A Proposed Digital Forensic Tool for Video Content Analysis in the Investigation Process: Overcoming Challenges and Improving Efficiency

Submitted by

Ruwa' Fayeq Suleiman Abu Hweidi

Thesis Advisor

Dr. Eman Daraghmi

The thesis was submitted in partial fulfillment of the requirements for the master's degree of Science in Cybercrimes and Digital Evidence Analysis

2024

Committee Decision

Main Advisor:	Dr. Eman Daraghmeh
Internal TEC member 1:	Prof. XXXX
External TEC member 1:	Prof. XXXX (optional)

Declaration

I, Ruwa' Fayeq Suleiman Abu Hweidi, declare that this thesis titled, "A Proposed Digital Forensic Tool for Video Content Analysis in the Investigation Process: Overcoming Challenges and Improving Efficiency" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

I hereby confirm the following:

- I hereby confirm that the thesis work is original and has not been submitted to any other University or Institution for higher degree purposes.
- I hereby grant SUTD the permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created in accordance with Policy on Intellectual Property, clause 4.2.2.

- I have fulfilled all requirements as prescribed by the University and provided 1 copy of my thesis in PDF.
- I have attached all publications and award list related to the thesis (e.g., journal, conference report, and patent).
- The thesis does / does not (delete accordingly) contain patentable or confidential information.
- I certify that the thesis has been checked for plagiarism via turnitin/ithenticate. The score is 100%.

Name and signature: Ruwa' F. S. Abu Hweidi

Date:

Abstract

The escalating reliance on digital media in criminal investigations has spurred the development of the Video Content Analysis Tool (VCAT), a sophisticated digital forensic solution designed to enhance video data analysis. VCAT integrates leading-edge machine learning models including GroundingDINO for object detection, PaddleOCR for optical character recognition, and Whisper AI for speech recognition, facilitating a comprehensive forensic approach. These technologies enable VCAT to effectively process video content from varied sources such as CCTV and social media, achieving high accuracy and efficiency. Additionally, VCAT's advanced text-based search functionality allows for targeted investigations using specific keywords or prompts. Rigorous experimental evaluations conducted using Google Collaboratory highlight VCAT's superior performance over standard forensic tools in detecting diverse video elements. Adhering strictly to NIST standards and maintaining the integrity of the chain of custody, VCAT ensures that its outputs are legally admissible. The final reports generated by VCAT are meticulously formatted via ReportLab to meet legal standards. Despite its robust capabilities, challenges like processing efficiency and handling low-quality video data were identified, which will guide future enhancements. VCAT marks a significant advancement in digital forensic technology, providing a scalable, efficient, and compliant tool that significantly enhances forensic analysts' ability to extract and analyze evidentiary data swiftly and accurately.

Keywords: Digital forensics, legal admissibility, machine learning, NIST standards, object detection, OCR, speech recognition, textbased search, video analysis

Publications

Journal Articles

1. Abu Hweidi, R. F., Jazzar, M., Eleyan, A., & Bejaoui, T. (2023). Forensics investigation on social media apps and web apps messaging in Android smartphone. 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), 1–7. https://doi.org/ 10.1109/SmartNets58706.2023.10216267

Conference Presentations

1. Hweidi, R. F., & Eleyan, D. (2023). Social engineering attack concepts, frameworks, and awareness: A systematic literature review. International Journal of Computing and Digital Systems, 13(1), 691–700.
2. Hweidi, R. F., Jazzar, M., Eleyan, A., & Bejaoui, T. (2023). Sata m.2 on forensics: Trim function effect on recovering permanently deleted files. 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), 1–6. https : / / doi . org / 10 . 1109 / SmartNets58706.2023.10215536

Acknowledgements

First and foremost, I express my deepest gratitude and praise to Allah for the vastness of the heavens and the earth, for His immense favor, and for blessing me with the success of completing this thesis. I pray that this work will serve as beneficial knowledge and earn His pleasure.

I would also like to thank my university, Palestine Technical University - Kadoorie (PTUK), for this pioneering educational opportunity and experience.

I am sincerely grateful to my advisor, Dr. Iman Daraghmi, for her invaluable guidance, constructive feedback, and unwavering support throughout this research. I also extend my thanks to my thesis committee members, Dr. Rami Debe'e and Dr. Mohammad Abu Omar, for their guidance and advice.

My heart never forgets to thank Gaza. Whenever I feel overwhelmed and unable to continue my thesis amidst my home suffering, your resilience and patience in the face of ongoing genocide inspire me. Thank you, Land of Rabat.

I am profoundly grateful to my family. To my dear mum and dad, whose hearts overflow with prayers and pride, to my beloved husband Alaa, for his boundless support, and to my precious children Naeem, Basil, and Yousof, for their patience, love, and encouragement. Your sacrifices and unwavering faith in me have been my greatest strength. May Allah reward you all abundantly. Moreover, to my siblings, my brothers' wives, and everyone who supported me in any way during the completion of this thesis, thank you.

Heartfelt thanks to my colleagues for their collaboration, encouragement, and moral support throughout this journey.

Contents

List of Figures

List of Tables

Table 1: List of Abbreviations

Abbreviation	Meaning
ACPO	**A**ssociation of **C**hief **P**olice **O**fficers
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
AP	**A**verage **P**recision
ASR	**A**utomatic **S**peech **R**ecognition
BPE	**B**yte **P**air **E**ncoding
BiLSTM	**B**idirectional **L**ong **S**hort-**T**erm **M**emory
CCTV	**C**losed-circuit **T**elevision
CNN	**C**onvolutional **N**eural **N**etwork
CPU	**C**entral **P**rocessing **U**nit
DBNet	**D**ifferentiable **B**inarization **N**etwork
DETR	Detection Transformer
DINO	**D**ETR with **I**mproved **D**enoising **A**nchor **B**oxes
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
EHR	**E**lectronic **H**ealth **R**ecord
FTK	**F**orensics **T**ool **K**it
FVA	**F**orensics **V**ideo **A**nalytic
GenAI	**G**enerative **a**rtificial **i**ntelligence
GLIP	**G**rounded **L**anguage-**I**mage **P**re-training
GMM	**G**aussian **M**ixture **M**odels
GPU	**G**raphics **P**rocessing **U**nit
GPT	**G**enerative **P**re-trained **T**ransformer
GUI	**G**raphical **U**ser **I**nterface
HOG	**H**istogram of **O**riented **G**radients
HDD	**H**ard **D**isck **D**rive
ICR	**I**ntelligent **C**haracter **R**ecognition
IoT	**I**nternet of **T**hings

Abbreviation	Meaning
IVR	**I**nteractive **V**oice **R**esponse
LMM	**L**arge **M**ultimodal **M**odels
LSTM	**L**ong **S**hort-**T**erm **M**emory
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
MICR	**M**agnetic **I**nk **C**haracter **R**ecognition
ML	**M**achine **L**earning
MLLM	**M**ultimodal **L**arge **L**anguage **M**odel
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
OCR	**O**ptical **C**haracter **R**ecognition
OSAC	**O**rganization of **S**cientific **A**rea **C**ommittees for **F**orensic **S**cience
RAM	**R**andom **A**ccess **M**emory
RCNN	**R**egion-based **C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
ROI	**R**egion of **I**nterest
RPN	**R**egion **P**roposal **N**etwork
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
SVM	**S**upport **V**ector **M**achine
SOAT	**S**tate-of-the-art
SSD	**S**olid-**S**tate **D**rive
SSMD	**S**ingle **S**hot **M**ultiBox **D**etector
SVTR	**S**cene **T**ext **R**ecognition with a **S**ingle **V**isual **M**odel
SWGDE	**S**cientific **W**orking **G**roup on **D**igital **E**vidence
UI	**U**ser **I**nterface
VCAT	**V**ideo **C**ontent **A**nalysis **T**ool
YOLO	**Y**ou **O**nly **L**ook **O**nce

For the souls who are martyred every day in our Palestine.

To my beloved family—my dear parents, cherished husband, children, and siblings—artifacts of Paradise in this holy land.

To all who have supported me with heartfelt prayers throughout this journey …

Chapter 1

Introduction

1.1 Introduction

Forensic science, with its roots deep in history, has evolved significantly from rudimentary beginnings to a critical component of criminal investigations. The practice began emphasizing physical evidence in the late 18th century. Expanding to include ink analysis and microscopic examination of bloodstains by the early 19th century, the advent of photography further transformed the field by enhancing suspect identification and crime scene documentation. Significant turning points occurred in the 20th century, such as formalizing forensic science education and establishing the first police crime lab in 1910 in Lyons (Wileman, 2015).

Recent advancements in digital technology and the internet throughout the 21st century have revolutionized the sharing and analysis of forensic data, significantly enhancing the effectiveness of the field (Jacob et al., 2010). The widespread use of digital devices in recent decades has played a central role in the development of digital forensics investigation science, which has extended to multiple branches of our lives. The abundance of mobile devices, such as smartphones, that contain a variety of social network applications, enables users to record and share media flexibly. Other fields, such as smart home systems, Internet of Things (IoT) devices, and Closed-circuit Television (CCTV), which is known as video surveillance, also contribute to this evolution (Mante & Khan, 2020; Xiao et al., 2019).

The rise in complexity and volume of digital evidence has led to the need for advanced tools tailored to various forensic scenarios. The proposed digital forensic **V**ideo **C**ontent **A**nalysis **T**ool (VCAT) presents a flexible system that lets investigators customize by enabling

or disabling specific analysis modules, such as object detection, Optical Character Recognition (OCR), and Speech Recognition, depending on the case requirements. This adaptability enhances the overall effectiveness and efficiency of the digital forensic process.

It is common to seize digital evidence at crime scenes, and this, along with the increase in the size of stored media, leads to the acquisition of many clues with numerous varying types of files that need to be investigated (Javed et al., 2021). Video files take up a lion's share of user interest in social media and marketing, and some popular social networks, such as YouTube and WhatsApp, depend on video (Rajeev & Raviraj, 2023).

In digital forensic investigations, acquiring forensic images from various storage devices presents challenges due to the diverse file types and large volumes of data involved. The process often necessitates data reduction to manage the overwhelming amount of information (Quick & Choo, 2016).

Artificial intelligence (AI) has become an essential tool in tackling the issues related to cybercrimes. It provides methods like clustering, filtering, and data content searching, which are instrumental in this field (Iqbal et al., 2020; Javed et al., 2021). By automating the investigation process through AI, results are achieved more quickly and cost-effectively, minimizing the reliance on human efforts (Jarrett & Choo, 2021).

Digital multimedia forensic analysis is a commonly employed technique for verifying the authenticity and consistency of evidence submitted in court during civil or criminal procedures (Zakariah et al., 2018). Many researchers are applying forensic investigation on video from different perspectives within the guidance of the challenges previously mentioned. Thus, a lot of studies try to solve gaps in video

content forensic investigation by focusing on data reduction of the data that needs to be analyzed (Quick & Choo, 2016), reusing speech-to-text tools (Negrao & Domingues, 2021), or using AI algorithms in machine learning (ML) and deep learning (DL) in other fields (Gad et al., 2022).

An essential aspect of the proposed tool architecture is the rigorous process of data integrity verification through hash checks. This ensures that from the moment data is acquired to its final analysis, its integrity is maintained, crucial for its admissibility in legal proceedings. This rigorous approach underscores the forensic soundness of the methodology and is fundamental in establishing the credibility of the evidence generated.

1.2 The Problem Statement

Digital forensic investigators often face significant challenges when extracting forensic images, particularly when targeting specific artifacts within video files. While tools like Forensics Tool Kit (FTK) ("FTK Forensics Toolkit - Digital Forensics Software Tools", 2024), Magnet AXIOM (the truth. Protect the innocent, 2023), OSForensics (investigation for a new era by PassMark Software, 2023), and Autopsy (Technology, 2023) provide some assistance, their capabilities in video content analysis are frequently limited.

Investigators usually depend on keyword searches within forensic tools by importing or adding keywords manually. However, this method is limited because it relies only on text, reducing its efficiency in analyzing varied video content. Videos can include events, speeches, objects, and written text, which may not be fully captured by keyword searches alone. As a result, manual analysis is often required to identify all potential artifacts, particularly if significant clues

were missed during initial reviews (Horsman, 2022).

Compounding these challenges is the sheer volume of video data encountered in forensic investigations. From small files on social media platforms like TikTok to extensive CCTV footage, investigators must navigate through a vast amount of data. While forensic tools help in filtering and keyword-based searching, manual video content review remains essential due to the nuanced complexities that automated tools may miss.

Moreover, analyzing video files requires sophisticated techniques such as metadata analysis, OCR, object detection, and speech recognition technology to extract relevant information accurately. Environmental factors like video quality and background noise further complicate forensic analysis, affecting the reliability of findings.

These complexities underscore the need for advanced methodologies and robust software solutions to analyze video evidence and produce reliable forensic reports effectively. However, current forensic tools face several limitations:

- Limited tool capabilities hinder comprehensive video content analysis.
- Keyword-based searches often lack effectiveness in diverse video content.
- Manual analysis is frequently needed for overlooked clues extraction.
- Vast video data volume complicates forensic investigation processes.
- Current tools may lack support for requiring sophisticated analysis techniques within video streams.

The proposed architecture is designed to address these challenges

partly by incorporating configurable settings that enable forensic investigators to adapt the tool to the specific demands of each investigation. This includes the ability to choose between different analytical modules like OCR, and speech recognition based on the needs of the case. Each module is optimized to handle the complexities of diverse video content effectively, enhancing the accuracy and efficiency of forensic analysis. By taking steps to overcome these obstacles, investigators can improve the efficacy of criminal investigations by increasing the precision and efficiency of digital forensic analysis.

1.3 The importance of the study

In the forensic investigation process, investigators go through specific steps to ensure the admissibility of the investigation in court. These steps include acquiring data, improving the integrity and validity of extracted data, and documenting the chain of custody for seized hardware or software elements (Bokolo & Liu, 2024). However, the existing software tools often lack comprehensive features for analyzing video content and producing admissible reports.

This study proposes a forensic tool that can extract video files from acquired forensic images, search for the content of video files using AI concepts such as open AI models, and then produce a court-admissible report to save time, cost, and effort.

The proposed tool integrates state-of-the-art (SOAT) AI technologies to enhance the analysis of video content. By leveraging advanced OCR, and speech recognition capabilities, the tool efficiently processes and analyzes complex video data. This ensures that the results are solid and legally defendable while also expediting the forensic analysis process and improving data extraction precision and dependability.

The proposed tool offers several advantages, such as reducing the amount of time needed to find the occurrence of specific keyword content (audio, OCR, event, or object) in video files, saving effort in finding results in large or multiple small video files, producing a court-admissible report containing filtered and clustered data, explaining the analyzed data related to the case, and reducing the cost of the video forensic investigation process. In forensic investigations, producing court-admissible reports is crucial for the admissibility of evidence in legal proceedings (Berghs et al., 2018; KAUR & DHAWAN, 2024). By streamlining the investigation process, the tool aims to enhance the efficiency and reliability of forensic analysis. ultimately contributing to the administration of justice (Berghs et al., 2018).

By bridging the gaps identified in existing forensic software and introducing advanced capabilities for video content analysis, this tool is poised to make a significant impact in the digital forensics field. The technological advancements encapsulated in this tool not only meet but exceed current forensic analysis standards, providing both; a more robust methodological approach and a more precise, reliable means of extracting and analyzing video data.

1.4 Research Questions and Objectives

This study addresses the complexities inherent in forensic investigations, particularly concerning analyzing video content. The primary objectives are designed to guide the technological developments of forensic tools and ensure their applicability in real-world legal settings, thereby contributing significantly to the field of forensic science. These objectives include:

- Developing a forensic tool capable of efficiently extracting and analyzing video files from forensic images.

- Implementing advanced AI concepts, such as open AI libraries, for content analysis within video files.
- Generating court-admissible reports containing filtered and clustered video data to streamline the investigation process.
- Enhancing the efficiency and effectiveness of video forensic analysis, thereby reducing investigation time, effort, and costs.

Building on these objectives, this section delineates specific research questions to address these complexities. By exploring these questions, this study seeks to develop innovative solutions that enhance the reliability, efficiency, and legal admissibility of forensic video analysis:

1. **RQ1:** What are the challenges faced by using keyword-based searches in video content analysis within forensic tools, and how do they affect the filtering of relevant artifacts?

2. **RQ2:** How can a forensic tool be designed and implemented to analyze video content more effectively, thereby addressing the limitations of keyword-based searches?

3. **RQ3:** What are the necessary steps and considerations for generating court-admissible reports based on evidence extracted through video content analysis tools?

These questions aim to tackle the core challenges in digital forensic video analysis by using advanced computational methods while ensuring the legal integrity of the investigative process. By aligning the tool's features directly with the research questions, this study seeks to improve forensic analysis and promote a deeper comprehension of how technology and law interact in forensic environments.

1.4.1 Addressing the Research Questions

RQ1: Challenges of Keyword-Based Searches in Video Content Analysis

Keyword-based searches in forensic tools often struggle with the diversity and complexity of video content. Enhancing keyword search with AI-driven context recognition can mitigate these issues by capturing non-textual elements such as actions, or specific visual scenes crucial in forensic analysis.

RQ2: Implementing a Video Content Analysis Tool

To address the challenges identified in RQ1, the proposed forensic tool will incorporate advanced ML algorithms and AI technologies. These will enable the tool to analyze video content beyond simple keywords, recognize speech, detected objects, and textual contents, and interface with existing forensic databases to enhance analytical capabilities.

RQ3: Generating Court-Admissible Reports

The design of the forensic tool will focus on extracting and analyzing data and organizing findings in a legally compliant format. This includes maintaining a detailed chain of custody, using verified algorithms acceptable in court, and providing clear, concise, and factual reporting. Training the tool to generate reports that adhere to legal standards automatically is essential for its adoption in legal proceedings.

The forensic tool will be designed to extract and analyze data, presenting findings in a legally compliant format. This involves ensuring a detailed chain of custody, utilizing court-accepted procedures, and producing clear, concise, and factual reports. Additionally, training

the tool to automatically generate reports that meet legal standards is essential for its use in legal proceedings.

By reducing reliance on manual searches and broadening the scope of automatic video analysis, the proposed tool aims to enhance both the speed and accuracy of forensic investigations, ultimately improving the forensic investigation process.

1.5 Contributions

This study significantly advances the field of digital forensics by developing a novel forensic tool tailored for video content analysis. The contributions of this proposed tool are directly aligned with the challenges and objectives identified earlier in the study, and include:

- Efficient Identification and Analysis: The tool enhances the capability to identify and analyze specific content within video files efficiently.
- Automated Processes: The tool streamlines the investigative process by automating the extraction and analysis of video content.
- Court-Admissible Report: It generates a court-admissible report that contains filtered and clustered video data. This report meets the legal standards necessary for evidence.
- Cost Reduction: The overall efficiency and effectiveness of the forensic tool significantly reduce costs associated with digital forensic investigations.

1.6 Thesis Organization

The thesis is organized in the following chapters:

1. **Chapter 2: Literature Review**
   This chapter critically examines the existing literature and related work in digital forensics, providing a comprehensive overview of current methodologies, tools, and challenges in video content analysis.

2. **Chapter 3: Methodology**
   Chapter 3 details the methodology employed in the development of the proposed forensic tool, outlining the process of the tool construction, experimentation, and validation.

3. **Chapter 4: Experiment Results**
   This chapter presents the empirical results obtained from the experimentation with the forensic tool and offers a detailed discussion of the findings, their implications, and potential limitations.

4. **Chapter 5: Discussion**
   This chapter explores the outcomes of experiments with digital forensic VCAT, evaluating its effectiveness, operational challenges, and implications for digital forensic investigations, and suggesting areas for enhancement within digital forensic science.

5. **Chapter 6: Conclusion and Future Work**
   The final chapter provides a comprehensive conclusion to the study, summarizing key findings, discussing their significance, and proposing avenues for future research and development in the field of forensic video analysis.

Chapter 2

Literature Review

2.1 Background

2.1.1 Exploring the Landscape of Digital Crime Scenes and Forensic Analysis

The field of computer forensics, introduced by (Collier & Spaul, 1992), has evolved into a crucial aspect of forensic science, encompassing the gathering, examination, analysis, and presentation of digital evidence in legal proceedings (Raghavan, 2013). Terms like ”computer forensics”, ”forensic computing”, and ”digital forensics” are often used interchangeably in this context (Schatz, 2007).

Cybercrime, driven by technological advancements, presents a significant threat globally, encompassing offenses from data breaches to cyber-attacks (Carroll, 2017). Modern cybercrimes, including social engineering and ransomware attacks, have become increasingly sophisticated, posing substantial financial and security risks (Hweidi & Eleyan, 2023). According to estimates, worldwide cybercrime might cost the economy ten trillion Dollars a year by 2025, a significant rise from earlier estimates (Morgan, 2020).

As cyber threats evolve, so do digital forensics techniques and technologies. Recent advancements in cryptographic techniques have significantly bolstered the security of digital evidence, ensuring its integrity from acquisition to courtroom presentation (Kolochenko, 2022). For instance, the use of blockchain technology in chain-of-custody protocols represents a cutting-edge approach to managing digital evidence securely and transparently.

Digital crime scenes present unique challenges to forensic investigators due to the intangible nature of digital evidence and its susceptibility to alteration (Cavigline et al., 2017; OliveiraJr et al., 2020). Unlike traditional crime scenes, digital crime scenes encompass vast

volumes of electronic devices, networks, and storage media, providing numerous potential sources for forensic examination (Andrijcic & Horowitz, 2006; Radanliev et al., 2020).

Various forms of digital evidence, including files (documents and media), emails, logs, metadata, and network traffic, require specialized techniques and tools for identification and analysis (Raghavan, 2013). Swift and efficient digital evidence collection is crucial to prevent loss or tampering, given its ephemeral nature (Reedy, 2023).

Meticulous documentation of the digital crime scene, including hardware configurations, network topologies, and potential points of compromise, is essential for establishing a chain of custody and ensuring evidence integrity (Bokolo & Liu, 2024; Reedy, 2023). However, investigators often lack a standardized model to follow (Matijević Gostojić & Vuković, 2023).

Digital forensic experts maintain evidence integrity by creating forensic images of seized hard drives in read-only mode, aiding in file analysis, and detecting remnants of deleted content (Anderson & ENISA, 2015; Kent et al., 2006).

The proliferation of digital devices and data volumes poses significant time constraints. To address this challenge, live system triaging allows for extracting evidence from volatile digital artifacts such as Random Access Memory (RAM) and network connections, enhancing efficiency (Cavigline et al., 2017).

Encryption, anonymization techniques, and anti-forensic tools present additional challenges to digital forensic investigations (Berghs et al., 2018). For instance, enabling the TRIM function on solid-state drives (SSD)s can internally wipe unused data blocks, potentially affecting data integrity (Hweidi et al., 2023; M. Kumar, 2021).

Staying abreast of evolving technologies and methodologies is crucial for investigators to counter these challenges effectively. The Association of Chief Police Officers (ACPO) (Casino et al., 2022). Proper utilization of formal acquisition methods, as outlined in ACPO principles for digital evidence examination, can help preserve data integrity by accessing valuable content and metadata not readily accessible through standard device operating procedures (Horsman, 2020).

2.1.2 The Digital forensic Process

This study followed and adhered to NIST (National Institute of Standards and Technology) standards and recommendations in digital forensic investigation (Kent et al., 2006). The most relevant NIST guidelines are 800-86, which provide a systematic approach to the digital forensic process as shown in Figure 2.1, encompassing four key steps:

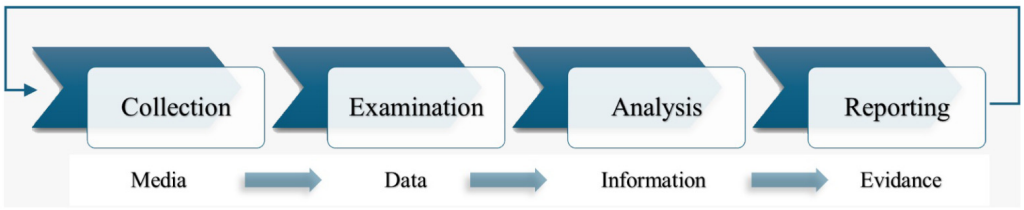


Figure 2.1: The Digital forensic Process (Kent et al., 2006).

1. **Collection**: This involves locating and recording credible sources of data relevant to the incident, followed by acquiring data from these sources while maintaining their integrity.

2. **Examination**: This entails assessing the data obtained during the collection phase, with a focus on extracting relevant information related to the incident while preserving the integrity of the data.

3. **Analysis**: During this step, the information extracted during the examination phase is studied to address pertinent investigative questions and determine if a conclusive or partial conclusion can be reached.

4. **Reporting**: Consists of preparing and presenting the investigation's procedures, methodologies, and tools utilized, along with the outcomes derived from the analysis phase.

Additionally, the study followed the Organization of Scientific Area Committees for Forensic Science (OSAC) standards, and the Scientific Working Group on Digital Evidence (SWGDE) guidelines for conducting digital forensic examinations and analysis for video files ("Standards open for comment | NIST", 2024). SWGDE is an organization engaged in the field of digital and multimedia evidence to foster communication and cooperation as well as to ensure quality and consistency within the forensic community, it has released several documents to provide the current best practices on a large variety of SOAT forensics subjects. This study ensured methodological rigor and reliability in the analysis of digital evidence, facilitating a thorough and objective investigation process.

