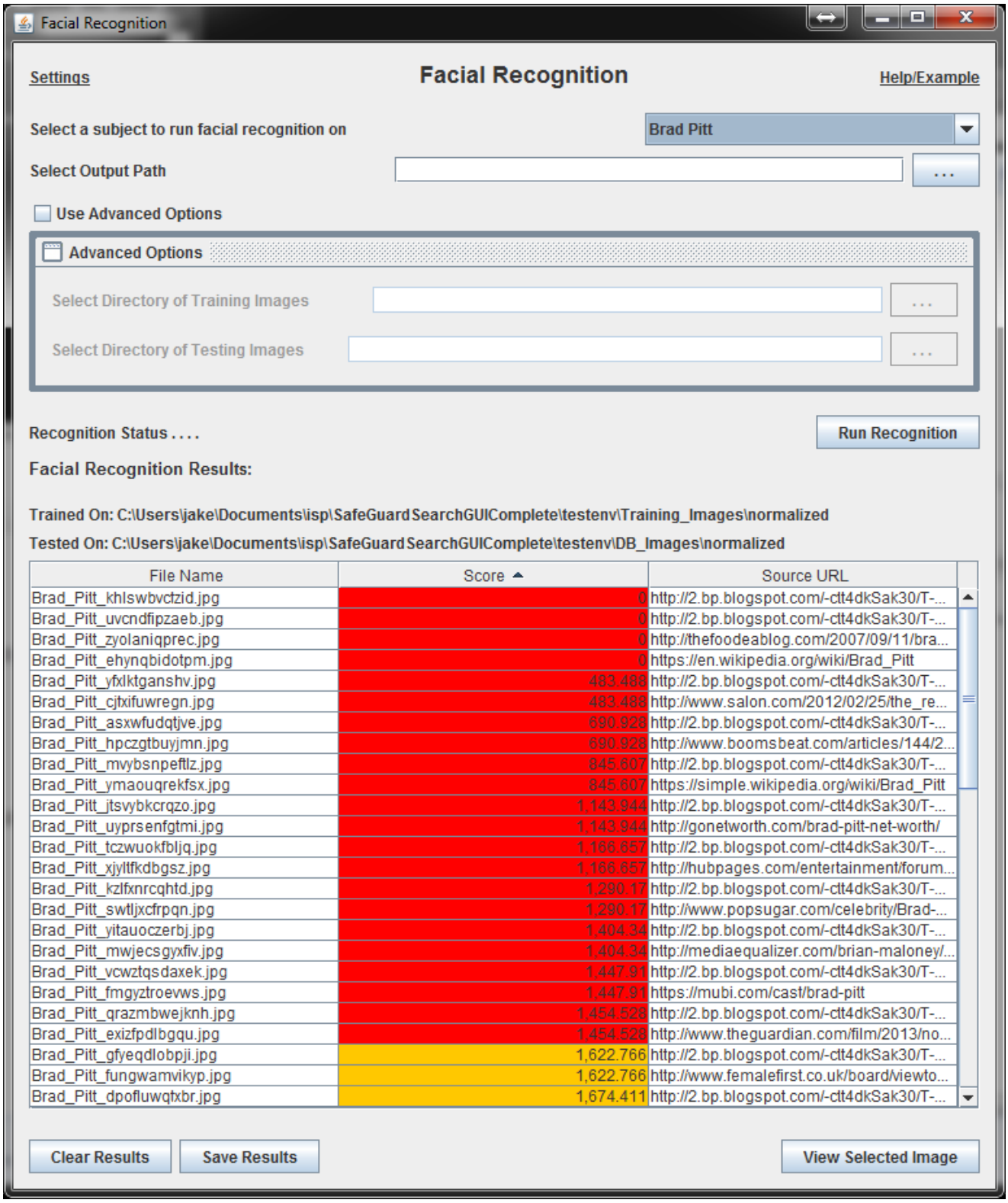

Figure 3.16: Facial Detection and Recognition Manager

The bottom half of the UI is the results table and it is divided into three columns: File Name, Score, and Source URL. The score of each image file is color coded in either red, yellow, or green. The color represents the danger the image potentially poses to online privacy. These scores represent the confidence to which the associated image is a match to the candidate's likeness. If a result is below 1.5 million in confidence, it is considered a likely match for the scoring method used by the default OpenCV algorithm (Eigenfaces). A score of 1.5 - 2 million is highlighted in yellow and anything above 2 million is green to indicate the file is most likely a negative match - and hence probably safe for use online. The source URL's of each file are listed on the right and indicate exactly where each file was collected from online.

Along the bottom of the of interface are three buttons that perform utility functions. 'Clear Results' omits everything in the results table, 'Save Results' will bring up a modal window that allows a user to save the results table to a text file. 'View Selected Image' will open the Image Viewer tool with whichever row is selected - just click the row to highlight/select it - to make the image viewable.

3.4.4 Image Viewer and Downloader

The Image Viewer tool (Figure 3.17) is a simple way to allow users to view images in a directory. Primarily oriented toward making the task of sifting through images pulled down by a web crawl easy and fluid, this piece of the application lists all files in a designated directory (top of the UI) on the the left hand sand of its interface with the majority of the window real-estate being used for the display of the image content of a selected file. The Previous and Next Image buttons in the upper right portion of the UI offer a quick way to iterate through a directory of images. In the bottom right corner of the window are two utility buttons. The first gives users the ability to open in a web browser the image's source page - the website where the crawler found and downloaded it. The second button offers quick access to the Facial Modification tool, which if clicked will bring up the tool with the currently selected image.

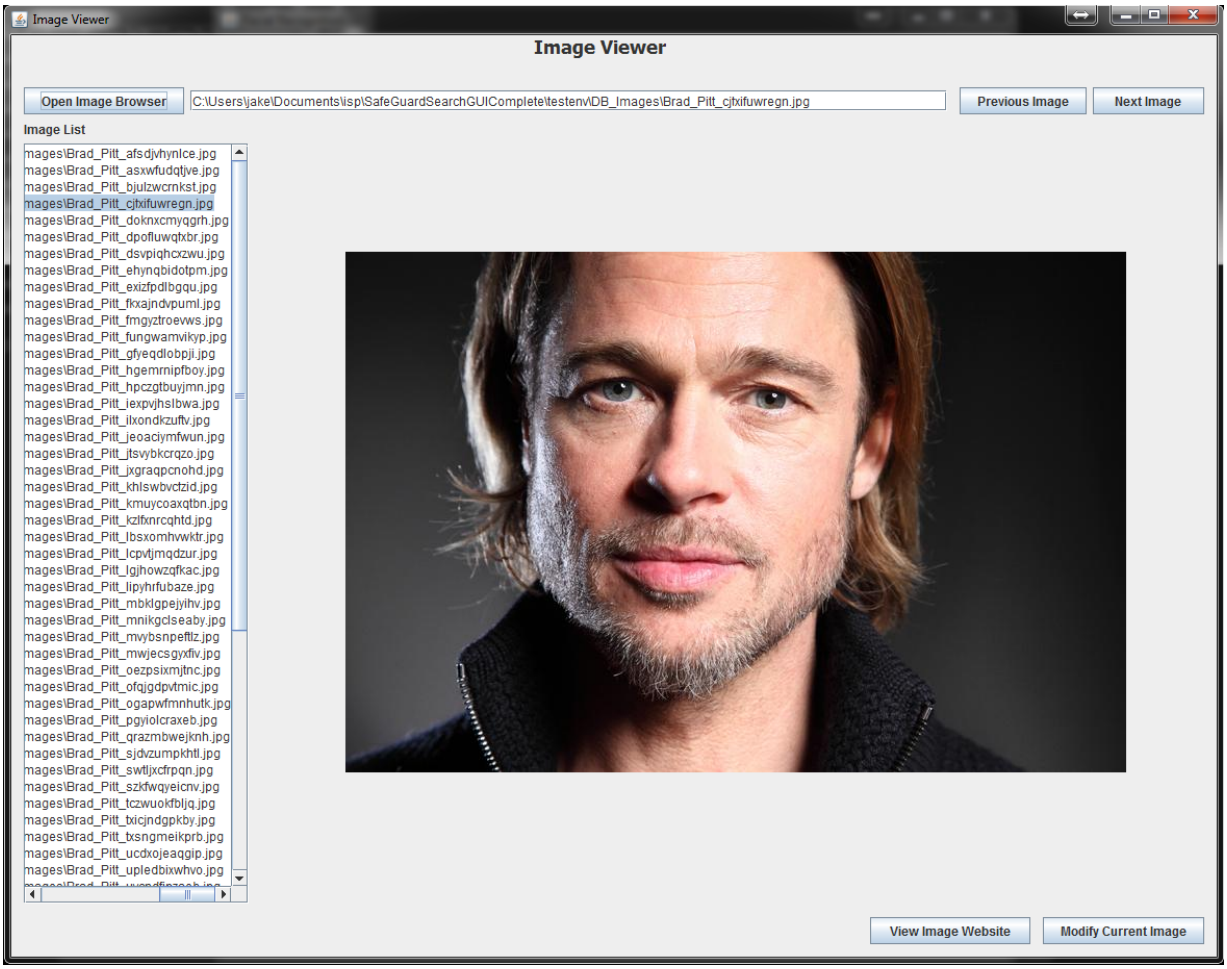

Figure 3.17: Image Viewer Tool

The Image Downloader (Figure 3.18) is a simple tool that allows users to download images from the Cloud Database. This means if a crawler on another physical machine pulled down an image of interest or if this particular image was found and cached at some point in the past, it can be easily retrieved. There are several options the interface provides for doing this: all images crawled/scraped for a profile can be pulled down, all images designated for training (Profile Photos) a facial recognition algorithm can be downloaded, a specific image can be downloaded, and finally a range of images can be downloaded by providing the beginning and ending row number for the database table in which they are stored. The bottom left and right hand corners of the UI contain radio buttons that allow users to choose between accessing the pool of scraped (left) or crawled (right) images. Both cannot be downloaded simultaneously.

