Identifying a Critical Threat to Privacy through Automatic Image Classification*

David Lorenzi
Rutgers University
1 Washington Park
Newark, NJ 07102
dlorenzi@cimic.rutgers.edu

Jaideep Vaidya
Rutgers University
1 Washington Park
Newark, NJ 07102
jsvaidya@business.rutgers.edu

ABSTRACT

Image classification, in general, is considered a hard problem, though it is necessary for many useful applications such as automatic target recognition. Indeed, no general methods exist that can work in varying scenarios and still achieve good performance across the board. In this paper, we actually identify a very interesting problem, where image classification is dangerously easy. We look at the problem of image classification, in the specific context of accurately classifying images containing highly sensitive data such as drivers licenses, credit cards and passports. Our key contribution is to build a Hierarchical Temporal Memory (HTM) network that is able to classify many sensitive images with over 90% accuracy, and use this to develop a system to automatically derive and transcribe sensitive information from image data. Our system classifies images into two groups – sensitive and non-sensitive. The group of sensitive images can then be further analyzed. This is a real world security issue that could easily lead to privacy problems such as identity theft, since scans of passports and drivers licenses are routinely emailed or kept in digital form, and many local documents are left unencrypted. Essentially, an attacker can use data mining and machine learning techniques very effectively to breach individual privacy. Thus, our main contribution is to demonstrate the efficacy of image classification for deriving sensitive information, which could also serve as a guide for other interesting applications such as document detection and analysis. Thus, it also serves as a warning against leaving data unencrypted and again proves that security through obscurity is simply not enough.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Object Recognition*; K.4.4 [**Computers and Society**]: Electronic Commerce—*security*; I.5.1 [**Pattern Recognition**]: Models—*Neural nets*

*This work is supported in part by the National Science Foundation under Grant No. CNS-0746943.

CODASPY'11, February 21–23, 2011, San Antonio, Texas, USA.

General Terms

Security

Keywords

Privacy, Image Classification, Neural Networks

1. INTRODUCTION

Image classification, in its various forms, has a wide variety of useful applications. For example, automatic target recognition[2] can be considered one of the most sought after military goals in image exploitation. However, no general methods exist that can work in varying scenarios and still achieve good performance - for target recognition, this is due to the ambiguity in defining "target". Even in general, image classification can easily be considered to be a hard problem. Therefore, a lot of research has been done to achieve good performance on specific applications, and in limited contexts[9]. In this paper, we actually identify a very interesting sub-problem for which image classification turns out to be dangerously easy. This is interesting since in general, the fact that classification works well is a cause for celebration. However, in this particular case, this fact can be used in a rather destructive fashion. Specifically, the application we consider is that of accurately classifying images containing highly sensitive data such as drivers licenses, credit cards, and passports. From the perspective of an attacker, being able to accurately classify and extract data from such images could easily enable fraud such as identity theft, or even worse. Thus, a vast potential for a breach of privacy exists through this route.

One may question, whether this problem is real. However, this can be easily demonstrated from the following scenario of email interception / trojan attack. The ubiquity of email cannot be disputed today. Almost everyone has access to some form of email, either through their work, school, or even free email accounts widely available on the web. Data privacy becomes an important issue when personal data is routinely disclosed via email. People take the information they are sending to each other through email for granted, because most casual users of computers and email are ignorant of computer security issues. It is not difficult for an unscrupulous individual to intercept unencrypted email messages and read them, nor is it difficult to gain access to someone's inbox and comb through it, garnering sensitive information in the process. Even if you consider such interception to be difficult, more worryingly, in many cases, individuals now scan and send digital copies of documents

(as TIFF images) to their colleagues or to officials. It is quite easy for people to become victims of trojan horse programs that are emailed to them. If an unscrupulous individual crafts an email mimicking the traits of a legitimate email and tricks the user into running a file that compromises their computer, the attacker now gets complete freedom of access, and is able to scan the local hard disk or other media for the sensitive images stored on it. Though this may be like searching for a needle in a haystack, when, as this paper shows, accurate and efficient detection and classification techniques exist, this becomes a huge problem. Other possible attack channels also include social networking and public file sharing (e.g., imageshack) websites, since sensitive images could automatically be shared without the user's knowledge.

Indeed, our main contribution in this paper is the observation that an attacker can apply data mining and machine learning techniques to detect and acquire sensitive data from images. This is a real world security issue, and shows that security through obscurity is simply not enough. In fact, we build a Hierarchical Temporal Memory (HTM) network that is able to classify many sensitive images with over 90% accuracy, and use this to develop a system to automatically derive sensitive information from image data. While the primary focus is on the ability to detect and classify images, not much more additional work is necessary for the attacker to derive sensitive textual information.

As stated above, the main classification technique used are Hierarchical Temporal Memory (HTM) Networks – a type of neural networks that are built according to biology and are well suited to image classification tasks. Our trained HTM network is able to classify many sensitive images with over 90% accuracy. As such, we identify a critical security and privacy problem since scanned images of driver's licenses and government issued passports can be picked up via the HTM, and the data contained within the image can be read and converted into textual form with an Optical Character Recognition program, ready to be cataloged in a database. While the paper primarily focuses on a proposed "attack" on a workstation using the trained HTM, we also discuss the security issues involved with an adversary using variations on the proposed method to attack on local computer systems, file servers, or the attacker building an identity theft network via trojan horses and botnets.

Thus, our contribution is two-fold. First, our system demonstrates the efficacy of image classification for deriving sensitive information, which could also serve as a guide for other interesting applications such as document detection and analysis. Secondly, it serves as a warning against unencrypted data and again proves that security through obscurity is simply not sufficient. The rest of the paper is organized as follows. Section 2 discusses Hierarchical Temporal Memory Networks, since the paper actively uses this technique. Section 3 looks at the related work. Section 4 presents the overall attack system that would be used by the attacker, and discusses the possible variants. Section 5 examines the actual images of interest, and looks at possible strong and weak features for identification. Section 6 presents the experimental analysis of the system. Finally, Section 7 concludes the paper and looks at future work.

2. PRELIMINARIES

In this section, we first discuss what are Hierarchical Temporal Memory (HTM) Networks[7], then present the Numenta Toolkit[11], which is a freely available implementation that we use for building the actual network.

