**Introduction**

Forensic Investigation involves three major steps; Acquisition, Analysis, and Reporting. The investigator has to get them right, every time, so there’s not much room for errors when people’s lives are on the line. Digital Forensics is the branch of Forensic Science that deals with the same process of investigation but in the context of cyberspace. Since past few years, the world has seen a boom in rise of computation power, storage space and variety of electronic devices. The rise of storage and media formats further increases types and the number of artefacts that one has to sift through to reconstruct the scenario and prepare a timeline for the chain of events. With the consumer grade products gaining storage capacity more than few terabytes and availability of applications for the ease of everyone for all day-to-day tasks the complexity, severity, and number of cybercrimes are increasing at an equal, if not at a faster rate. Almost every type of crime today has a digital touch to it, some related artefact residing in suspect or victim or any other person of interest’s computing device.

Digital Forensics has become an inevitable tool in the identification of cybercriminal activities due to its ability to extract valuable information and evidence from computing devices in a legally acceptable manner (Casey, 2010). However, over the course of past few years it has been facing a rise in challenges including but not limited to increased investigation time, explosion in data storage capacity leading to increased error rate and lack of automation. These are some of the limitations that can be overcome with the judicious use of Machine Learning & Artificial Intelligence algorithms by automating parts of digital forensic investigation that are otherwise performed manually. The idea is to pull together Acquisition, Analysis & Reporting phases and developing a solution that helps in setting a baseline for the investigators by automating time-consuming steps to speed up overall investigative process.

Some of the tasks can be automated by writing elegant algorithms that perform a series of pre-defined steps. Some, like image processing, can be achieved with the help of Machine Learning. The analysis phase is usually the most time-consuming one, it requires a lot of patience and searching to find evidence among artefacts. One of the tasks during this phase is suspect identification; while, for a human it is trivial when the number of images is few, but the difficulty of correct identification exponentially grows with the scale of case. This paper aims to solve above mentioned problems by providing an AI driven digital forensics framework to streamline the investigation process, by narrowing down the amount of evidence to be processed manually to an acceptable proportion.

The rest of the paper is structured with sections, “Related Work”, it presents existing use cases of automation in digital forensics and the ways its impacting data retrieval, analysis and data enrichment. “Methodology”, the process and algorithm involved in developing this software including programming languages and libraries used during the development phase. It’s subsection “Automated PoI Identifier” discusses an empirical method that applies deep learning and computer vision algorithms which are then evaluated in after mentioned test-cases. The next section follows an extensive discussion about the “Result” before walking into “Conclusion” and “Future Work”.

**Related Work**

This section will talk about rise of automation in digital forensics through machine learning and other methods. The section will talk about various ways through which the process of investigation has been made easier. This will range from big data analytics, document forgery detection, facial morphing detection, face recognition, adaptive forensics and challenges that being faced by detectives. Digital Forensics is an ever-growing field and so are its challenges, they include IoT Devices, Cloud Services and a variety of other computing devices with ever growing volume of data and lack of standardization in file formats that make things challenging and time consuming for investigators, that generates a backlog of cases. This study by ***[1]David Lillis et al. (2016)*** explains how those challenges can be solved by automating repetitive and boring tasks that hog a lot of precious time. Time that can be utilised better in establishing more concrete timeline of events to speed up both criminal investigation and civil litigation. Another way of achieving in automation of digital forensics is making it more adaptive and requirements driven. This method involves initial requirements modelling and configuration by security admins in the software but it pays off by shortening the cycles of event reconstruction ***[2]Liliana Pasquale et al. (2013)****.* Although above mentioned methods do help in speeding up the investigative process, they still require human intervention and configuration. This rule-based automation can be updated by deploying machine learning and artificial intelligence in the field of digital forensics to provide a better automation experience for the blue team.