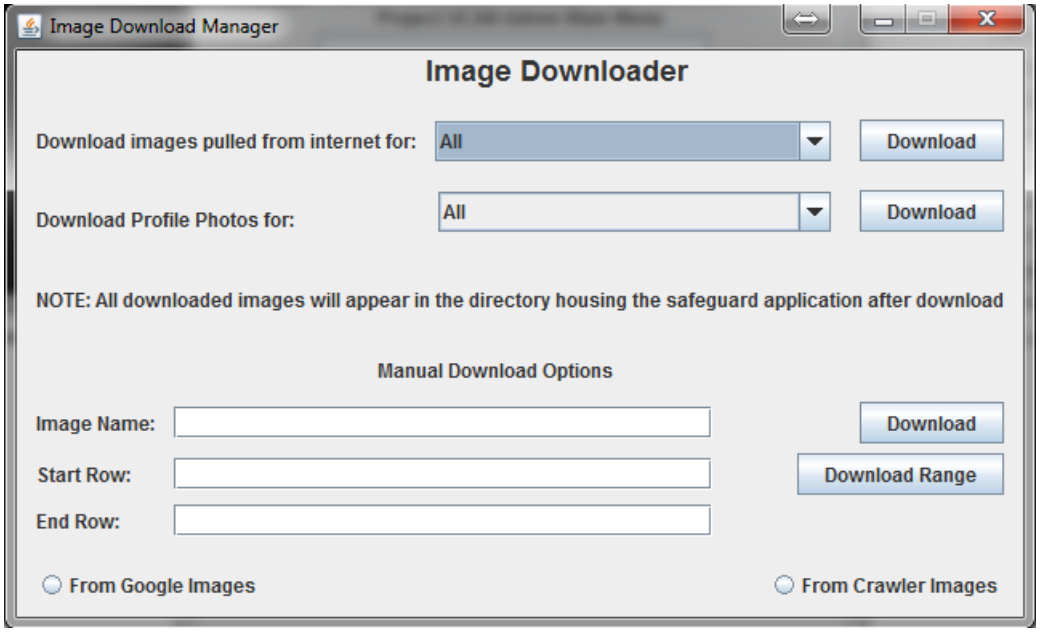

Figure 3.18: Image Downloader Tool

3.4.5 Image Modification

The Image Modification tool is offered as a mitigation strategy for any image found by the Facial Recognition tool to be a positive match and then deemed a threat by a user due to a risk it poses toward a candidate's online anonymity. Figure 3.19 is a screen-shot of the default user interface a user is presented with which uses a generic place holder image. This place holder/default image is unique in that rather than having to open the image file, modifications can be previewed on it in real time in the Facial Modification window. This is useful as it allows the user performing the modification to first visually test the changes they plan to make to ensure that the face does does not loose its plausibility.

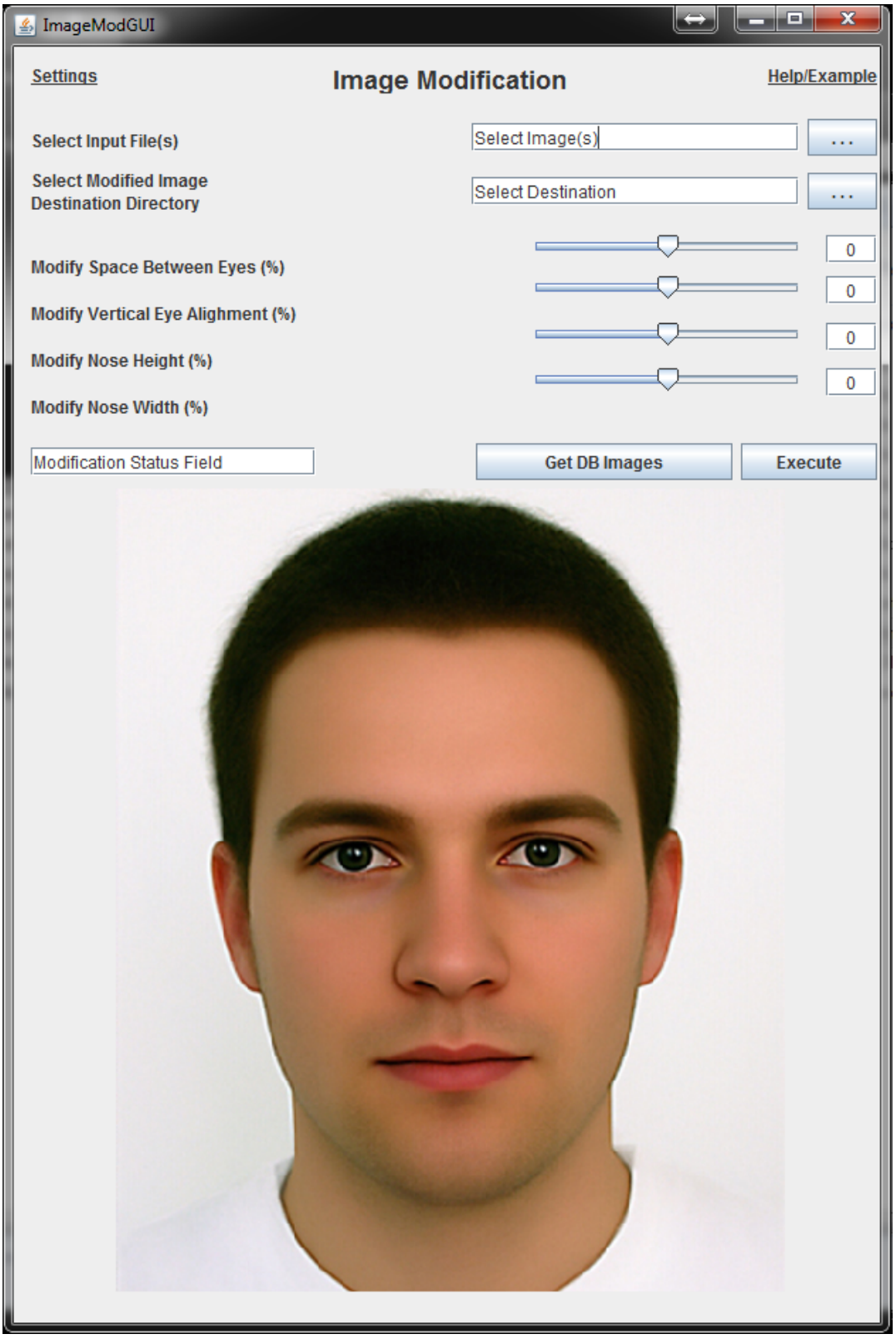

Figure 3.19: Default Image Modification Tool

As per usual, the interface is laid out vertically and in order of necessary operation from top to bottom. Users are first required to select the image they wish to modify - multiple files can be selected and iterated through automatically, with the same facial modification being performed on all of them - and then designate a directory to store each newly created modified image. Once an image is selected the default image is no longer displayed and is replaced with the selected image. A working example of this can be seen in Figure 3.20. The next four UI elements are sliders that adjust four aspects of the face by a given positive or negative percentage: the space between the eyes, the vertical eye alignment, the height of the nose on the face and the width of the nose. The 'Execute' button will will perform these operations by calling upon two Python scripts. For more information on this process please refer to [Wolen's Thesis] and [Poster's Thesis] - these documents detail the underlying technology in use. Lastly, the 'Get DB Images' button brings up the Image Downloader tool so users have an easy way of accessing images that need modified that aren't locally stored without leaving the modification tool.

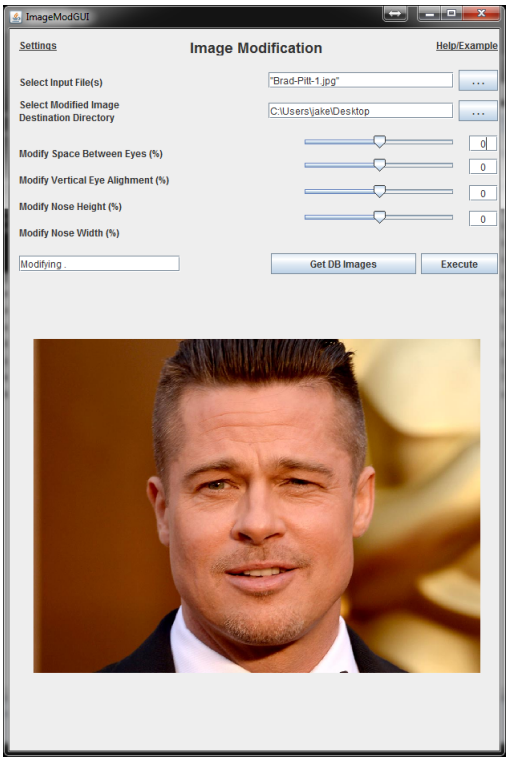

Figure 3.20: Image Modification Tool - Working Example

3.4.6 Settings

The Settings Menu is only available to users with administrative (admin) privileges with the system. There are four tabs that make up this interface: Add User, Facial Recognition, OpenCV, and Databases. Figure 3.21 shows the 'Add User' tab which allows a system administrator to create a new user account with system. Figure 3.22 allows for the swapping of Facial Recognition algorithm. If a new system is developed, purchased, or obtained freely for use as is, it can be assigned for use with the Facial Recognition tool here by providing the file path to an executable file for it (.jar, .exe, etc.).