2.1 Numenta HTM's and Image Recognition

Since our problem is to differentiate between "sensitive" images and "non-sensitive" images, this can be considered as a standard classification problem. As such, once we decide how to generate / extract features from the underlying images, we could easily utilize any of the standard classification algorithms such as decision trees or k-nearest neighbor classification, etc. Indeed, since neural networks are also one of the most used techniques for this process[16, 4], these are quite suitable for our purpose as well.

In our case, we are particularly interested in the ability to generate a classifier that works like a human brain, for the express purpose of distinguishing certain types of images from each other. The recently developed Hierarchical Temporal Memory (HTM) actually fits the bill quite well. HTM networks are actually a type of neural network that try to approximate the vision cortex. The formalized mathematical model underlying this is based on the Memory-Prediction framework[10] developed by Jeff Hawkins.

An HTM network created for vision tasks is identical to the high level structure of a mammalian visual cortex. The network receives an image as input, performs a set of preprocessing operations on it, and passes the result through multiple levels of processing. As the image passes through each level, the HTM builds successively more abstract hierarchical representations, with the highest level representing global image properties and shape. These various levels of representations allow the network to be invariant to small changes in the input and increase the robustness of the system overall. For categorization tasks, these high-level representations are fed through a supervised classifier at the top of the network. The overall system performs static inference, that is, there is a single upward pass through the hierarchy. For example, Figure 1 shows an example HTM network for vision. The sensor (bottom level) receives the image, which is processed by the levels above. The selected node at level 4 (green) receives input from the blue nodes below it.

The Numenta Toolkit implements the HTM idea and makes it freely available for academic research. Therefore, we use this implementation, though other implementations could equally easily be used. However, unlike a full-fledged HTM network, the academic version used here for vision tasks has no feedback connections, temporal inference, or attention mechanisms. The version of the vision toolkit used in the experiments discussed in the paper operates on 200x200 pixel grayscale images. Larger images are down sampled to this resolution and converted to grayscale before they are fed into the HTM for classification, categorization etc. [11].

The structure of the hierarchy is very important in the case of recognizing "sensitive" images like passports and drivers licenses, and as such, lends itself to accomplishing the task of image recognition very well. For our task, we are looking for image data that is very regular in its structure, that is, it follows a set of rules/guidelines for its structure and does not deviate much from the prescribed format between instances. For example, a U.S. passport has a head and shoulders picture of the individual it belongs to on the upper left side, a machine readable code that is two lines long that extends the length of the bottom of the passport, and

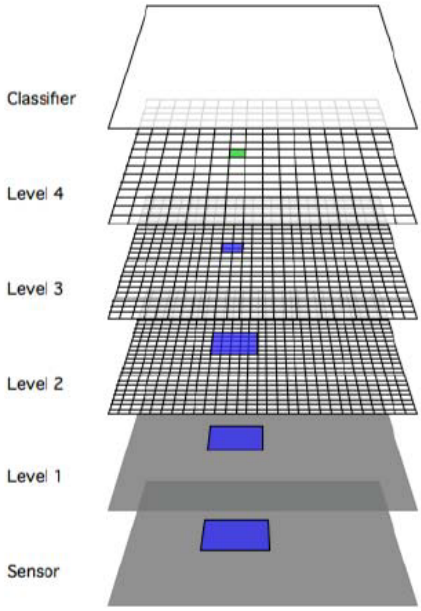


Figure 1: Example HTM Network for vision

a block of text containing the individual's information that is left justified bordering the headshot photograph. These are features that are inherent to all passports and vary negligibly between individual passports. Where the input data varies between images, things like: the textual data itself, personal facial features, different state seals etc. play a role in making the image unique, but do not affect the global structure of the image in a meaningful way. Due to this, the image can still be positively identified as a driver's license or a passport with a high degree of accuracy and few false positives.

2.2 HTM Network Implementation and Training

The HTM is useless to us unless we give it image data to train and test on, in order to understand how it is learning and to test if what we are teaching it is being used in an effective/efficient manner. Without restating the entire paper dedicated to the design and operation of HTM's, we need to understand the course that the input takes to generate the output.

The process of training an HTM model with spatio-temporal data is the process of knowing the state of the coincidence patterns and Markov-chains in each node at every level of the hierarchy. Although algorithms of varying levels of sophistication can be used to learn the states of an HTM node, the basic process can be understood using two operations, the first being memorization of coincidence patterns, and the second being a learning of a mixture of Markov chains over the space of coincidence patterns[7]. In the case of a simplified generative model, an HTM node remembers all the coincidence patterns that are generated by the generative model. In real world cases, where it is not possible to store all coincidences encountered during learning, storing a fixed number of a random selection of the coincidence patterns is sufficient as long as multiple coincidence patterns are allowed to be active at the same time[7].

The coincidence patterns enable the image identification algorithm to account for the inherent differences between each unique image, while still classifying it correctly. A 4 layer network, as shown in Figure 1 is sufficient for most image classification tasks, and additional layers do not significantly improve the classification[11]. The HTM networks are trained in a level-by-level manner, starting with the coincidence patterns and Markov chains at the first level and then moving up the hierarchy.

The specific network used is specialized to grayscale images. In this network, the first level of coincidences are replaced with Gabor filters of different orientations. At all levels, the coincidence patterns were restricted to have spatial receptive fields smaller than that of the Markov chains. In our paper, we build upon this replacement network, utilizing it to process images of driver's licenses, passports and any other images of interest we train for (e.g credit cards, student id's, social security cards) and discern them from other common images. This downward resolution resampling turns out to be helpful when processing a variety of images from different sources for the fact that no matter what resolution the image is fed into the model at, it will be shrunk down to a standardized size and then grayscaled by the preprocessor.