2.1.3 The Crucial Role in Digital Forensic Analysis of Seized Videos

Video Quality and Compression Issues

Low-quality footage and compression can hinder forensic analysis by causing the loss of crucial details and making recovery difficult. Variations in brightness require manual adjustments for accurate analysis. Additionally, factors like camera angles, object rotation, and the number of visible features significantly impact the analysis's effectiveness (Xiao et al., 2019).

Forensic Identification and Emerging Technologies

Forensic identification is challenging due to heterogeneous face recognition and inadequate reference samples. Emerging technologies like IoT and social networks require advanced forensic tools, which current techniques do not fully support. Effective connection recognition with various situations and databases is also essential for comprehensive analysis (Javed et al., 2021).

Digital Forensics: Acquisition, Representation, and Storage

In digital forensics, video acquisition, representation, and storage are crucial for maintaining the integrity and usability of visual data for analysis and evidence.

Acquisition

Acquisition involves capturing digital images and videos from sources like digital cameras, smartphones, CCTV systems, and drones. Preserving the original content and metadata without alteration is vital for forensic authenticity. Accurate timestamps and geolocation metadata are pivotal for verifying event timelines or locations in criminal investigations (Daraghmi & Shawahna, 2023).

Representation

Videos must be encoded and compressed using standard formats like MPEG or H.264 for both analysis and storage. Maintaining high fidelity to the original scene is crucial for tasks like facial recognition and license plate identification. Lossy compression can distort details and introduce artifacts that mislead investigations. Proper format and compression settings ensure critical details are preserved while maintaining manageable file sizes (Xiang, 2024).

Storage

Storing videos securely, accessibly, and efficiently is essential. Forensic data needs protection against unauthorized access and corruption to ensure court admissibility. This involves robust file systems, secure backups, and often redundant storage mechanisms. The choice of storage technology, such as SSDs, Hard Disk Drives (HDD)s, or cloud storage, depends on access speed, data volume, and long-term preservation (Javed et al., 2021).

Advancements in cloud storage offer scalable and secure options for storing large data volumes. Redundant storage systems and error-correcting codes mitigate challenges like data degradation, ensuring data recovery and integrity (Javed et al., 2021).

By adhering to best practices in acquisition, representation, and storage, forensic professionals can maintain the chain of custody and ensure the reliability of digital evidence throughout investigations (Bokolo & Liu, 2024). This supports thorough and accurate analysis and strengthens the legal admissibility of evidence. Therefore, these methods are integral to the success of forensic investigations and subsequent judicial proceedings.

2.1.4 Digital Forensic Tools

Autopsy

Autopsy is an open-source digital forensics platform developed by Basis Technology. Known for its comprehensive and efficient capabilities, Autopsy is designed to handle hard drive investigations with speed and thoroughness. It offers features similar to those found in commercial forensic tools, making it a versatile solution that evolves with users' needs. Autopsy does analyze video content, allowing investigators to extract and review metadata, frame-by-frame analysis,

and other video-related evidence to support their cases. It supports metadata analysis, flagging frames, and the use of third-party modules (Technology, 2023).

OSForensics

OSForensics is a digital investigation tool designed for quick and efficient extraction of forensic data from computers. It aims to uncover hidden information within PCs, streamlining the forensic process for investigators. OSForensics does not specifically emphasize video content analysis but focuses broadly on data retrieval from various computer components. It does not support metadata analysis, flagging frames, or third-party modules for video content analysis (investigation for a new era by PassMark Software, 2023).

EnCase

EnCase Forensic by OpenText is a widely used digital investigation tool, known for its robust data collection and investigation capabilities. It supports a wide range of file systems and encryption tools, making it ideal for comprehensive disk-level investigations (OpenText, 2023). While EnCase does not offer built-in video content analysis features, it includes functionalities for detailed examination of video files and metadata and supports scripting and third-party modules for advanced analysis.

MAGNET AXIOM

MAGNET AXIOM by Magnet Forensics integrates data from mobile, cloud, computer, and vehicle sources into a single case file. It features intuitive analytical tools to surface relevant evidence quickly. AXIOM has limited support for video content analysis, including geolocation data review, which can help in locating CCTV footage. It also

supports scripting and third-party modules for advanced study. However, specific capabilities like object detection, speech recognition, or OCR are not explicitly mentioned in their site (the truth. Protect the innocent, 2023).

MOBILedit Forensic

MOBILedit Forensic is an all-in-one data extraction tool for phones, smartwatches, and cloud services. It supports physical and logical data acquisition and excels in application analysis and deleted data recovery (Labs, 2023). MOBILedit Forensic does not inherently support video content analysis features like object detection, speech recognition, or OCR. It also does not support metadata analysis, flagging frames, or third-party modules for video content analysis.

FINALMobile Forensics

FINALMobile Forensics offers advanced data-carving tools, making raw data comprehensible information with minimal effort. It is adept at recovering deleted entries and analyzing specific data patterns within mobile devices (Finaldata, 2023). However, it does not support video content analysis. It also does not support metadata analysis, flagging frames, or third-party modules for video content analysis.

Forensic Toolkit (FTK)

FTK by Exterro is a comprehensive digital forensics tool for its image labeling and categorization capabilities. FTK provides context for images by reconstructing user activity leading up to and following the creation of the image through built-in mini timelines. It also includes investigator wellness settings to reduce exposure to sensitive content

and offers AI support for advanced analysis ("FTK Forensics Toolkit - Digital Forensics Software Tools", 2024).

Several forensic tools share common capabilities such as metadata analysis, flagging frames, and the use of third-party modules for enhanced functionality. These tools, including Autopsy, EnCase, MAGNET AXIOM, and FTK, allow investigators to extend their analysis capabilities beyond the built-in features. While most tools do not support object detection, speech recognition, or OCR, they provide robust frameworks for integrating specialized analysis tools. Autopsy is free, while OSForensics, EnCase, MAGNET AXIOM, MOBILedit Forensic, FINALMobile Forensics, and FTK are commercial products, often with support for AI-driven analysis and automation.

Generative Pre-Trained Transformers (GPTs) Tool

In addition to traditional forensic tools, various video analysis programs are designed for specific purposes. GPTs for video analysis can summarize content, extract key information, and provide insights, making long-form content more accessible. Available to ChatGPT Plus subscribers, these GPTs operate within the ChatGPT interface and are mostly free, though some offer premium features ("Best GPTs for Video Analysis With ChatGPT", 2024). Despite their utility, these tools face challenges in digital forensic analysis, particularly in producing reports with a clear chain of custody, which is crucial for judicial acceptance.

Forevid

Forevid is a non-commercial tool designed for enhancing and analyzing video files from sources like CCTV, surveillance cameras, mobile devices, and more. It includes features such as screen recording, frame exporting, bookmarking, applying filters, encoding videos, viewing

metadata, and editing frames. It is available for Windows and MacOS with a free trial option ("Video Analyzer Software: Forensic Video Analyzer Software - Analysis Video File Formats", 2024).

Videoma Intelion

Videoma Intelion is a high-cost video and audio analyzer for law enforcement and intelligence agencies. It automates the review and documentation of video and audio from various sources, using advanced AI to classify information and locate targets in near real-time. Features include face biometry, object recognition, speaker ID, audio fingerprinting, speech-to-text, and automatic translation ("Videoma Intelion", 2024).

Metadata

Metadata includes data about data, such as creation date, location, frame rate, and device information, which is crucial in forensics for verifying the authenticity and integrity of evidence, and aids in establishing the chain of custody and authenticity of digital evidence, making it invaluable in legal contexts. Tools like Exiftool are used to write and extract metadata, providing critical information that supports forensic analysis (Harvey, 2024).

2.1.5 The Admissibility of AI Evidence in Forensic Investigations

Despite the rapid integration of AI into various sectors, the legal standards for its admissibility in civil and criminal trials remain underdeveloped. As AI technology advances quickly, the legal framework struggles to keep pace, leading to uncertainties about how AI outputs should be evaluated in court. Many experts have raised concerns about the challenges this poses for the admissibility of AI evidence, yet few

court decisions have directly addressed these issues under existing evidence rules. This gap in the legal framework necessitates that legal professionals gain a foundational understanding of AI operations, its scientific assessment, and the factors affecting its reliability and validity to judge its evidentiary value competently. Currently, there are no specific rules in the Federal Rules of Evidence that directly cater to AI evidence, leaving legal practitioners to rely on existing rules to navigate these new challenges. This scenario calls for a critical examination of AI's role in legal disputes and prepares the ground for potential future regulations that could provide clearer guidance. This discussion aims to equip lawyers and judges with the necessary knowledge to handle AI evidence judiciously, ensuring fair trial outcomes where AI evidence is involved (Grimm et al., 2021).

2.1.6 Integrating Artificial Intelligence in Digital Forensics: A New Era of Investigation

The field of digital forensics has significantly advanced with the integration of AI, enhancing forensic capabilities and introducing new investigative techniques (Karie et al., 2019). Understanding AI's concepts, history, and applications is crucial for grasping its impact on forensic investigations.

2.1.6.1 Definition

AI refers to creating computer systems that can perform tasks traditionally requiring human intelligence, such as reasoning, learning, problem-solving, perception, and language understanding. These systems mimic human cognitive functions, allowing them to analyze complex data, identify patterns, and make autonomous decisions. AI technologies are not only capable of operating at high speeds and efficiency but can also engage in human-like interactions to enhance user engagement through conversation that simulates natural dialogue,

making these applications both effective and user-friendly (Kubassova et al., 2021).

2.1.6.2 History

The concept of AI, formalized by John McCarthy in 1956, has evolved significantly, offering powerful tools for forensic investigations while raising ethical concerns about privacy and algorithmic bias (McCarthy et al., 2006). AI's future in forensics may include predictive policing, using data to forecast criminal activities and optimize resource allocation.

This research has undergone periods of intense growth and relative stagnation, known as AI springs and winters. Breakthroughs have included the development of expert systems in the 1970s and significant advancements in neural networks and natural language processing in the 1980s (Fradkov, 2020; Schneider et al., 2024).

The 1990s marked a shift towards practical applications with the introduction of the Support Vector Machine (SVM) algorithm by (Cortes & Vapnik, 1995), which greatly enhanced both practical applications and theoretical understanding in AI (Fradkov, 2020). The rise of Big Data in the early 21st century required new analytical approaches (Montasari, 2023), supported by advancements in parallel computing, memory technologies, and the introduction of graphics processing unit (GPU) enhancements by companies like Nvidia. These developments facilitated the launch of technologies such as Apache Spark in 2014, optimizing the processing of unstructured data (Fradkov, 2020).

AI is divided into two types: weak AI (narrow AI) and strong AI (general AI). Weak AI performs specific tasks within a limited scope, like chatbots and recommendation systems (Alrajhi, 2020). Strong

AI, aiming to replicate human cognitive abilities, remains theoretical and unrealized (Fähndrich et al., 2023).

2.1.7 Terms of Artificial Intelligence (AI)

AI encompasses diverse terms, each addressing distinct facets of intelligent behavior:

2.1.7.1 Machine Learning

ML is a field that combines computer science and statistics to create algorithms enabling computers to learn from data and make decisions. ML integrates concepts from psychology, neuroscience, and adaptive control theory, demonstrating its interdisciplinary nature (Harrington, 2012; Jordan & Mitchell, 2015). It processes and analyzes large datasets, learning from experiences to perform tasks. ML is defined as enhancing performance through data-driven predictions and decisions (Sarker, 2021). IBM describes ML as part of AI, focusing on using data and algorithms to improve learning and accuracy (IBM, 2024).

The success of ML is contingent on algorithms' ability to generalize from training data to unseen situations, with models chosen based on the specifics of the data and the desired outcome (Barbierato & Gatti, 2024). Algorithms range from simple linear regressions for small datasets to complex methods like DL for larger sets (Maabreh & Almasabha, 2024; Maulud & Abdulazeez, 2020; Tyralis & Papacharalampous, 2024). Despite its advances, ML must be applied cautiously to avoid illogical outcomes, emphasizing the importance of sound principles in model training (Pugliese et al., 2021). Furthermore, the field is divided into subfields based on the type of learning task, each suited to different kinds of data signals and feedback mechanisms. ML techniques are categorized into four types, based on how

they learn and process data: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Barbierato & Gatti, 2024; Bell, 2022; Pugliese et al., 2021).

1. **Supervised Learning:** This method uses labeled data to predict outcomes. Algorithms like Naïve Bayes, linear regression, and SVMs learn by mapping inputs to the correct output using known input-output pairs (Barbierato & Gatti, 2024; Bell, 2022). This process involves minimizing errors between predicted and actual outputs (Pugliese et al., 2021). Common applications include email filtering, face and speech recognition, and data center optimization (Bell, 2022). The primary advantage is explicitly classifying data into known categories, which is critical for tasks like predictive modeling. However, the downside is that supervised learning requires extensive computation and cannot discover new categories or features independently (Pugliese et al., 2021).

2. **Unsupervised Learning:** Unlike supervised learning, unsupervised learning algorithms identify patterns in unlabeled data, clustering similar items together based on inherent similarities without prior labels (Barbierato & Gatti, 2024; Bell, 2022; Pugliese et al., 2021). This approach is useful in fields like social network analysis, market segmentation, and computational biology. One of its strengths is the ability to uncover hidden patterns that are not immediately obvious, offering valuable insights for anomaly detection like fraud (Pugliese et al., 2021). However, it often requires more human intervention to make sense of the output and lacks the precision of supervised methods due to the absence of labeled data to guide the learning process (Barbierato & Gatti, 2024; Bell, 2022; Pugliese et al., 2021).

3. **Semi-Supervised Learning:** This method combines both labeled and unlabeled data, which helps in improving learning accuracy. It is less data-intensive than supervised learning but offers more control than unsupervised learning, making it suitable for tasks like text classification and semantic scene classification. However, results can be unstable, and it generally offers lower accuracy compared to fully supervised learning (Barbierato & Gatti, 2024; Bell, 2022; Pugliese et al., 2021).

4. **Reinforcement Learning:** This type focuses on making sequences of decisions, learning from the consequences of actions to maximize some notion of cumulative reward, and is particularly effective in dynamic environments where the algorithm must make decisions in real-time, such as in autonomous driving and gaming (Barbierato & Gatti, 2024; Bell, 2022; Pugliese et al., 2021). Reinforcement learning mimics human learning behaviors, adjusting its strategies based on the feedback from the environment to optimize outcomes. The challenges include managing the complexity of different states and the high data and computational needs (Pugliese et al., 2021).

2.1.7.2 Deep Learning

DL represents a sophisticated branch of ML and AI, characterized by its use of artificial neural networks (ANNs) Inspired by the human brain's neuronal structures (Shahinzadeh et al., 2024). These networks use multiple layered architectures to extract and learn higher-level features from raw data inputs. This capability enables significant advancements in tasks like image and speech recognition, machine translation, and autonomous driving (Heaton, 2018; LeCun et al., 2015).

The following sections detail the primary architectures of ANNs and highlight their specific applications and impacts across various domains.

- **Feedforward Neural Networks (FFNNs)** are the simplest type of ANNs where connections between the nodes do not form a cycle. Primarily used for basic prediction and classification tasks, FFNNs process information in a straightforward manner, with data moving in one direction from input to output layers (Bebis & Georgiopoulos, 1994; Sanger, 1989). This structure makes them particularly effective for static pattern recognition and simple predictive modeling tasks.
- **Convolutional Neural Networks (CNNs)** are designed for visual data analysis and pattern recognition, leveraging the spatial structure of images through convolutional layers that automatically detect features like edges, textures, and objects (Albawi et al., 2017; Shrestha & Mahmood, 2019). These networks are advantageous due to their ability to capture complex spatial hierarchies through hierarchically structured convolutional layers, making them highly suitable for image analysis. Additionally, the use of reduced weights in convolutional filters reduces the number of parameters and enables efficient feature detection (Krichen, 2023). Applications of CNNs include object detection and lane detection in self-driving cars, which aid autonomous vehicles in navigating safely (Alabyad et al., 2024). In medical image analysis, CNNs assist in diagnosing conditions from radiology images, while in facial recognition, they help identify and verify individuals based on their facial features (Albawi et al., 2017).
- **Recurrent Neural Networks (RNNs)** are designed for sequential data like speech or text, with internal memory to process

varying sequence lengths (De Mulder et al., 2015; Mikolov et al., 2010). Advanced variants like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) address vanishing gradient issues using memory cells and gates to control information flow (Biniyaz & Liu, n.d.; Hochreiter & Schmidhuber, 1997; Shrestha & Mahmood, 2019). Bidirectional RNNs pass information forward and backward, providing context from both past and future states. Transformers, with attention mechanisms, enable focusing on specific input sequence parts (Fu et al., 2024; Vaswani et al., 2017). RNNs excel at contextual learning, widely used in Natural Language Processing (NLP) for tasks like language modeling, text generation, machine translation, and question answering (De Mulder et al., 2015; Sutskever et al., 2014). In speech recognition, they convert speech to text and identify speakers (Graves & Jaitly, 2014; Shafey et al., 2019). They are crucial in time series forecasting, predicting future values in financial data, weather patterns, and anomaly detection (Hochreiter & Schmidhuber, 1997). In video analysis, RNNs facilitate activity recognition, video captioning, and action prediction (Abbas et al., 2018; Donahue et al., 2017).

In general, DL's enhanced algorithms for trading and fraud detection in finance (Barongo & Mbelwa, 2024; Huang et al., 2020), improved healthcare diagnostics and treatment personalization (Lei et al., 2024; Orji & Ukwandu, 2024), strengthened cybersecurity measures and digital forensics investigation (Chakraborty et al., 2023; Gonaygunta, 2023), and automated content moderation on social media (Karabulut et al., 2023). Although big deep neural network (DNN) models trained on large amounts of data have recently achieved the best accuracy on hard tasks, the distributed training DL models require significant computational resources and vast data, challenging

scalability and efficiency (Yan et al., 2015).

2.1.7.3 Foundation Models in Computer Vision

Introduced by Stanford's Human-Centered AI Institute in 2021, foundation models redefine AI's capabilities. These models, characterized by their training on vast, diverse data via self-supervision, are adaptable to a wide array of downstream tasks (Bommasani et al., 2022). Foundation models like Transformers exhibit a high level of homogenization and can perform an extensive range of tasks without further fine-tuning (Vaswani et al., 2017). Foundation Models represent a shift from traditional models by focusing on four critical areas:

- **Model Architecture:** The architecture of foundation models is designed for parallel processing, which allows them to be trained efficiently on supercomputers. These models integrate seamlessly into various AI applications due to their flexible and scalable nature (Bommasani et al., 2022; Devlin et al., 2019; Zhou et al., 2024).
- **Data Utilization:** Leveraging the largest data sources created during the digital age, these models learn from a vast array of un-labeled data, which significantly diminishes the costs associated with data annotation (Bommasani et al., 2022).
- **Computational Resources:** The development of foundation models has been propelled by unprecedented advancements in hardware capabilities, such as GPUs, which facilitate the training of these highly demanding models (Bommasani et al., 2022; Vaswani et al., 2017).
- **Prompt Engineering:** Foundation models can be prompted (textual, visual, and others) to perform tasks using natural language, making them extremely versatile in applications (Awais et al., 2023; Bommasani et al., 2022) such as Whisper, The

Grounded Language-Image Pre-training (GLIP) and GroundingDINO (Awais et al., 2023). However, structuring these prompts requires an understanding of the underlying meta-language used to interact with these models (Radford et al., 2021).

The application of foundation models extends across various domains, from simple object recognition tasks to complex scenarios involving multiple modalities including but not limited to Open AI which is explained in the section Generative artificial intelligence (GenAI), OpenAI (Brown et al., 2020; Meng et al., 2023). These models not only enhance the current capabilities in fields like object detection and segmentation but also pave the way for innovations in 3D understanding and commonsense reasoning within AI (Vaswani et al., 2017). As AI continues to evolve, the role of foundation models becomes increasingly integral, necessitating a thorough understanding of their taxonomy, capabilities, and potential impacts on society (Rothman & Gulli, 2022). With ongoing research and development, foundation models are expected to become more sophisticated, offering even more possibilities for enhancement and application in AI (X. Li et al., 2023; Schneider et al., 2024).

2.1.7.4 Generative artificial intelligence (GenAI)

GenAI focuses on developing systems that can independently create new content, representing a significant advancement in AI (Feuerriegel et al., 2024). A prime example is Generative Adversarial Networks (GANs), which are trained to generate realistic images, texts, or musical pieces by analyzing large datasets of existing content (Goodfellow et al., 2014). GANs enhance their performance by recognizing patterns within these datasets. Similarly, Large Language Models

(LLMs) use transformer architecture to perform tasks such as text generation, language translation, creative writing, and answering complex questions (Hadi et al., 2023; Luitse & Denkena, 2021).

This transformative wave in AI began with OpenAI's introduction of GPTs, which utilize a neural network architecture designed for NLP tasks (Hadi et al., 2023; Hassija et al., 2023). This architecture enables computers to understand, interpret, and generate human language, supporting applications across machine translation, sentiment analysis, and text summarization (Luitse & Denkena, 2021).

Simultaneously, the field of Computer Vision strives to enable machines to interpret and understand visual information from the real world (Gollapudi, 2019; Szeliski, 2022). Key applications such as object detection, image classification, and facial recognition drive advancements in this area (Szeliski, 2022; Voulodimos et al., 2018). Traditionally, DL models in this domain have relied on extensive, annotated datasets for training, a process that can be both costly and time-consuming. Recent trends, however, show a shift towards pre-training these models on large-scale, uncurated web data, seeking to leverage the vast availability of digital content to enhance learning efficiency and model robustness (Voulodimos et al., 2018).

2.1.7.5 OpenAI: Pioneering the Future of Artificial Intelligence

OpenAI, established in 2015 as a non-profit research organization, quickly gained attention for its groundbreaking advancements in AI (Brockam et al., 2015). Within five days of its launch, the OpenAI language model attracted over one million users, setting a record for rapid user adoption (Buchholz, 2023).

Since introducing ChatGPT in November 2022, OpenAI has drawn significant interest from knowledge workers, developers, and internet

users (Rudolph et al., 2023). Beyond ChatGPT, OpenAI has been a leader in AI research and development for years, being among the first platform companies to offer access to GenAI through simple REST API endpoints (Roumeliotis & Tselikas, 2023).

Alongside other influential entities like Google DeepMind, Facebook AI Research, and Microsoft Research, OpenAI has played a pivotal role in shaping the modern AI era (Aprill et al., 2024). The organization's GPT series, DALL-E, and CLIP are notable projects, demonstrating its commitment to advancing AI technologies (Bommasani et al., 2022; Chakraborty et al., 2023). These models, including ChatGPT built on GPT-3.5 and GPT-4 foundation models, have been integrated into various applications (Bommasani et al., 2022).

Following the development of the GPT series, OpenAI transitioned from a non-profit to a "capped" for-profit model to enhance its ability to attract venture capital and top talent, a move that supports its ambitious projects (Andhov, 2024; Aprill et al., 2024). Today, OpenAI offers a broad array of products, including the GPT-3 and GPT-4 APIs, Whisper for audio processing (speech-to-text and translation tasks), DALL-E for image creation, and ChatGPT for conversational AI. Additionally, OpenAI's influence extends to products like Google Gemini, Microsoft Bing (Copilot), and Duolingo Max, demonstrating its expansive impact across various fields (Rudolph et al., 2023).

Key attributes of OpenAI models include high accuracy, rapid processing speeds, scalability, and sophisticated NLP capabilities. Trained on extensive datasets, these models are adept at making precise predictions and efficiently handling large volumes of data, making them ideal for real-time applications (Alto, 2023). The scalability of OpenAI models ensures adaptability to varying project demands, while their advanced NLP features allow for effective management of

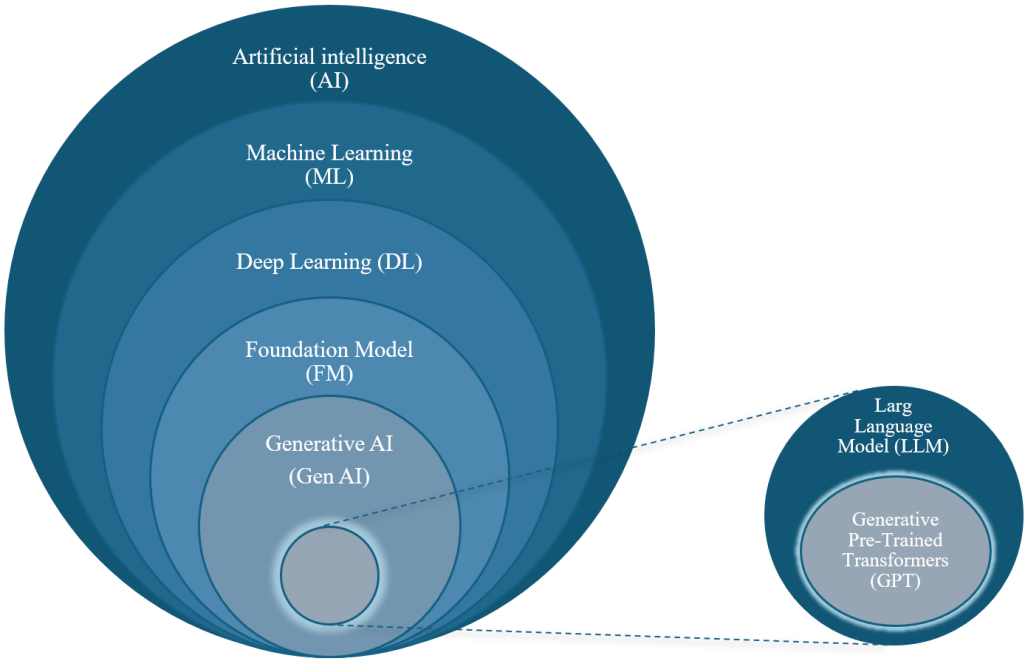


Figure 2.2: The Layers of AI.

complex linguistic tasks (Haque, 2023).