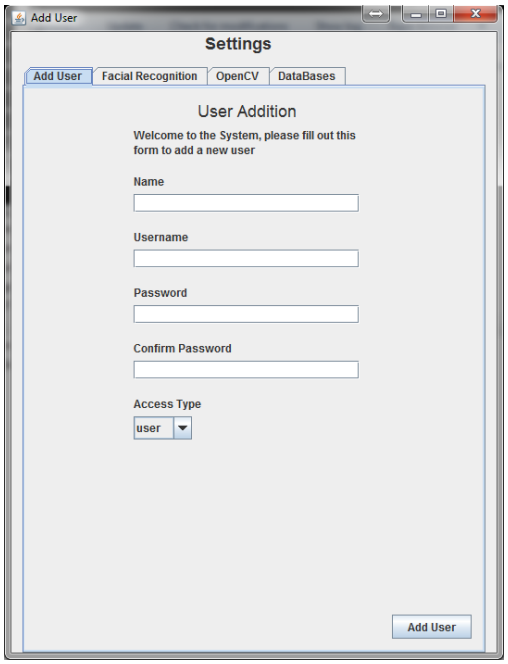

Figure 3.21: Social Media Information Tab

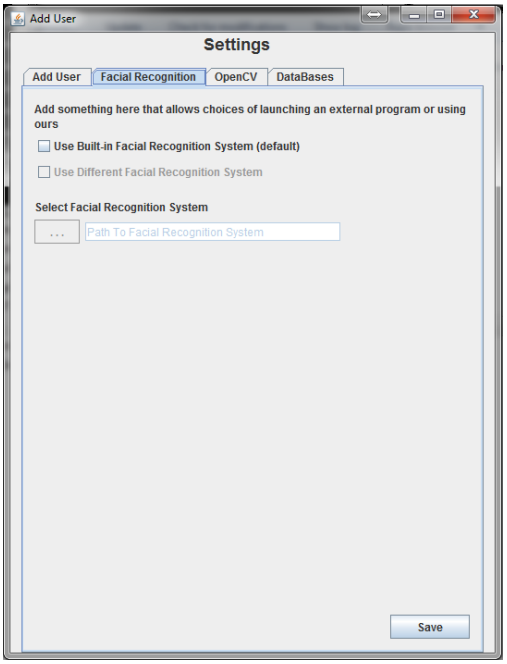

Figure 3.22: Extra Curricular Activities Tab

The OpenCV tab in Figure 3.23 allows users to modify specific aspects of the OpenCV framework, which is the default framework used by system for performing facial recognition and modification. Lastly the Databases tab (Figure 3.24) has options for setting new database pointers if the database that maintains user accounts or the image and profiling data changes servers or if an alternative database is wished to be used. There is also a ping button to test the connection with the database entered in the above fields.

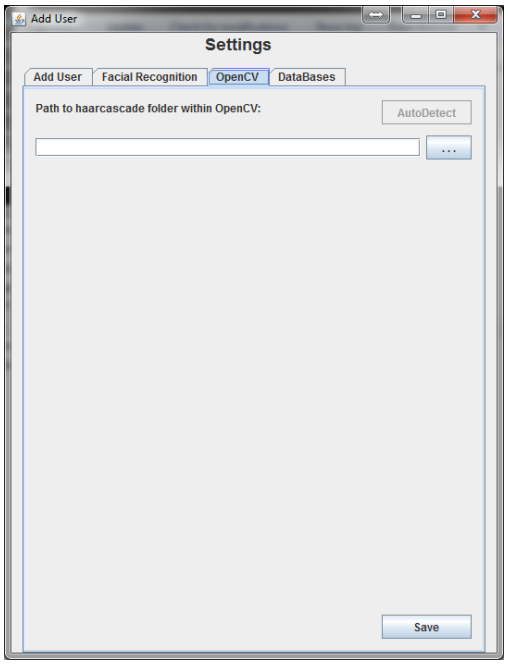

Figure 3.23: Social Media Information Tab

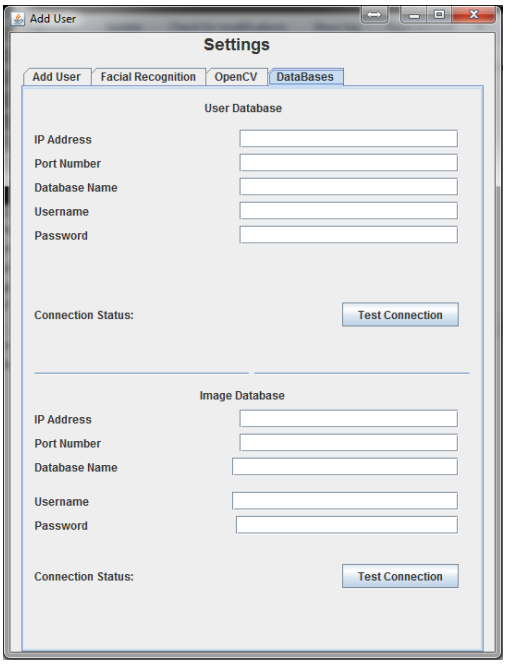

Figure 3.24: Extra Curricular Activities Tab

3.5 System Technical Specifications

The system was developed in Linux - Ubuntu 14.04 LTS but is designed to run on Windows 7 or higher, OS-X 10.9 Mavericks or higher. An i3 dual core processor with 4GB of RAM and a 256GB HDD are recommended as the minimum hardware specifications for running the system smoothly. The multi-threading of the web crawler can take advantage of more robust hardware by increasing the number of active threads per crawl. Running the client on a machine requires the Java Runtime Environment (JRE) and version 2.7 or higher of Python to be installed.

3.5.1 Versions and Dependencies

- Client Graphical User Interface
  - Developed in Java v.1.751
  - Java Swing Libraries (native) for GUI
  - Makes use of Deprecated Google Search API
- SmallScrape.py
  - Python v.2.7
  - Beautiful Soup Libraries v.3.2.1
- Web Crawler
  - Developed in Java v.1.7.51
  - JDBC PostgreSQL Driver to Connect to Cloud Database
  - Crawler4J Open Source Web Crawler Framework
- Cloud Database
  - Private Ubuntu 14.04LTS Server
  - Monit for monitoring of open ports for use of SSH and SCP

  - PostgreSQL v.9.3.1

- Image Modification/Facial Recognition/Facial Detection

  - Python v.3.3.x

  - OpenCV v.2.4.10

Chapter 4

Experiment Methodology

This chapter outlines the systematic process of an experiment that seeks to validate the effectiveness of the capabilities of the Internet Search and Facial Recognition tools to provide an exposure score of candidates run through the UCAN system using only a single photo for training the facial detection and recognition algorithms to simulate the baseline scenario of only having a social media profile picture or surveillance photo - any photo taken by a public-facing camera - to use as a means of identifying an individual. This serves a dual purpose of simulating the two plausible scenarios of someone building and using a similar system with malicious intent or with intent to vet a single photo for use online. The The two main aspects to be tested are how well the web crawler/scraper work at downloading images that are positively associated/relevant to a candidate and how well the chosen implementation of facial recognition performs at automating the identification of the images downloaded by the web crawler/scraper. Both of these aspects are contributing attributes to a candidates overall exposure score as defined by equation 4.1.

The validation of the Image Modification tool can be found in Wolen [19] and Poster's [7] work respectively as it is their systems that run underneath the UI. The successful interaction of these systems to completion when running the experiment inherently proves that a cohesive system built with readily available technologies can be used to provide fluid automation of photographic PII data gathering into biometric analysis for evaluation of exposure.

4.1 Preparatory Analysis and Setup

There were several key decisions that needed to be made regarding the specific configuration of the technologies used to fill each part of the UCAN System before performing this experiment. The first is which readily available facial recognition platform should be used - the default as aforementioned is OpenCV's Eigenfaces implementation. The second decision is dependent on the first, being that it is the acceptance thresholds for facial detection and confidence in recognition scores. The third decision is the time epoch - or period of time - used to standardized the intervals of crawling for the web crawler. The final is which search engine API should be used by the search request form and web scraper to provide the web crawler with seeds.

4.1.1 Web Scraper and Crawler Configuration

The sole parameter to be determined for the SmallScrape.py web scraper is which search engine API it would utilize. This decision influences the first batch of images downloaded into the database as well as the websites given to the crawler as seeds. Since Google is by far the most popular search engine in the world (with a reported %72 of the global market share in search) and their proprietary Pagerank algorithm already takes into account a page's exposure to a certain degree they were the easy choice to go with [20]. SmallScrape was configured to use the top 100 results of a Google Image search and the first 4 websites that appear from a standard "All" search - arbitrary assumption being anything outside of the top 100 would be irrelevant or excessive and would hence produce unnecessary noise and volume in the data to be processed.