3. RELATED WORK

Within image classification, automatic target recognition[2] has been well studied over many years. Neural networks for automated target recognition have been experimented with for quite a while[14]. Different conceptual ideas we see implemented in one form or another in the HTM for vision tasks can be found in other papers published, like a coarse-to-fine strategy for multiclass shape detection[1], or using a Bayesian based hierarchical approach for target recognition[15]. As of late we have seen work specifically in the area of unsupervised learning of invariant feature hierarchies with applications to object recognition[13], and finally even direct work with HTM's via Content-based image retrieval of architectural drawings[3, 5]. One point that we would like to emphasize is the blending of disciplines involved in the formulation of the process proposed in this paper. We demonstrate the combination of techniques and research from a multitude of areas of study, especially in the areas of computer vision, image processing and machine learning. The paper also borrows ideas and concepts from the computer security community, using classifiers in intrusion detection systems for wireless networks[17] and using neural networks to detect system anomalies and system abuse[8]. Essentially, this paper really highlights the security issues inherent in digitizing personal information, specifically from the image processing sense, and shows how the power of classifiers can be utilized in a destructive way to breach privacy and security. We believe that this particular application of classifiers to security issues is a unique and original contribution to each of the communities respectively, and worthy of exploring as a legitimate new security threat.

4. ATTACK SYSTEM & EXPERIMENTAL DATA

Several types of image data can actually have sensitive personal information. As such, a wide variety of data was chosen to test the capabilities of HTM based neural net-

works, such as social security cards, passports, drivers' licenses, student IDs, and credit cards. These were selected due to the high probability that they would be stored on a computer in some form, as either strings of text, or as a scan in an image file. Disturbingly enough, a majority of this data has become searchable online, and all datasets for the experiments in this paper were garnered and compiled from a Microsoft Bing image search. As such, this paper also further illustrates the lack of importance people place on information security and ease of theft and resulting abuse that exists with this type of data.

4.1 Attack Algorithm Overview

The proposed attack begins with training and testing HTM neural networks to seek out and positively identify "sensitive images". These networks operate in a binary manner, determining whether a particular image can be classified as an instance of a particular type of "sensitive" image. Figure 2 gives an overview of the entire attack timeline. We discuss the details below.

4.1.1 Pre-Attack

The pre-attack phase consists of building the HTM networks for the types of data that are to be discovered and identified on the target machine. This requires the attacker to find training and testing images of the requisite type, then using them to build the relevant HTM network. The attacker must then package the HTM networks together, as each of these networks must run in parallel, because each image on the victim's machine needs to be run through each type of HTM to determine if it meets the specified criterion. The complete HTM Networks themselves are very small in size, approximately 2MB each. If an attacker was to launch the attack described in the paper, it would be approximately 10MB worth of data (for the 5 classification networks), as well as the space required for the actual classifier itself (which may vary based on how stripped down it is).

4.1.2 Attack

The attack phase begins with a gathering of images by a regex expression in a script or batch file that simply fetches each of a relevant type of image file format (e.g *.gif or *.jpg). These images are then fed into the HTM networks and processed. Positive hits are identified and pulled. Repeat these steps until every type of image file format desired has been searched or until the information desired is acquired.

When the attack is initiated, the attacker might want to run the HTM network with both the highest number of predicted total hits and probability of hits first. However, as positive identifications are made, the sample space of potential images shrinks accordingly. There exists the fact that images which are false positives of one group could in fact be an instance of another sensitive image category that are now removed from the sample space. If this is an issue, the attacker can simply rerun every image mutually exclusive of the outcomes from a particular HTM network and have no "filter down heuristic" running. If intrusion detection systems are not expected to be encountered, running the HTM's mutually exclusive will produce the most information, but result in the greatest use of system resources. The HTM's and regex expressions can be modified so as to not trigger some types of intrusion detection systems that monitor CPU cycle use, memory usage, and disk I/O usage, by throttling the number of images the HTM is processing, or limiting the scope of the image search. The Trojan can even go so far as to attempt to fake a valid signature certificate signature to fool the operating system into thinking it is a legitimate signed process, if the process is not entirely stealth from the beginning. It is up to the attacker to determine this information and adjust accordingly before implementing the attack.

4.1.3 Post Processing & Database Creation

In the post processing step, an optical character recognition (OCR) program can be used to improve accuracy. The OCR program can translate the data garnered into text strings, and place them into a database, along with the original image from which they were extracted. For example, if an image of a credit card is targeted by the HTM, and is then found on the target system and selected for export to the OCR program, the number from the card can then be translated from the image into a text string and stored in a database along with the person's name on the card and any other relevant data needed to use the card. All of this can be done with off the shelf, open source programs that are widely available to the public for Windows or Unix.

4.2 Scope: Wide Area Attack vs. Personalized Attack

The type and style of attack on the target depends on how much prior knowledge the attacker has while compiling the attack package. For example, if the attacker knows his target is one person with an image of a passport on his computer, he can tailor the HTM package and the wordlists to only go after passport images and word search for passports. This makes the attack stealthier because the delivery package is smaller in terms of bytes and utilizes less system resources than a larger more generalized attack looking for more types of data. It also greatly decreases the time required for a successful attack, as fewer classifiers need to run.

On the other hand, attacking large, centralized stores of personal information can prove more fruitful for the attacker, as attacking something like a DMV would provide not only an implicit guarantee of "sensitive data" e.g. drivers license number, but also potentially other relevant information about the subject (age, sex, eye color, hair color, home address, etc) at which point an identity thief could begin scanning social networks for enough information to build an accurate personal profile and steal your identity. A company's HR database is also a prime target for attack, as it contains important information such as SSN's, bank account numbers (if the individual has direct deposit), home address, salary earned(only target employees' with a high salary), at which point the attacker has enough information to scan social networks for pictures of the individual, create a false driver's license with the attackers face allowing them to execute identity theft.

The true danger lies in that an attacker only needs certain pieces of the puzzle to be able to make highly probable guesses and/or turn to the Internet's social networks to flesh out the remaining information.