Machine Learning has revolutionized technology in many aspects, forensic investigation is no stranger to the game. Machine Learning is transforming digital forensics triage by automating the tedious tasks of sorting out potential evidence from a pool of artefacts. Algorithms like Naive Bayes Networks, Locally Weighted Learning, Support Vector Machines and Decision Trees can help in classification of different types of forensic artefacts to help in cases involving copyright infringement and child pornography among others. ***[3](Fabio Marturana and Simone Tacconi, 2013)***. Another use of machine learning in digital forensics focuses on the use of SOM neural network, in this approach a number of files’ metadata is sent through the network to form clusters of files. Different clusters of files form different groups/classes of artefacts during an investigation. Those clustered files help in establishing a timeline of events by distinguishing between important and noise files ***[4]M. Al Fahdi et al. (2017)***. With the rise of higher capacity storage devices, the very amount of data is crossing terabytes, this increases analyst’s cognitive load. A framework suggested by ***[5]Hussam Mohommad et al. (2016)***focuses on data volume and heterogeneity by extracting metadata of files from different data sources. The semantic web ontologies help in avoiding ambiguities while extracting features. These features are pushed into AI based models to support automated identification and correlation of artefacts. Forensic Investigation involves anomaly detection, a task that is perfect for machine learning. But with the rise of so many methods including but not limited to k-nearest neighbours, support vector machines and decision trees, it’s hard to figure out which method to use when. A comparative study done by ***[6]Nandan Kumar et al. (2017)***tries to find out best method suited for outlier classification in network forensics (a branch of digital forensics) and data accuracy for digital forensic use. Going through papers related to machine learning, deep learning and automation for sensitive tasks like criminal investigation may raise questions about their accuracy and relevance in real world cases. A study done by ***[7]Adita K et al.(2018)***tries to enable trust in Deep Neural Networks by developing an Adversary Testing Framework that helps in testing robustness of deep learning methods.

Questioned Documents are often processed by the analysts to find inconsistencies to detect forgeries, one such scenario was tackled by ***[8](Micah K. Johnson and Hany Farid, 2007)****,* where document forgery was detected by calculating the source of light and the position of eyes relative to the light source, thus proving whether a certain picture was a digital composite or not. But document forgeries are not limited by doctored images, another form of forgery involves face morphing. Automatic generation of morphs was achieved by utilizing Benford features calculated by quantized DCT coefficients of JPEG-compressed images. A linear Support Vector Machine is trained on Benford features and is used to automatically detect morphed faces in images by ***[9]Andrey Makrushin et al. (2017)****.* Face morphing being a common cybercrime, there are some other methods for the detection of morphed faces. One such method relies on continues image degradation. This degradation approach creates multiple artificial self-references that eventually help in classification of faces in an image as stated by ***[10](Tom Neubert, 2017).***A deep learning based method also exists for forgery detection that makes use of pre-trained Convolutional Neural Networks which feeds a Support Vector Machine that that is trained on hierarchical representations from RGB images to help in detection of image forgery by copy move or image splicing methods ***[11]Yuan Rao et al. (2016)****.*

Images have become one of the most mainstream methods of sharing ideas and communicating thoughts, and so it is important to identify whether the given image is original or processed can help in progressing investigation. These processes can be identified by a multi-class classification neural network that identifies whether an image has been processed and classify most common post-processing methods ***[12]Tian Huang et al. (2018)****.* At times, identification of a camera's make and model can turn the tables during case examination. Images from different cameras can be sent through a multi-class SVM classifier, based on extracted features can help in identification of camera. Authors ***[13]Amel Tuama et al. (2016)***, reached more than 95% accuracy with their database. This work can lead to ease in PoI (Person of Interest) identification by connecting images to the camera that they were shot from where camera can be further be connected to the person responsible. Moving forward with images and document forgery, a need for identification of fake faces, facial morphs and just facial recognition in general has become paramount in forensic investigation. With the advancement in the field of image processing, machine learning and artificial intelligence it is now easier than even to create realistic looking fake faces by employing Generative Adversarial Networks(GAN) or skilled graphic designers. A research done by ***[14]Shahroz Tariq et al.(2018)***proposes a neural network that helps in detection of fake or generated images by humans or by machines without resorting to any metadata information.