These characteristics underscore OpenAI's pivotal role in pushing the boundaries of AI technology, a development that has significant implications for the era of digital forensic investigation. The advanced capabilities of OpenAI models, particularly in processing large datasets swiftly and handling complex language tasks, are reshaping how digital evidence is analyzed, offering more sophisticated tools for extracting, processing, and interpreting data in forensic contexts.

Figure 2.2 encompasses the hierarchy and relationship between various AI technologies.

2.1.8 Models in Artificial Intelligence

2.1.8.1 History and Evolution of Object Detection

Object detection is a fundamental topic in computer vision, where advancements in DL have significantly revolutionized the field. The

history of object detection spans several decades with continuous research and development exploring various methodologies and models:

Initial Concepts (1960s - 1990s): Early research explored fundamental ideas like feature extraction and template matching (Brunelli, 2009). Techniques such as the Hough Transform and Scale-Invariant Feature Transform (SIFT) enabled rudimentary object localization by matching fixed patterns to specific image features (Calitz, 1995).

Machine Learning Era (2000s): This era saw significant advancements:

- Viola and Jones (2001) introduced a face detection framework using Haar features and AdaBoost, marking a major leap in detection efficiency.
- Dalal and Triggs (2005) introduced the Histogram of Oriented Gradients (HOG) feature descriptor, which significantly improved object detection performance.
- Felzenszwalb et al. (2010) built on HOG features with Deformable Part Models, which handled part-based object detection effectively.

Deep Learning Revolution (2010s - Present):

- Regions with Convolutional Neural Networks (RCNN): Girshick et al. (2016), Girshick et al. (2014) proposed RCNN, which used selective search to generate region proposals and CNNs to classify and refine them.
- Fast RCNN: Improved by introducing Region of Interest (ROI) pooling (Girshick, 2015).
- Faster RCNN: Integrated Region Proposal Network (RPN) for end-to-end region proposal generation (Ren et al., 2017).

- You Only Look Once (YOLO): Redmon and Farhadi (2018) introduced a unified, real-time object detection model, improving localization and classification accuracy through batch normalization and multi-scale predictions in YOLOv1 to v3. Through YOLOv4's enhanced speed, YOLOv5's model variants, and YOLOv6's multi-head detection head, YOLOv7 achieved SOAT results in object detection (Bochkovskiy et al., 2020; Terven et al., 2023; C.-Y. Wang et al., 2023).
- Single Shot MultiBox Detector (SSMD): W. Liu et al. (2016) proposed SSMD, which provided real-time detection using a multi-scale feature map approach.

Recent Advances SOAT (2020s):

- **YOLOv8**: proposed by Ultralytics in 2023, offers unparalleled speed and accuracy in real-time object detection and image segmentation, adaptable across diverse hardware platforms with AI flexibility and prompt engineering (Kang & Kim, 2023; Sandhya & Kashyap, n.d.; Ultralytics, 2024).
- **Detection Transformer (DETR)**: Carion et al. (2020) introduced DETR, leveraging transformer architecture for end-to-end object detection. This model simplifies the object detection pipeline by eliminating the need for non-maximum suppression and anchor generation, streamlining the entire process. It matches the accuracy and runtime performance of the established Faster RCNN on the COCO dataset.
- **DETR with Improved Denoising Anchor Boxes (DINO)**: Introduced by Zhang et al. in 2022, is an enhanced iteration of the original DETR model that includes better optimization techniques. This model notably improves object detection performance through several key innovations: it uses a contrastive method to reduce noise during training, applies a blended query

selection for effective anchor initialization, and incorporates dual future prediction strategies to accurately predict bounding box locations.

After completing pre-training on the Objects365 dataset using a SwinL backbone, DINO demonstrated remarkable performance on the COCO dataset. It achieved an Average Precision (AP) of 63.2% on the val2017 split and 63.3% on the test-dev split. Additionally, DINO significantly reduces both model size and pre-training data requirements compared to earlier models, providing a more efficient solution for high-performance object detection (Zhang et al., 2022).

- **GLIP Model**: GLIP signifies a substantial advancement in visual representations by integrating object-level precision, language sensitivity, and semantic depth. Unlike traditional object detection methods that assign categorical classes to detected (Zareian et al., 2021), GLIP merges detection with phrase grounding during its pre-training phase, redefining detection as a grounding task. This innovative approach utilizes combined datasets from both; detection and language processing tasks, enhancing the model's performance across both domains and fostering a robust grounding framework (L. H. Li et al., 2022).

  GLIP exploits extensive image-text pairs, autonomously generating grounding boxes during self-training. This method enriches the semantic granularity of the learned representations, significantly enhancing the model's capabilities. In its pre-training phase, GLIP processed a comprehensive dataset comprising 27 million grounding data points, including 3 million human-annotated and 24 million web-crawled image-text pairs. This extensive training has equipped GLIP with formidable zero-shot and few-shot capabilities, evident in its performance across various object recognition tasks. Notably, GLIP achieves

49.8 AP on the COCO dataset (Lin et al., 2014), and 26.9 AP on LVIS dataset (Gupta et al., 2019), outperforming several supervised models without prior exposure to these datasets during training. When fine-tuned on COCO, GLIP attains 60.8 AP on the validation set and 61.5 on test-dev, surpassing previous SOAT models (L. H. Li et al., 2022).

Furthermore, GLIP demonstrates remarkable adaptability in a one-shot scenario across 13 downstream object detection tasks, rivaling fully supervised models. The unified framework of GLIP reduces costs by leveraging scalable image-text data and enhances flexibility and applicability to a broad spectrum of detection scenarios. This model sets new benchmarks in object detection, significantly improving efficiency and adaptability in visual recognition tasks (L. H. Li et al., 2022).

- **GroundingDINO SOAT in Open-Set Object Detection**: GroundingDINO, developed by IDEA Research in 2023, sets a new standard in open-set object detection with its advanced visual foundation model(S. Liu et al., 2023). This model architecture as shown in 2.3 effectively integrates a text-guided grounding strategy using Grounded Language-Image Pretraining (GLIP) (L. H. Li et al., 2022), and incorporates foundational principles of DINO (Zhang et al., 2022). A standout feature of GroundingDINO is its proficiency in zero-shot object detection, allowing it to identify new object classes without needing pre-labeled training data. This capability is further enhanced by sophisticated prompt engineering, emphasizing the need for detailed, descriptive prompts to avoid non-detection and ensure optimal performance (Son & Jung, 2024).

  GroundingDINO diverges from traditional object detection models by not requiring extensively annotated datasets. Instead, it recognizes objects from unannotated classes through

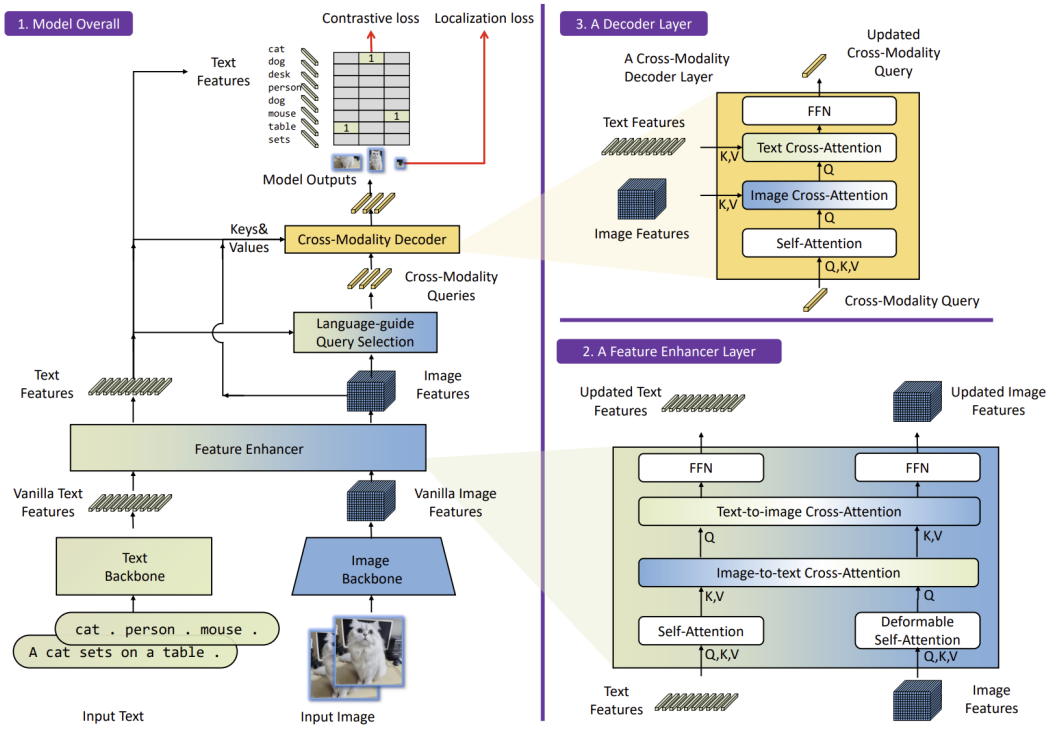


Figure 2.3: GroundingDINO Architecture (S. Liu et al., 2023).

a strategic process of 'grounding' visual features with corresponding class names, facilitated by the pre-trained DINO model (S. Liu et al., 2023). This approach promotes a dynamic interaction between visual and textual data, enhancing the model's ability to adapt to new and unseen object categories via a Cross-modality decoder that analyzes each query's self-attention and focuses on relevant image features based on the query.(S. Liu et al., 2023; Zhang et al., 2022).

The model's architecture, based on transformers, aligns textual prompts with image features (like Swin Transformer) and utilizes multimodal transformers to improve text-guided detection (like BERT). GroundingDINO achieves enhanced accuracy through efficient Non-Maximum Suppression and streamlined

training strategies (S. Liu et al., 2023; Zhang et al., 2022). Performance benchmarks indicate significant superiority over existing models, including notable improvements in mean Average Precision (mAP) scores—achieving 63.2% after fine-tuning with the COCO dataset and 52.5% on the COCO zero-shot transfer benchmark, thus surpassing DINO by 5% (S. Liu et al., 2023).

2.1.8.2 History and Evolution of Optical Character Recognition (OCR)

OCR is a crucial technology in computer vision, converting different types of documents, such as scanned paper documents, PDF files, images, or video captured by a digital mobile, into editable and searchable data. The OCR process is divided into transforming images into editable text, following a multi-step approach. First, it pre-processes the image to enhance clarity. Then, it segments the image to isolate the text regions (Memon et al., 2020). Next, it extracts features that define the characters.

Subsequently, it recognizes these features and translates them into letters. Finally, it post-processes the recognized text for accuracy (Avyodri et al., 2022). The evolution of OCR spans several decades, marked by significant advancements that have enhanced its accuracy and applications in various fields, including digital forensics:

Initial Concepts (1950s): In the 1950s Early OCR systems were designed to recognize specific fonts and were used primarily for reading text from printed documents (Memon et al., 2020). The first commercial OCR system was introduced by Reader's Digest in the 1950s, named GISMO, which used optical scanning to read characters printed in a particular typeface (Islam et al., 2017)

Pattern Recognition and New Technologies (1960s - 1970s): Innovations such as Intelligent Character Recognition (ICR) and Magnetic Ink Character Recognition (MICR) technologies emerged, enhancing the capabilities of OCR systems. These technologies enabled better recognition of printed text and were particularly useful in banking for processing checks (Islam et al., 2017)

Template Matching and Feature Extraction (1980s - 1990s): Advanced OCR techniques based on template matching and feature extraction allowed the recognition of multiple fonts and styles, although systems still struggled with variations in handwriting and image quality (Mori et al., 1992; J. Wang, 2023)

Machine Learning Era (2000s): Combining machine learning algorithms revolutionized OCR by enhancing adaptability and accuracy. Neural networks enabled OCR systems to learn from large datasets, improving performance over time (J. Wang, 2023). In 2007, Smith developed Tesseract, an open-source OCR engine that utilized machine learning techniques to achieve high accuracy, particularly in recognizing complex and variable fonts (Memon et al., 2020)

Deep Learning Revolution (2010s - Present): The integration of deep learning, particularly CNNs, LSTM networks, brought about significant improvements in OCR capabilities, enabling the handling of diverse text styles and qualities with high accuracy by learning hierarchical features from raw image data (Memon et al., 2020; J. Wang, 2023). End-to-end OCR approaches enhance text detection and transcription by jointly optimizing both components (H. Li et al., 2017). Techniques such as using a differentiable region of interest slide operator to address distorted text (W. Feng et al., 2019) and integrating region proposal networks with text and character segmentation (Liao

et al., 2021), effectively reduce errors (A. Kumar et al., 2023; Subramani et al., 2020).

Present-day OCR systems leverage foundational models to provide enhanced document, image, and video processing capabilities, transcribing a wide range of languages and fonts, including non-Latin scripts, with remarkable accuracy (Chaudhury et al., 2022).

State-of-the-Art (SOAT) Architectures: Jiao et al. (2024) conducted a thorough empirical investigation to integrate both the object detection model (GroundingDINO) and OCR model (PaddleOCR) into Multimodal Large Language Models (MLLM)s. Additionally, they explored the potential of open-set object detection models to enable question-driven detection and validated the sustained efficacy of their method following the substitution of detection models. The resulting enhanced MLLMs outperform SOTA models on 9 out of 10 benchmarks, achieving an improvement of up to 12.99% on the normalized average score, marking a notable advancement in multimodal understanding.

Wadhawan et al. (2024) introduced CONTEXTUAL, a dataset designed to evaluate text-rich visual reasoning in large multimodal models (LMM). Unlike prior efforts that primarily test reading skills in visual contexts, CONTEXTUAL includes novel and challenging instructions that require models to understand the context in which text is presented in an image. Human participants were asked to solve the tasks in the dataset, and human annotators were used to evaluate model responses. The experiments reveal that modern LMMs, both proprietary and open models, struggle with the dataset, whereas humans perform well. Additionally, they conducted fine-grained evaluation and qualitative analysis to identify gaps in the model capabilities (Wadhawan et al., 2024). This aligns with findings by (M. Li et al., 2023),

who demonstrated the effectiveness of transformer-based models in OCR tasks, emphasizing the need for contextual understanding in text recognition.

PaddleOCR developed by Baidu's PaddlePaddle team, is known for its high accuracy, efficiency, and extensive multilingual support. It integrates cutting-edge deep learning techniques, offering exceptional performance in text detection and recognition. PaddleOCR's support for multiple languages makes it versatile for global applications. It is optimized for both the Central Processing Unit (CPU) and GPU usage, ensuring fast and precise text extraction. Its end-to-end OCR capabilities encompass all stages from text detection to recognition, providing a seamless and highly efficient process (Du et al., 2021).

The evolution of PaddleOCR through PP-OCRv2, PP-OCRv3, and recently PP-OCRv4 has introduced significant enhancements, establishing it as a leading choice for lightweight and efficient OCR systems. The PP-OCRv4 system block diagram is shown in 2.4 (PaddlePaddle, 2024).

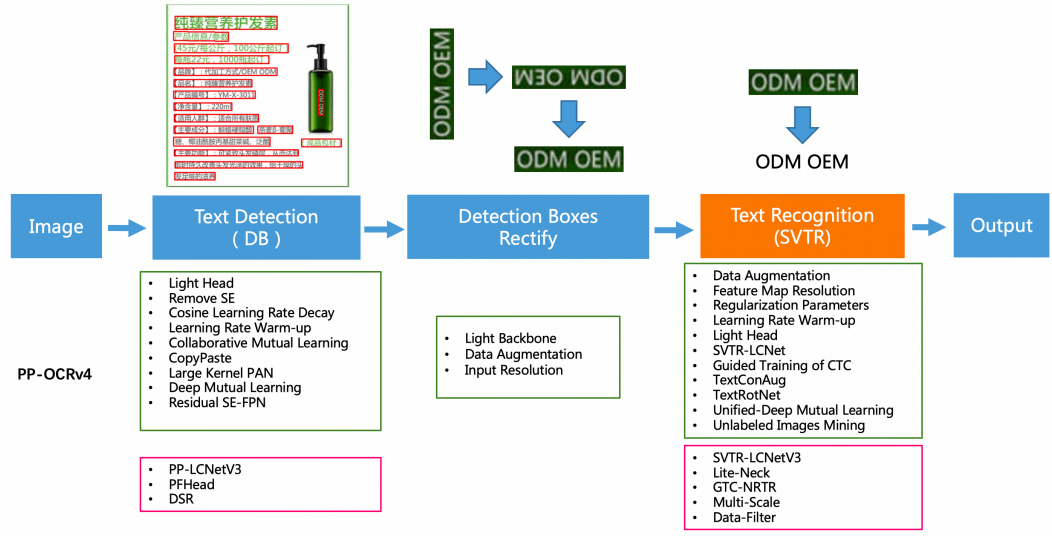


Figure 2.4: PaddelOCRv4 model Architecture (PaddlePaddle, 2024).

PP-OCRv2 employs a ”bag of tricks” approach to optimize the model size and inference time while maintaining high accuracy, including improved backbone networks and various optimization techniques (Du et al., 2021). PP-OCRv3 builds on this by introducing advanced detection algorithms, enhanced recognition models, and fine-grained optimizations to balance model size, speed, and accuracy. Specifically, the detection network is still optimized based on Direct Broadcast Network (DBNet). and the recognition network's base model is upgraded from CRNN to SVTR (C. Li et al., 2022; Research, 2022). Which can process videos, and live video streams in addition to images (Du et al., 2022)

Tesseract OCR, developed by Google, is a well-known open-source OCR engine recognized for its high accuracy and support for over 100 languages. Python-tesseract, a wrapper for Tesseract, enables it to read various image types. However, it only works on CPUs and struggles with blurry, noisy images and complex forms. EasyOCR, a Python-based library, supports over 70 languages and is optimized for speed and ease of use. but offers limited customization options. KerasOCR, leveraging the Keras deep learning framework, provides high accuracy and extensive customization options, but supports only a few languages and has limited community support (Dag, 2023). Compared to Tesseract, EasyOCR, and PaddleOCR offer superior accuracy, extensive language support, and efficient processing, making them leading choices for modern OCR applications (Dag, 2023; Ujwal Karanth et al., 2023). When dealing with video analysis in big data, especially in seized evidence cases, the ability to use GPU for processing and fine-tuning models is crucial for both speed and cost-effectiveness.

OCR plays a vital role in digital forensics by enabling the extraction and analysis of textual information from various types of documents

and images. Key applications include:

- Document Analysis: OCR technology is beneficial for reading car plates and signs, aiding the analysis of evidence in legal investigations.
- Automated Data Entry: OCR streamlines the process of extracting information from forms, invoices, and other documents, reducing manual effort and minimizing errors.
- Evidence Authentication: By converting handwritten or printed documents into digital format, OCR aids in verifying the authenticity of evidence and detecting tampering.
- Enhanced Search Capabilities: OCR enables the indexing and searching of large volumes of data, allowing investigators to locate relevant information quickly.

2.1.8.3 History and Evolution of Speech Recognition

The advent of AI has revolutionized many sectors, with transcription services emerging as one of the primary beneficiaries. This shift has significantly reduced the need for traditional roles such as stenographers and typists, as AI technologies can automatically convert video and audio into text within minutes.

Speech recognition technology, known as automatic Speech recognition (ASR), translates spoken language into text, enabling computers to recognize and interpret human speech as either commands or transcriptions (Juang & Furui, 2000). This technology has revolutionized human-computer interaction by allowing machines to process human language in various contexts, thereby enhancing user experience and accessibility (Davis et al., 1952).

The journey of speech recognition technology began in the 1950s with Bell Laboratories 'Audrey' which could recognize spoken digits

(0-9) for a single speaker (Anusuya & Katti, 2010). A decade later, IBM's 'Shoebox' expanded this capability to understand 16 English words (Shaikh, 2023).

In the 1970s to 1980s: Technological advancements enabled systems like Carnegie Mellon's "Harpy" to handle a few thousand words. The 1990s marked a turning point with the advent of personal computing. Dragon Systems introduced 'Dragon Dictate' the first consumer speech recognition software, and later 'Dragon Naturally Speaking' a product that recognized continuous speech at a rate of 100 words per minute (H. Li et al., 2013). By the early 2000s, accuracy rates had soared close to 80%.

Modern Era: Deep Learning and Diverse Applications (2010s-Present): The 2010s ushered in the era of deep learning, dramatically enhancing speech recognition systems' accuracy and speed. Products like Apple's Siri, introduced in 2011, and Google's Voice Search became industry benchmarks, leveraging sophisticated neural network architectures to improve user interaction (Hinton et al., 2012).

Contemporary speech recognition technology leverages advanced deep learning models such as Bidirectional Long Short-Term Memory (BiLSTM) networks, CNN, and Transformers (Mishra et al., 2024). These models are particularly adept at recognizing nuances in speech such as accents and intonations, thanks to their training on extensive and diverse datasets (Grigaliūnaitė, 2022). Notably, end-to-end models like Google's Speech-to-Text API, OpenAI's Whisper, and other prominent tools such as Vosk, Sphinx, and Google Scribter API process speech directly without the need for intermediate phonetic transcription, setting new benchmarks in the field (Kim et al., 2024; Samin et al., 2024; Serdyuk et al., 2018).

Moreover, speech recognition technology is now widely used in

various applications, including smartphones, virtual assistants such as Alexa, Siri, and Google Assistant, as well as smart speakers and other voice-controlled devices. These tools have transformed user interfaces and enabled innovative multimodal applications that combine speech with visual and contextual data. The versatility and growing capabilities of AI are clearly shown in everyday technology, improving user experiences and setting the stage for future advancements (He et al., 2023; Mehrish et al., 2023). The development of pre-trained models, such as those offered by Google and Open AI, further exemplifies the sophistication, and reach of current speech recognition technologies, enabling developers and researchers to push the boundaries of what's possible with voice-activated systems (Devlin et al., 2019; Kim et al., 2024).

Whisper, developed by OpenAI, epitomizes the integration of cutting-edge AI in speech recognition. Its architecture is based on a transformer framework, which has been fine-tuned for extensive, robust speech recognition. It has been trained on a diverse, multilingual dataset of supervised data sourced from the web, enabling high accuracy across various languages and accents. Notably proficient in noisy environments, Whisper's performance is pivotal for practical applications where clarity and accuracy are paramount (Radford et al., 2023).

Traditional speech recognition systems typically consist of several key components: signal processing to convert raw audio into a digital format, feature extraction techniques like Mel-Frequency Cepstral Coefficients (MFCC) to identify distinct features in speech, acoustic modeling using techniques such as Gaussian Mixture Models (GMM) or deep neural networks, language modeling to predict word sequences, and decoding to interpret audio signals and generate text or commands (Deng, 2016; Kim et al., 2024; H. Li et al., 2013).

In contrast, modern systems like Whisper by OpenAI epitomize a more streamlined, end-to-end approach. Whisper's architecture as summed in 2.5 is based on a transformer framework, fine-tuned for robust speech recognition across diverse languages and accents. Instead of relying on separate stages for feature extraction and modeling, Whisper integrates these processes within a single, ANN model. This innovation allows Whisper to handle complex tasks, such as recognizing speech in noisy environments, with higher accuracy and efficiency, reflecting the significant advancements in AI-driven speech recognition technologies (Lyu et al., 2024). Practically, The Whisper model employs a robust encoder-decoder Transformer architecture, namely A sequence-to-sequence Transformer model. It processes audio sampled at 16,000 Hz into 80-channel log-magnitude Mel spectrograms over 25-millisecond windows with a 10-millisecond stride. Feature normalization scales inputs between 1 and 1. The encoder includes two convolutional layers, followed by sinusoidal position embeddings and Transformer blocks with pre-activation residuals and final layer normalization. The decoder mirrors the encoderś structure and uses learned position embeddings. Utilizing the GPT-2 byte-level Byte-Pair Encoding (BPE) text tokenizer for English and a refitted vocabulary for multilingual models as 2.5 showed (Radford et al., 2022; Radford et al., 2023).