4.3 Styles of Attack

There are many methods by which an attacker could pack-

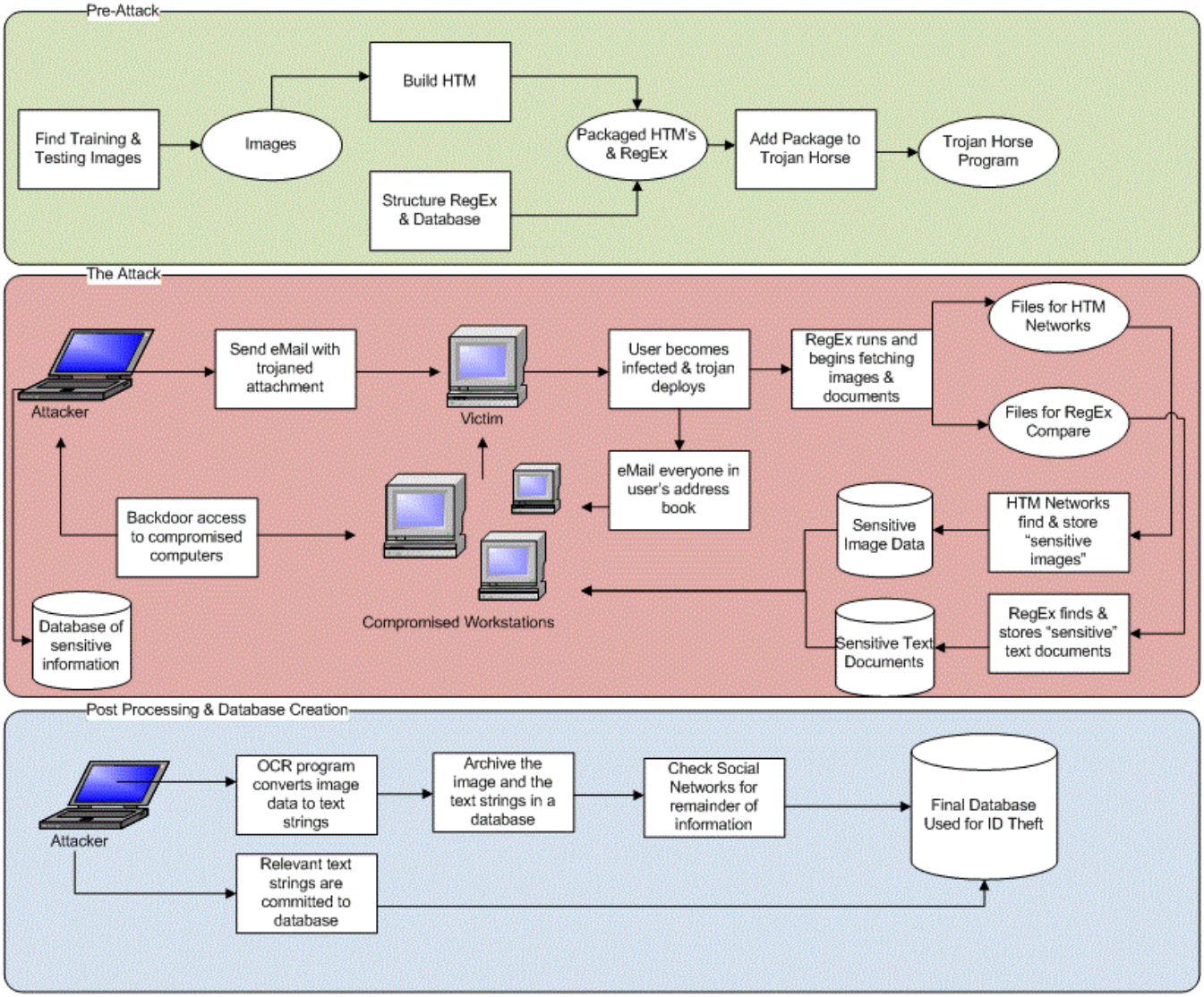


Figure 2: Flowchart for attack with HTM Networks

age and deploy these programs to garner information. One possibility, as described earlier, involves a Trojan horse program, in which a user downloads and runs a false program that actually contains the expressions and HTM's that run under the cover of the process of the fake program. This is the most effective method, because once a machine is successfully compromised, it is not much more work to install a backdoor at which point the machine can be added to a global botnet where its system resources can be utilized and commanded from a centralized attack server. The other advantage to this method is each of the other infected computers can act as intermediary drops points for the sensitive information, so that the loss of any one machine does not affect the ID theft network as a whole. This also helps to avoid the traditional centralized drop off server, which would be a target for other identity thieves as well as law enforcement and perhaps even some of the more sophisticated users that could backtrack from an infection. The best and most traditional method of distribution is via email using links to redirect someone to a compromised site or embedding the trojan in some form of attachment (word processor document, spreadsheet, etc).

The Trojan can then have a worm component and begin emailing everyone in the infected machine's address book an infected file at which point they are added to the botnet and their machine is scanned for sensitive images. This attack very similar to other older, more traditional Trojan/worm combinations, however now the attacker has "intelligent agents" (the HTM's & regex) acting on his behalf to find sensitive information instead of having to comb through datafiles/images manually. This is essentially a sophisticated automation of such an attack, essentially a "fire and forget" Trojan. It is really up to the attacker to determine the best infection vectors, as these will be adapted according to the target and the constraints of the systems. The attacker can then periodically check the drop-off points for information to complete a profile for an identity theft.

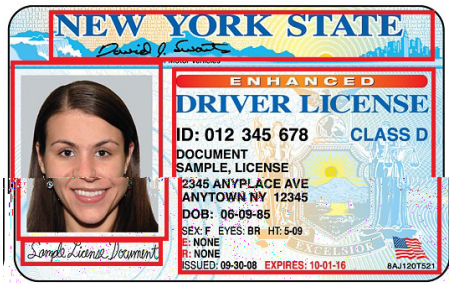


Figure 3: "Strong Features" of NY Drivers License

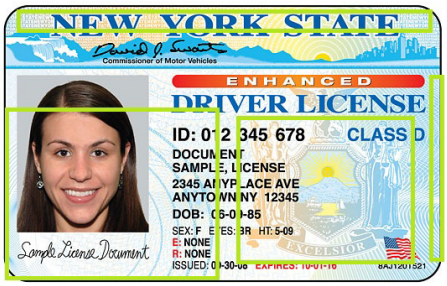


Figure 4: "Weak Features" of NY Drivers License

5. IMAGES OF INTEREST

We now examine the specific types of images of interest, from which sensitive information can be extracted. We discuss possible features of interest that might be useful in accurately classifying such images, though the HTM network may actually be finding and using other features. First, we look at Drivers licenses, followed by U.S. Passports. Other data types will also be discussed, though not at length, since it would be relatively redundant.