Some of the use cases where computer vision related algorithms can help in digital forensics include detection of child pornography, child sexual abuse and related crimes. A research that focuses on features based on iris geometry helps in automatic detection of children in images. This approached proposes Face to Iris Area Ratio(FIAR) is based on the fact that iris size remains almost constant in childhood. The algorithm provides a robust method for identification of underage children in digital images ***[15]Vassilios Chatzis et al. (2016).*** Closed-Circuit Television (CCTV) cameras are used almost everywhere these days for security surveillance. This makes them one of the most common providers for digital evidence. At times footage from these cameras can make or break the case, which makes it vital to quality check these cameras. A framework developed by ***[16]Mohamad Firham Efendy Md Senan et al.(2017)*** created a framework for testing different cameras for their video quality, forensic value and scope for face recognition from the videos produced by those cameras. The framework helps in identifying ideal camera resolution and distance for higher forensic value and face recognition accuracy. Different image processing, computer vision and machine learning algorithms produce different accuracy of results during face verification and face recognition. A comprehensive comparative study on those algorithms help in identification of limitations for state-of-the-art neural networks providing valuable insights for future research in the field ***[17]Giuseppe Amato et al.(2019).*** Facial recognition works very well as long as images are not tampered with, but it’s hard to find any images without any traces of retouching in today’s date. Social media has changed the way conduct their daily life and how they take images. A novel approach based on supervised deep learning can help in classification of original and retouched images. This helps in detection of retouched images with software, detection of makeup in face images and distinguishing between original and modified images ***[18]Aparna Bharti et al(2016).***

A lot of work has been carried out to improve automation in this niche field of digital forensics. Sometimes through rules and configuration and other times through the implementation of machine learning and artificial intelligence. Research work ranging from document forgery detection to extensive comparative analysis of multiple camera models used for security monitoring. This paper will propose a methodology that will help in speeding up forensic investigation through the use of state-of-the-art machine learning and computer vision algorithms along with some rule-based automation for data acquisition, file-type identification and person of interest identification.

**Methodology**

This section will explain the development tools, environment and the phases that this proposed software goes through to provide an automated experience during a digital forensics' investigation. Starting off with the development tools, following programming languages were used to write this program:

1. GoLang
2. Python

These languages were selected due to their unique qualities, the idea was to use the right tool for the right task. GoLang***[19](golang.org)*** was selected for systems programming part due to its simple learning curve, fast execution and compilation speed that allows writing efficient and reliable software. GoLang also has the smallest memory footprint which is another added advantage. Python***[20](python.org)*** was selected due to its supremacy in data science related tasks. GoLang’s concurrency model, fast runtime and simplicity sets it apart from other languages ***[21](K.L. Clark & F.G. McCabe, 2004).*** Number of readily available libraries for machine learning and artificial intelligence related tasks also helped in building automation. Notably, dlib & face\_recognition libraries which were very helpful in face verification part. That, combined with python’s wide acceptance in the industry makes it go-to language for statistical analysis and machine learning ***[22](Abhinav Nagpal & Goldi Gabrani, 2019)***. In next few sub-sections, the paper will go on to explain how traditional investigation process works with all three phases that are involved in it followed by machine learning based automation that the software provides, proposing an approach that helps in minimizing human effort. The paper will also get into details about the correct usage of the software along with its available CLI commands, flags and options.

**Traditional Method**

The classical method of digital forensic investigation involves three separate processes, namely, acquisition, analysis & reporting. The process is mostly manual and involves a lot of human effort. This can lead to delays in investigative process which may result to increase backlogs of pending cases. **Figure 1** is a flowchart explaining traditional digital forensics investigation process.