The web crawler needs to be run for a uniform amount of time for all candidates processed through the system in this experiment. A standardized amount of time needs to be defined to use as an epoch for calculating the overall exposure scores. This is difficult to define given the randomized delays per HTTP request made by the crawler and the unpredictable variance in size of the images encountered by the crawler. Download speed and outages also play into this as these vary given the amount of strain on a given network being used to make and serve the requests produced by the crawler. To compensate for the high level of unpredictability,

the amount of time chosen erred on the side of excess as one epoch was defined as four times the upper limit it takes to download 1000 images with an average download speed of 2Mbits/s (the specified rate by the ISP of the WAN the hardware used was connected to), an average file size being 50MB, and each request delay being the maximum seven seconds. Given these values, it takes 3 seconds to download plus the time of the delay to make the initial request, a total of 10 seconds per image and thus 10,000 seconds to download 1000 images. This is equivalent to 2.8 hours, 3 hours being the nearest whole value, and then four times that being 12 hours. Every 12 hours is defined as one epoch.

4.1.2 Face Identification Configuration

A licenced copy of VeriLook 5.4 from Neurotechnology was chosen as the platform for performing facial identification. This decision was made given that preliminary informal runs of the default OpenCV Eigenfaces algorithm was observed to perform abysmally when only a single photo was provided for training - only identical photos were identified as matches (score of 0 meaning exact matches) with the rest receiving a score of over 3,000,000. Given that VeriLook was affordable to the average consumer, is very popular, and offered out of the box one-to-many facial recognition it was decided as the replacement for the default [9].

The default thresholds for image quality score and confidence were kept as part of the configuration. The default score threshold is 128, anything above the threshold can be considered a positive match with higher values signifying greater confidence. Anything less than this threshold will be deemed as a negative match and hence disregarded. It is also strongly recommended that an image's resolution be at minimum 640x480 for VeriLook to have enough information in the image for feature extraction [12].

A calibration test was run on one of the candidates using a separate bust photo than the one to be used in the experiment. Figure 4.1 shows the image was successfully marked for feature detection by VeriLook for use as a training template. This verifies a proper setup and configuration of the VeriLook SDK.

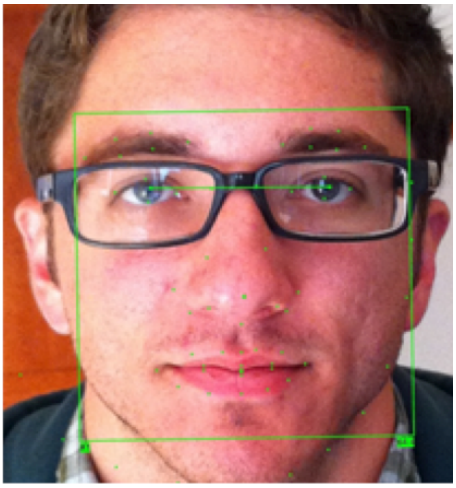

Figure 4.1: VeriLook Calibration Image

4.1.3 Assumptions

It can be assumed that the following factors will influence the ability for the facial identification algorithm(s) to produce reliable results: the resolution of the image, lighting within the image, background faces in the image / more than one face, aging variance, obstruction of the face from clothing or environment, and posture of the face in the image [8]. It is also highly likely that the majority of the images downloaded by the web crawler will have these factors negatively affecting the performance of the VeriLook one-to-many facial identification algorithm.

As for the web scraper and crawler, it can be assumed that most immediate positive results will come from the web scraper since Google's Image search results are the product of their industry standard setting web indexing. The web crawler will also more than likely encounter one or more images that are very big in size (GB's) and take a considerable amount of time to download. It can also be assumed that the uniqueness of the name - which is dependent on the candidate - used to perform the search generate the web crawler seeds will help to steer results to a more accurate position because there is a natural constraint on the size of the search space.

The longer the web crawler goes - the more epochs observed - the less granular the web crawl domains become. Hence the assumption can be made that greater periods of time will

yield more irrelevant data, or noise, within a candidate's photo set. This creates a "Faces in the Wild" type scenario for the facial recognition where the search/test space for the facial recognition algorithm is unconstrained [21].

4.2 Running the Experiment

Three candidates were chosen to be processed through the system each with a different size of digital footprint. The three candidates chosen were each representing a respective category of high, medium, and low levels of assumed usage of online services as qualitatively determined by the amount the candidates admitted to/or were known to be using. The 44th President of the United States Barrack Obama was the high exposure candidate, as we know he has an extremely high digital footprint due to the celebrity that comes with holding the US Presidency and would provide a baseline for maximum risk. WVU Graduate student Elliott V. Iannello (the author of this thesis) and WVU Professor Dr. Roy S. Nutter were the medium and low candidates respectively.

Two time epochs (24 hours) were observed for each candidate by the web crawler's active crawl time (an epoch being defined in Section 4.1) with 4 threads initiated for each candidate's respective crawl. All crawls will be performed on the same network and machine but it should be noted that rates will still vary unpredictably per candidate as different size images will be encountered.

Facial Recognition is configured to currently use the entire database of photos collected across all crawls run to simulate an unconstrained environment similar to a "Faces in the Wild" scenario [21]. This means there is a chance that a result is returned that doesn't have a Google Relevance value associated to the candidate since the seed that generated the image came from another unassociated web crawl.

Table 4.1 describes the algorithmic process and associated actions taken by a user of the UCAN system to perform the data gathering, data analysis, and the scoring to evaluate each candidate's level of online exposure in the proposed experiment.

Step in Algorithm	User Action	UCAN Tool
1 Form Candidate Profile	Fill in SRF and Upload Training/Profile Photo(s)	Internet Search
2 Construct Search Term	Combine SRF Data with Boolean Operators	Internet Search
3 Search Term Yields Corpus of Domains	Run Search on Search Term	Internet Search
4 Crawl Corpus of Domains for Online Photographic Data	Configure and Run Web Image Crawler	Internet Search
5 Train Facial Identification Algorithm to Identify Candidate	Use Profile Photos from Step 1 to create VeriLook Template	Facial Recognition (VeriLook)
6 Perform Facial Recognition using gathered Online Photographic Data	Configure and Run Facial One-to-Many Facial Recognition	Facial Recognition (VeriLook)
7 Score the Results	Use the Confidence Scores as input into Equation 4.1 and Calculate	No Tool (Analyst Driven)

Table 4.1: Algorithmic Process to User Action Relation

4.2.1 Data Input

Generation of the data to be used as input begin with each candidate filling out a Search Request Form so a simple Boolean Search can be run against them using just their first name, last name, and middle initial (if provided). A single training image will be used to create a template in VeriLook for matching for each separate run of a candidate through the system. Each of these photos will be uploaded using the photos tab of the Internet Search tool (Figure 3.10). Each training image was taken using a cellular device - an iPhone 5S without HDR enabled with the exception being Barrack Obama whose training images were acquired from the Official WhiteHouse.gov administration page (his official work photo). Each training image will be a forward facing bust shot of the individual in a well-illuminated

environment with as solid and consistent a background as possible. This was setup to simulate the scenario of a malicious individual using a standard social media profile picture to scour for more information about a candidate. There are intermittent forms of output acting as input for other subsystems that make up the whole of UCAN System, please refer to Figure 3.2 for greater clarification.

4.2.2 Data Output

There are technically four tiers of output - the first is the corpus of domains produced by the Search Request Form tool that is given to act as seeds for the Web Crawler tool. The second tier are the images collected by the crawler and scraper. The third tier of output are the results of the Facial Recognition tool which are given as confidence values - representing the algorithm's (in this case defined by VeriLook) certainty that a given image contains the likeness of the candidate. The fourth and final tier of output, which is also the goal, is the exposure score assigned using Equation 4.1. The score is the desired and most informative piece of information that can be derived from the system about a candidate. Once again, for a clearer idea of how all the data flows together please review Figure 3.2 as a visual aid.

4.2.3 Candidate Exposure Scoring

A candidate's exposure score is defined to approximately be the summed total of each associated image's exposure (E_i) times the facial recognition's normalized confidence score (in relation to the minimum threshold score for being a positive match (T_{min})) for that image ($conf_i$) all divided by the number of standardized time increments, or time epochs (t_{st}), that passed to obtain the total number of images collected (n). Exposure is defined as the product of the image's Google Relevance rating (G_{rel}) and the average number of hits per month for its source site (H_{pm}). Hits per month of the source site is the best metric for determining the amount of web traffic a website sees since it is the most holistic - there is no discrimination between actual users and automated ones (the automated ones, or bots, could be doing a web crawler of a system trying to perform the same task as the UCAN system with malicious intent). An image's Google Relevance is a normalized value (ranging

zero to one) determined by the rank of the seed (R_i) among the seeds (search results) given (I_{total}) by Google associated to the image (the rank in Google's results is determined by Pagerank) [20]. For example the first image listed in a Google Image result is the top image and its source URL is used as a seed for the web crawler. That image and any image found by crawling the same source URL or any other URL that can be traced back to the source URL as a point of origin receives the same Google Relevance rating.

The formulaic representation of the exposure score and it's components are seen in Equations 4.1, 4.2, 4.3, and 4.4.

$$S_c \approx \frac{1}{t_{st}} \sum_{i}^{n} E_i \times conf_i \tag{4.1}$$

$$E_i = H_{pm} \times G_{rel} \tag{4.2}$$

$$G_{rel} = \frac{I_{total} - (R_i - 1)}{I_{total}} \tag{4.3}$$

$$conf_i = \frac{X_{conf}}{T_{min}} \tag{4.4}$$

Manual Application / Calculation of each candidates overall exposure score was necessary due to values that yield E_i. It was not a part of the UCAN system's scope to automate the retrieval of the data for hits per month (H_{pm}) of a source website. Additionally it was not possible to automate calculation of the Google Relevance due to the web scraper not being able to reliable order each image due to randomized timing delays and variance in image size effecting download speeds. Google also doesn't provide the result rank as part of the meta-data when an image is scraped (no meta-data is technically provided when scraping).