5.1 Drivers License

We first train HTMs to detect images of drivers' licenses. Both real and fake images are used to train and test the HTM's for the attack methods for finding and identifying "sensitive" images described earlier in the paper. There are two primary reasons for this. First, while the original goal was to use U.S. passports for the experiments, it was challenging to find a sufficient number of images of real passports for training and testing. Currently, there are simply not enough scans of passports available on the web, whereas there are plenty of scans and mockups of drivers licenses images that are freely available for gathering off of your search engine of choice. Additionally, it is more challenging to build a good classifier for classifying drivers licenses, since these show a wide degree of variability across states (each state has its own format for a drivers license). On the other hand, since the US Passport has a fixed format (and does not vary by state), it should be easier to detect it.

Privacy is also a concern when conducting experiments such as these that deal with sensitive data, so only images that were indexed in a search engine were considered for use. Microsoft's Bing search engine was used to gather the sets of drivers license images used in the training and testing of the HTM's for image recognition in our experiments. We now give examples of the strong and weak features typically found in the average drivers' license in the United States, using a new proposed drivers license to demonstrate these features.

5.1.1 Strong Features of Drivers License

In Figure 3, we see prominent features in a regular pattern that can easily be picked up by edge detection algorithms or other image recognition filters. The left justified headshot along with a handwritten signature make up the prominent left side features with the personal information left justified a few millimeters away from the headshot. The personal information is in a regular typeface that can be easily read by an OCR program and takes up a majority of the space of the license. In the lower right hand corner there is a prominently featured American flag with some typeface characters underneath. Finally at the top the name of the state is in large font that can be OCR'ed. This image is meant to demonstrate traits that are common amongst all drivers licenses that will be picked up by the HTM when the image is processed.

5.1.2 Weak Features of Drivers License

This particular drivers license is very advanced from an anti-counterfeiting perspective, and thus, scanning it with an edge detection algorithm or applying a filter to it can yield some interesting results. Figure 4 shows some of the "weak features" of the drivers license. There are concentric ring designs that intersect with headshot picture on the left side of the license. There is a large state symbol watermark that features text and bold edges. There is a strong possibility that an OCR may pick up on the watermark text, depending on the thresholds set for detection upon scanning of the image. There is also very small "New York" printed repeatedly along the top length of the license, again perhaps causing some OCR programs some grief (although, the garbage text could easily be filtered out with some extra work). However, none of these features will hinder overall detection by the HTM (due to the hierarchy and the Gabor filter), but it will make the extraction of data step more difficult, and perhaps produce some foiled attempts when large scale image dumps are scanned and individual licenses cannot be checked by the attacker.

5.2 United States Passport

We are not concerned with all of the information on the passport, merely its "defining features"; more specifically, we are just concerned with identifying whether or not the image we are looking at is a U.S. passport based on these features.

5.2.1 Strong Features of U.S. Passport

Figure 5 illustrates these "strong features". The term "strong features" is used because these are the objects that will most likely be picked up by any edge detection algorithm. These features include the "unique" information of the individual's passport, the machine readable code at the bottom of the page, the facial features of the person, the three red nautical stars on the left edge of the photograph, and the letters printed at the top of the page. Most font typefaces are regular in their structure and edges, so it is easy to detect them and use optical character recognition methods to translate them from an image to textual data. Another important distinction is the location and layout of the passport. Specific information will always be in the same

Figure 5: "Strong Features" of U.S. Passport

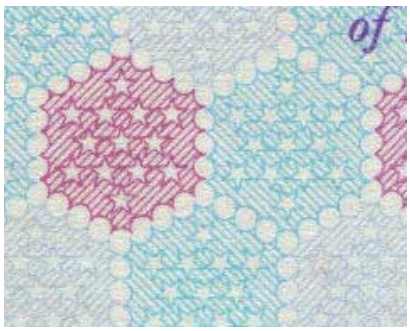


Figure 6: Background Pattern of U.S. Passport

location, so looking for it is relatively easy and systematic, however, it can become difficult based on the number of pixels in the scanned image (e.g. the image resolution may be higher or lower).

5.2.2 *U.S. Passport Background Pattern*

The background pattern (shown in Figure 6) consists of a mesh of hexagons with five circles for edges, filled with 13 five sided stars and a diagonal line pattern. The red hexagons have lines slanting to the left, the blue hexagons have lines slanting to the right, and the white hexagons have lines slanting left. Some of the red hexagons will become a blend of red and blue toward the middle of the document, but the same rules as normal red hexagons apply. This pattern is a blend of strong and weak features.

5.2.3 *Weak Features of U.S. Passport*

The term "weak features" refers to features that are present in the passport, but are not as bold or straightforward as the strong features. These features are mainly designed to prevent counterfeiting, tampering, and alterations to the documents. The U.S. State department put these features into the document deliberately, and in our case, they do cause some interference with the strong features, making it more difficult to identify and extract the features of interest. These designs are very minute and intricate. For example, the blue waves placed over the image of the individual on the passport, obscuring the facial features and breaking up the lines of the face. The watermark of an eagle surrounded by a wreath of stars that overlaps the unique personal data and the image of the face, as well as the watermark wavy texts stating "The United States of America" and the three "seals" in the center right of the document. Our goal is to be able to stratify the strong features from the weak features, because both are vital for positive identification, yet both can interfere with the detection of each other. This can be achieved via image processing techniques, as discussed later.

Figure 7: "Weak Features" of U.S. Passport

6. EXPERIMENTAL ANALYSIS

We now discuss the specific experiments conducted to create and evaluate the classifier. As discussed earlier, while any image classification algorithm could be used, we actually train a neural network for classification. In specific, we use the Numenta software[1] to train a HTM Network for vision tasks designed to sort images into one of two bins, either sensitive images (bin1) or not sensitive images (bin2). The goal was to see at what point, in terms of number of training and testing images required, we could feasibly sort between the two with around 90% rate of success. Four experimental groups were created and named accordingly. As a control, the images from the experiment "puppies" included in the Vision Toolkit by Numenta were used to represent non-drivers license pictures in all four experiments. The images that were contained in the control group feature pictures such as that of landscapes and family photos, typical of common images found on users' systems.