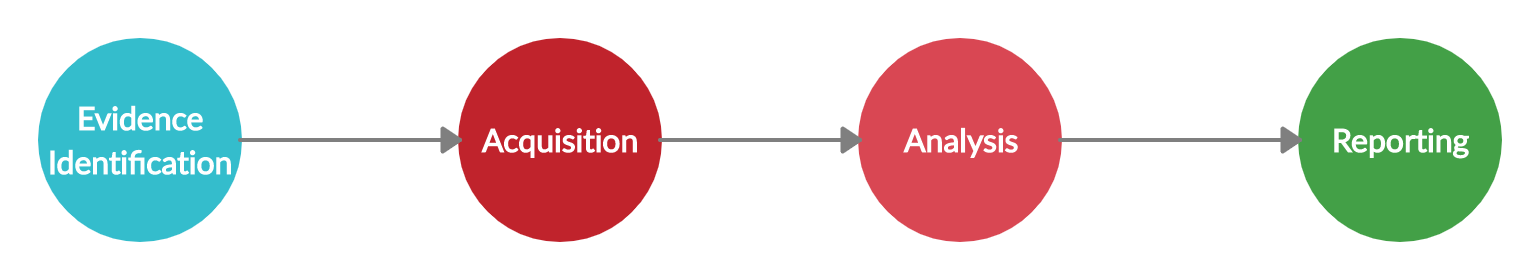


Figure 1

In the traditional process, the investigator must collect the physical device while maintaining the Chain of Custody and forensic integrity of the physical devices. This task is typically achieved by first responders that arrive at the scene. This evidence is then sent to the analysts who perform acquisition, analysis write the report for the same. This usually is mostly done manually which increases human effort unnecessarily. Several parts can be automated and if they are done correctly then that can become deciding factor between increasing backlogs and solved cases. **Figure 2** is a pie-chart representing the amount of effort exerted by an analyst during a typical forensic analysis process. The pie-chart is based on experience of authors of this paper.

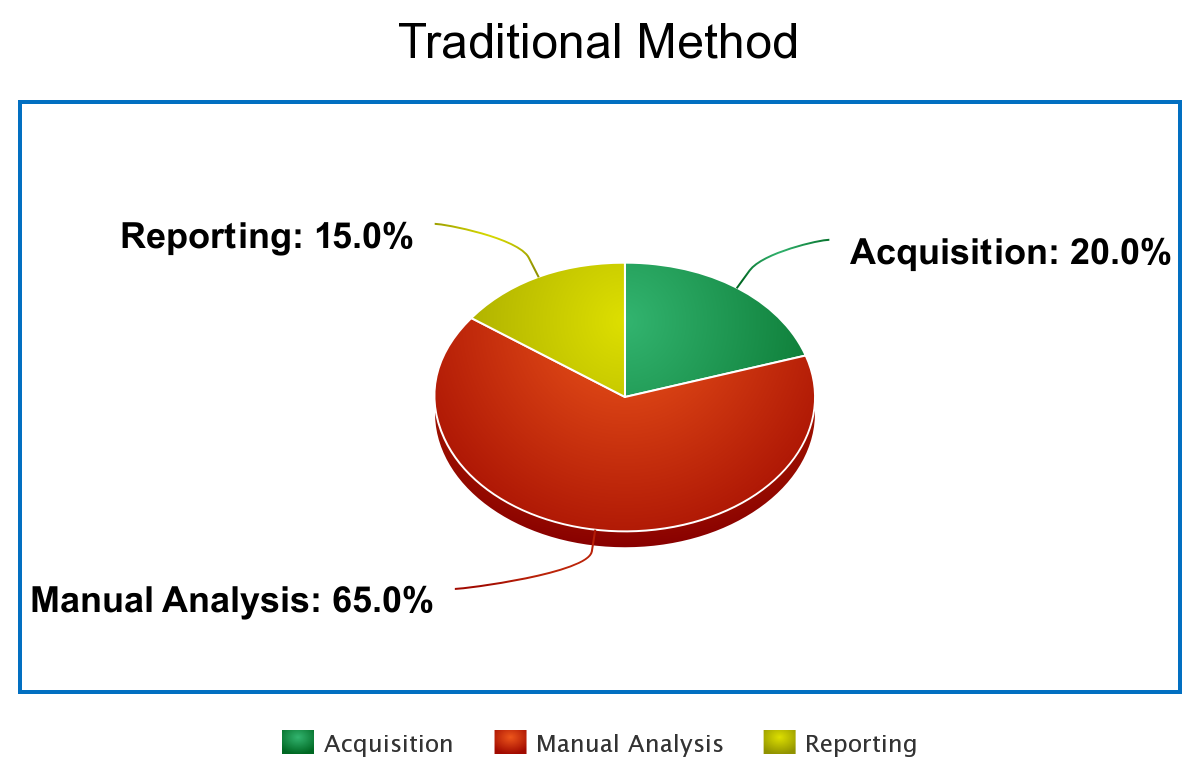


Figure 2

Above chart is a graphical representation of the sheer volume of human effort that goes into the analysis, making up the largest chunk of any digital forensic investigation process. This becomes the bottleneck that slows down entire examination. To help overcome this barrier, SynFO provides an ML based intelligent triage system that helps in acquisition, extraction & preliminary analysis of acquired evidence.