Although this scoring algorithm is tailored to the UCAN System and was derived by examining the nature of its comprised parts and the ordering in which they are normally used by an analyst, it can still be adapted for use on data that is generated by an alternative entity.

Chapter 5

Results and Analysis

The main goal of this research was to use the experimental methodology and technology described in the previous two chapters to prove or disprove each of the following three hypothesises:

1. The overall exposure score for each candidate will correlate respectively to their pre-disclosed frequency of use of online services (i.e. high, medium, and low).

2. The number of images successfully identified as a match by the facial recognition software for each respective candidate should correlate to each candidate's respective pre-disclosed frequency of use of online services (i.e. high, medium, and low), meaning more images should be positively identified for a candidate who admits to higher usage of online services.

3. The success of the system to act cohesively to produce a result where a candidate is successfully identified in one or more gathered online photos validates the system as both a potential threat and vetting tool for acting as an automated agent toward online photographic privacy.

Each candidate's results from the experiment are broken down in the following sections with an aggregate analysis following that will ultimately address each of the three aforementioned hypothesises as confirmed, denied, or inconclusive. To preface the results disclosed in each section, a few matters should be noted. The web scraper results were limited to the

first 100 images (as mentioned in Chapter 4) and the 4 results used as web crawler seeds from the Google API, which are the top four results of their "All" search, were appended as 101-104 in terms of their relevance since they did not focus on images. Only the top 15 results were analyzed for each candidate due to a number of factors. First being the necessity of manual analysis to calculate the overall exposure scores making it too time consuming to perform on the volume of data collected. The second factor being that candidate one (assumed high digital footprint) was the only candidate that had positive matches from VeriLook (the threshold score being 128 or above). The other two candidates had abysmally low scores with neither of them breaking 40 except for one image from candidate three, meaning that it rationally follows that an analyst or malicious online user would ignore the majority of results all together.

All data for the Hits per Month columns in the results tables were ascertained from `http://www.trafficestimate.com`. For the sake of absolute clarity, constants across all three candidates include the minimum threshold (T_{min}) of 128 for use in Equation 4.4 for the calculation of the confidence score of each image; the I_{total} for Equation 4.3 across each candidate was 104 - this represents the total number of seeds used for each web crawl run for each respective candidate - and the value for t_{st} was 2 since each candidate was given a 24 hour web crawl.

5.1 Candidate One

Candidate One was the assumed high digital footprint candidate due to frequent use of online services. This candidate was a control for what the high bar for an exposure score should be. Barack Obama, 44th President of the United States, was chosen as the high candidate as it can be safely assumed that the celebrity associated with his position would yield an overwhelming amount of photographic data.

The search ran to generate the candidate's web crawler seeds was: Barack Obama. Figure 5.1 was the training image provided via the Search Request Form that was used by VeriLook to perform one-to-many matching on the 13,956 total images that were gathered by the web scraper and web crawler. Of the 13,956 images gathered, 7168 were associated to candidate one - this is 51% of the total search space.

Figure 5.1: Candidate One (Assumed High Exposure) - Barack Obama

Table 5.1 shows the top 15 results the system yielded based on the VeriLook Score. Candidate One had every single result in the top 15 be above the 128 threshold and each image was human-verified to be a positive match for Barack Obama. Only one result, row 14 of the table, had an unattainable Average on the Hits per Month of its source website - this is most likely due to consistently low amounts of web traffic.

Table 5.1: Candidate One Results

File Name in Data Base	Source URL	VeriLook Score	Avg. Hits per Month Source Website	Google Result Rank
060976d443c74c7db00afa1db36d559e.jpg	https://www.whitehouse.gov/ administration/president-obama	9334	3,607,500	1
1b29fbe38eb44d02a1d3e235d702fd8c.jpg	https://www.whitehouse.gov/ 1600/presidents/barackobama	9334	3,607,500	2
1573f5290e39460ea937609e1536b84c.jpg	http://www.biography.com/ people/barack-obama-12782369	9334	3,703,000	3
18fdf2ea954646d397c9f60eb667c713.jpg	http://www.newsweek.com/ barack-obama-one-question-presidential-candidates-409396	3010	3,834,700	8
2fbcf994b27f4a0e967ae7c34510ce50.jpg	http://www.biography.com/ people/barack-obama-12782369	2967	3,703,000	4
5799ce694cc941f093d05f23725047ab.jpg	http://abovethelaw.com/ barack-obama/	2745	767,600	6
70ed1415b8e341deb544eb5d7358c9ee.jpg	http://www.forbes.com/ profile/barack-obama/	2733	60,229,000	5
a294b0051f5c4dfd8ae61e66cf19e11c.jpg	http://www.usmagazine.com/ celebrities/barack-obama	2065	8,732,700	9
86c68d02f9394fee86940e903e56e975.jpg	http://wallpapersdsc.net/celebrities/ barack-obama-11412.html	2065	361,700	12
26fb4af5bc774f25b4765b426d568470.jpg	https://plus.google.com/ +BarackObama	1647	4,840,295,000	7
94ee3e57563142e88697c2899f97c193.jpg	http://inhabitat.com/ tag/president-barack-obama/	902	2,178,200	11
d111781df6044a4eaef2bd81f7e6a176.jpg	http://www.billboard.com/ articles/videos/6745316/obama-thriller-michael-jackson	902	7,519,700	14
571d1c38285e40c0bf17707473117d95.jpg	https://www.whitehouse.gov/	367	3,607,500	10
eb8d9dce6642439bba61f8b6b0c3d129.jpg	http://www.boomerslife.org/ barack_obama_early_days_career_bio.htm	354	N/A	13
e89656e4b9f647e08c211742aa821bec.jpg	https://twitter.com/ thepresobama	236	732,947,000	15

The results of calculating each image's exposure (E_i) and image score ($E_i \times conf_i$) can be found in Table 5.2. The sum of the Image Score column divided by the amount of time it took to gather the images gives the Overall Exposure Score (S_c) of the candidate. A $S_c \approx$ **31,135,176,519.00** was yielded for candidate one. Looking at Table 5.2 it is easy to see that the main contributors to this extremely high score are the images associated to rows 7, 10, and 15 in Table 5.1. Even though these images were on the mid to low range of the listed VeriLook scores for the respective candidate, two out of three of the source websites are OSN's (Twitter and Google Plus) with the the remaining source being a popular main stream publication (Forbes). The very high amount of web traffic these three sources draw a month compensated for the low - but still positive - facial recognition scores giving a

justifiably larger Image Score for each of these results. It can be said that candidate one's S_c successfully sets the high benchmark for comparison as the facial recognition algorithm in use by the UCAN system performed well on the images gathered, 100% of the top 15 being positive matches and each had an associated source website with marginal to very high volumes of monthly hits.

Table 5.2: Candidate One Individual Image Scores

Associated Row in Table 5.1	Exposure of Image (E_i)	Image Score($E_i \times conf_i$)
1	3607500.00	263065664.10
2	3572812.50	260536186.50
3	3631788.46	264836824.20
4	3576595.19	84105871.32
5	3596182.69	83358391.00
6	730696.15	15670007.36
7	57912500.00	1236522363.00
8	8060953.85	130045857.00
9	323443.27	5218049.62
10	4561047212.00	58687849667.00
11	1968757.69	13873589.36
12	6579737.50	46366587.70
13	3295312.50	9448278.81
14	0.00	0.00
15	634281057.70	1169455700.00

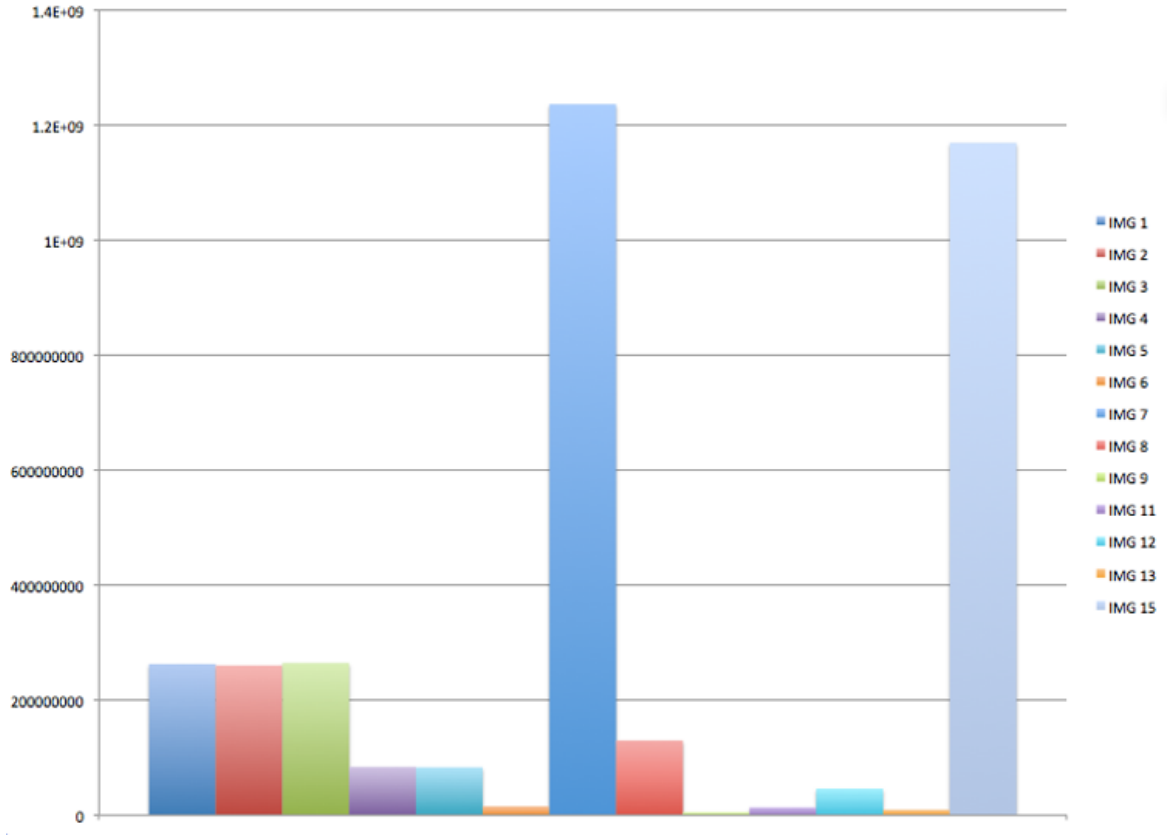

Figure 5.2: Table 5.2 Image Score Magnitudes Barring Outliers (10 and 14)