Despite advancements, there are still issues with speech recognition. The comprehension of varied speakers can be impeded by accents and dialects. Additionally, background noise reduces accuracy, necessitating improved noise reduction techniques (Grigaliūnaitė, 2022). Rapid, overlapping speech and homophones—words with the same sound but different meanings—make identification even more difficult. Precise interpretation depends on context-aware processing. Processing in real-time on devices with limited resources

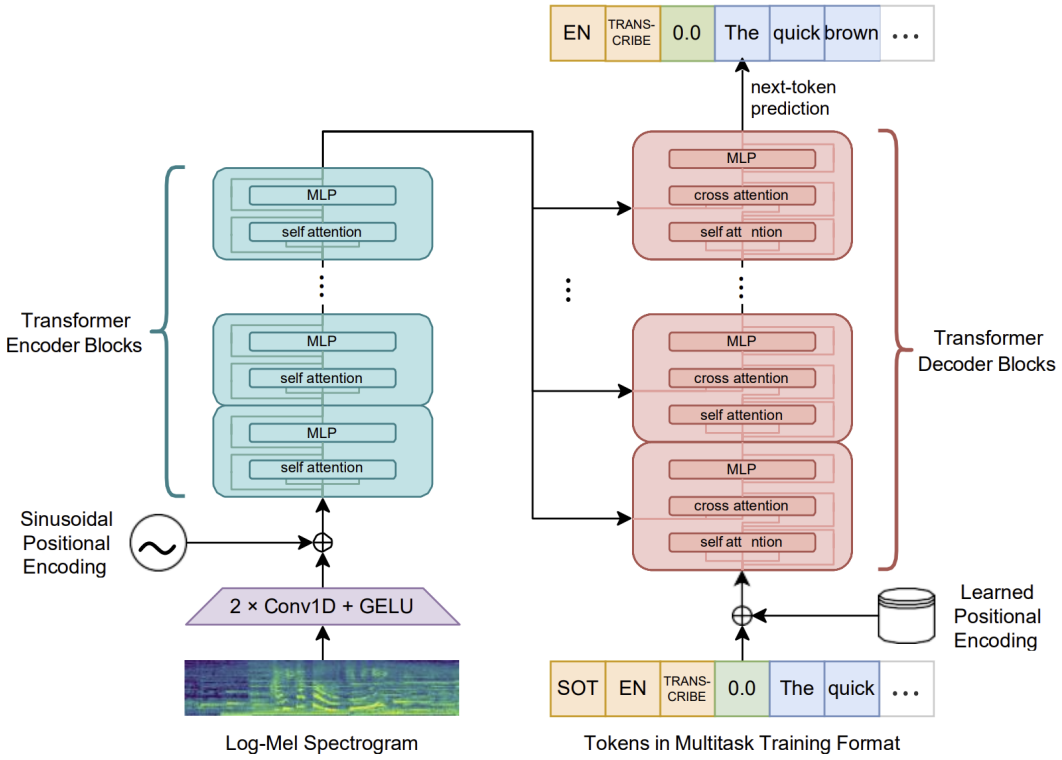


Figure 2.5: The Automated Speech Recognition Whisperś Architecture (Radford et al., 2022).

continues to be a difficulty (Davis et al., 1952). Misunderstandings can also result from the natural intricacies of language, such as idioms and slang. The key to obtaining wider and more accurate speech recognition is to address these issues through improvements in algorithms, models, and training data (Sumon et al., 2024). there is a need for more parameter-efficient, interpretable models and the potential of DL for multimodal ASR (Mehrish et al., 2023).

Speech recognition technology is employed across various domains, each with unique requirements and benefits (Mehrish et al., 2023; Mishra et al., 2024):

- **Healthcare:** Used in medical transcription services, enabling doctors to dictate notes and patient data efficiently, which are then transcribed automatically into electronic health records (EHRs).

- **Customer Service:** Powers interactive voice response (IVR) systems and virtual assistants to enhance customer interaction. such as Amazon's Alexa, Google Assistant, and Apple's Siri are commonplace in homes and smartphones, helping users perform tasks through voice commands.
- **Automotive:** Facilitates hands-free controls and communication systems in vehicles.
- **Home Automation:** Integrates with smart home devices to enable voice-activated control.
- **Education:** Assists in educational tools and accessibility features for students.
- **Accessibility:** Assists individuals with disabilities by enabling voice-based control over computers and mobile devices, significantly improving accessibility.
- **Legal and Security:** Supports law enforcement and legal professionals in documentation and security monitoring.

Speech recognition technology is invaluable in forensic applications for several compelling reasons. First, it automates the transcription of audio from surveillance and interrogations, which makes the data quickly searchable and analyzable, streamlining the process of evidence collection. Additionally, it aids in the enhanced analysis of spoken content in multimedia, thereby improving the identification of suspects and analysis of spoken evidence in legal contexts. Finally, speech recognition technology significantly speeds up the processing of vast amounts of audio data, increasing the overall efficiency of forensic investigations.

The integration of ASR with other AI-driven technologies like object detection and OCR further enhances the capabilities of digital forensic tools, providing a comprehensive approach to evidence analysis and investigation.

2.2 Related work

Since the video comprises a variety of types of data, such as text, speech, and objects, as well as surveillance systems, software packages, and applications that rely on video recordings, the researchers were interested in analyzing and examining the video from various perspectives in digital forensic investigation.

In the field of audio forensics, (H. Feng et al., 2011) proposed a framework to classify audio in the video stream on CCTV NEWS of TRECVID and music CDs by extracting the audio data and clustering it via audio feature extraction and SVM into three groups: silence, speech, and sound. Then, they converted the speech to text using ASR. (Ozkan & Barkana, 2019) introduced a framework for smart surveillance systems' forensic audio analysis and event detection. Through spectral domains, they examined the properties of sound occurrences such as dog barking, screams, gunshots, explosions, and house alarm sounds. Meanwhile, (Xiang et al., 2022) analyzed the audio to extract the history from the audio file to determine whether temporal regions of an audio stream had been compressed once or more at the temporal frame level.

Negrao and Domingues (2021) presented a novel software module called Speech to Text for the widely used Autopsy forensic software. This module utilizes ASR technology to identify and transcribe speech in audio recordings, all while maintaining data privacy by avoiding the need to upload potentially sensitive files to the cloud. The performance of Speech to Text was evaluated using a diverse collection of audio files from 14 popular Android applications such as Facebook and WhatsApp, including 12 common communication applications. The evaluation utilized several open-source libraries, namely INA's 'Ina Speech Segmenter' for speech detection and Mozilla's

Deep Speech for speech transcription. Lastly, the accuracy and speed of the software were assessed by comparing the transcriptions with the test-clean set from the LibriSpeech corpus.

Researchers in the video analysis field are interested in dense video captioning to describe video events that contain important complementary information to video content. (Mun et al., 2019) introduced a new method for dense video captioning that captures the temporal dependency between events in a video. RNN generates captions for event sequences rather than isolated events, using the preceding events and captions as context. Their method involves three stages: First, an event proposal network extracts a set of candidate events from the input video. Second, an event sequence generation network dynamically selects a subset of sequential events from the candidates. Third, a sequential captioning network creates captions for the selected events using a RNN. Then, (Iashin & Rahtu, 2020) emphasized the importance of using multi-modal input in a dense video captioning method for event description. They apply an ASR system to produce a temporally aligned textual description of the speech (similar to subtitles) and treat it as a separate input along with video frames and the corresponding audio track. Also, the captioning task is formulated as a machine translation problem, and the method employed Transformer architecture to convert multimodal input data into textual descriptions.

In OCR era, (Daraghmi & Shawahna, 2023) proposed that the dashcam forensics framework significantly improves the extraction and analysis of evidential text and speech from dashcam videos. By using accurate OCR methods like Tesseract and effective speech-to-text APIs such as Google's and Mozilla's DeepSpeech, the framework provides a comprehensive tool for investigators. It enhances accuracy and efficiency in reviewing evidence, making it a more effective solution

compared to existing digital forensic tools. The framework's integration of spatial and temporal data with mapping technology offers a detailed overview of the vehicle's trip, supporting thorough forensic analysis.

Jeran Ratnarajah et al. (2023) proposed, the development of advanced Forensic Video Analytic (FVA) Software is crucial for enhancing the efficiency and effectiveness of evidence extraction in law enforcement. The proposed tool in this project addresses the limitations of existing FVA software by utilizing a combination of machine learning techniques, GPU acceleration, and integrated architecture development. The methods employed include CNN for object detection, Gaussian Mixture Models GMM for anomaly detection, and multithreading for improved processing efficiency. OpenCV C++ coding was used to build the software, ensuring robustness and flexibility. The tool enhances capabilities in multiple object tracking, tampering detection, activity recognition, and video synopsis, resulting in a more efficient and reliable forensic analysis process.

Rashmi et al. (2021) and T. Wang et al. (2021) introduced end-to-end video captioning with parallel decoding, which simplifies dense video captioning by directly producing a set of sentences with event localization and captioning running in parallel. In more detail, (T. Wang et al., 2021) proposed a real-time monitoring system using transfer learning on YOLOv3, which gives results in the form of recognition and localization by analyzing people's actions in video content from CCTVcameras. Most of these studies were conducted employing the Activity Net caption (Iashin & Rahtu, 2020; Mun et al., 2019; Rashmi et al., 2021), and YouCook2 (Rashmi et al., 2021) dataset.

Chapter 3

Methodology

3.1 Introduction

The primary objective of this study is to develop and validate a comprehensive Digital Forensics Video Content Analysis Tool (VCAT) that addresses key challenges in the forensic analysis of video content, utilizing various AI technologies to enhance the forensic analysis of video data. As identified in chapter1, forensic investigators face significant obstacles when analyzing digital video evidence, such as the limitations of keyword-based searches and the need for tools capable of maintaining the forensic integrity of evidence while providing in-depth content analysis. Chapter1 reviews the SOAT that underpins our proposed methodology, including advanced techniques in object detection, optical character recognition, and speech recognition, crucial for extracting and analyzing complex digital evidence from video files.

3.2 VCAT Architecture

In this research, the proposed VCAT tool is designed to ensure the systematic processing of digital forensic videos as explained in chapter 2 through NEST standards, focusing on improving the accuracy, preserving the integrity, and legal admissibility. This chapter outlines the structured approach VCAT uses to address the research questions posed in 1, aiming to enhance the efficiency and effectiveness of video content analysis in forensic investigations. Figure 3.1 presents the architecture of the proposed tool. Figure 3.1 describes the methodology of the proposed tool that is consistent with the digital forensic process and maintains the chain of Custody without breaking smoothly and reliably. The VCAT consists of four main phases: Configuration Input, Digital Forensic Image Extraction, Video Analysis Process, and Reporting. The following subsections illustrate each phase in detail.

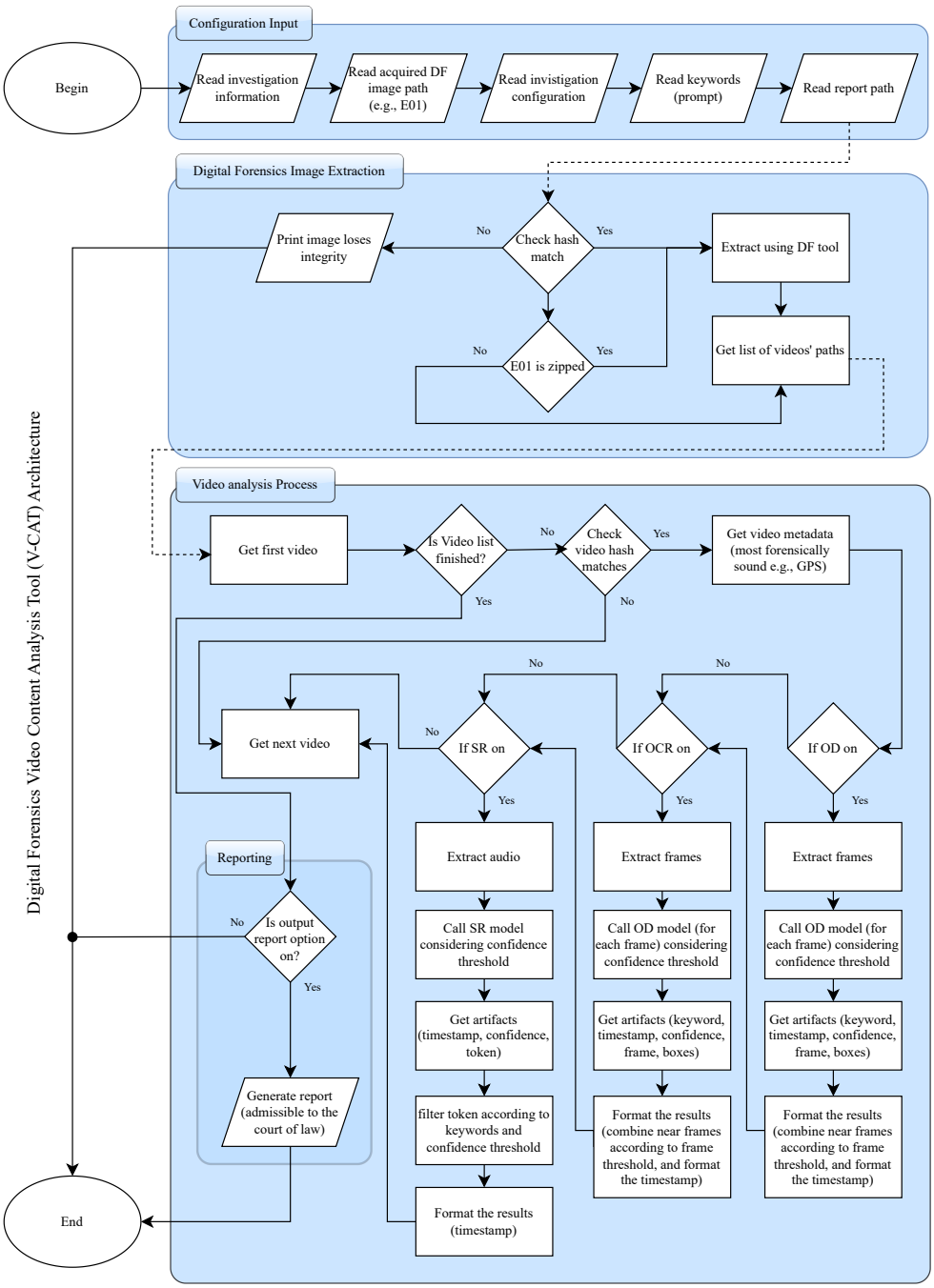


Figure 3.1: The Digital Forensics Video Content Analysis Tool (VCAT) Architecture

3.2.1 Input Configuration

This initial phase involves setting up the parameters and preferences required for the subsequent processes. It includes the following steps:

- **Read Investigation Information:** The system reads detailed investigation information, such as case numbers and investigator details, which are crucial for contextualizing the analysis.
- **Read Investigation Configuration:** Concurrently, the system loads the investigation settings that dictate how the analysis should be tailored to the specific requirements of the case. This includes determining which analytical modules (e.g., Object Detection [OD], Optical Character Recognition [OCR], Speech Recognition [SR]) to activate.
- **Read Keywords (Prompt):** This step involves loading specific keywords or phrases that the OCR and speech recognition modules will use to filter relevant content during the analysis.
- **Read Report Path:** The path for the output report is set up, ensuring that all outputs from the analysis are systematically organized and stored.
- **Read Acquired Digital Forensics Image Path:** This step involves reading the path to the forensics image file (e.g., an E01 file) and depends on the acquisition method to determine the starting point for analysis.

3.2.2 Digital Forensics Image Extraction and Integrity Verification

In this phase, digital forensic techniques are employed to extract images from the video sources. This involves the following steps to ensure that the digital evidence is captured, preserved, and its integrity verified, ensuring it remains unaltered and admissible in a legal context:

- **Check if The Image is Zipped:** Immediately following the path reading, the system checks if the forensics image is zipped, a common condition for forensics archives, to prepare for extraction.
- **Extract Using Digital Forensics Tool:** If the image file is confirmed as zipped, the VCAT proceeds to extract it using specialized forensics tools that ensure data integrity during the decompression process.
- **Check Hash Match:** Post-extraction, a critical hash check is performed to verify that the data has not been altered from its original state. This step is crucial for maintaining the chain of custody of the evidence and forensics soundness.

3.2.3 Video Analysis Process

The actual analysis of the video images that were retrieved is included in this core step. To find pertinent information, spot trends, and derive useful insights from the video data, a thorough set of procedures must be followed:

- **Get a List of Videos' Paths:** The system compiles a list of all video file paths from the extracted data, setting the stage for sequential video processing.
- **Iterative Video Processing:** For each video in the list, a hash check is performed to ensure its integrity before proceeding.
- **Metadata Extraction:** If the video passes integrity checks, metadata (like GPS data from video files) is extracted to aid in forensics analysis.
- **Process Video Content:** Depending on the activated modules:
  - **Object Detection (OD):** If enabled, each video frame undergoes object detection analysis. % using the GroundingDINO model.

  - **Optical Character Recognition (OCR):** If enabled, OCR is performed.% using Paddler to extract textual data from video frames.
  - **Speech Recognition (SR):** If enabled, audio tracks are analyzed to detect and transcribe spoken words that match the loaded keywords. % using OpenAI's Whisper model.
- **Format Results:** After all videos are processed, the detected artifacts (objects, text, speech) are formatted and arranged into coherent keywords, confidence, and timelines based on their timestamps, with specifications for OD and OCR in frame and boxes, and the token for speech recognition.

3.2.4 Reporting

The final phase involves compiling the findings from the video analysis into a comprehensive report. This report includes detailed documentation of the methods used, the results obtained, and any conclusions or recommendations. It is designed to be clear and thorough, providing a valuable resource for legal proceedings, investigations, or further analysis.

- **Generate Report:** A comprehensive report is generated if the reporting option is enabled. This report includes detailed findings and is structured to meet the standards required for court admissibility.

3.2.5 Completion

The process concludes after all videos are analyzed and the final report is generated, marking the end of the VCAT analysis for that case.

3.3 Development Resources

The development and implementation of the VCAT methodology utilized a variety of Python packages and models, which enhanced the tool's functionality and integration with the digital forensics environment. Below is a list of the primary packages used:

VCAT incorporates a variety of standard and specialized libraries to facilitate its operations effectively. Libraries such as `datetime` and `time` manage system dates and times, while `Exiftool` is utilized to extract metadata from videos. The OpenCV library, often imported as `cv`, is crucial for image and video processing tasks. Additionally, `numpy` is employed for numerical operations. Other important libraries include `io`, `os`, `fnmatch`, `json`, and `re` for regular expressions, supporting various file operations and data handling.

The most significant packages integrated into the proposed tool are:

- **GroundingDINO**: This comprehensive package includes all necessary commands and libraries for establishing the GroundingDINO environment. Essential activities include cloning the repository, installing dependencies, and configuring the model for object detection. This repository is sourced from the IDEA-Research GitHub project (IDEA-Research, 2023).
- **PaddleOCR**: Utilized for optical character recognition, PaddleOCR is adept at extracting textual content from images and video frames, enhancing the tool's ability to analyze visual information.
- **OpenAI's Whisper**: This model is specifically designed for speech recognition, converting spoken content from audio tracks into text, which is pivotal for comprehensive multimedia analysis.

- **ReportLab**: Employed for generating PDF reports, ReportLab facilitates the inclusion of text, tables, and images, enabling detailed documentation of findings.

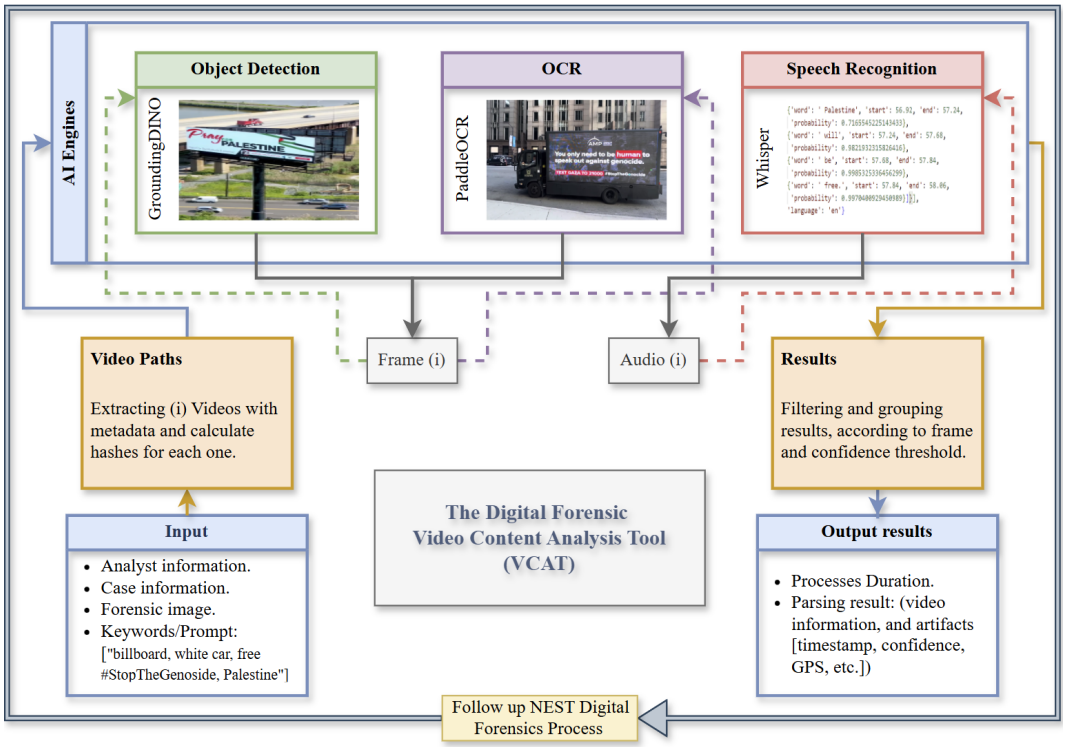


Figure 3.2: The flowchart of Digital Forensics Video Content Analysis Tool (VCAT) process.

Figure 3.2 shows the practical example flowchart of VCAT Process. Each library and package is carefully chosen for its robust performance in its respective function, ensuring that VCAT operates efficiently and effectively across a broad range of tasks.

These libraries and packages were selected for their robust performance, rich functionality, and relevance to the specific needs of digital forensic video analysis for integration into the VCAT methodology. See Appendix 6.2 for the full source code. The `datetime` and `time` libraries were chosen for their ability to manage system dates and

times, essential for timestamp analysis in forensic investigations. Using ExifTool for extracting metadata from videos ensures comprehensive data retrieval, aiding in contextualizing and validating digital evidence. OpenCV, a widely used library for image and video processing, provides a versatile toolkit for analyzing visual content, including tasks such as object detection and frame manipulation. Numpy's efficient numerical operations support various mathematical computations required in data analysis tasks. Additionally, libraries such as `io`, `os`, `fnmatch`, `json`, and `re` offer essential functionalities for file operations, data handling, and regular expressions, facilitating seamless integration with the digital forensic environment. The selection of specialized packages such as GroundingDINO, PaddleOCR, OpenAI's Whisper, and ReportLab was based on their demonstrated effectiveness in addressing specific challenges within forensic video analysis, such as object detection, optical character recognition, speech recognition, and report generation, respectively.

Chapter 4

Experiment and Results

This experiment performed for this research aims to validate the effectiveness of the proposed digital forensic VCAT in the investigation process. This chapter details the experimental setup, data collection methods, procedures followed, tools and techniques tested, and choosing the best models and libraries that satisfy the tool objective by following SOAT techniques, execution of the experiment, and the result of analysis. This experiment demonstrates show the proposed tool can overcome the challenges identified in the study and improve the efficiency of forensic investigations.

4.1 Resources and Tools

4.1.1 Dataset/ videos

Due to the challenge of obtaining pre-made digital forensic images for this study, two versions were prepared. The first version, derived from a previous study, included the contents of a smartphone (Abu Hweidi et al., 2023). The second version originated from a laptop. Both versions contained videos captured from real-life events and social media platforms, and both were acquired using the FINALMobile Forensics tool(Finaldata, 2023). Additionally, individual images and audio files were chosen from online datasets, such as COCO (Lin et al., 2014), and previous research was conducted for various purposes in ML. Some of these elements were combined using the Camtasia program to create experimental setups and to address the challenge of limited free CPU availability on Google Colab.

4.1.2 Coding Resources

The VCAT methodology was implemented using Google Colaboratory (Colab), a cloud-based extension of the Jupyter Notebook environment designed specifically for interactive computing and the development of Python-based ML models. Colab provides a user-friendly

platform for creating, sharing, and managing computational documents. It includes a complimentary virtual machine equipped with an Intel Xeon CPU, 2 virtual CPUs with a 56320 KB cache, and 13GB of RAM. It comes pre-installed with leading ML libraries, including TensorFlow and PyTorch, running on top of Python 3.10.12 on the Linux operating system. This setup facilitates extensive data processing and complex model training without the need for local computational resources. Additionally, Colab grants access to advanced hardware, such as CPUs and TPUs, which are crucial for accelerating tasks integral to the VCAT methodology. For this study, the free version of Colab was utilized, although upgrades to higher CPU configurations are available at a cost (Google Colab, 2019).