**Proposed Method: SynFO**

An ML based intelligent triage forensic software to provide a better investigation experience by curbing redundant manual effort by combining ML & rule-based triage analysis. This in turn changes the process by taking care of most common parts including disk image acquisition, extracting relevant files & PoI identification. **Figure 3** is a flowchart explaining how SynFO changes the picture of digital forensic investigation.

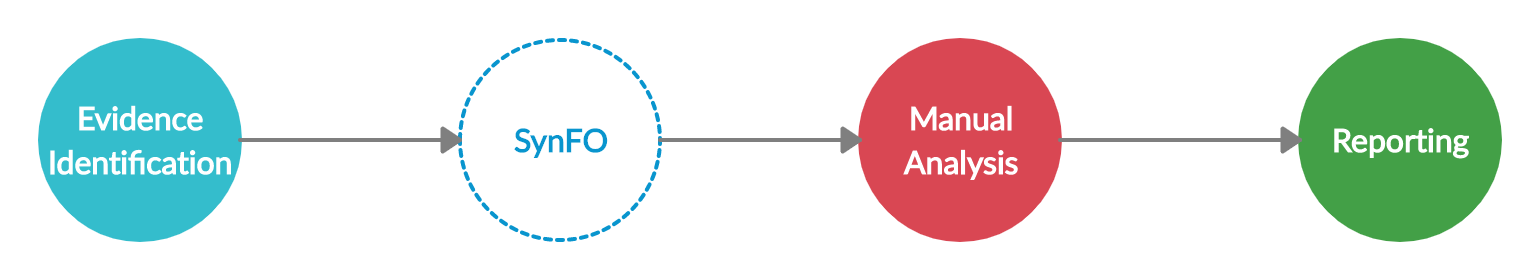


Figure 3

By Involving SynFO into the picture, human effort can be cut down to 15-10%. Some cases, that this software can help in investigating include suspect identification, forged documents, missing person identification, kidnapping and etc. **Figure 4** is another pie-chart explaining how inclusion of SynFO changes the investigation scenario.