To get a better sense of the magnitude of each score in relation to one another, Figure 5.2 was created as a visual aid. It dismisses images associated to rows 10 and 14 of Table 5.1 as statistical outliers with row 10 being an image with a score in the tens of billions and row 14 being an image where no hits per month information could be found so its score was zero. Images 7 and 15 tower over the other scores due to the aforementioned reason of their source websites being very high traffic sites. The top three images in terms of biometric scoring all have relatively the same score and are perhaps not coincidentally the top three images of the Google Image search ran on the candidate. This could suggest that Google is using a similar approach algorithmically in terms of incorporating weight from facial recognition alongside Pagerank scoring into the ranking of results.

5.2 Candidate Two

Candidate Two was the assumed medium digital footprint candidate. This candidate was to be an individual who used one or more OSN's a day, preferably using a feature or outlet that involves the posting of images with some frequency. Elliott Iannello, the author of this thesis, was chosen as the medium candidate as he met this daily use criteria.

The search ran to generate the candidate's web crawler seeds was: Elliott Iannello. Figure 5.3 was the training image provided via the Search Request Form that was used by VeriLook to perform one-to-many matching on the 13,956 total images that were gathered by the web scraper and web crawler. Of the 13,956 images gathered, 3043 were associated to candidate two - this is 22% of the total search space.

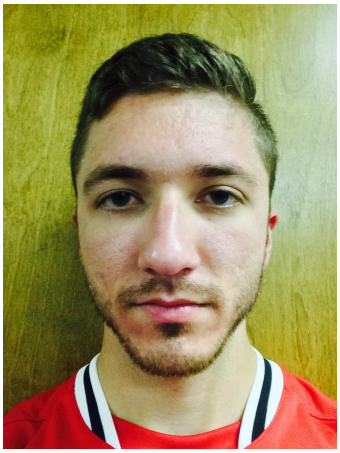

Figure 5.3: Candidate Two (Assumed Medium Exposure) - Elliott Iannello

Table 5.3 shows the top 15 results the system yielded based on the VeriLook Score. Candidate Two had every single result in the top 15 be well below the 128 threshold and each image was human-verified to be a negative match for Elliott Iannello. This was unexpected given the medium categorization. The SmallScrape.py results were manually checked and two images were found to be humanly-verified as matches for the candidate. Upon closer inspection, these images were well under the recommended 640x480 resolution being 200x200 and 100x100 respectively. Both images also had the head turned to showcase more of one side of the face than the other. These two factors combined were more than likely the reason VeriLook was unable to successfully identify a human face in either. Even still, the web

scraper was only able to create seeds from two images that had photographic relevance to the candidate with the remaining 98 not containing the candidate's likeness (human-verified).

Table 5.3: Candidate Two Results

File Name in Data Base	Source URL	VeriLook Score	Avg. Hits per Month Source Website	Google Results Rank
b8c139eff43a463ab97a664055455783.jpg	http://vibes.ng/wp-content/uploads/2014/08/michael-jackson-d.jpg	25	105,100	N/A
1925022b9a4e49daad00f22a235795e9.jpg	http://www.history.com/s3static/video-thumbnails/AETN-History_VMS/21/154/History_Becoming_Barack_Obama_SF_HD_still_624x352.jpg	25	4,578,500	N/A
bdc5ec2ec3c14234b09182c1d059887d.jpg	http://www.channelingerik.com/wp-content/uploads/2015/03/PeaceSignSilly.jpg	19	72,700	N/A
a76b96d2eaa0492e8f8278c48da4f61f.jpg	http://www.boomerslife.org/barack_obama-2.jpg	18	N/A	N/A
02bfa58faac84e78a5f46938ba96e4a5.jpg	http://www.boomerslife.org/barack_obama-2.jpg	18	N/A	N/A
7b3d2ff8796b4021acd2f6cd5e88085e.jpg	https://ronemy927charlotte.files.wordpress.com/2012/06/michael-jackson.jpg	18	40,887,000	N/A
8c6cf6f0face4a7191a13ffffa6736b4.jpg	http://images1.laweekly.com/imager/a-rabbi-remembers-his-relationship-with-mi/u/original/4873860/bad25.jpg	17	1,318,800	N/A
4c3c29da661745bb9a9ad6d4d4b111b9.jpg	http://upload.wikimedia.org/wikipedia/commons/e/e9/Official_portrait_of_Barack_Obama.jpg	17	41,160,000	N/A
3a0f8ac92ee643bb995fbaa7b58638cc.jpg	https://ordinaryevil.files.wordpress.com/2014/08/jackson-2s.jpg	16	40,887,000	N/A
571d1c38285e40c0bf17707473117d95.jpg	http://www.mjworld.net/wp-content/uploads/brunei-300x169.jpg	15	N/A	N/A
7687ae45f3f5440ebd4ee8414804a980.jpg	http://elitedaily.com/wp-content/uploads/2013/01/barack-obama-aged-elite-daily.jpg	15	3,466,900	N/A
18fdf2ea954646d397c9f60eb667c713.jpg	http://upload.wikimedia.org/wikipedia/commons/7/7d/Michael_Jackson_1988.jpg	15	41,160,000	N/A
85ddb54e7d4e4a6aa9fa0ecd39afc048.jpg	http://television.mxdwn.com/wp-content/uploads/2015/05/1431541464_the-catch-zoom.jpg	14	103,600	N/A
36d9dd59309a4e2d9c200d81cfbd8529.jpg	http://hdwallpapersfit.com/wp-content/uploads/2015/01/barack-obama-wallpapers.jpg	14	N/A	N/A
599f40cd834b43148c1885e9e50c042c.jpg	http://nyopoliticker.files.wordpress.com/2012/10/obama-smiling-getty.jpg	14	40,887,000	N/A

What VeriLook listed as the top 15 results for candidate two all pertain to candidate one, who had a verifiable larger and stronger presence within the testing/search space for VeriLook given the results show in Section 5.1. The top image scored a 25 from the facial recognition algorithm, meaning it was only about 19% confident to even begin considering it as a positive match. None of the results listed in the top 15 for candidate two came from

seeds generated by the search on this candidate - effectively meaning the Google Relevance (G_{rel}) for each image in Table 5.3 is zero.

The sum of the Image Score column in Table 5.4 is obviously zero thus defaulting the Overall Exposure Score (S_c) to be: **0.00**. This means that within the context and configuration of this experiment, Candidate Two has no online photographic exposure detectable by an automated agent.

Table 5.4: Candidate Two Individual Image Scores

Associated Row in Table 5.3	Exposure of Image (E_i)	Image Score($E_i \times conf_i$)
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	0.00	0.00
12	0.00	0.00
13	0.00	0.00
14	0.00	0.00
15	0.00	0.00

5.3 Candidate Three

Candidate Three was the assumed low digital footprint candidate due to infrequent use of online services. This candidate was ideally someone who had only one OSN profile and either avoided or was apathetic to posting images of them self online. Dr. Roy Nutter, a Professor in WVU's Lane Department of Computer Science and Electrical Engineering, was chosen as the ideal candidate for the low category as he admitted to infrequent use of a single OSN account (Facebook) and avoided sharing images of himself online.

The search ran to generate the candidate's web crawler seeds was: Roy Nutter. Figure 5.4 was the training image provided via the Search Request Form that was used by VeriLook to perform one-to-many matching on the 13,956 total images that were gathered by the web scraper and web crawler. Of the 13,956 images gathered, 3745 were associated to candidate three - this is 27% of the total search space.

Figure 5.4: Candidate Three (Assumed Low Exposure) - Roy Nutter

Table 5.5 shows the top 15 results for Candidate Three based on VeriLook Score. Candidate Three had every single result in the top 15 be well below the 128 threshold except the top result - which was a positive match scoring 237 and was human verified to be the second result listed in a Google Image search meaning it had an assigned G_{rel} of 2. Having just this one image be detected by an automated agent puts the score of candidate three, categorized as low usage, above that of candidate two's who was assumed to have medium online service usage. However poor the results may have been for candidates two and three, this does mean the system was able to provide an assessment that indicates a more accurate picture of canidate exposure than what was assumed.