6.1 Experimental Images

The images that were used to conduct these experiments are categorized into different groups based on what type of information is contained in the image. The groups are: documents with plain text, pure pictures, documents with pictures and plain text. Samples of these are shown in Figure 8. The networks were trained and tested by splitting the images into 4 groups, testing and training of drivers licenses, and testing and training of not drivers licenses. The drivers license group consists of documents with pictures and plain text, consistent with the layout and composition of a drivers license. The not drivers license group features pure images, containing no text of any kind, only objects, people or landscapes. The same experimental structure is used for all subsequent data styles (credit cards, social security cards, student ID's, passports). The deliberate focus on drivers' licenses is due to the fact that they have the highest rate of success amongst all the data types, and serve as a great example of structured images and textual data.

In each experiment, they were split 50/50 at random into training and testing groups from 1 large group of 197 images, with the extra image always going to the training group. The

[1]Numenta can be downloaded from http://www.numenta.com/vision/vision-toolkit.php.

Figure 8: Sample Images

(a) Image with Text

(b) Pure Image – Family Vacation

(c) Pure Image – Landscape

images used in the experiments were pulled from a common web search for drivers licenses. This is important for the fact that the pool of images contains drivers license from many different states, and some of the images are completely computer generated, while others are scans of real drivers licenses or are fake licenses altogether. The image quality, file format, compression algorithm used on the image, and pixel count varies wildly between each license. This is a benefit as it is indicative of real world possibilities encountered when searching for potential scans of "sensitive" information. There is no guarantee that it will be in a specific format or size. The specific details for each of the four datasets are given in Table 1 [2].

6.2 HTM Training and Test Results

Figures 9-12 shows the results from the experimental trials of each group of images. Each of the four sets was run through 4 tests, one train & test which trains the network on the training images, and then checks its accuracy on the test images. This was performed again with the training options turned on, these options include additional training to handle shifts, size changes, mirroring, and small rotations. Finally, two optimization runs were conducted, one with the training options on and one with the training options off. Optimization finds the best set of parameters for the network based on the features found in training images, and then tests the optimized network on the test images for accuracy. Table 2 gives the detailed set of system parameters used in each different run. The same set of parameters are used in each corresponding run over the different datasets. The networks were created by an Intel Core 2 Duo 2.26Ghz with 4 gigabytes of memory, and the times involved in creating each network are included in the tables.

[2]The actual compiled datasets can be found at http://cimic.rutgers.edu/~dlorenzi.

Table 1: Datasets(# of img in each category)

(a) Drivers License Image Data

Group	Category	Training	Testing
DL25	Drivers License	25	25
	Not Drivers License	99	98
DL50	Drivers License	50	50
	Not Drivers License	99	98
DL75	Drivers License	75	75
	Not Drivers License	99	98
DL100	Drivers License	100	100
	Not Drivers License	99	98

(b) Credit Card Image Data

Group	Category	Training	Testing
CC25	Credit Card	25	25
	Not Credit Card	99	98
CC50	Credit Card	50	50
	Not Credit Card	99	98
CC75	Credit Card	75	75
	Not Credit Card	99	98
CC100	Credit Card	100	100
	Not Credit Card	99	98

(c) Student ID Image Data

Group	Category	Training	Testing
SID25	Student ID	25	25
	Not Student ID	99	98
SID50	Student ID	50	50
	Not Student ID	99	98
SID75	Student ID	75	75
	Not Student ID	99	98
SID100	Student ID	100	100
	Not Student ID	99	98

(d) Social Security Card Image Data

Group	Category	Training	Testing
SSN25	Social Security Card	25	25
	Not Social Security Card	99	98

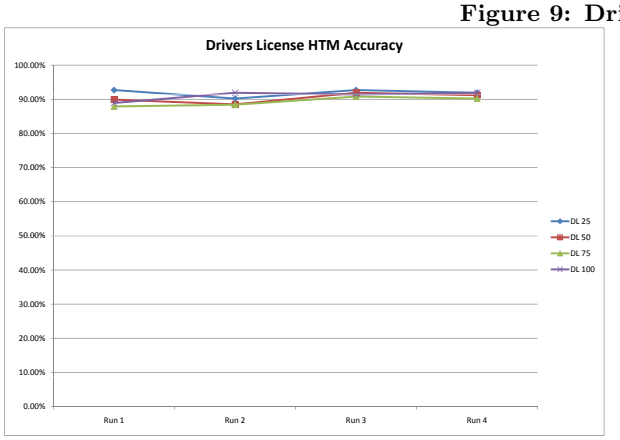
(e) Passport Image Data

Group	Category	Training	Testing
PP25	Passport	25	25
	Not Passport	99	98

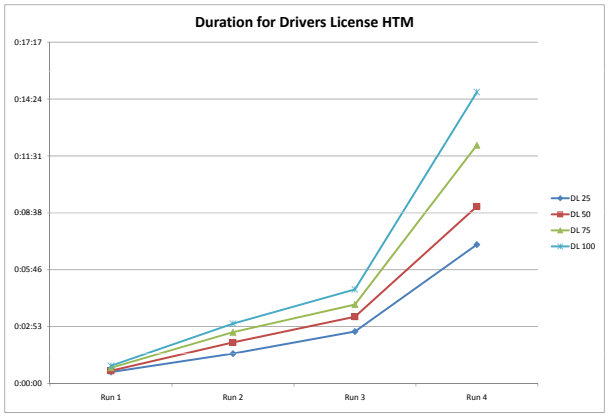
6.3 Results Analysis

As discussed above, each experiment was performed with a set number of experimental images along with a control group of 197 images. Each experimental image block was run 4 times, with a train and test run followed by a train and test run with parameterization of the training set. The networks were then subsequently optimized based on standard optimization, and concluded with an optimization that took account for the parameterization of the experimental images (shift, size changes, mirroring, and small rotations). The number of images was increased by 25 each time, up to a maximum of 100, to see what detection rate could be achieved. The goal was to find a relative minimum number of images for this task to reliably detect and identify our "sensitive" image of interest. This is important for an adversary, as it makes it easier to target images of interest without