FORENSIC UNIT'S HEADER

Video Analyst Information
Analyst Name: Ruwa' Abu Hweidi
Analyst Organization: PTUK
Analyst Oepartment: Cybercrime_PTUK
Case Information
Case Number: 7102024
Report Number: 1948
Date: 28/05/2024
Video Content Analysis Report
Work Time
Start Date: 00:31:57 28/05/2024
End Date: 00:39:49 28/05/2024
Duration: 00:07:51.98
Forensic Image Information
Forensic Image Path: /content/drive/MyDrive/VCAT/project/data
Forensic Image Size: 390.6GB
MD5 Hash Value: 7D1D5D54E9FE1E94CEEE88120BEC4A56
Note:

Figure 4.1: The first page in VCAT-report

VCAT uses a simple graphical user interface (GUI) within the Colab environment, as illustrated in Figure 4.2. This interface is designed for easy navigation and user interaction, optimizing workflow efficiency by providing quick access to key features, as shown in Figure 4.1. The GUI allows the analyst to input essential information and configure the analysis process through the following steps:

1. Video Analyst Information: Input the analyst's name, organization, and department.
2. Case Information: Enter the case number and report number.
3. Directory Paths: Specify the root directory path for the digital forensic image or separate video files, and the output path for the report. In this study, Google Drive was used for storage.
4. Keyword/Prompt: Enter the keyword or prompt to be searched for.
5. Search Type Choosing: Decide the type of search for the keyword/prompt used in the analysis (e.g., object detection, OCR, and speech detection).

.

4.1.3 Preparing Data for Analysis

After obtaining the extracted version of video files as mentioned above, VCAT reads the root directory and searches for all video file paths to be analyzed, storing them in a list. Due to hardware resource limitations and time constraints, a sample of video files was selected for analysis.

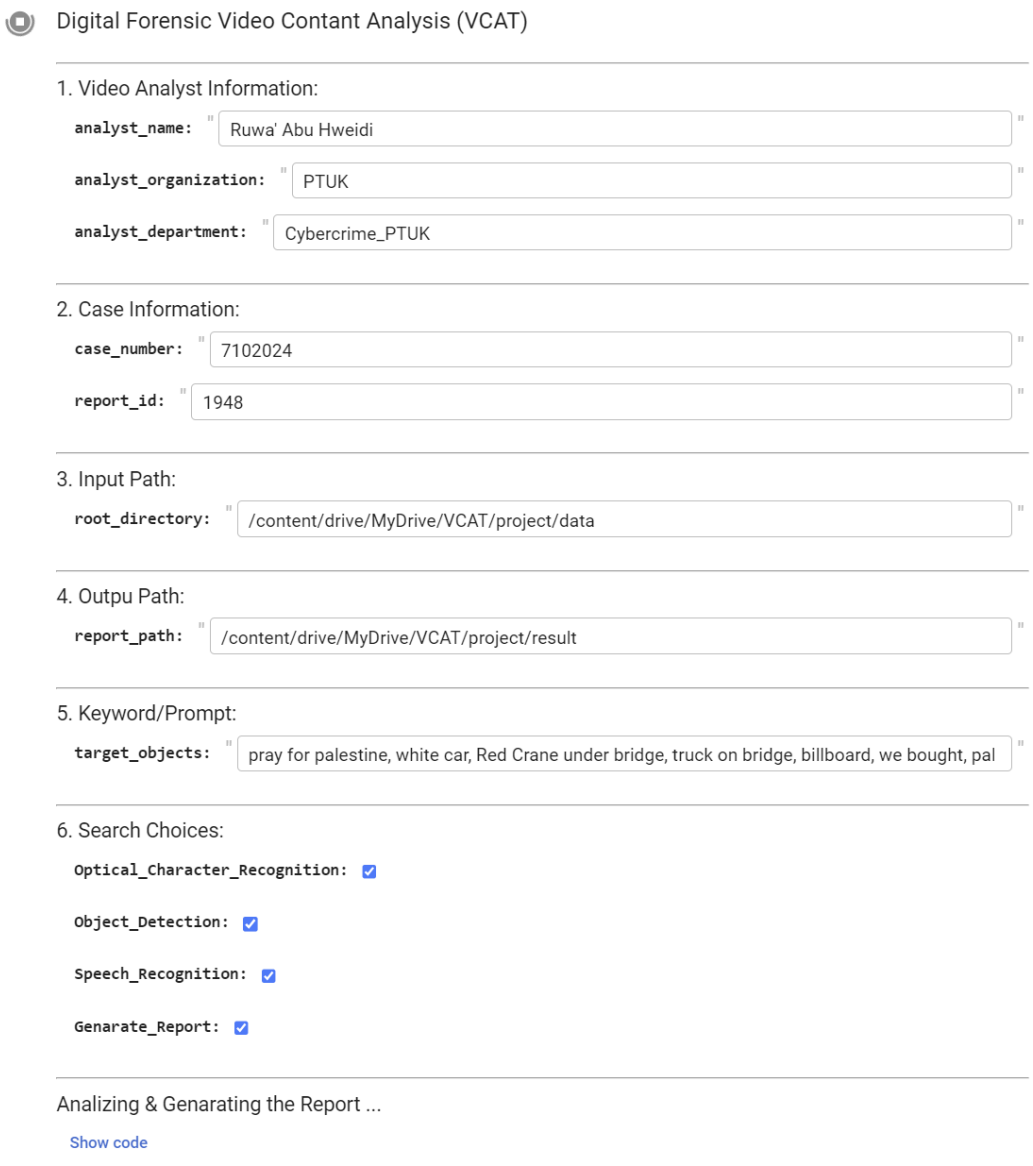


Figure 4.2: The Interface of VCAT.

4.1.4 Starting Video Analysis

For every video file in the video paths list, VCAT reads the metadata using `Exiftool` and then stores it as a `JSON` file. Figure 4.3 shows a sample of the extracted data. Afterward, the hash value of each video is calculated.


```
{
    "SourceFile": "/content/drive/MyDrive/VCAT/project/data/VID_20240202_162857.mp4",
    "ExifToolVersion": 12.4,
    "FileName": "VID_20240202_162857.mp4",
    "Directory": "/content/drive/MyDrive/VCAT/project/data",
    "FileSize": "16 MiB",
    "FileModifyDate": "2024:02:02 14:29:05+00:00",
    "FileAccessDate": "2024:05:26 12:18:10+00:00",
    "FileInodeChangeDate": "2024:05:26 12:18:10+00:00",
    "FilePermissions": "-rw-------",
    "FileType": "MP4",
    "FileTypeExtension": "mp4",
    "MIMEType": "video/mp4",
    "MajorBrand": "MP4 v2 [ISO 14496-14]",
    "MinorVersion": "0.0.0",
    "CompatibleBrands": [
        "isom",
        "mp42"
    ],
    "MovieHeaderVersion": 0,
    "CreateDate": "2024:02:02 14:29:05",
    "ModifyDate": "2024:02:02 14:29:05",
    "TimeScale": 1000,
    "Duration": "6.55 s",
```


Figure 4.3: The Interface of VCAT

Depending on the chosen processes to be performed among the three options—object detection, OCR, and speech recognition—the analysis process begins as follows:

4.1.4.1 Object Detection Process

The study evaluated multiple object detection models, including YOLOv3, YOLOv8, Clip OpenAI, and Zero-shot GroundingDINO. GroundingDINO was selected for the experiment due to its superior

performance and its unique ability to interpret and respond to expressions, effectively detecting objects based on these nuanced inputs. Other models were assessed as follows:

- **YOLOv3:** This model is confined to the COCO dataset, limiting its capability to recognize only the specific words included in the dataset, without the ability to understand synonyms, semantics, or expressions.
- **YOLOv8 (including YOLOv8n and YOLOv8m):** Known for its stability, YOLOv8 is the SOAT within the YOLO series. While its performance is comparable to GroundingDINO, it does not support the detection of expressions.
- **Clip OpenAI:** Capable of searching for multiple objects, this model assigns a confidence score ranging from zero to one for each detected object. However, it occasionally produces errors in the output.

Among the tested models, YOLOv8 and GroundingDINO were deemed most suitable for this study, each with its advantages and drawbacks. GroundingDINO was ultimately chosen for its ease in searching for expressions, enhancing investigator efficiency. Table 4.1 presents a comparison of these models based on their response to a set of keywords/prompts. Although all YOLO versions tested showed variability in detecting the keyword 'truck', they struggled with precisely identifying color-associated terms such as 'white car'. GroundingDINO correctly identified 'bridge' and 'billboard' but faltered with 'red crane', mistaking other objects for cranes. This discrepancy highlights the challenges faced by video analysts in selecting suitable search terms and adjusting confidence thresholds to achieve consistent results. Notably, all models required longer processing times compared to YOLOv3 but demonstrated enhanced detection capabilities, with CPU processing achieving speeds up to eight times

faster than CPU within the tested methodology.

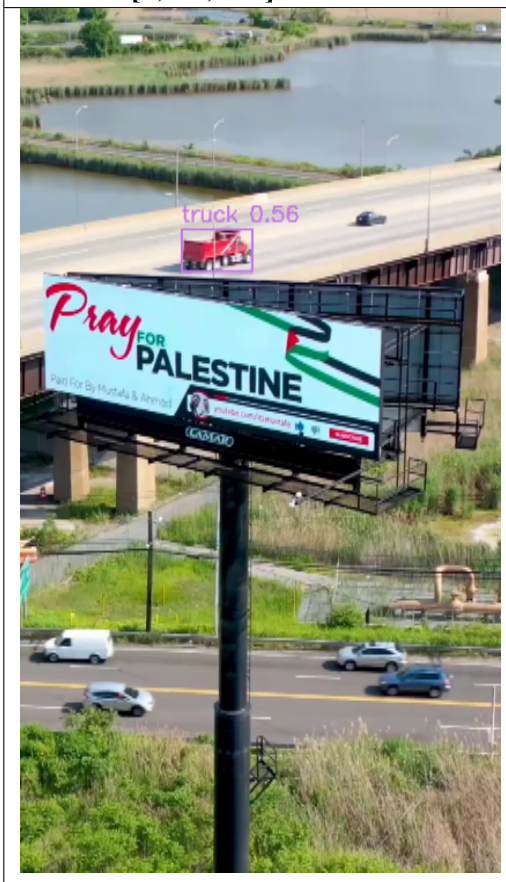
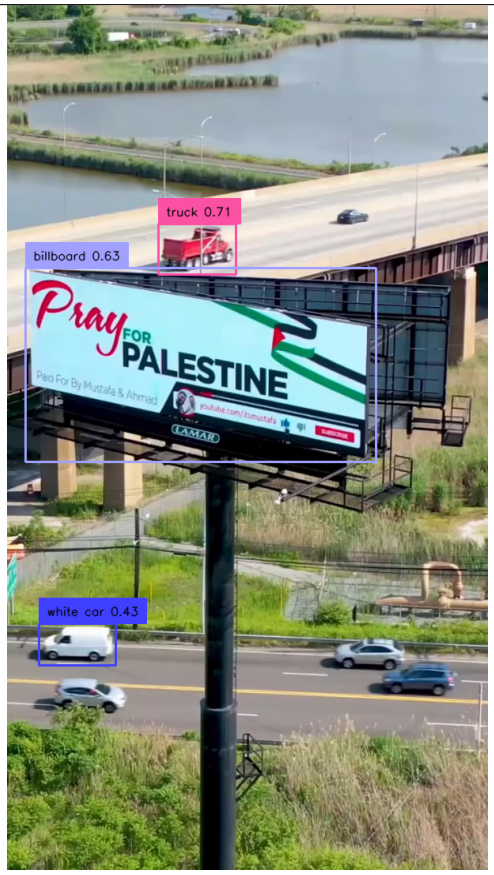
Tested Model	
keyword/prompt [“bridge, truck, white car, red crane, billboard”]	
YOLOv[3, 8n, 8m]:	**GroundingDINO:**
	
Detected Words	
truck [for all tested YOLO’s versions with differences in confidence]	bridge, truck, white car, red crane, billboard

Table 4.1: Comparison between object detection models results.

Furthermore, GroundingDINO’s superior capability in prompt engineering, which allows it to interpret complex queries and execute detailed object recognition, also introduces potential complications. This sophisticated feature, although beneficial in enhancing the model’s responsiveness to nuanced expressions, may inadvertently lead to intricate and ambiguous results. Such complexity can be problematic, particularly in digital forensic investigations where precision

and unambiguous clarity are paramount. Misinterpretations or over-generalizations by the model could result in outputs that are not only irrelevant but could also obscure critical evidence or lead to erroneous conclusions. This poses significant risks in forensic contexts, where the accuracy and reliability of object detection are crucial for the integrity of the evidence and the success of the investigation.

Notable is that the accuracy rate depends on the keyword used to describe the subject of the search. As an illustration, the percentage rose from 45.2% to 59.6%, with “child” used in the initial percentage and “little girl” in the second one in the same timestamp by using keywords/prompt [”child, little girl”], despite the writing mistake in the word ”little”. Table 4.2 shows the effect of choosing the various expressions to parse synonyms. This underscores the importance of exercising care in choosing the keywords or prompts that are entered, as VCAT offers the capability of searching using a combination of key phrases to describe the desired search.

4.1.4.2 OCR Process

: This study tested several OCR models, including Tesseract, EasyOCR, and PaddleOCR, to identify the most effective solution for the VCAT objective. PaddleOCR was selected due to its high performance and robust results in detecting handwritten text. Table 4.3 presents samples from PaddleOCR using various resources on YouTube, highlighting its proficiency with keywords like ”step, from, parking, robotic, machine, today, 003-ycs, and Washington bivd”. whilst the other models have the following characteristics:

- **Tesseract** offers moderate to high accuracy for printed text but requires significant customization to detect handwriting effectively.

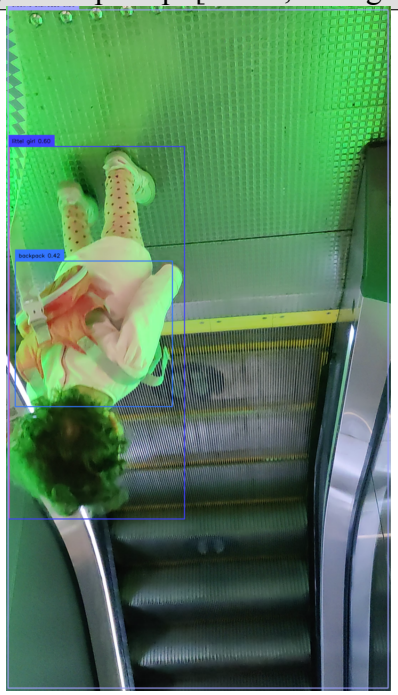
keywords/prompt [”child, little girl”]		
		
Keyword	**Confidence**	**Timestamp**
Child	0.4519	Start: 00:00:00 - End: 00:00:07.52
Little girl	0.5963	Start: 00:00:00 - End: 00:00:07.52

Table 4.2: the effect of choosing the various expressions to parse synonyms

- **EasyOCR** provides higher accuracy and supports multilingual text recognition for printed materials. It is also faster than Tesseract, yet its capability to detect handwriting is limited.

Comparing PaddleOCR with EasyOCR in detecting handwriting, which is the main challenge in this field, shows the vast difference in the results, Figure 4.4 shows a handwritten image, Figure 4.5 and Figure 4.6 show the results of OCR operations made by PaddleOCR and EasyOCR respectively. these figures show that PaddleOCR detects words and phrases with high accuracy and precision, while EasyOCR detects words or chunks of letters with poor results and incorrect characters in many cases. the figure of EasyOCR results shows that the only correct word was (today), while PaddleOCR's correct results were far more.

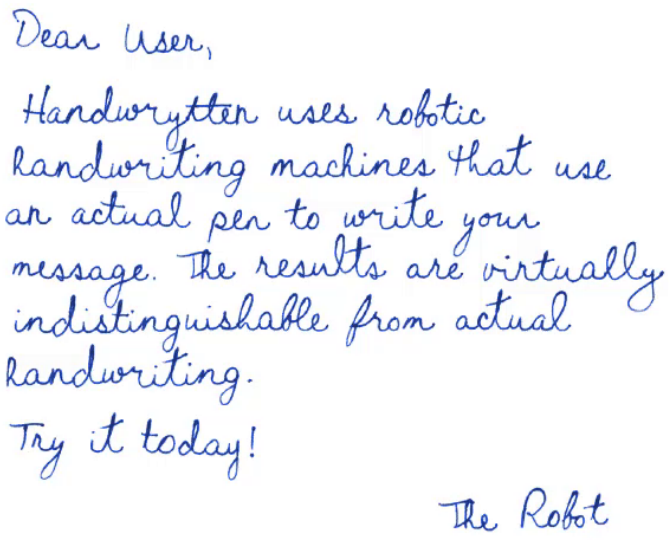


Figure 4.4: The handwritten frame.


```
[[[[[56.0, 123.0], [445.0, 127.0], [445.0, 163.0], [56.0, 159.0]],
   ('Handlwryten uses robotic', 0.9286580085754395)],
   [[[42.0, 165.0], [553.0, 165.0], [553.0, 210.0], [42.0, 210.0]],
    ('Randwriting machines that use', 0.9453631043434143)],
    [[[36.0, 202.0], [497.0, 213.0], [496.0, 261.0], [35.0, 250.0]],
     ('an actual pen to write you', 0.9293038845062256)],
     [[[46.0, 298.0], [521.0, 295.0], [521.0, 341.0], [46.0, 344.0]],
      ('inolistinguishable from actual', 0.9821996092796326)],
      [[[40.0, 344.0], [236.0, 349.0], [235.0, 387.0], [39.0, 383.0]],
       ('Randwriting', 0.9344900250434875)],
       [[[46.0, 401.0], [254.0, 408.0], [253.0, 454.0], [45.0, 447.0]],
        ('Thy it today!', 0.9840268492698669)],
        [[[404.0, 466.0], [550.0, 462.0], [551.0, 502.0], [405.0, 505.0]],
         ('Ihe Rofot', 0.8851025104522705)]]]
```


Figure 4.5: PaddleOCR result on handwriting.

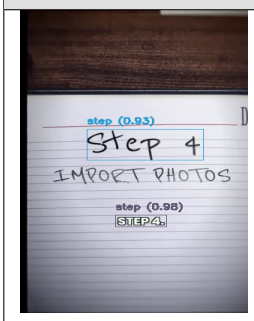
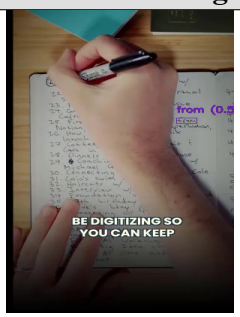
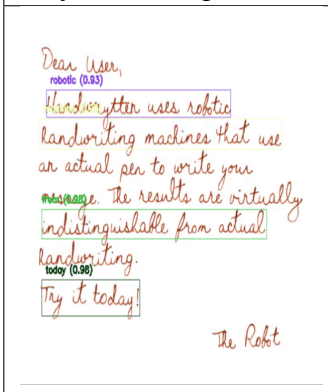
OCR Images		
		
Keyword: step	**Keyword:** from	**Keyword:** parking
		
Keyword: robotic, machine, from, today	**Keyword:** 033-ycs	**Keyword:** Washington bivd

Table 4.3: Group of OCR Images results

The differences in performance are substantial when comparing PaddleOCR and EasyOCR for detecting handwriting, which remains a major challenge in this field. Figure 4.4 displays an image of handwritten text. Figure 4.5 and Figure 4.6 illustrate the OCR results obtained with PaddleOCR and EasyOCR, respectively. These figures reveal that PaddleOCR detects words and phrases with high precision, while EasyOCR often produces results with poor accuracy, detecting only chunks of letters and frequently misinterpreting characters. Notably, the only word correctly identified by EasyOCR was ”today,” whereas PaddleOCR achieved far more accurate results.

PaddleOCR was integrated into the VCAT tool for its superior speed and accuracy, especially in handling handwritten text, speed, and ability to detect characters in little blurry or pixelated frames. Although

```
[([[133, 68], [222, 68], [222, 109], [133, 109]], 'Ule ,', 0.16416927395476186),
 ([[268, 136], [342, 136], [342, 164], [268, 164]], 'WL', 0.21940837122127355),
 ([[348, 125], [453, 125], [453, 168], [348, 168]], 'holtic', 0.2599561858394219),
 ([[27, 105], [270, 105], [270, 234], [27, 234]], 'uty-', 0.06432921439409256),
 ([[260, 160], [494, 160], [494, 212], [260, 212]], 'mackinea Hat', 0.5541282138941162),
 ([[502, 184], [556, 184], [556, 210], [502, 210]], 'Ue', 0.8093929182520277),
 ([[38, 224], [82, 224], [82, 248], [38, 248]], 'ah', 0.971027485038819),
 ([[87, 209], [499, 209], [499, 275], [87, 275]], 'actall_pea to iute you', 0.3661512651079754),
 ([[184, 248], [368, 248], [368, 298], [184, 298]], 'Te 1eanta', 0.11045994176519365),
 ([[376, 268], [430, 268], [430, 294], [376, 294]], 'ahe', 0.5619611692599433),
 ([[434, 252], [606, 252], [606, 324], [434, 324]], 'tutoally', 0.2441865443360808),
 ([[17, 246], [338, 246], [338, 423], [17, 423]], '355', 0.10059373840128864),
 ([[323, 291], [527, 291], [527, 351], [323, 351]], 'hom adual', 0.11574178937278995),
 ([[37, 395], [149, 395], [149, 461], [37, 461]], 'Ty *', 0.8566514032116881),
 ([[406, 460], [558, 460], [558, 510], [406, 510]],'Te Rokt', 0.3764794459679711),
 ([[45.8418973634698, 55.13596892657316], [128.74341649025257, 66.41886116991581],
  [122.1581026365302, 106.86403107342683], [38.25658350974743, 96.58113883008419]],
  'Deas', 0.314683198928833),
 ([[150.91936183808474, 392.0839157637423], [265.6407830863536, 409.7864056378821],
   [253.08063816191526, 469.9160842362577], [138.3592169136464, 452.2135943621179]],
  "today'", 0.42165973088482095)]
```


Figure 4.6: EasyOCR result on handwriting.

EasyOCR and Tesseract offered reliable results for printed text, PaddleOCR's efficiency and potential for fine-tuning made it a more suitable choice for the comprehensive demands of forensic analysis in this study.

4.1.4.3 Speech Recognition Process

: The third component of our study involved the evaluation of various speech recognition models. These models differ significantly in terms of approach and sophistication while testing them. Advanced technologies like OpenAI's Whisper and Google's speech-to-text technology leverage deep learning techniques to deliver high-precision transcriptions, typically accompanied by confidence scores to indicate accuracy. These models benefit from extensive pre-training on diverse datasets and are fine-tuned to perform optimally across various audio conditions and languages. In contrast, models like wav2vec by Meta, which learn from unlabeled data in a self-supervised manner and then fine-tune on labeled data, were also tested for their innovative transcription capabilities

Our evaluation criteria focused on accuracy, detail of transcription outputs, and the model's ability to handle real-world audio data. Whisper markedly outshone its competitors in the experiment due to its exhaustive output details. It transcribes entire sentences with high accuracy and provides intricate details such as the start and end times for each sentence and word. This level of detail is crucial for VCAT, requiring precise timing and transcription accuracy. An example of Whisper's capability is demonstrated in the transcription of the phrase "Careful. That's dangerous." where each word is segmented with timestamps and probability scores, underscoring the model's confidence in its accuracy as 4.7


```
{'id': 0, 'seek': 0, 'start': 0.0, 'end': 3.74, 'text': " Careful. That's dangerous.",
 'tokens': [50388, 7276, 913, 13, 1320, 338, 4923, 13, 50613], 'temperature': 0.0,
 'avg_logprob': -0.5277482509613037, 'compression_ratio': 0.7647058823529411,
 'no_speech_prob': 0.019361425191164017,
 'words': [
     {'word': ' Careful.', 'start': 0.0, 'end': 0.7, 'probability': 0.3792660557082854},
     {'word': " That's", 'start': 3.38, 'end': 3.48, 'probability': 0.8943972289562225},
     {'word': ' dangerous.', 'start': 3.48, 'end': 3.74, 'probability': 0.9985926747322083}
]}
```


Figure 4.7: Whisper model result.

The keywords or prompts used in the VCAT for this experiment were: "noor, watch out, dangerous" while the entire spoken script was: "Noor watch out, careful, that's dangerous, Noor let's go." Whisper's proficiency in recognizing two words: 'dangerous' is detailed in Figure 4.8 with timestamp(start-end) of its occurrence and confidence (64%) within the video stream.

keyword: that's dangerous

Timestamp:
Start: 00:00:00.00 - End: 00:00:03.48
Confidence: 0.6366864829087717
The Context: careful. that's dangerous.

Figure 4.8: VCAT speech recognition result.

On the other hand, Google Recognizer, tested under similar conditions with the same keywords, failed to yield any transcription results. This highlights Whisper's enhanced ability to handle nuanced audio inputs—a capability that Google Recognizer lacks. On other tests, Google Recognizer shows results in form segment–confidence pairs, with no word level results as Whisper.

Lastly, wav2vec showed the worst results in the experiment, they were irrelevant, and most of them were weird words, chunks, or segments. so it was excluded.