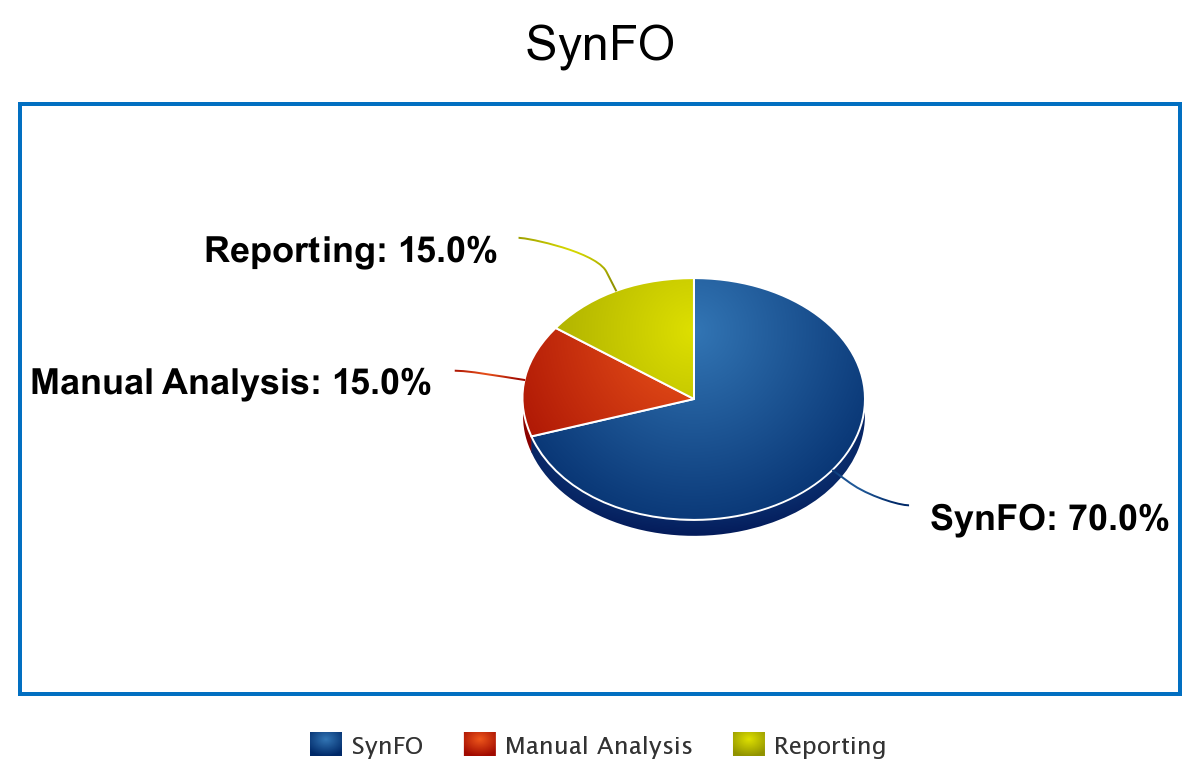


Figure 4

Above pie chart shows how SynFO helps through ML based intelligent triage system. **Figure 5** helps by putting things into perspective through a line diagram that depicts comparison between traditonal manual process and SynFO aided forensic analysis.

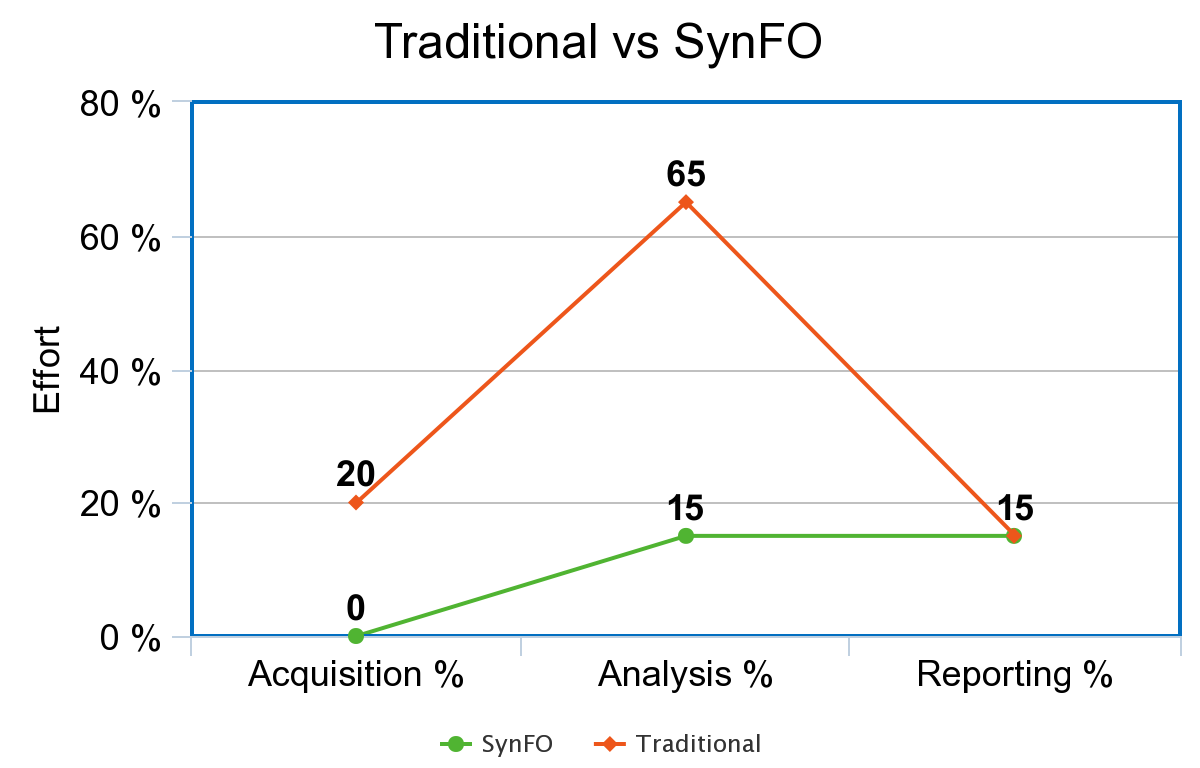


Figure 5(less is better)