Table 5.5: Candidate Three Results

File Name in Data Base	Source URL	VeriLook Score	Avg. Hits per Month Source Website	Google Results Rank
beef123ddba548e49c0c57726e012e76.jpg	http://wvutoday.wvu.edu/ n/2014/08/11/ increase-in-hacking-cases-brings-urgency- to-cybersecurity-says-wvu-expert	237	1,025,900	2
448730c5a90242ed8eb7e0148499b6f6.jpg	https://groundcontrolparenting.files.wordpress.com/ 2014/02/barack-obama-with-united-states-flag.jpg	30	40,887,000	N/A
3cadedba3c29404eb02fae4c913aa2a9.jpg	http://thewestsidestory.net/ wp-content/uploads/2014/12/Barack-Obama.jpg	29	27,900	N/A
2761d81c87804769a375412a4044e743.jpg	http://www.phootoscelebrities.com/ wp-content/uploads/2014/06/Barack-Obama-images-.jpg	29	N/A	N/A
e95c91917b884236be66519c9a57f85a.jpg	http://blackstonian.com/ info/wp-content/uploads/2013/06/Barack-Obama_81.jpg	28	N/A	N/A
01475df849c148deaead52f8a61c0076.jpg	http://blogs.r.ftdata.co.uk/ photo-diary/files/2013/05/obama.jpg	28	12,600	N/A
1a349077228d46cf9c1a7b81f3b2664b.jpg	http://quietmike.org/ wp-content/uploads/2014/03/BARACK-OBAMA.jpg	27	7,600	N/A
ca63577c63db4fd68bae0f8af6975209.jpg	http://www.slate.com/ content/dam/slate/blogs/behold/2015/05/ Southern%20Rites/7.jpg.CROP.promo-large.jpg	26	14,714,000	N/A
fc3d23cbb47f4ba2b3681b1465b9b4ed.jpg	http://www.lasportsanostra.com/ wp-content/uploads/2015/04/ daredevil-netflix-awesome-600x400.jpg	25	N/A	N/A
55411057a1d648f6a191792d3a386cdb.png	http://s1.firstpost.in/ wp-content/uploads/2014/07/ObamaAFP.jpg	25	300,400	N/A
9b342a19851d4ffb98062c643d401008.jpg	http://www.hdwallpapersinn.com/ wp-content/uploads/2014/07/michael-jackson-7.jpg	25	N/A	N/A
9c9f86c2b00141e18b3cd3108903b625.jpg	http://keri22.free.fr/ mjackson/Michael-Jackson.jpg	25	NA	N/A
c768003ec7ba44e6918c301ce0a808ef.jpg	http://www.hdwallpapersinn.com/ wp-content/uploads/2014/07/michael-jackson-6.jpg	24	N/A	N/A
4fbdab367fb64d54a4b9eee0e74624b4.jpg	http://cdn.hitfix.com/ photos/5553035/Michael-Jackson.jpg	24	1,072,300	N/A
d9777d7e9e45495fb66e5fcefa22c08c.jpg	http://www.mxdwn.com/ wp-content/uploads/2014/08/michael1.jpg	24	103,600	N/A

The singular image found to have be a positive match was farther examined and was found to be a work related image with a resolution of 500x281. This is still below the recommended 640x480 but is considerably higher than the images collected that were human-verified to be positive matches for candidate two. The face of candidate three in the matched photo is also postured very similarly to the training photo (Figure 5.4). After visiting the source website it can also be assumed that candidate three had very little control over the online publication of this image given it was included as media on an article that was run online by Dr. Nutter's employer, West Virginia University.

The Image Score column in Table 5.6 was summed and divided by the number of epochs the crawler was run, which was two, to yield an Overall Exposure Score (S_c) of: **940,626.69** for candidate three.

Table 5.6: Candidate Three Individual Image Scores

Associated Row in Table 5.5	Exposure of Image (E_i)	Image Score($E_i \times conf_i$)
1	1016035.58	1881253.37
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	0.00	0.00
12	0.00	0.00
13	0.00	0.00
14	0.00	0.00
15	0.00	0.00

5.4 Overall Results Analysis

Although some aspects of the data created by this experiment were mentioned in passing in the prior sections, Table 5.7 gives a quick glance at the pertinent information that is considered holistic of all three candidates.

Table 5.7: Overall Results and Averages

	Candidate One (High)	Candidate Two (Medium)	Candidate Three (Low)
Top 15 Face Recognition Positive Matches	15	0	1
Manually Identified Positive Matches that Appear in Scraper Results (first 100 results from Google Image Search)	100	2	8
Average Face Recognition Score	3199.67	17.33	47.07
Overall Candidate Exposure Score (S_c)	31,135,176,519.00	0.00	940,626.69
Normalized S_c	1	0	0.0000302
Total Images Associated	7168	3043	3745

The second row of Table 5.7 shows that VeriLook was unable to identify images that were verified to be a part of the test space. It was unable to match two images found for candidate two and seven images for candidate three. With regards to candidate one (Barack Obama), 22 images total were scored above the 128 threshold by VeriLook out of the 100 total that were human-verified to be candidate one (that's just 22%). Considering just the information shown in the top two rows of Table 5.7 and the additional seven images found for candidate one, the true acceptance rate of VeriLook 5.4 is at best 21%. A total of 23 images out of 13,956 scored above the threshold all together - meaning only 0.002% of the images collected were able to have the chosen automated agent successfully identify them as one of the candidates processed through the system.

Considering the total 13,956 images gathered over 24 hours, candidate one easily produced the majority. Figure 5.5 shows the breakdown by percentage of each candidate's image contribution to the database. Just as the Overall Candidate Exposure Score for candidate three was unexpectedly found to be higher than that of candidate two's, so was the distribu-

tion of the biometric search space in favor of candidate three - more images were generated by the scraper and crawler for the low candidate than the medium.

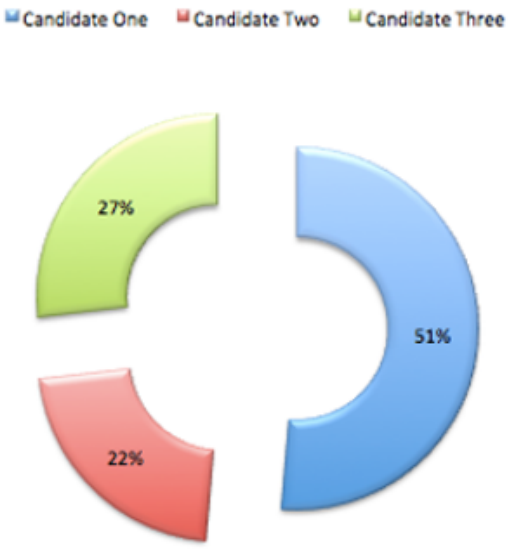

Figure 5.5: Distribution of Biometric Search Space by Candidate

Figure 5.5 is also a bit of a preamble for the comparison of each of the Overall Exposure Scores for each of the candidates; just as the search space was dominated by candidate one, so were these scores. Following equation 5.1, the scores were normalized with candidate one's score acting as the X_{max} and zero acting as the minimum (meaning absolute no exposure to an automated agent).

$$\frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{5.1}$$

The normalized values of the Overall Exposure Score (S_c) for candidates one, two, and three respectively are 1,0, and 0.0000302 - candidate's two and three are dwarfed by candidate one's exposure.

5.5 Observations

A few remarks with regard to some of the observed behaviors of the UCAN System. The SmallScrape.py web scraper pulled down way more relevant images than the web crawler in terms the ease of bio-metrically identifying them as a candidate. This was probably due in part to the fact the a web scraper by its very nature is utilizing another website's content and can take advantage of the algorithms already run on the data being displayed. It is also worth mentioning that there were duplicate images observed to be ascertained from two different websites, meaning that the web crawler potentially found a URL that the web scraper's website was using as source for the image.

As expected small, thumb nail size images didn't register with face rec even though they were human verifiable and were obtained since they were within the first 100 images of a Google Image search (hence the web scraper got them).

Social Media photos and photos pertaining to work and education were most prominently found. This means if a search term were to be expanded beyond a first and last name, incorporating a place of work or attended University using the boolean operator AND could help improve results by narrowing the search space to be more specific to a candidate.

Chapter 6

Conclusion

6.1 Summary

With regard to each of the hypothesises stated at the beginning of Chapter 5, the experiment showed the following

1. The overall exposure score for each candidate did not correlate to their respectively disclosed frequency of use of online services. The low candidate scored above the medium candidate with a score of 940,626.69 to 0.00.

2. The number of images successfully identified as a match by the facial recognition software for each respective candidate did not match each candidates pre-disclosed frequency of use of online services (i.e. high, medium, and low). No photos were found for the medium candidate, while one was matched for the low candidate.

3. The system configured for the experiment did act cohesively and did produce a total of 23 results where a candidate was successfully identified in one or more gathered online photos. This validates the UCAN system as both a potential threat and vetting tool for acting as an automated agent toward online photographic privacy.

The most notable conclusion though pertaining to the poor quality of the facial recognition results is that one-to-many facial recognition with VeriLook 5.4 grossly underperformed on the images gathered with a mere 21% at best true acceptance rate. The "Faces in the

Wild" data scenario undermined VeriLook's one-to-many matching abilities by providing too many images below the recommended resolution of 640x480 along with too many "noisy" images - images with more than one face or a lot of face-like shapes that throw off the algorithm. It can be said that it's specific implementation and use in this experiment is not ideal for rating exposure; simultaneously it can be said that it poses little biometric threat to photographic information online. This is subject to change as VeriLook is updated and its facial recognition capabilities enhanced. It is recommended that a training set of multiple images be used in place of one-to-many matching.

The web scraper and web crawler performed well enough, but are slow and do not utilize any type of discrimination techniques to predicatively evaluate image data before it is downloaded. Improving the scraper and crawler would help to restrict the search space for a facial recognition algorithm, potentially weeding out unwanted noise in the data and producing faster results.