The efficacy of Whisper in accurately capturing spoken words, along with their precise timings and confidence levels made it the best choice to adapt to VCAT,

This depiction provides a comprehensive comparison of the limited capabilities of Google Recognizer and wav2vec models compared to Whisper in Appendix 6.2.

4.2 Generating Output Report

The final component of this experiment was to generate VCAT reports. To achieve this, the study used the ReportLab library in Python. This library enables the programmatic creation of high-quality PDF reports. The VCAT approach has been carefully designed to comply with NIST standards. This ensures the integrity of the chain of custody for digital evidence.

Using the ReportLab library, VCAT has implemented a reporting system that dynamically generates detailed and formatted reports based on the analysis conducted by VCAT. Key features of VCAT implementation include:

- **Dynamic Content Generation:** The system is capable of automatically incorporating analysis results, metadata, and visual elements into the reports.
- **Customizable Templates:** Templates can be customized according to the specific requirements of the investigation or the standards prescribed by the governing bodies.
- **Graphical Capabilities:** Integration of charts, images, and complex graphical representations of data analysis.
- **PDF Output:** Reports are generated in PDF format, ensuring they are portable and secure, which is critical for sharing and archival purposes.

To ensure compliance with NIST standards, the VCAT report generation module emphasizes several key design principles. First, VCAT prioritizes accuracy and reliability, ensuring all data representations and findings are highly accurate and verifiable. This commitment extends to maintaining consistency, as the format of the reports adheres to standardized NIST guidelines, which enhances the integrity and readability of the outputs. Additionally, VCAT implements comprehensive traceability; every step from data input to report output is meticulously logged and traceable. This level of documentation is crucial for supporting the chain of custody.

4.2.1 Report Structure

- **Video Analyst Information:** contains Analyst Name, Analyst organization, Analyst department.
- **Case Information:** Case Number, Report Number, Date.
- **Video Content Analysis Report:**
  - Work Time: Start Date, End Date, Duration.
  - Forensic Image Information: Forensic Image Path, Forensic Image size, MD5 Hash Value, Note.

- **Parsing Result of [Object Detection][OCR][Speech Recognition]**
  - Video Path
  - video information: Video Name, File Type, Video File Size, MD5 Hash Value, Frame Rate, Frame Size, Duration, Creation Time, Last Access Time, Modification Date, Device Make, Device Model, Location, Google Map Link, keyword, Timestamp (Start-End), Confidence, The Context [for Speech Recognition], Captured Frame [for Object Detection and OCR]

Parsing Result of Optical Character Recognition

Video Path: /content/drive/MyDrive/VCAT/project/data/VID_20240523_200756.mp4

Video Information

Video Name: VID_20240523_200756.mp4
File Type: MP4
Video File Size: 13 MiB
MD5 Hash Value: 80adb5bb6a7b60cce2076e580dddd7bc
Frame Rate: 30.243
Frame Size: 1920x1080
Duration: 5.17 s
Creation Time: None
Last Access Time: 2024:05:27 23:42:50+00:00
Modification Date: 2024:05:23 17:08:03
Device Make: None
Device Model: None
Location: 32 deg 13' 14.88" N, 35 deg 13' 57.36" E
Google Map Link:
"https://www.google.com/maps?q=32.220800000000004,35.232600000000005"

Keyword: 6746-94
Timestamp:
Start:00:00:01.79 - End: 00:00:04.33
Confidence: 0.8817408084869385

Figure 4.9: VCAT speech recognition result.

Figure 4.9 illustrates a part of the parsing result of OCR, that is the most significant details about the identified word or expression, as derived from the aforementioned video analysis process utilizing VCAT. This tool offers a digital forensic sound, including details such as the timestamp for starting and ending time, the context of the occurrence or identification, accuracy, and the geographical position of the location, if available. This position is subject to various factors, including

the camera type and settings utilized by the user, and other crucial data.

Tool	Metadata	OD	OCR	SR	Report	Free
Autopsy	✓	✗	✗	✗	✓	✓
OSForensics	✓	✗	✗	✗	✓	✗
EnCase	✓	✗	✗	✗	✓	✗
Magnet AXIOM	✓	✓	✓	✓	✓	✗
MOBILedit	✓	✗	✓	✗	✓	✗
FINALMobile	✓	✗	✓	✗	✓	✗
FTK	✓	✗	✗	✗	✓	✗
GPTs Tool	✓	✓	✓	✓	✓	✗
Forevid	✓	✗	✗	✗	✗	✓
Videoma Intelion	✓	✓	✗	✓	✓	✗
VCAT (our's)	✓	✓	✓	✓	✓	✓

Table 4.4: Comparative Table of Digital Forensic Tools with VCAT

4.2.2 Comparing with Other Digital Forensic Tool

Most current forensic tools divert in the ability to detect objects, recognize characters, or convert speech to text. The comparative analysis of various digital forensic tools highlights their diverse features and functionalities. Autopsy is a free tool that offers comprehensive metadata analysis but lacks object detection, OCR, and speech recognition. OSForensics and EnCase are commercial products providing robust metadata analysis without object detection, OCR, or speech recognition capabilities, but they are suitable for court reporting. Magnet AXIOM stands out with full support for metadata, object detection, OCR, and speech recognition, making it a powerful commercial tool for comprehensive investigations. MOBILedit Forensic and FINALMobile Forensics, also commercial tools, support metadata and OCR but lack object detection and speech recognition. FTK (Forensic

Toolkit) excels in metadata analysis but does not support object detection, OCR, or speech recognition. GPTs tools are a versatile commercial tool offering full support for metadata, object detection, OCR, and speech recognition. Forevid, a free tool, supports metadata but lacks object detection, OCR, and speech recognition and is not suitable for court reporting. Videoma Intelion, a high-cost commercial tool, offers extensive features including metadata analysis, object detection, and speech recognition, but not OCR. Lastly, VCAT, our proprietary tool, provides comprehensive support for metadata, object detection, OCR, and speech recognition, and is designed for court reporting, making it a robust non-commercial solution for digital forensic investigations, Table 4.4 summarizes the comparison between VCAT and other Digital forensics tools. It is worth noting that some of these tools offer a trial version such as MOBILedit, or a version with limited capabilities like GPTs tools.

Chapter 5

Discussion

This chapter delves into the outcomes derived from the experiments conducted with VCAT, focusing on their implications for digital forensic investigations. The discussion aims to contextualize the performance of VCAT within the broader framework of digital forensic science, evaluating its effectiveness, identifying operational challenges, and suggesting potential areas for enhancement. By analyzing the effectiveness of VCAT in addressing the challenges identified in digital forensic investigations.

5.1 Object Detection Performance

The experimental findings from the VCAT's deployment revealed significant advancements in object detection capabilities within digital forensic contexts. The integration of advanced models such as YOLOv8 and GroundingDINO facilitated a marked improvement in detection accuracy. These models were specifically chosen for their robustness and high performance in object recognition tasks, which are critical in forensic investigations where precision is paramount. GroundingDINO excels YOLOv8 in handling compound expression while YOLOv8 results are more robust.

The experimental findings from the VCAT's deployment revealed significant advancements in object detection capabilities within digital forensic contexts. The integration of advanced models such as YOLOv8 and GroundingDINO facilitated a marked improvement in detection accuracy. These models were specifically chosen for their robustness and high performance in object recognition tasks, which are critical in forensic investigations where precision is paramount. GroundingDINO excels YOLOv8 in handling compound expression while YOLOv8 results are more robust.

To address the shortcomings observed in the experimental phase, the study suggests several enhancements for future development:

1. **Model Training and Optimization:** Continued training of the object detection models with expanded datasets that include a broader variety of real-world scenarios, especially challenging conditions such as low-light and high-movement environments.
2. **Algorithm Refinement:** Development of more sophisticated algorithms capable of distinguishing between nuanced movements and background variations to reduce false positives and improve the accuracy of detection.
3. **Preprocessing Techniques:** Implementation of advanced preprocessing modules that enhance video quality through stabilization, lighting correction, and noise reduction, ensuring that VCAT operates with the highest possible input quality

5.2 Optical Character Recognition Performance

VCAT's OCR technology integrates advanced image preprocessing techniques to enhance text clarity before analysis, a factor critical in managing the varied quality of video content encountered in forensic investigations. Despite its overall success, the results highlighted some challenges in maintaining consistency across all text scenarios. Particularly, text overlaid on complex backgrounds or displayed in low-light conditions often resulted in lower recognition accuracy. This variance underscores the need for adaptive algorithms capable of dynamic adjustment based on the characteristics of the input video.

In the experiments, VCAT's OCR function was tested with a range of keywords and phrases commonly encountered in forensic investigations, such as vehicle identifiers and personal names. The system's

ability to detect and recognize these keywords accurately was measured against a set of control images known for their text clarity and complexity. The results, as documented in Chapter 4, showed that while VCAT could effectively identify and parse high-contrast text, performance discrepancies were evident in cases involving symbols, cursive handwriting, or stylized fonts.

The study also compared VCAT's OCR model performance with other leading OCR models to benchmark its capabilities. While tools like EasyOCR provided rapid processing times, they often sacrificed accuracy, particularly with handwritten or poorly contrasted texts. In contrast, VCAT's integration of PaddleOCR allowed for more refined character recognition, albeit with slightly longer processing times. This trade-off between speed and accuracy highlights the importance of selecting the right OCR tool based on the specific needs of forensic analysis.

Moreover, VCAT enhances the reliability of OCR outputs by providing detailed confidence scores for each detected text segment. These scores are crucial for forensic analysts, who require a quantifiable measure of certainty in the evidence they present in court reports. The confidence scores not only aid in assessing the reliability of the OCR results but also help in determining whether additional verification steps are necessary before finalizing a forensic report.

5.3 Speech Recognition Performance

Whisper was able to transcribe complex dialogues with a high degree of accuracy, providing detailed outputs that included the confidence levels and precise timestamps for each spoken word. This functionality is crucial for forensic analysis, where understanding the context and exact timing of spoken words can be pivotal.

Despite its overall superior performance, Whisper exhibited some variability in accuracy depending on the clarity of speech and the speaker's accent. This was observed in experiments where phrases with less articulation or heavier accents resulted in lower confidence scores and some inaccuracies in transcription. Such results highlight the importance of continuing to refine the model's training dataset—incorporating a wider variety of accents and speech patterns—to enhance its universal applicability. In contrast, Google Recognizer, while also tested under similar conditions, did not yield any viable transcription results for the same keywords in the experiment. This stark difference not only highlights the robustness of Whisper but also its adaptability to complex audio scenarios which Google Recognizer could not match. Furthermore, the lack of some detailed output from Google Recognizer, such as missing timestamps and confidence scores for individual words, limited its utility compared to Whisper.

The detailed and accurate results provided by Whisper are of particular importance in the legal context, where the precision of evidence can determine the outcome of cases. VCAT's integration of Whisper ensures that outputs are not only useful for preliminary analyses but are also robust enough to stand up to scrutiny in court settings. The ability to pinpoint the exact moment and confidence level of each spoken word allows forensic analysts to construct a more reliable narrative of events. These functionalities allow forensic analysts to pinpoint relevant forensic details quickly, reducing the time spent on manual video analysis and increasing the overall throughput of forensic investigations.

5.4 Reporting and Chain of Custody

In the realm of digital forensics, the integrity of the chain of custody for evidence is essential. VCAT addresses this critical aspect through

its sophisticated reporting module, which ensures that all outputs adhere strictly to NIST standards for digital evidence. VCAT's reporting system is designed to automatically generate detailed, tamper-evident reports that document every step of the video content analysis. This feature is crucial for maintaining the legality and integrity of the evidence throughout the forensic process. Each report generated by VCAT includes comprehensive metadata, analysis results, and a full audit trail, which are essential for establishing the provenance and authenticity of digital evidence.

The reports are structured to provide unambiguous information that can withstand legal scrutiny. This includes detailed descriptions of the methodologies employed during the analysis, the exact parameters set by the analyst, and the specific results obtained. For instance, in object detection tasks, the reports not only list the objects detected but also annotate each instance with precise timestamps and confidence scores. This level of detail ensures that the evidence is both traceable and verifiable, which is essential for legal proceedings. Furthermore, VCAT enhances the chain of custody by embedding security features into the reports, such as digital signatures and time-stamping. These features help to establish a secure audit trail that logs each interaction with the data, from acquisition through to reporting. This process guards against unauthorized modifications and establishes a clear timeline of custody, which is critical in legal contexts.

In summary, VCAT's reporting capabilities are meticulously engineered to support the chain of custody and ensure compliance with forensic standards. The tool's ability to generate detailed, secure, and comprehensive reports ensures that the evidence it produces is not only scientifically reliable but also legally sound. This makes VCAT an invaluable tool in the field of digital forensics, where the accuracy and integrity of evidence are of utmost importance.

5.5 Integration and User Interface (UI)

The UI of VCAT was designed for simplicity and efficiency, which facilitates its adoption by forensic professionals. However, feedback from trial users suggests that additional customization options could be provided to allow analysts to tailor the tool's functionality more closely to specific case requirements.

The UI of VCAT, as detailed in Chapter 4, was engineered to facilitate ease of use and efficient navigation. This interface enables analysts to initiate video content analysis configuration through a series of straightforward steps, from inputting case details to selecting analysis types, such as object detection, OCR, and speech recognition. One of the key strengths highlighted in the results chapter was VCAT's capacity to integrate various forensic analysis tools into a single, cohesive platform.

This integration reduces the operational complexity and time required to switch between different tools, enhancing the overall efficiency of the forensic process. The GUI supports this by providing intuitive access to all functionalities, which allows analysts to easily manage their tasks.

However, the experiment also revealed areas for improvement in the UI design. The interface could benefit from additional features that allow more advanced customization options. Such as video playback, analysis sensitivity, and specifying parameters for object detection, OCR, and speech recognition directly from the main interface. Moreover, the experiment demonstrated that VCAT's GUI effectively handles the input and organization of large volumes of video data but could be enhanced to better support batch processing and simultaneous analysis of multiple video files. Enhancing these capabilities would

significantly reduce the time required for case preparations and allow for quicker turnaround times in urgent investigative scenarios.

5.6 Expanded Functional Capabilities of VCAT

VCAT not only addresses traditional forensic needs but also extends its utility to various domains such as investigative journalism, regulatory compliance in educational and medical fields, and more. This versatility is made possible through its ability to analyze not just video but also audio and image extensions, providing a comprehensive multimedia analysis suite. Moreover, VCAT enhances the usability and accessibility of its analyses by saving the extracted metadata in a JSON file, which facilitates data integration with other digital forensic tools and systems.

VCAT's ability to output results in multiple formats—including text, images, or video files with overlaid boxes for object detection, OCR, and recognized speech—significantly aids in the interpretation and presentation of evidence. These features make VCAT an adaptable tool that can meet the diverse needs of various stakeholders involved in forensic investigations, from technical analysts to legal professionals.

5.7 Comparing with existing tools

VCAT's integration of state-of-the-art (SOAT) artificial intelligence technologies significantly advances its capabilities beyond those of existing digital forensic tools. Through rigorous testing, as outlined in Chapter 4, VCAT has proven to effectively combine object detection, OCR, and speech recognition—facilitating a comprehensive analysis that addresses complex forensic needs. This robust integration of AI technologies not only streamlines the investigative process but also enhances the accuracy and efficiency of forensic analyses.

While VCAT demonstrates superior capabilities, it is also important to consider the cost-effectiveness of integrating such advanced technologies. Commercial tools like Magnet AXIOM and Videoma Intelion, although offering extensive features, come at a high cost, which can be prohibitive for many forensic departments. In contrast, VCAT provides a commercially viable solution that balances advanced functionality with cost-efficiency. This is particularly significant when compared to free tools like Autopsy, which, while cost-effective, lack the comprehensive video analysis capabilities found in VCAT. VCAT is a promising option for integration with free open-source tools like Autopsy due to the high prices of similar commercial tools.

Despite its advantages, VCAT faces several challenges, particularly when integrating with existing digital forensic infrastructures. These challenges include compatibility with different data formats and the need for extensive computational resources, e.g., CPU, and RAM to handle large datasets effectively. Additionally, the constant evolution of AI technologies necessitates ongoing updates and training to ensure that VCAT remains at the forefront of forensic technology.

To address these challenges and enhance its accessibility and utility, there is potential for integrating VCAT with existing open-source tools like Autopsy. Such integration would combine VCAT's advanced video content analysis capabilities with Autopsy's well-established forensic suite, providing a comprehensive, cost-effective solution for forensic investigations. This would not only expand VCAT's applicability but also leverage Autopsy's strong user base and open-source development model to foster continuous improvement and community-driven enhancements.

5.8 Limitations and Challenges

This section critically examines the limitations and challenges encountered during the deployment and testing of VCAT as detailed in the results chapter. These challenges highlight specific areas where VCAT's current capabilities can be enhanced to better meet the needs of digital forensic investigations.

1. **Dependency on Internet Connectivity:** VCAT's performance is significantly influenced by the quality of Internet connectivity since it relies on cloud-based resources for processing. This dependency poses a challenge in environments where internet access is unreliable or unavailable, potentially limiting the tool's utility in remote or field conditions.

2. **Computational Resource Requirements:** The intensive computational demands of VCAT, particularly when processing large volumes of high-definition video data, necessitate substantial hardware capabilities. While the study utilized Google Colab's virtual machines equipped with advanced CPUs, the free version of these resources is limited. This limitation could restrict the tool's accessibility for some users without the means to upgrade to more powerful processing options.

3. **Performance in Varied Conditions:** The experiments conducted revealed that VCAT's object detection and OCR capabilities are less effective in low-light conditions or with low-resolution videos. These conditions frequently occur in real-world scenarios, suggesting a need for improved algorithms that can adapt to and compensate for suboptimal visual data quality.

4. **Speech Recognition Variability:** Although VCAT integrates Whisper for speech recognition, which generally outperforms

other models, the accuracy remains variable depending on the clarity of speech, background noise, and speaker accents. These factors can severely affect the tool's ability to accurately transcribe spoken words, which is critical in forensic analysis.

5. **Chain of Custody Assurance:** While VCAT is designed to adhere to the stringent chain of custody requirements essential for court admissibility, maintaining this integrity across all stages of data handling poses a constant challenge. Ensuring that all interactions with the data are logged and traceable requires robust system architecture and continuous monitoring to prevent any potential breaches that could compromise the evidence.

6. **Scalability and Flexibility:** The current implementation of VCAT may not easily scale to handle simultaneous multiple analyses due to resource constraints. Moreover, the tool's flexibility in integrating new functionalities or updates is crucial as forensic technology evolves. Ensuring that VCAT can adapt to include emerging technologies and methodologies without extensive overhauls is essential for its long-term viability. However, VCAT's performance is influenced by the specific characteristics of the input media, such as the type of mobile device and camera features used to capture the data. This dependency can significantly impact the quality of the metadata extracted, affecting the accuracy and reliability of the subsequent analysis.

These limitations underscore the need for ongoing development to address specific challenges. Future versions of VCAT will require enhancements in hardware compatibility, algorithm optimization for diverse environmental conditions, and improved scalability and flexibility to ensure it remains a valuable tool in the rapidly evolving field of digital forensics.

Chapter 6

Conclusion and Future Work

6.1 Conclusion and Future Work

VCAT's application of cutting-edge models like YOLOv8 and GroundingDINO for object detection provided notable improvements in accuracy and efficiency. These models were proven to significantly outperform traditional techniques, especially in recognizing complex visual elements from video data. However, the performance variations in scenarios involving obscured or partially visible objects highlight the need for ongoing model training and algorithm optimization.

In terms of OCR capabilities, VCAT, leveraging PaddleOCR, achieved a high level of accuracy in text recognition. This functionality proved crucial for extracting readable content from video frames, supporting the extraction of critical evidence such as signage and license plates in forensic contexts. Nevertheless, the variable results across different text backgrounds and qualities suggest the enhancement of image preprocessing techniques could further bolster OCR accuracy.

The implementation of Whisper for speech recognition within VCAT also marked a significant advancement. This model delivered superior transcription accuracy compared to other evaluated models, such as Google Recognizer, particularly in handling diverse audio environments and multiple speaker inputs. However, inconsistencies in recognizing accents and speech in noisy backgrounds point to the need for further refinement of the speech recognition capabilities.

Moreover, VCAT's reporting functionalities, designed to comply with NIST standards and chain of custody requirements, ensure that the output is not only comprehensive but also legally admissible. This adherence to stringent standards is critical for maintaining the integrity of forensic evidence throughout the investigative process.

Despite these advancements, VCAT faced several challenges during testing. The tool's dependency on robust internet connectivity for cloud-based operations and the requirement for substantial computational resources to process large video datasets are areas that could be improved. The performance in low-light conditions and with low-resolution video also needs enhancement to broaden VCAT's applicability in various forensic situations.

In conclusion, while VCAT has established a robust framework for analyzing video content and represents a significant step forward in digital forensic technology, there are multiple avenues for future improvement. The results of this study underscore the need to continue to develop ML models and their application in forensic tools to keep pace with evolving technological requirements and complex investigative needs. Ongoing refinement of VCAT will focus on addressing these challenges to further enhance its effectiveness and reliability in forensic investigations.

6.2 Future Work

Looking forward, the continual evolution of ML models and their application within VCAT offers a path toward even more automated and precise forensic analyses. As VCAT continues to evolve, the study advocates for integrating unsupervised learning models to boost the tool's adaptability across varying scenarios and further refine its dynamic algorithms to enhance processing speed, particularly for real-time video analysis. A significant area for future development includes advancing VCAT's search functionalities, notably by integrating voice and image-based search capabilities. These features would allow users to initiate searches using voice commands or by uploading images, thereby broadening the tool's applicability and ease of use in complex forensic investigations. Additional efforts will also focus

on improving VCAT's performance in challenging environments such as low-light conditions and high-noise backgrounds and expanding its capacity to manage larger and more intricate datasets. These technological enhancements are aimed at cementing VCAT's status as a sophisticated, multifaceted forensic solution tailored to meet the dynamic demands of modern law enforcement and legal sectors.

Bibliography

Abbas, Q., Ibrahim, M. E. A., & Jaffar, M. A. (2018). Video scene analysis: An overview and challenges on deep learning algorithms. *Multimedia Tools and Applications*, *77*(16), 20415–20453. https://doi.org/10.1007/s11042-017-5438-7

Abu Hweidi, R. F., Jazzar, M., Eleyan, A., & Bejaoui, T. (2023). Forensics investigation on social media apps and web apps messaging in android smartphone. *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 1–7. https://doi.org/10.1109/SmartNets58706.2023.10216267

Alabyad, N., Hany, Z., Mostafa, A., Eldaby, R., Tagen, I. A., & Mehanna, A. (2024). From vision to precision: The dynamic transformation of object detection in autonomous systems. *2024 6th International Conference on Computing and Informatics (ICCI)*, 332–344. https://doi.org/10.1109/ICCI61671.2024.10485026

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Alrajhi, A. M. (2020). A survey of artificial intelligence techniques for cybersecurity improvement. *Int. J. Cyber-Secur. Digit. Forensic*, *9*, 34–41.

Alto, V. (2023). *Modern generative ai with chatgpt and openai models: Leverage the capabilities of openai's llm for productivity and innovation with gpt3 and gpt4*. Packt Publishing. https : / / books . google . ps / books ? id = mcnAEAAAQBAJ

Anderson, P., & ENISA. (2015, May 22). *Electronic evidence - a basic guide for first responders* (Project Report) [Catalogue number TP-05-14-116-EN-N]. European Network and Information Security Agency (ENISA). https://doi.org/10.2824/068545

Andhov, A. (2024). Openai's transformation: From a non-profit to a 100 billion valuation. https://doi.org/10.2139/ssrn.4750197

Andrijcic, E., & Horowitz, B. (2006). A macro-economic framework for evaluation of cyber security risks related to protection of intellectual property. *Risk Analysis*, *26*(4), 907–923. https : / / doi . org / https : //doi.org/10.1111/j.1539-6924.2006.00787.x

Anusuya, M. A., & Katti, S. K. (2010). Speech recognition by machine, A review. *CoRR*, *abs/1001.2267*. http://arxiv.org/abs/1001.2267

Aprill, E. P., Loui, R. C., & Horwitz, J. R. (2024). Board control of a charity's subsidiaries: The saga of openai [UCLA School of Law, Law-Econ Research Paper No. 24-01, Loyola Law School, Los Angeles Legal Studies Research Paper No. 2024-04]. *Tax Notes Federal*, *182*. https://ssrn.com/abstract=4720202

Avyodri, R., Lukas, S., & Tjahyadi, H. (2022). Optical character recognition (ocr) for text recognition and its post-processing method: A literature review. *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, 1–6. https://doi.org/10.1109/ICTIIA54654.2022.9935961

Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H., & Khan, F. S. (2023). Foundational models defining a new era in vision: A survey and outlook. *arXiv e-prints*. https://doi.org/10.48550/arXiv.2307.13721

Barbierato, E., & Gatti, A. (2024). The challenges of machine learning: A critical review. *Electronics*, *13*(2). https://doi.org/10.3390/electronics13020416

Barongo, R. I., & Mbelwa, J. T. (2024). Using machine learning for detecting liquidity risk in banks. *Machine Learning with Applications*, *15*, 100511. https://doi.org/https://doi.org/10.1016/j.mlwa.2023.100511

Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, *13*(4), 27–31. https://doi.org/10.1109/45.329294

Bell, J. (2022). What is machine learning? In *Machine learning and the city* (pp. 207–216). John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781119815075.ch18

Berghs, S., Morrison, G. S., & Goemans-Dorny, C. (2018). Electronic evidence: Challenges and opportunities for law enforcement. In M. A. Biasiotti, J. P. Mifsud Bonnici, J. Cannataci, & F. Turchi (Eds.), *Handling and exchanging electronic evidence across europe* (pp. 75–123). Springer International Publishing. https://doi.org/10.1007/978-3-319-74872-6_6

Best gpts for video analysis with chatgpt [Retrieved from OpenAI website]. (2024). https://chat.openai.com/

Biniyaz, A., & Liu, Z. (n.d.). Multi-step ahead prediction of freezing depth via deep learning with long short-term memory. In *Geo-congress 2024* (pp. 742–750). https://doi.org/10.1061/9780784485330.075

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Bokolo, B. G., & Liu, Q. (2024). Artificial intelligence in social media forensics: A comprehensive survey and analysis. *Electronics*, *13*(9). https://doi.org/10.3390/electronics13091671

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., … Liang, P. (2022). On the opportunities and risks of foundation models.