**Acquisition**

In the initial phase, any external storage device that makes connection via a USB interface will be connected to the forensic workstation. This external storage will stay connected, but it will not be mounted so that integrity can be preserved. The software will take the special device file of this external storage as an input. A special device file is an intermediary that provides the Kernel and the device driver to communicate. After that, the software will proceed to start reading raw bytes from the device file, creating a bit-stream copy of the same. Default buffer size is 10240 bytes but that can be changed via command line flags. After the bit-stream copy, a single image file will be written on the disk. This entire imaging process will be timed, and an elapsed time will be presented in a human readable format, so that chain of custody can be maintained, and imaging time can be documented. Post-imaging, the software will move on to calculate MD5 and SHA256 hashes of the image file. These hashes will go on to help maintain integrity of the evidence through the investigation process.

**Extraction**

The second phase is about extracting files of interest. The software will mount the image that was created in the previous step and then proceed to walk through the file system starting from the root. It’ll go through all the directories and sub-directories to make sure all the files are considered. Each file in the entire filesystem will be opened in read-only mode to check if their magic numbers match a certain criterion or not. These magic numbers are present in headers of almost all the files. And they indicate what type of file they are. Each magic string uniquely identifies a type of file. The software provides support to find pictures, audios, videos and archive files. When identified, these files (depending upon user selection), will be extracted and copied. Extraction process also carving images out of unallocated space, log files, pcap files, common documents and pdf files. **Table 1** will present a list of supported file formats that can be carved out of unallocated space, log files and pcap files. The software parses unallocated space files looking for header and footer to rebuild and carve out interesting files.

|  |  |  |
| --- | --- | --- |
| **Source Type** | **File Type** | **Support** |
| **DD** | PNG | YES |
|  | GIF | YES |
|  | JPG | YES |
|  |  |  |
| **ISO** | PNG | YES |
|  | GIF | YES |
|  | JPG | YES |
|  |  |  |
| **LOG** | PNG | YES |
|  | GIF | YES |
|  | JPG | YES |
|  |  |  |
| **PCAP** | PNG | YES |
|  | GIF | YES |
|  | JPG | YES |
|  |  |  |

Table 1

**Table 2** represents a list of file formats that are supported for embedded image extraction. The software goes through different files, finds embedded images within those files and extracts them. It supports most common file formats that are currently in use.

|  |  |
| --- | --- |
| **File Type** | **Support** |
| PDF | YES |
| PPT | YES |
| PPTX | YES |
| DOC | YES |
| DOCX | YES |
| XLS | YES |
| XLSX | YES |
| ODT | YES |
| ODP | YES |
| ODF | YES |
| EPUB | PARTIAL |

Table 2

**Automated PoI Identification**

This is the phase where a handoff between Go and Python takes place. The compiled go binary calls python script. This script makes use of dlib and face\_recognition, two battle-tested machine learning and computer vision libraries that help in development of face verification, detection and recognition related software. The face\_recognition library provides a high-level application interface that interacts with the underlying dlib library that was written in C++, this face\_recognition library can make use of either Histogram of Gradients or Convolutional Neural Networks to detect faces in an image. After face detection, face landmarks are calculated and presented in a 128-dimension feature space. These landmarks are then used to compare to other faces to verify whether two faces match or not. The script in use here leverages these libraries to provide following functionalities:

1. One to One Matching
2. One to Many Matching
3. Many to One Matching
4. Many to Many Matching

Provided two datasets of images, the database of images with the forensic investigator and unknown images, the software will try to find number of faces in each instance. The result could either be 0, 1 or many. In either case, the software will match every face from unknown set of images with every image of known set of images. At the end the control will be transferred back to the go binary where it’ll display elapsed time for PoI identification process. These libraries that are being used currently claim 99.38% accuracy on LFW dataset. All the results from poi identification script will be written into a text file for easy referencing and documentation. The software provides command line arguments to select method for detecting faces, the methods being a rule based approach using HoG(histogram of gradients) & a deep learning based approach using a CNN(Convolutional Neural Network).