Figure 9: Drivers Licenses

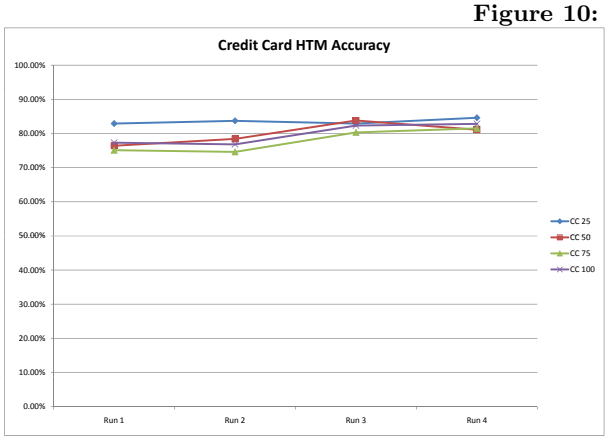


(a) Accuracy

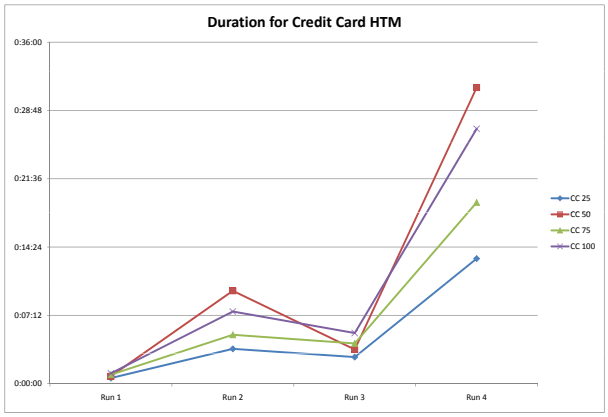


(b) Duration

Figure 10: Credit Cards



(a) Accuracy



(b) Duration

Table 2: System Parameters for Experimental Runs

	Run 1	Run 2	Run 3	Run 4
Action	Train and Test	Train and Test	Optimize	Optimize
Shift	n	y	n	y
Size Changes	n	y	n	y
Mirroring	n	y	n	y
Small Rotations	n	y	n	y

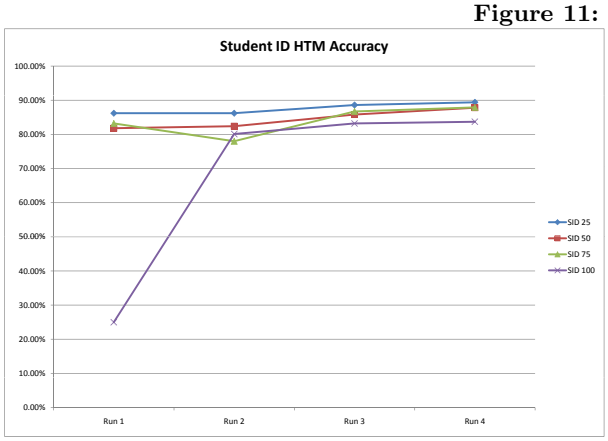
needing an overly large sample size of images to train the HTM as well as cutting down on the training time required for a high degree of positive identifications.

The data demonstrates that when you train the network to look for more subtle things like shifts, size changes, mirroring and small rotations, accuracy is sacrificed due to the additional requirements placed on the network, making it more difficult for a positive identification to occur. However, given a larger sample size, scanning for these nuances will aid in network robustness. It is also useful in gaining accuracy when you have numerous images of the same "object" from multiple angles and in varying sizes. However, in the case of our experiment, optimizing would seem to be more of a hindrance to improving accuracy due to the large variance among our image data, as the results demonstrate no particular trend towards improvement.

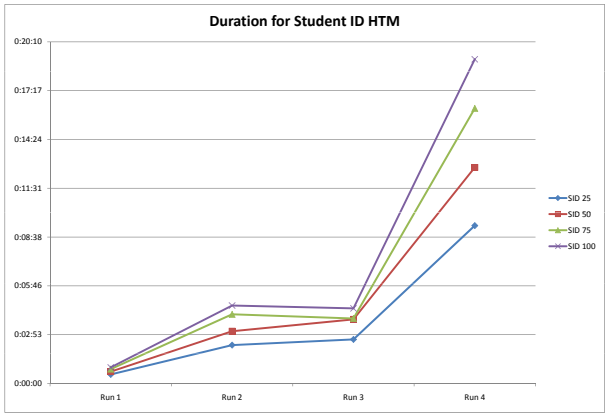
For an attacker, in cases where the total number of images scanned by the network is small, it is better to have false positives turn up in the list of image hits, because a quick visual confirmation of the image will determine if it contains sensitive information that has been obfuscated by anti-counterfeiting techniques implemented in the image or if it is an image of no value.

It is worth spending some time discussing the Gabor filter[6], as it does play an important role in image recognition. A Gabor filter is a linear filter used in image processing for edge detection[12]. Its impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. Usually Gabor filters are used to detected edges at specific angles, i.e. a 90 degree filter will pick up all edges that run at that angle. A number of these

Figure 11: Students IDs

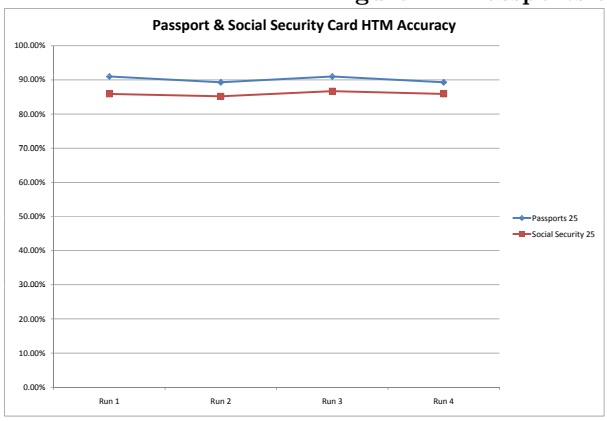


(a) Accuracy

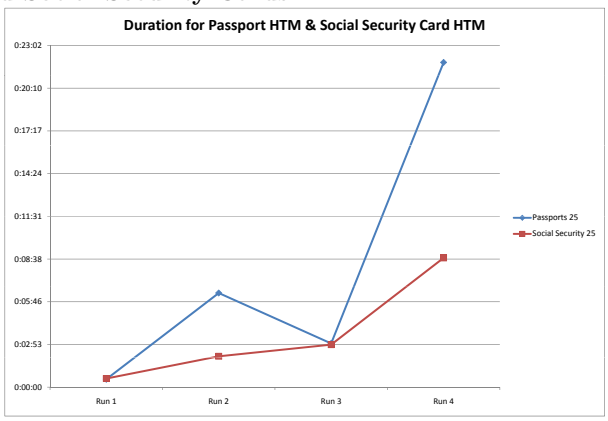


(b) Duration

Figure 12: Passports and Social Security Cards



(a) Accuracy



(b) Duration

filters are created for different angles and each generates an image. Then resulting images are then put back together and closely represent the original image that was processed with the filters. While Numenta doesn't tell us what degree filters they use in their Vision Toolkit as it is proprietary software and they want to keep their trade techniques secret, they do tell you that the first level of coincidences in the HTM model are replaced with Gabor filters of different orientations. As the general global image properties migrate to the upper levels of the HTM, the model is built, and can handle the differences between each license while still being able to tell that it is a license.