Brockam, G., et al. (2015). Introducing openai. https://openai.com/blog/introducing-openai

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *CoRR*, *abs/2005.14165*. https://arxiv.org/abs/2005.14165

Brunelli, R. (2009). *Template matching techniques in computer vision: Theory and practice*. Wiley. https://books.google.ps/books?id=AowB9dRNTqYC

Buchholz, K. (2023). Threads shoots past one million user mark at lightning speed. https://www.statista.com/chart/29174/time-to-one-million-users/

Calitz, M. F. (1995). *Image understanding and feature extraction for applications in industry and mapping* (Doctoral dissertation). University of Cape Town. University of Cape Town. http://hdl.handle.net/11427/15942

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision – eccv 2020* (pp. 213–229). Springer International Publishing. https://github.com/facebookresearch/detr

Carroll, O. (2017). Challenges in modern digital investigative analysis. *US Att'ys Bull.*, *65*, 25.

Casino, F., Dasaklis, T. K., Spathoulas, G. P., Anagnostopoulos, M., Ghosal, A., Borocz, I., Solanas, A., Conti, M., & Patsakis, C. (2022). Research trends, challenges, and emerging topics in digital forensics: A review of reviews. *IEEE Access*, *10*, 25464–25493. https://doi.org/10.1109/ACCESS.2022.3154059

Cavigline, L., Wendzel, S., & Mazurczyk, W. (2017). The future of digital forensics: Challenges and the road ahead. *IEEE Security & Privacy*, *15*(6), 12–17. https://doi.org/10.1109/MSP.2017.4251117

Chakraborty, U., Roy, S., & Kumar, S. (2023). *Rise of generative ai and chatgpt: Understand how generative ai and chatgpt are transforming and reshaping the business world (english edition)*. Bpb Publications. https://books.google.ps/books?id=Sfu0EAAAQBAJ

Chaudhury, A., Mukherjee, P. S., Das, S., Biswas, C., & Bhattacharya, U. (2022). A deep ocr for degraded bangla documents [Article No.: 98]. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *21*(5), 1–20. https://doi.org/10.1145/3511807

Collier, P. A., & Spaul, B. J. (1992). A forensic methodology for countering computer crime. *Artificial intelligence review*, *6*, 203–215.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Dag, U. (2023). Comparison of paddle ocr, easyocr, kerasocr, and tesseract ocr [Retrieved May 23, 2024, from Plugger.ai website]. https://www.plugger.ai/blog/comparison-of-paddle-ocr-easyocr-kerasocr-and-tesseract-ocr

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Daraghmi, Y.-A., & Shawahna, I. (2023). Digital forensic analysis of vehicular video sensors: Dashcams as a case. *Sensors*, *23*(17). https://doi.org/10.3390/s23177548

Davis, K. H., Biddulph, R., & Balashek, S. (1952). Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America*, *24*(6), 637–642. https://doi.org/10.1121/1.1906946

De Mulder, W., Bethard, S., & Moens, M.-F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*,

30(1), 61–98. https://doi.org/https://doi.org/10.1016/j.csl.2014.09.005

Deng, L. (2016). Deep learning: From speech recognition to language and multimodal processing. *APSIPA Transactions on Signal and Information Processing*, *5*, e1. https://doi.org/10.1017/ATSIP.2015.22

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2017). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 677–691. https://doi.org/10.1109/TPAMI.2016.2599174

Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., Du, Y., & Jiang, Y.-G. (2022). Svtr: Scene text recognition with a single visual model. *International Joint Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:248496672

Du, Y., Li, C., Guo, R., Cui, C., Liu, W., Zhou, J., Lu, B., Yang, Y., Liu, Q., Hu, X., Yu, D., & Ma, Y. (2021). Pp-ocrv2: Bag of tricks for ultra lightweight ocr system. *ArXiv*, *abs/2109.03144*. https://api.semanticscholar.org/CorpusID:237431121

Fähndrich, J., Honekamp, W., Povalej, R., Rittelmeier, H., Berner, S., & Labudde, D. (2023). Digital forensics and strong ai: A structured literature review. *Forensic Science International: Digital Investigation*, *46*, 301617. https://doi.org/https://doi.org/10.1016/j.fsidi.2023.301617

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010). Cascade object detection with deformable part models. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2241–2248. https://doi.org/10.1109/CVPR.2010.5539906

Feng, H., Jiang, C., & Yang, X. (2011). An audio classification and speech recognition system for video content analysis. *2011 International Conference on Multimedia Technology*, 5272–5276. https://doi.org/10.1109/ICMT.2011.6002093

Feng, W., He, W., Yin, F., Zhang, X.-Y., & Liu, C.-L. (2019). Textdragon: An end-to-end framework for arbitrary shaped text spotting. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9075–9084. https://doi.org/10.1109/ICCV.2019.00917

Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative ai. *Business & Information Systems Engineering*, *66*(1), 111–126. https://doi.org/10.1007/s12599-023-00834-7

Finaldata. (2023). *Finalmobile forensics* [Retrieved from FINALMobile website]. https://finaldata.com/mobile/?ckattempt=1

Fradkov, A. L. (2020). Early history of machine learning [21st IFAC World Congress]. *IFAC-PapersOnLine*, *53*(2), 1385–1390. https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.1888

Ftk forensics toolkit - digital forensics software tools [Retrieved from Exterro website]. (2024). https://www.exterro.com/digital-forensics-software/forensic-toolkit

Fu, R., Liang, H., Wang, S., Jia, C., Sun, G., Gao, T., Chen, D., & Wang, Y. (2024). Transformer-bls: An efficient learning algorithm based on multi-head attention mechanism and incremental learning algorithms. *Expert Systems with Applications*, *238*, 121734. https://doi.org/https://doi.org/10.1016/j.eswa.2023.121734

Gad, G., Gad, E., Cengiz, K., Fadlullah, Z., & Mokhtar, B. (2022). Deep learning-based context-aware video content analysis on iot devices. *Electronics*, *11*(11). https://doi.org/10.3390/electronics11111785

Girshick, R. (2015). Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384

Girshick, R., Donahue, J., Darrell, T., Malik, J., & Mercan, E. (2014). R-cnn for object detection. *IEEE Conference*.

Gollapudi, S. (2019). Artificial intelligence and computer vision. In *Learn computer vision using opencv: With deep learning cnns and rnns* (pp. 1–29). Apress. https://doi.org/10.1007/978-1-4842-4261-2_1

Gonaygunta, H. (2023). Machine learning algorithms for detection of cyber threats using logistic regression. *International Journal of Smart Sensor and Adhoc Network*, *3*(4). https://doi.org/10.47893/IJSSAN.2023.1229

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

Google Colab. (2019). Google colab [Retrieved from Google Colaboratory website]. https://colab.research.google.com/

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (pp. 1764–1772). PMLR. https://proceedings.mlr.press/v32/graves14.html

Grigaliūnaitė, J. (2022). *Accent identification using machine learning* (Master's thesis) [Available through eLABa – National Academic Electronic Library of Lithuania.]. Vilniaus universitetas. Vilnius. https://virtualibiblioteka.vu.lt/primo-explore/fulldisplay?docid=ELABAETD146235383&vid=VU&lang=en_US

Grimm, P. W., Grossman, M. R., & Cormack, G. V. (2021). Artificial intelligence as evidence. *Nw. J. Tech. & Intell. Prop.*, *19*, 9.

Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hadi, M. U., Tashi, Q. A., Qureshi, R., Shehab, L. A., Alyousef, H., Alamer, A., Akram, S., Mostafa, R., Taqa, E., & Zafar,

A. (2023). Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. *TechRxiv*. https://doi.org/10.36227/techrxiv.23589741.v4

Haque, M. A. (2023). A brief analysis of "chatgpt" – a revolutionary tool designed by openai. *EAI Endorsed Transactions on AI and Robotics*, *1*, e15. https://doi.org/10.4108/airo.v1i1.2983

Harrington, P. (2012). *Machine learning in action*. Manning Publications Co. https://doi.org/https://doi.org/10.5555/2361796

Harvey, P. (2024). Exiftool [Download Version 12.84 (7.1 MB) - Apr. 23, 2024]. https://exiftool.org

Hassija, V., Chakrabarti, A., Singh, A., Chamola, V., & Sikdar, B. (2023). Unleashing the potential of conversational ai: Amplifying chat-gpt's capabilities and tackling technical hurdles. *IEEE Access*, *11*, 143657–143682. https://doi.org/10.1109/ACCESS.2023.3339553

He, Y., Seng, K. P., & Ang, L. M. (2023). Multimodal sensor-input architecture with deep learning for audio-visual speech recognition in wild. *Sensors*, *23*(4). https://doi.org/10.3390/s23041834

Heaton, J. (2018). Ian goodfellow, yoshua bengio, and aaron courville: Deep learning. *Genetic Programming and Evolvable Machines*, *19*(1), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.

IEEE Signal Processing Magazine, *29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Horsman, G. (2020). Acpo principles for digital evidence: Time for an update? *Forensic Science International: Reports*, *2*, 100076. https://doi.org/https://doi.org/10.1016/j.fsir.2020.100076

Horsman, G. (2022). Conducting a 'manual examination' of a device as part of a digital investigation. *Forensic Science International: Digital Investigation*, *40*, 301331. https://doi.org/https://doi.org/10.1016/j.fsidi.2021.301331

Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, *14*(1), 13. https://doi.org/10.1186/s11782-020-00082-6

Hweidi, R. F., & Eleyan, D. (2023). Social engineering attack concepts, frameworks, and awareness: A systematic literature review. *International Journal of Computing and Digital Systems*, *13*(1), 691–700.

Hweidi, R. F., Jazzar, M., Eleyan, A., & Bejaoui, T. (2023). Sata m.2 on forensics: Trim function effect on recovering permanently deleted files. *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 1–6. https://doi.org/10.1109/SmartNets58706.2023.10215536

Iashin, V., & Rahtu, E. (2020). Multi-modal dense video captioning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

IBM. (2024). What is machine learning (ml)? [Retrieved from IBM website]. https://www.ibm.com/topics/machine-learning

IDEA-Research. (2023). Groundingdino: Official implementation of the paper "grounding dino: Marrying dino with grounded pre-training for open-set object detection" [Retrieved from GitHub website]. https://github.com/IDEA-Research/GroundingDINO

investigation for a new era by PassMark Software, D. (2023). *Osforensics* [Retrieved from OSForensics website]. https://www.osforensics.com/

Iqbal, F., Debbabi, M., & Fung, B. C. M. (2020). Artificial intelligence and digital forensics. In *Machine learning for authorship attribution and cyber forensics* (pp. 139–150). Springer International Publishing. https://doi.org/10.1007/978-3-030-61675-5_11

Islam, N., Islam, Z., & Noor, N. (2017). A survey on optical character recognition system. *CoRR*, *abs/1710.05703*. http://arxiv.org/abs/1710.05703

Jacob, L., Thomas, K., & Savithri, M. (2010). Ai in forensics: A data analytics perspective. In *Artificial intelligence for cyber defense and smart policing* (pp. 41–60). Chapman; Hall/CRC.

Jarrett, A., & Choo, K.-K. R. (2021). The impact of automation and artificial intelligence on digital forensics. *WIREs Forensic Science*, *3*(6), e1418. https://doi.org/https://doi.org/10.1002/wfs2.1418

Javed, A. R., Jalil, Z., Zehra, W., Gadekallu, T. R., Suh, D. Y., & Piran, M. J. (2021). A comprehensive survey on digital video forensics: Taxonomy, challenges, and future directions. *Engineering Applications of Artificial Intelligence*,

106, 104456. https://doi.org/https://doi.org/10.1016/j.engappai.2021.104456

Jeran Ratnarajah, A., Goonetilleke, S., Tissera, D., Balagopalan, K., & Rodrigo, R. (2023). Forensic Video Analytic Software. *arXiv e-prints*, Article arXiv:2401.02960, arXiv:2401.02960. https://doi.org/10.48550/arXiv.2401.02960

Jiao, Q., Chen, D., Huang, Y., Li, Y., & Shen, Y. (2024). Enhancing multimodal large language models with vision detection models: An empirical study. *ArXiv*, *abs/2401.17981*. https://api.semanticscholar.org/CorpusID:267335231

Jordan, M. I., & Mitchell, T. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Juang, B.-H., & Furui, S. (2000). Automatic recognition and understanding of spoken language - a first step toward natural human-machine communication. *Proceedings of the IEEE*, *88*(8), 1142–1165. https://doi.org/10.1109/5.880077

Kang, C. H., & Kim, S. Y. (2023). Real-time object detection and segmentation technology: An analysis of the yolo algorithm. *JMST Advances*, *5*(2), 69–76. https://doi.org/10.1007/s42791-023-00049-7

Karabulut, D., Ozcinar, C., & Anbarjafari, G. (2023). Automatic content moderation on social media. *Multimedia Tools and Applications*, *82*(3), 4439–4463. https://doi.org/10.1007/s11042-022-11968-3

Karie, N. M., Kebande, V. R., & Venter, H. (2019). Diverging deep learning cognitive computing techniques into cyber forensics. *Forensic Science International: Synergy*, *1*,

61–67. https://doi.org/https://doi.org/10.1016/j.fsisyn.2019.03.006

KAUR, G., & DHAWAN, A. (2024). *Laws of electronic evidence and digital forensics*. PHI Learning Pvt. Ltd. https://books.google.ps/books?id=zUEDEQAAQBAJ

Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). Guide to integrating forensic techniques into incident response. https://doi.org/10.6028/nist.sp.800-86

Kim, E., Tang, Y., Ki, T., Neelagiri, D., & Apsingek, V. R. (2024). Joint end-to-end spoken language understanding and automatic speech recognition training based on unified speech-to-text pre-training. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10971–10975. https://doi.org/10.1109/ICASSP48485.2024.10447509

Kolochenko, I. (2022). *Framework proposal to regulate lawful hacking by police within criminal investigations* (Doctoral dissertation No. 29992059). Capitol Technology University. ProQuest Dissertations Publishing.

Krichen, M. (2023). Convolutional neural networks: A survey. *Computers*, *12*(8). https://doi.org/10.3390/computers12080151

Kubassova, O., Shaikh, F., Melus, C., & Mahler, M. (2021). Chapter 1 - history, current status, and future directions of artificial intelligence. In M. Mahler (Ed.), *Precision medicine and artificial intelligence* (pp. 1–38). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-820239-5.00002-4

Kumar, A., Singh, P., & Lata, K. (2023). Comparative study of different optical character recognition models on handwritten and printed medical reports. *2023 International*

Conference on Innovative Data Communication Technologies and Application (ICIDCA), 581–586. https://doi.org/10.1109/ICIDCA56705.2023.10100213

Kumar, M. (2021). Solid state drive forensics analysis—challenges and recommendations. *Concurrency and Computation: Practice and Experience*, *33*(24), e6442. https://doi.org/https://doi.org/10.1002/cpe.6442

Labs, C. (2023). *Forensic — mobiledit* [Retrieved from MOBILedit website]. https://www.mobiledit.com/

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lei, Q., Hou, X., Liu, X., Liang, D., Fan, Y., Xu, F., Liang, S., Liang, D., Yang, J., Xie, G., Liu, Z., & Zeng, C. (2024). Artificial intelligence assists identification and pathologic classification of glomerular lesions in patients with diabetic nephropathy. *Journal of Translational Medicine*, *22*(1), 397. https://doi.org/10.1186/s12967-024-05221-8

Li, C., Liu, W., Guo, R., Yin, X., Jiang, K., Du, Y., Du, Y., Zhu, L., Lai, B., Hu, X., Yu, D., & Ma, Y. (2022). Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *ArXiv*, *abs/2206.03001*. https://api.semanticscholar.org/CorpusID:249431435

Li, H., Wang, P., & Shen, C. (2017). Towards end-to-end text spotting with convolutional recurrent neural networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 5248–5256. https://doi.org/10.1109/ICCV.2017.560

Li, H., Ma, B., & Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the*

IEEE, *101*(5), 1136–1159. https://doi.org/10.1109/JPROC.2012.2237151

Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., & Gao, J. (2022). Grounded language-image pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10955–10965. https://doi.org/10.1109/CVPR52688.2022.01069

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). Trocr: Transformer-based optical character recognition with pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(11), 13094–13102. https://doi.org/10.1609/aaai.v37i11.26538

Li, X., Tian, Y., Ye, P., Duan, H., & Wang, F.-Y. (2023). A novel scenarios engineering methodology for foundation models in metaverse. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(4), 2148–2159. https://doi.org/10.1109/TSMC.2022.3228594

Liao, M., Lyu, P., He, M., Yao, C., Wu, W., & Bai, X. (2021). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(2), 532–548. https://doi.org/10.1109/TPAMI.2019.2937086

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, *abs/1405.0312*.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2023).

Grounding dino: Marrying dino with grounded pre-training for open-set object detection [Provided by the SAO/NASA Astrophysics Data System]. *arXiv e-prints*, Article arXiv:2303.05499, arXiv:2303.05499. https://doi.org/10.48550/arXiv.2303.05499

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multi-box detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – eccv 2016* (pp. 21–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Luitse, D., & Denkena, W. (2021). The great transformer: Examining the role of large language models in the political economy of ai. *Big Data & Society*, *8*(2), 20539517211047734. https://doi.org/10.1177/20539517211047734

Lyu, K. M., Lyu, R. Y., & Chang, H. T. (2024). Real-time multilingual speech recognition and speaker diarization system based on whisper segmentation. *PeerJ Computer Science*, *10*, e1973. https://doi.org/10.7717/peerj-cs.1973

Maabreh, M., & Almasabha, G. (2024). Machine learning regression algorithms for shear strength prediction of sfrc-dbs: Performance evaluation and comparisons. *Arabian Journal for Science and Engineering*, *49*(4), 4711–4727.

Mante, R. V., & Khan, R. (2020). A survey on video-based evidence analysis and digital forensic. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 118–121. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00024

Matijević Gostojić, M., & Vuković, Ž. (2023). A knowledge-based system for supporting the soundness of digital

forensic investigations. *Forensic Science International: Digital Investigation*, *46*, 301601. https://doi.org/https://doi.org/10.1016/j.fsidi.2023.301601

Maulud, D., & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, *1*(2), 140–147.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, *27*(4), 12. https://doi.org/10.1609/aimag.v27i4.1904

Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R., & Poria, S. (2023). A review of deep learning techniques for speech processing. *Information Fusion*, *99*, 101869. https://doi.org/https://doi.org/10.1016/j.inffus.2023.101869

Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, *8*, 142642–142668. https://doi.org/10.1109/ACCESS.2020.3012542

Meng, Y., Michalski, M., Huang, J., Zhang, Y., Abdelzaher, T., & Han, J. (2023). Tuning language models as training data generators for augmentation-enhanced few-shot learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 24457–24477). PMLR. https://proceedings.mlr.press/v202/meng23b.html

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech*, *2*(3), 1045–1048.

Mishra, A., Dhanda, N., Gupta, K. K., & Verma, R. (2024). Speech recognition using machine learning techniques. *2024 2nd International Conference on Disruptive Technologies (ICDT)*, 1142–1146. https://doi.org/10.1109/ICDT61202.2024.10489508

Montasari, R. (2023). The application of big data predictive analytics and surveillance technologies in the field of policing. In *Countering cyberterrorism: The confluence of artificial intelligence, cyber forensics and digital policing in us and uk national cybersecurity* (pp. 81–114). Springer International Publishing. https://doi.org/10.1007/978-3-031-21920-7_5

Morgan, S. (2020). The world will store 200 zettabytes of data by 2025 [Retrieved May 3, 2024]. https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/

Mori, S., Suen, C., & Yamamoto, K. (1992). Historical review of ocr research and development. *Proceedings of the IEEE*, *80*(7), 1029–1058. https://doi.org/10.1109/5.156468

Mun, J., Yang, L., Ren, Z., Xu, N., & Han, B. (2019). Streamlined dense video captioning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Negrao, M., & Domingues, P. (2021). Speechtotext: An open-source software for automatic detection and transcription of voice recordings in digital forensics. *Forensic Science International: Digital Investigation*, *38*, 301223. https://doi.org/https://doi.org/10.1016/j.fsidi.2021.301223

OliveiraJr, E., Zorzo, A. F., & Neu, C. V. (2020). Towards a conceptual model for promoting digital forensics experiments. *Forensic Science International: Digital Investigation*, *35*, 301014. https://doi.org/https://doi.org/10.1016/j.fsidi.2020.301014

OpenText. (2023). *Forensic suite – e-forensic services* [Retrieved from EnCase website]. https://e-forensic.ca/products/encase-forensic-suite/

Orji, U., & Ukwandu, E. (2024). Machine learning for an explainable cost prediction of medical insurance. *Machine Learning with Applications*, *15*, 100516. https://doi.org/https://doi.org/10.1016/j.mlwa.2023.100516

Ozkan, Y., & Barkana, B. D. (2019). Forensic audio analysis and event recognition for smart surveillance systems. *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–6. https://doi.org/10.1109/HST47167.2019.9032996

PaddlePaddle. (2024). Paddlepaddle pp-ocrv4 introduction [Retrieved from GitHub website]. https://github.com/PaddlePaddle/Paddle/releases/tag/v2.4.0

Pugliese, R., Regondi, S., & Marini, R. (2021). Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management*, *4*, 19–29. https://doi.org/https://doi.org/10.1016/j.dsm.2021.12.002

Quick, D., & Choo, K.-K. R. (2016). Big forensic data reduction: Digital forensic images and electronic evidence. *Cluster Computing*, *19*(2), 723–740. https://doi.org/10.1007/s10586-016-0553-1

Radanliev, P., Roure, D. C. D., Nurse, J. R. C., Montalvo, R. M., Cannady, S., Santos, O., Maddox, L., Burnap, P.,

& Maple, C. (2020). Future developments in standardisation of cyber risk in the internet of things (iot). *SN Applied Sciences*, *2*(2), 169. https://doi.org/10.1007/s42452-019-1931-0

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 8748–8763). PMLR. https://proceedings.mlr.press/v139/radford21a.html

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv e-prints*, Article arXiv:2212.04356, arXiv:2212.04356. https://doi.org/10.48550/arXiv.2212.04356

Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 28492–28518). PMLR. https://proceedings.mlr.press/v202/radford23a.html

Raghavan, S. (2013). Digital forensic research: Current state of the art. *CSI Transactions on ICT*, *1*(1), 91–114. https://doi.org/10.1007/s40012-012-0008-7

Rajeev, A., & Raviraj, P. (2023). An insightful analysis of digital forensics effects on networks and multimedia applications. *SN Computer Science*, *4*, 186. https://doi.org/10.1007/s42979-022-01599-8

Rashmi, M., Ashwin, T. S., & Guddeti, R. M. R. (2021). Surveillance video analysis for student action recognition and localization inside computer laboratories of a smart campus. *Multimedia Tools and Applications*, *80*, 2907–2929. https://doi.org/10.1007/s11042-020-09741-5

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement.

Reedy, P. (2023). Interpol review of digital evidence for 2019–2022. *Forensic Science International: Synergy*, *6*, 100313. https://doi.org/https://doi.org/10.1016/j.fsisyn.2022.100313

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Research, B. (2022). Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system [Published on 2022-05-17]. http://research.baidu.com/Blog/index-view?id=168

Rothman, D., & Gulli, A. (2022). *Transformers for natural language processing: Build, train, and fine-tune deep neural network architectures for nlp with python, hugging face, and openai's gpt-3, chatgpt, and gpt-4*. Packt Publishing. https://books.google.ps/books?id=u9FjEAAAQBAJ

Roumeliotis, K. I., & Tselikas, N. D. (2023). Chatgpt and open-ai models: A preliminary review. *Future Internet*, *15*(6). https://doi.org/10.3390/fi15060192

Rudolph, J., Tan, S., & Tan, S. (2023). War of the chatbots: Bard, bing chat, chatgpt, ernie and beyond. the new ai gold rush

and its impact on higher education. *Journal of Applied Learning and Teaching*, *6*(1).