**How to Use**

SynFO provides easy to use commands and flags. The software tries to keep number of flags to a minimal to keep a balance between ease of use and customizability. Following are some command examples:

1. **File Extraction**

|  |  |  |
| --- | --- | --- |
| |  | | --- | | $ sudo ./synfo\_v1 ext -src <path\_to\_device\_file> -dst <path\_to\_evidence\_file> |   **OR**   |  | | --- | | $ sudo ./synfo\_v1 ext -src <path\_to\_device\_file> -dst <path\_to\_evidence\_file> -bs 50000 | |

1. **Automated PoI Identification**

|  |  |  |
| --- | --- | --- |
| |  | | --- | | $ sudo ./synfo\_v1 apd -src <path\_to\_device\_file> -dst <path\_to\_evidence\_file> -poi <path\_to\_image\_db> |   **OR**   |  | | --- | | $ sudo ./synfo\_v1 apd -src <path\_to\_device\_file> -dst <path\_to\_evidence\_file> -poi <path\_to\_image\_db> -model {hog|cnn} |   **Figure 6, Figure 7 & Figure 8** are examples of how the software can be used with above commands and other existing options. The command-line utility has a help option that makes the software easy to use.    Figure 6    Figure 7    Figure 8 |

SynFO comes with two commands: ext & apd. The ext command will perform acquisition and extraction while providing options between picture, video, audio and archive files. This was designed to quickly pull out files of interest during investigation. Two optional flags are available, the bs flag with this command will provide way to set buffer size, ft flag with this command will provide a way to set the file type to be extracted. Other command, apd, will perform acquisition, extraction and automated PoI identification. This command allows use of two optional flags, those flags being bs for buffer size and model. This model flag provides a selection between hog & can models to be used while detecting faces in pictures. If no commands or flags are provided, then SynFO shows a help screen with all examples for all possible commands and flags.

**Result**

In this section, the paper will represent results provided by running SynFO through example cases. These scenarios will be presented to help bridge the gap between example use-cases and the real world. Results are based on two cases; each one has been run on following devices:

1. 16 GB SD Card
2. 32 GB USB Drive

Case 1:

A disgruntled employee of cleanyCleanCleaners.co decided to create the fake profile of their CEO. Being a small startup, everyone knew everyone’s name in the company, regardless of their position and rank. This employee decided to create multiple fake accounts in the name of the CEO and then started writing hate-speech content on public forums and went rogue on popular social media sites like Facebook. To achieve this task, the employee had to download several images of their CEO. When this employee was caught, and first responders arrived on the scene they found a secondary storage device containing all the images. Without SynFO just gathering all the images would be a time-consuming task, identification of the person of interest among those images is whole another process. But by involving this software, all that manual effort it cut down. **Table 3** shows how SynFO performs in this scenario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device** | **Size** | **Imaging Time** | **Data Pull Time** | **PoII Time** |
| SD Card | 16 GB | 00:22:22 | 00:01:15 | 00:00:54 |
| USB Drive | 32 GB | 00:40:23 | 00:02:40 | 00:01:54 |

Table 3

Case 2:

An organized crime unit based out of a certain city launders money and forges fake identities. Their routine tasks involve creating replicas of important documents including but not limited to PAN Card, Aadhar Card, and Passport. These documents are then used to help transport high value targets all over the globe. The documents also serve for human trafficking. At times it’s hard to pin down the organizations due to the amount of time it takes to identify evidence type, people and faces within the evidence. Due to this plethora of different document types, embedded images, deleted data and other anti-forensic techniques used by cyber criminals it’s hard to perform timely analysis. In a normal scenario, the first responders will try to collect as much evidence as possible without disturbing the scene and bring in a lot of data for the analysts to sift through. That can take a considerable amount of time which ends up slowing down entire investigation process. The first responders can easily run SynFO which will take care of data acquisition, relevant file extraction and person of interest identification. **Table 4** shows a similar scenario along with the results that were observed by running this software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device** | **Size** | **Imaging Time** | **Data Pull Time** | **PoII Time** |
| SD Card | 16 GB | 00:22:23 | 00:02:52 | 00:02:47 |
| USB Drive | 32 GB | 00:40:39 | 00:02:56 | 00:03:03 |

Table 4

**Conclusion**

This section