The image data itself is worth discussing, due to the nature of the applications of sensitive personal data and the societal and personal costs involved with sensitive information disclosure. As a consequence of this, the image data used in the experiments to train these networks is of what could generally be considered low-grade images. Most of the image data is extremely heterogeneous, and as a consequence of this lack of standardization amongst the training data is a significantly more difficult to train network, because there are not as many common traits for the HTM to focus on. If legitimate scans of real drivers licenses were gathered and a "lossless" (one that does not produce artifacts when the image is compressed) image file format was used to store these images, results in terms of accuracy would increase by a sizable margin. However, some of the robustness of the system comes from the generation of a generalized model. There is always a trade-off between accurate, specific models designed to capture one type of sensitive image and a generalized model that can grab many types of images. However, it is worth noting that approximately 85 to 90% accuracy can be achieved with this classifier. This serves to validate the fact that image classification algorithms are very effective at identifying such sensitive images. Since the emphasis of this paper was not really on creating new classification algorithms, and given that we have already achieved extremely good results, we did not go ahead and test the system with other classifiers. Based on our observations, we do expect that a focused attacker could achieve even better performance and that this is a very real significant threat.

One may ask at this point, that given this problem, is there any way of foiling such attacks. The HTM system proposed in this paper has a few weaknesses that can easily be exploited to foil the detection algorithm. Encrypting the files themselves or storing them in an encrypted container is the strongest and best security assurance that you can have against this type of attack. If the HTM (or indeed any

other classifier) cannot read the file, then it cannot perform the preprocessing and edge detection algorithms required to discover the image properties to classify it correctly. You can use a masking program as well to alter the image data and prevent it from being recognized, but it will require such large scale alterations as to mask the global structure of the image that it becomes pointless, and significantly more time consuming on a per image basis than just encrypting the image files in an encrypted container. To test the encrypted files, a known image of a drivers license was encrypted and an attempt was made to run the file through the HTM for detection. The HTM simply crashed, as the file could not be read properly. Given that many tools exist for modern encryption and are freely and easily available, there is no justification for not encrypting such critical data. Indeed, this process can easily be automated, by simply setting up the scanner to automatically deposit the scanned images into a mounted encrypted folder (for example, simply ensure that the "\My Scans" folder on Windows based systems is encrypted).

Another possibility is to ensure that scanned sensitive images are somehow made self-destructing. Typically, the use of such images is often for a limited time, at the end of which they should be destroyed. This can also be easily done, simply by tagging the file to be securely erased by the system after a specific period of time has elapsed. Finally, if the system cannot be penetrated, it cannot be scanned. Thus, another way of foiling these attacks is to ensure that the system is never penetrated in the first place. However, in general, this is much more difficult to ensure, and also limits security to that particular system (i.e., if the data is backed up, it may still be vulnerable at a different site). All of these options should be explored.

7. CONCLUSION

In this paper, we have made two significant contributions. Our first contribution is more abstract – we posit that a modern attacker can use sophisticated image classification techniques to accurately filter out images containing sensitive data, and use data extraction tools to breach individual privacy. To validate this hypothesis, we have built a classification system based on Hierarchical Temporal Memory that is able to classify sensitive images with over 90% accuracy. We believe that this can also serve as a guide for other interesting applications such as document detection and analysis. Secondly, our work once again proves that proves that security through obscurity is simply not enough and robust countermeasures such as encryption must be utilized to protect security and privacy. In the future, we plan on expanding our search to other types of sensitive documents, and images, and build more advanced tools to automatically extract data – this may even be useful from a counter-intelligence perspective.

8. REFERENCES

[1] Y. Amit, D. Geman, and X. Fan. A coarse-to-fine strategy for multiclass shape detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1606 – 1621, 2004.

[2] B. Bhanu. Automatic target recognition: State of the art survey. *IEEE Transactions on Aerospace and Electronic Systems*, 22:364 – 379, 1986.

[3] B. A. Bobier and M. Wirth. Content-based image retrieval using hierarchical temporal memory. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 925–928, New York, NY, USA, 2008. ACM.

[4] G. A. Carpenter. Large-scale neural systems for vision and cognition. In *IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks*, pages 3542–3547, Piscataway, NJ, USA, 2009. IEEE Press.

[5] R. Datta, J. Li, and J. Z. Wang. Content-based image retrieval: approaches and trends of the new age. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262, New York, NY, USA, 2005. ACM.

[6] H. G. Frichtinger and T. Stroher. *Gabor analysis and algorithms, theory and applications*. Birkhauser, 1998.

[7] D. George and J. Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS Comput Biol*, 5(10):e1000532+, October 2009.

[8] A. K. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 12–12, Berkeley, CA, USA, 1999. USENIX Association.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[10] J. Hawkins and S. Blakeslee. *On Intelligence*. Times Books, 2004.

[11] N. Inc. Numenta vision toolkit documentation tutorial.

[12] Y. Ji, K. H. Chang, and C.-C. Hung. Efficient edge detection and object segmentation using gabor filters. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 454–459, New York, NY, USA, 2004. ACM.

[13] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[14] M. W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on Neural Networks*, 1:28–43, 1990.

[15] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1372 – 1384, 2006.

[16] C.-F. Tsai, K. McGarry, and J. Tait. Image classification using hybrid neural networks. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 431–432, New York, NY, USA, 2003. ACM.

[17] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283, New York, NY, USA, 2000. ACM.