The scores produced by Equation 4.1 did accurately reflect, within the limited context of the experiment detailed in Chapters 4 and 5, the exposure relation between candidates and proved that a candidate's assumption of their digital footprint size isn't always accurate. Candidate three was predicted to have the lowest score but ended up being rated higher than the medium candidate due a photo posted online that was related to their employment. Equation 4.1 needs more testing and application before a conclusion can be made with regard to its universal accuracy and the relational weight of scores between candidates deemed appropriate.

6.2 Threats to Validity

The biggest threat to the validity of this thesis is the breadth of scope encompassed by the UCAN system and its numerous online interactions. The full depth of each sub-system that comprises the whole was unexplored and simple, stable, and previously verified techniques were favored when building each piece that makes up the UCAN system. This made little room for variation in experimentation as the priority became building a full-thread, highly stable, product. Hence, only one technique was used for web crawling and only one facial recognition algorithm used for candidate identification (VeriLook 5.4's one-to-many).

There was also a heavy reliance on manual analysis and calculation. Images were human-verified to have been gathered by either the web scraper or crawler and all of the scoring calculations were done outside of the UCAN system by an analyst due to the lack of automation in retrieving a source website's hits per month data and the inability for a web scraper to provide the rank for a downloaded search engine result.

There is an inherent threat to validity within the nature of a web crawler and scraper. It is very difficult to repeat the results detailed in this thesis given the ever changing topography of the Internet. What a crawler/scraper will encounter over a given amount of time of activity is unpredictable and the size of the data it encounters makes it difficult to predict a speed at which it can gather images reliably. There is also a bias created by two factors: geographic location and robots.txt file restrictions. Since the web crawler is dependent on a search engine - in this case Google - for generating its seeds it hence follows that the search engine returns results tailored to the physical location of the requesting user. This means ranking and results change depending on where you are on the earth. The latter of the two factors, robots.txt files, are restrictions given to a crawler on a site-by-site basis that informs the crawler what it can and cannot touch. This prevents in some cases the automated retrieval of some images that may be viewable by a human user.

6.3 Future Work

Future research endeavors need to address the aforementioned threats to validity. There needs to be more variation in technique and algorithm with respect to each of UCAN's sub-systems. Giving the web crawler the ability to discern (discriminate) the relevance of an image given the textual information found on the page it resides could drastically reduce crawl times and constrain the test/search space for a facial recognition algorithm improving thus its results. More types of facial recognition algorithms need to be used to perform identification with specific consideration given to the use of a training set of images. Using alternative web crawlers and scrapers to iteratively build a training set for many-to-many matching could be a potential way to drastically increase a facial recognition algorithm's effectiveness over time. "Exposure" is a bit of an aqueous term within the context of the web and as of now there is no clearly defined set of variables that define what it means within the online space. The scoring algorithm defined in this thesis needs to be further explored and empirically verified through use with other search engines and metrics for measuring a website's traffic. There also needs to be special consideration to the weight given to images found on Online Social Networks. The topography of an OSN is different than a website designed solely for content delivery or application use. There is a willingness to share on the part of the user and the information they release has an awareness to it (depending on the user's privacy settings) given it is a node within a directed structure.

References

[1] R. Bodle, "The ethics of online anonymity or zuckerberg vs. moot," *ACM SIGCAS Computers and Society - Selected Papers from The Ninth International Conference on Computer Ethics: Philosophical Enquiry*, vol. 43, no. 1, pp. 22–35, 2013.

[2] L. Jin, H. Takabi, and J. B. Joshi, "Towards active detection of identity clone attacks on online social networks," *CODASPY'11*, pp. 27–38, 2011.

[3] J. Maheswaran and D. I. Wolinsky, "Crypto-book: An architecture for privacy preserving online identities," *HotNets-XII*, 2013.

[4] T. Bennouas and F. de Montgolfier, "Random web crawls," *WWW'07*, pp. 451–460, 2007.

[5] N. Fujimoto and K. Hagihara, "A personal system for web image retrieval," *WISICT'05*, pp. 209–216, 2005.

[6] I. Varlamis, N. Tsirakis, V. Poulopoulos, and P. Tsantilas, "An automatic wrapper generation process for large scale crawling of news websites," *PCI'14*, pp. 1–6, 2014.

[7] D. Poster, "Digital eye modification a countermeasure to automated face recognition," Master's thesis, West Virginia University, United States - West Virginia, 2015.

[8] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003. [Online]. Available: http://doi.acm.org/10.1145/954339.954342

[9] Opencv - about. Accessed: 2016-11-12. [Online]. Available: http://opencv.org/about.html

[10] Opencv - face detection using haar cascades. Accessed: 2015-10-13. [Online]. Available: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html

[11] Opencv - face recognition with opencv. Accessed: 2015-10-16. [Online]. Available: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

[12] Neurotechnology - verilook sdk. Accessed: 2016-11-15. [Online]. Available: http://www.neurotechnology.com/verilook.html

[13] N. Nuno Gomes de Andrade, A. Martin, and S. Monteleone, "All the better to see you with, my dear: Facial recognition and privacy in online social networks," *IEEE Security and Privacy*, vol. 11, no. 3, pp. 21–28, 2013.

[14] A. A. E. Masri and J. P. Sousa, "Limiting private data exposure in online transactions: A user-based online privacy assurance model," *International Conference on Computational Science and Engineering, 2009. CSE '09*, 2009.

[15] K. Xu and V. Li, "Privacy exposure of online social search," *GLOBECOM 2010*, 2010.

[16] Y. Wang, R. K. Nepali, and J. Nikolai, "Social network privacy measurement and simulation," *ICNC'14*, 2014.

[17] P. D. Giang, L. X. Hung, and R. A. Shaikh, "A trust-based approach to control privacy exposure in ubiquitous computing environments," *IEEE International Conference on Pervasive Services*, 2007.

[18] S. Anthony. (2014) Facebook's facial recognition software is now as accurate as the human brain, but what now? Accessed: 2014-11-16. [Online]. Available: https://www.extremetech.com/extreme/\178777-facebooks-facial-recognition-software-is-\now-as-accurate-as-the-human-brain-but-what-now

[19] J. Wolen, "Impact assessment of facial recognition algorithms' performance when modifying nose dimensions," Master's thesis, West Virginia University, United States - West Virginia, 2015.

[20] I. Rogers. Pagerank explained - the google pagerank algorithm and how it works. Accessed: 2016-11-18. [Online]. Available: http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm

[21] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. Marseille, France: Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Oct. 2008. [Online]. Available: https://hal.inria.fr/inria-00321923

Appendix A

Meta Data

A.1 Web Crawl Logs

Candidate One - Barack Obama

```
=========================================================
Search Query String: Barack Obama
Time search run: 2015-05-20 10:30:58.67
=========================================================
        Title: <b>Barack Obama</b>
        URL: https://www.barackobama.com/
        -------------------------------------------------
        Title: <b>Barack Obama</b> - Wikipedia, the free encyclopedia
        URL: http://en.wikipedia.org/wiki/Barack_Obama
        -------------------------------------------------
        Title: <b>Barack Obama</b> (@<b>BarackObama</b>) | Twitter
        URL: https://twitter.com/barackobama
        -------------------------------------------------
        Title: <b>Barack Obama</b> - Biography - U.S. Representative, U.S. President
<b>...</b>
        URL: http://www.biography.com/people/barack-obama-12782369
        -------------------------------------------------
=========================================================
```

Candidate Two - Elliott Iannello

```
============================================================
Search Query String: Elliott Iannello
Time search run: 2015-05-21 13:32:04.574
============================================================
        Title: <b>Elliott Vincent Iannello</b> | Free Listening on SoundCloud
        URL: https://soundcloud.com/elliott-vincent-iannello
        ------------------------------------------------------
        Title: Hemingway | Facebook
        URL: https://www.facebook.com/hemingwayva
        ------------------------------------------------------
        Title: A Story Told | Facebook
        URL: https://www.facebook.com/astorytoldband
        ------------------------------------------------------
        Title: <b>Vincent Iannello</b> | Facebook
        URL: https://www.facebook.com/vincent.iannello
        ------------------------------------------------------
============================================================
```

Candidate Three - Roy Nutter

```
============================================================
Search Query String: Roy Nutter
Time search run: 2015-05-21 13:31:54.147
============================================================
        Title: AC Milan <b>nutter</b> Gennaro Gattuso charged as Spurs <b>...</b> - Daily
Mail
        URL: http://www.dailymail.co.uk/sport/football/article-1357861/Gennaro-Gattuso-
charged-headbutt-Joe-Jordan.html
        ------------------------------------------------------
        Title: Tottenham coach Joe Jordan blasts Milan <b>nutter</b> <b>...</b> - Daily
Mail
        URL: http://www.dailymail.co.uk/sport/football/article-1358138/Spurs-coach-Jordan-
blasts-Gattuso-racism-claim.html
        ------------------------------------------------------
        Title: Major success for the <b>nutter</b> with a putter | Daily Mail Online
        URL: http://www.dailymail.co.uk/sport/football/article-437702/Major-success-
nutter-putter.html
        ------------------------------------------------------
        Title: 43 Arrest Warrants Executed In Year-Long Crystal Meth Ring <b>...</b>
        URL: http://5newsonline.com/2015/02/13/43-suspects-arrested-in-year-long-crystal-
meth-ring-investigation/
        ------------------------------------------------------
============================================================
```

The system is too large to enclose all of the source code economically. The Full Source Code is available upon Request.