Samin, A. M., Kobir, M. H., Rafee, M. M. S., Ahmed, M. F., Hasan, M., Ghosh, P., Kibria, S., & Rahman, M. S. (2024). Banspeech: A multi-domain bangla speech recognition benchmark toward robust performance in challenging conditions. *IEEE Access*, *12*, 34527–34538. https://doi.org/10.1109/ACCESS.2024.3371478

Sandhya & Kashyap, A. (n.d.). Real-time object-removal tampering localization in surveillance videos by employing yolo-v8. *Journal of Forensic Sciences*, *n/a*(n/a). https://doi.org/https://doi.org/10.1111/1556-4029.15516

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*(6), 459–473. https://doi.org/https://doi.org/10.1016/0893-6080(89)90044-0

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*, 160. https://doi.org/10.1007/s42979-021-00592-x

Schatz, B. L. (2007). *Digital evidence : Representation and assurance* (Doctoral dissertation). Queensland University of Technology. https://eprints.qut.edu.au/16507/

Schneider, J., Meske, C., & Kuss, P. (2024). Foundation models. *Business & Information Systems Engineering*, *66*(2), 221–231. https://doi.org/10.1007/s12599-024-00851-0

Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B., & Bengio, Y. (2018). Towards end-to-end spoken language understanding. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,

5754–5758. https://doi.org/10.1109/ICASSP.2018.8461785

Shafey, L. E., Soltau, H., & Shafran, I. (2019). Joint speech recognition and speaker diarization via sequence transduction. *CoRR*, *abs/1907.05337*. http://arxiv.org/abs/1907.05337

Shahinzadeh, H., Mahmoudi, A., Asilian, A., Sadrarhami, H., Hemmati, M., & Saberi, Y. (2024). Deep learning: A overview of theory and architectures. *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 1–11.

Shaikh, S. J. (2023). Artificially intelligent, interactive, and assistive machines: A definitional framework for intelligent assistants. *International Journal of Human–Computer Interaction*, *39*(4), 776–789. https://doi.org/10.1080/10447318.2022.2049133

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Son, J., & Jung, H. (2024). Teacher–student model using grounding dino and you only look once for multi-sensor-based object detection. *Applied Sciences*, *14*(6). https://doi.org/10.3390/app14062232

Standards open for comment | NIST. (2024). https://www.nist.gov/organization-scientific-area-committees-forensic-science/standards-open-comment

Subramani, N., Matton, A., Greaves, M., & Lam, A. (2020). A survey of deep learning approaches for ocr and document understanding. *ArXiv*, *abs/2011.13534*. https://api.semanticscholar.org/CorpusID:227209404

Sumon, R. I., Uddin, S. M. I., Akter, S., Mozumder, M. A. I., Khan, M. O., & Kim, H.-C. (2024). Natural language processing influence on digital socialization and linguistic interactions in the integration of the metaverse in regular social life. *Electronics*, *13*(7). https://doi.org/10.3390/electronics13071331

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

Szeliski, R. (2022). *Computer vision: Algorithms and applications*. Springer International Publishing. https://books.google.ps/books?id=QptXEAAAQBAJ

Technology, B. (2023). *Autopsy | digital forensics* [Retrieved from Autopsy website]. https://www.autopsy.com/

Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

the truth. Protect the innocent, M. F. |. U. (2023). *Magnet axiom* [Retrieved from Magnet Forensics website]. https://www.magnetforensics.com/products/magnet-axiom/

Tyralis, H., & Papacharalampous, G. (2024). A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review*, *57*(4), 94.

Ujwal Karanth, K. V., Sujan, A. T., Thanay Kumar, Y. R., Joshi, S., Asha Rani, K. P., & Gowrishankar, S. (2023).

Breaking barriers in text analysis: Leveraging lightweight ocr and innovative technologies for efficient text analysis. *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 359–366. https://doi.org/10.1109/ICACRS58579.2023.10404305

Ultralytics. (2024). *Home* [Accessed: 2024-04-17]. Retrieved April 17, 2024, from https://docs.ultralytics.com/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Video analyzer software: Forensic video analyzer software - analysis video file formats [Retrieved from Forevid website]. (2024). https://forevid.com/

Videoma intelion [Retrieved from ISID website]. (2024). https://www.isid.com/videoma-intelion/

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I. https://doi.org/10.1109/CVPR.2001.990517

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review (D. Andina, Ed.). *Computational Intelligence and Neuroscience*, *2018*, 7068349. https://doi.org/10.1155/2018/7068349

Wadhawan, R., Bansal, H., Chang, K.-W., & Peng, N. (2024). Contextual: Evaluating context-sensitive text-rich visual reasoning in large multimodal models.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475.

Wang, J. (2023). A study of the ocr development history and directions of development. *Highlights in Science, Engineering and Technology*, *72*, 409–415. https://doi.org/10.54097/bm665j77

Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., & Luo, P. (2021). End-to-end dense video captioning with parallel decoding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6827–6837. https://doi.org/10.1109/ICCV48922.2021.00677

Wileman, J. (2015). *Past crimes: Archaeological & historical evidence for ancient misdeeds*. Pen & Sword Books Limited. https://books.google.ps/books?id=jC11CQAAQBAJ

Xiang, Z. (2024). Multimedia Forensics Using Metadata. https://doi.org/10.25394/PGS.25246828.v1

Xiang, Z., Bestagini, P., Tubaro, S., & Delp, E. J. (2022). Forensic analysis and localization of multiply compressed mp3 audio using transformers. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2929–2933. https://doi.org/10.1109/ICASSP43922.2022.9747639

Xiao, J., Li, S., & Xu, Q. (2019). Video-based evidence analysis and extraction in digital forensic investigation. *IEEE Access*, *7*, 55432–55442. https://doi.org/10.1109/ACCESS.2019.2913648

Yan, F., Ruwase, O., He, Y., & Chilimbi, T. (2015). Performance modeling and scalability optimization of distributed deep learning systems. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1355–1364. https://doi.org/10.1145/2783258.2783270

Zakariah, M., Khan, M. K., & Malik, H. (2018). Digital multimedia audio forensics: Past, present and future. *Multimedia Tools and Applications*, *77*, 1009–1040. https://doi.org/10.1007/s11042-016-4277-2

Zareian, A., Rosa, K., Hu, D., & Chang, S. (2021). Open-vocabulary object detection using captions. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14388–14397. https://doi.org/10.1109/CVPR46437.2021.01416

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., & Shum, H.-Y. (2022). Dino: Detr with improved denoising anchor boxes for end-to-end object detection [Provided by the SAO/NASA Astrophysics Data System]. *arXiv e-prints*, Article arXiv:2203.03605, arXiv:2203.03605. https://doi.org/10.48550/arXiv.2203.03605

Zhou, J., Chen, Y., Hong, Z., Chen, W., Yu, Y., Zhang, T., Wang, H., Zhang, C., & Zheng, Z. (2024). Training and serving system of foundation models: A comprehensive survey. *arXiv e-prints*. https://doi.org/10.48550/arXiv.2401.02643

Appendices

Appendix A

Python Code of the digital forensics VCAT experiment.

```
"""Thesis Title"""
# Final Project// VCAT/2024
## Ensure GPU Existance
!nvidia-smi
## Note: before start -> Prepare Paddleocr Packages from Processing OCR Request, because it needs restarting the system
! pip install paddleocr paddlepaddle
## Connect Colab to Drive
import sys
if 'google.colab' in sys.modules:
    %env GOOGLE_COLAB=1
    from google.colab import drive
    drive.mount('/content/drive', force_remount=True)
else:
    %env GOOGLE_COLAB=0
    print("Warning: Not a Colab Environment")
## Extract Video Paths
# Preparing OS and Extracting videos Pathes
import os
import fnmatch

#  Getting a list of file names from nested folders after extracting digital forensic image
def find_videos(root_path):
    video_extensions = ['*.mp4', '*.mkv', '*.avi', '*.flv', '*.mov', '*.wmv', '*.mpeg', '*.ogv']
    video_files = []
    # Walking through the directory and its subdirectories
    for path, dirs, files in os.walk(root_path):
        for extension in video_extensions:
            for file in fnmatch.filter(files, extension):
```

```
                video_files.append(os.path.join(path, file))

    return video_files
## Get MetaData of Video File
# Preparing exiftool
!apt-get install exiftool
import subprocess
import json

def extract_video_metadata(video_path):
    """Run ExifTool command and return the output as JSON."""
    command = ['exiftool', '-json'] + [video_path]
    result = subprocess.run(command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)
    if result.stdout:
        metadata = json.loads(result.stdout)
        if metadata:
            filename = video_path + 'metadata.json'
            with open(filename, 'w') as file:
                json.dump(metadata, file, indent=4) # indent=4 for pretty printing
            return metadata[0]

    else: return {}
## Prepering Helping Functions for Numerecal convergens and frame Combining
import numpy as np
import cv2

def time_converter(seconds):
    hours, seconds = np.divmod(seconds, 3600)
    minutes, seconds = np.divmod(seconds, 60)
    time_ = f'{hours:02.0f}:{minutes:02.0f}:{seconds:05.2f}'
    return time_
```

```
60
61 # Combine adjasent frames in one output intity,
   considering threshold value
62 def process_frames(frame_data, frame_threshold,
   frame_rate):
63     output_frame = []
64     for _ in frame_data.keys():
65         first_frame = -1
66         last_frame = -1
67         cur = -1
68         cur_frame = None
69         current_confidence = 0
70         for frame_id, confidence, frame in frame_data[_
   ]:
71             if first_frame == -1:
72                 first_frame = frame_id
73                 cur = frame_id
74                 cur_frame = frame
75                 cur_confidence = confidence
76             else:
77                 if frame_id - cur > frame_threshold:
78                     last_frame = cur
79                     output_frame.append([_,
   time_converter(first_frame/frame_rate),
   time_converter(last_frame/frame_rate), cur_frame,
   cur_confidence])
80                     first_frame = frame_id
81                     cur_frame = frame
82                     cur_confidence = confidence
83                 cur = frame_id
84         if first_frame != -1:
85             last_frame = cur
86             output_frame.append([_, time_converter(
   first_frame/frame_rate), time_converter(last_frame/
   frame_rate), cur_frame, cur_confidence])
87     return output_frame
88
```

```
import re
def convert_coordinates_gps(dms):
    parts = re.findall(r'\d+\.\d+|\d+', dms)
    degrees = float(parts[0])
    minutes = float(parts[1])
    seconds = float(parts[2])
    return degrees + minutes / 60 + seconds / 3600

import hashlib
def md5(fname):
    hash_md5 = hashlib.md5()
    with open(fname, "rb") as f:
        for chunk in iter(lambda: f.read(4096), b""):
            hash_md5.update(chunk)
    return hash_md5.hexdigest()
## Processing OD Request
# Preparing Dino Model
HOME = os.getcwd()
%cd {HOME}

!git clone https://github.com/IDEA-Research/
   GroundingDINO.git
%cd {HOME}/GroundingDINO
!pip install -q -e .

CONFIG_PATH = os.path.join(HOME, "GroundingDINO/
   groundingdino/config/GroundingDINO_SwinT_OGC.py")

# set model configuration file path
%cd {HOME}
!mkdir {HOME}/weights
%cd {HOME}/weights
!wget -q https://github.com/IDEA-Research/GroundingDINO
   /releases/download/v0.1.0-alpha/
   groundingdino_swint_ogc.pth

# set model weight file
```

```
WEIGHTS_NAME = "groundingdino_swint_ogc.pth"
WEIGHTS_PATH = os.path.join(HOME, "weights", WEIGHTS_NAME)

%cd {HOME}/GroundingDINO
from groundingdino.util.inference import load_model, load_image, predict, annotate
import groundingdino.datasets.transforms as T
import groundingdino.util.inference
from PIL import Image
def get_object(dino_model, transform, predict, annotate, video_path, target_objects):
    BOX_TRESHOLD = 0.35
    TEXT_TRESHOLD = 0.25
    target_objects = ', '.join(target_objects)
    print('-----------------------------------', target_objects)
    detected_objects = {}
    frame_id = 0
    cap = cv2.VideoCapture(video_path)
    assert cap.isOpened(), "Error reading video file"
    while True:
        ret, frame = cap.read()
        if ret:
            image_transformed, _ = transform(Image.fromarray(frame).convert("RGB"), None) # PIL obj img to tensor
            boxes, logits, phrases = predict(
                model=dino_model,
                image=image_transformed,
                caption=target_objects,
                box_threshold=BOX_TRESHOLD,
                text_threshold=TEXT_TRESHOLD,
                # device = 'cpu'   # Commet if run in GPU
            )
```

```
151             annotated_frame = annotate(image_source=
    frame, boxes=boxes, logits=logits, phrases=phrases)
152             %matplotlib inline
153             # import supervision as sv
154             # sv.plot_image(annotated_frame, (8, 8))
155             for i in range(len(phrases)):
156                 phrase = phrases[i]
157                 if phrase not in detected_objects:
158                     detected_objects[phrase] = []
159                 detected_objects[phrase].append([
    frame_id, float(logits[i]), annotated_frame])
160             frame_id += 1
161             if cv2.waitKey(1) == 27:
162                 break
163         else: break
164     cap.release()
165     return detected_objects
166 ## Processing OCR Request
167 # Preparing PaddleOCR packages
168 # ! pip install paddleocr paddlepaddle
169 from paddleocr import PaddleOCR  # PaddleOCR for OCR
    tasks
170
171 def get_detected_ocr(ocr, frame_id, frame,
    target_objects, ocr_detected_objects):
172     ocr_result = ocr.ocr(frame, cls=True)  # Perform
    OCR on the saved image # this line for paddler
173     # ocr_result = ocr.readtext(frame, detail=1,
    paragraph=True) # this line for easyosr
174     colors = np.random.uniform(0, 255, size=(100, 3))
175     if ocr_result[0] is None:
176         # print('none none')
177         return ocr_detected_objects
178     # frame_changed = False
179     # Iterate through OCR results
180     for i, line in enumerate(ocr_result[0]):  # this
    line for paddler
```

```
181     # for i, (box, text, confidence) in enumerate(
    results):  # this line for easyosr
182         box = line[0]       # this line for paddler
183         text = line[1][0]   # this line for paddler
184         conf = line[1][1]   # this line for paddler
185         words = text.lower().strip() #.split()
186         for target in target_objects:
187             if target in words and conf>=0.1:
188                 # frame_changed = True
189                 # Draw the bounding box
190                 start_point = (int(box[0][0]), int(box
    [0][1]))  # Top-left corner
191                 end_point = (int(box[2][0]), int(box
    [2][1]))  # Bottom-right corner
192                 cv2.rectangle(frame, start_point,
    end_point, colors[i], 1)
193                 cv2.putText(frame, f'{target} ({conf:.2
    f})', (start_point[0], start_point[1] - 10), cv2.
    FONT_HERSHEY_SIMPLEX, 0.5, colors[i], 2)
194                 if target not in ocr_detected_objects:
195                     ocr_detected_objects[target] = []
196                 ocr_detected_objects[target].append([
    frame_id, conf, frame])
197     return ocr_detected_objects
198
199 def  get_ocr(ocr, video_path, target_objects):
200     ocr_detected_objects = {}
201     cap = cv2.VideoCapture(video_path)
202     assert cap.isOpened(), "Error reading video file"
203     frame_id = 0
204     while True:
205         ret, frame = cap.read()
206         if ret:
207             ocr_detected_objects = get_detected_ocr(ocr
    , frame_id, frame, target_objects,
    ocr_detected_objects)
208
```

```
209             if cv2.waitKey(1) == 27:  # ESC key to
    break
210                 break
211         else: break
212         frame_id += 1
213     cap.release()
214     return ocr_detected_objects
215 ## Processing SR Requests
216 # base one for transcribting & searching exact word and
     phrases
217 !pip install openai-whisper
218 import whisper
219
220 def process_speech(wsr, option, video_path,
    target_objects):
221     spoken_words = []
222     detected_speech = {}
223     confidence = 0
224     start_time = 0
225     end_time = 0
226     spoken_words = wsr.transcribe(video_path,
    word_timestamps=True)
227     for indx, segment in enumerate(spoken_words['
    segments']):
228         segment['text'] = segment['text'].strip().lower
    ()
229         for target in target_objects:
230             position_target = segment['text'].find(
    target)
231             if(position_target !=-1):
232                 posistion_word = segment['text'].count(
    ' ',0, position_target)
233                 sum_probability = 0
234                 count_probability = 0
235                 word = segment['words'][posistion_word
    + count_probability]
```

```
                start_time = time_converter(word['start'])
                for target_word in target.split():
                    # print('posistion_word + count_probability= ',posistion_word, count_probability)
                    word=segment['words'][posistion_word + count_probability]
                    sum_probability += word['probability']
                    count_probability +=1
                    end_time = time_converter(word['end'])
                if count_probability > 0:
                    confidence = sum_probability / count_probability
                    token = segment['text'].strip().lower()
                    if target not in detected_speech:
                        detected_speech[target] = []
                    detected_speech[target].append([indx+1, start_time, end_time, confidence, token]) # indx= segment No.
                else:
                    confidence = 0
     return detected_speech
## The Output Report Generation
!pip install reportlab
from reportlab.lib.pagesizes import letter, A4
from reportlab.platypus import SimpleDocTemplate, Table, TableStyle, Paragraph, Spacer, Image as RLImage
from reportlab.lib import colors
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib.utils import ImageReader
from reportlab.lib.units import inch, cm  # Import inch for image sizing
```

```
260 import io
261
262 # Reporting .PDF
263 def make_pdf(configuration):
264     pdf_path = project_configuration['report_path']+ "/
    Report.pdf"
265     investigation_info = configuration['
    investigation_info']
266     doc = SimpleDocTemplate(pdf_path, pagesize=A4)
267     elements = []
268     # styles
269     styles = getSampleStyleSheet()
270     video_path_style = ParagraphStyle(
271         'VideoPath',  # Name of the style
272         parent=styles['Normal'],  # Base style
273         textColor='blue',  # Text color (e.g., red)
274         fontName='Helvetica-Bold',
275         fontSize=10
276     )
277     sub_title = ParagraphStyle(
278         'sub_title',  # Name of the style
279         parent=styles['Normal'],  # Base style
280         fontName='Helvetica-Bold',
281         fontSize=12,
282     )
283     sub_title_result = ParagraphStyle(
284         'sub_title_result',  # Name of the style
285         parent=styles['Normal'],  # Base style
286         fontName='Helvetica',
287         fontSize=10,
288         spaceBefore=6,
289         spaceAfter=6,
290         )
291     table_style = TableStyle([
292         ('BACKGROUND', (0, 0), (-1, 0), colors.
    lightblue),  # Header background color
```

```
293         ('BACKGROUND', (0, 5), (-1, 5), colors.
    lightblue),  # Header background color
294         ('BACKGROUND', (0, 12), (-1, 12), colors.
    lightblue),  # Header background color
295         ('BACKGROUND', (0, 17), (-1, 17), colors.
    lightblue),  # Header background color
296         ('TEXTCOLOR', (0, 0), (-1, 0), colors.black),
    # Header text color
297         ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
298         ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),  #
    Center align vertically
299         ('FONTNAME', (0, 0), (-1, 0), 'Helvetica-Bold')
    ,  # Header font
300         ('FONTNAME', (0, 1), (0, -1), 'Helvetica'),  #
    Subheader font
301         ('FONTSIZE', (0, 0), (-1, -1), 12),
302         ('BOTTOMPADDING', (0, 0), (-1, -1), 12),
303         ('GRID', (0, 0), (-1, -1), 0, colors.black),
304         ('INNERGRID', (0, 0), (-1, -1), 0.5, colors.
    grey),
305      ])
306      table_style_result = TableStyle([
307         ('BACKGROUND', (0, 0), (-1, 0), colors.
    lightblue),
308         ('TEXTCOLOR', (0, 0), (-1, 0), colors.
    whitesmoke),
309         ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
310         ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),  #
    Center align vertically
311         ('FONTNAME', (0, 0), (-1, 0), 'Helvetica-Bold')
    , # 'tradbdo'),
312         ('FONTNAME', (0, 1), (0, -1), 'Helvetica'),
313         ('FONTSIZE', (0, 0), (-1, -1), 12),
314         ('BOTTOMPADDING', (0, 0), (-1, 0), 12),
315         ('BACKGROUND', (0, 1), (-1, -1), colors.white),
316         ('GRID', (0, 0), (-1, -1), 0, colors.black),
```

```
317         ('INNERGRID', (0, 0), (-1, -1), 0.5, colors.
    grey),
318     ])
319     # Adding a title
320     title = "FORENSIC 'UNITS HEADER"
321     title_style = styles['Title']  # Default title
    style, you can customize it as needed
322     title_paragraph = Paragraph(title, style=
    title_style)
323     spacer = Spacer(1, 0.25 * inch)  # Adds some space
    after the title
324     # Creating a table for constant info
325     analyst_name = investigation_info['analyst_name']
326     analyst_organization = investigation_info['
    analyst_organization']
327     analyst_department = investigation_info['
    analyst_department']
328     sys_date = investigation_info['sys_date']
329     duration = investigation_info['duration']
330     case_number = investigation_info['case_number']
331     report_id = investigation_info['report_id']
332     start_date = investigation_info['start_date']
333     end_date= investigation_info['end_date']
334     Forensic_Image_Path= investigation_info['
    Forensic_Image_Path']
335     Forensic_Image_size= investigation_info['
    Forensic_Image_size']
336     MD5_Hash_Value = investigation_info['MD5_Hash_Value
    ']
337     data = [
338         ['Video Analyst Information'],
339         [f'Analyst Name: {analyst_name}'],
340         [f'Analyst organization: {analyst_organization}
    '],
341         [f'Analyst department: {analyst_department}'],
342         [],
```

```
343         [Paragraph('Case Information', style=sub_title)
    ],
344         [f'Case Number: {case_number}'],
345         [f'Report Number: {report_id}'],
346         [f'Date: {sys_date}'],
347         [],
348         [Paragraph('Video Content Analysis Report',
    style=title_style)],
349         [],
350         [Paragraph('Work Time', style=sub_title)],
351         [f'start_date: {start_date}'],
352         [f'end_date: {end_date}'],
353         [f'duration: {time_converter(duration)}'],
354         [],
355         [Paragraph('Forensic Image Information', style=
    sub_title)],
356         [f'Forensic_Image_Path: {Forensic_Image_Path}'
    ],
357         [f'Forensic_Image_size: {Forensic_Image_size}'
    ],
358         [f'MD5_Hash_Value: {MD5_Hash_Value}'],
359         [f'Note: \n\n']
360      ]
361      data_ocr = []
362      data_object = []
363      data_speech = []
364      if configuration['object']:
365          print('--------------------------',
    configuration['result_object'])
366          data_object.append([Paragraph('Parsing Result
    of Object Detection', style=sub_title)])
367          for video in configuration['result_object']:
368              data_object.append([Paragraph('Video Path:
    '+ video[0], style=video_path_style)]) #
    ########################################
369              file_md5 = md5(video[0])
370              # extracted video information
```

```
371             metadata = extract_video_metadata(video[0])
372             imgwidth = metadata.get('ImageWidth')
373             imgheight = metadata.get('ImageHeight')
374             FileName = metadata.get('FileName')
375             FileType = metadata.get('FileType')#
376             FileSize = metadata.get('FileSize')
377             FrameRate = metadata.get('VideoFrameRate')#
378             FrameSize = f"{imgwidth}x{imgheight}"#
379             Duration = metadata.get('Duration')#
380             FileCreateDate = metadata.get('
    FileCreateDate')
381             FileAccessDate = metadata.get('
    FileAccessDate')
382             ModificationDate = metadata.get('ModifyDate
    ')#
383             DeviceMake = metadata.get('Make')#
384             DeviceModel = metadata.get('Model')#
385             # extracted GPS video information
386             GPSPosition = metadata.get('GPSPosition')
387             if GPSPosition is not None:
388                 latitude = metadata.get('GPSLatitude')
389                 longitude = metadata.get('GPSLongitude'
    )
390                 latitude_decimal =
    convert_coordinates_gps(latitude)
391                 longitude_decimal =
    convert_coordinates_gps(longitude)
392                 google_maps_link = f"\"https://www.
    google.com/maps?q={latitude_decimal},{
    longitude_decimal}\""
393             else:
394                 google_maps_link = "-"
395             data_object.append([Paragraph('Video
    Information', style=sub_title)])
396             data_object.append([
397                 '\nVideo Name: '+ str(FileName) +
398                 '\nFile Type: '+ str(FileType) + #
```

```
399                 '\nVideo File Size: '+ str(FileSize) +
400                 '\nMD5 Hash Value: '+ str(file_md5) +
401                 '\nFrame Rate: '+ str(FrameRate) + #
402                 '\nFrame Size: '+ str(FrameSize) + #
403                 '\nDuration: '+ str(Duration) + #
404                 '\nCreation Time: '+ str(FileCreateDate
    ) +
405                 '\nLast Access Time: '+ str(
    FileAccessDate) +
406                 '\nModification Date: '+ str(
    ModificationDate) + #
407                 '\nDevice Make: '+ str(DeviceMake) + #
408                 '\nDevice Model: '+ str(DeviceModel) +
    #
409                 '\nLocation: '+ str(GPSPosition) +
410                 '\nGoogle Map Link: ' + str(
    google_maps_link) +
411                 '\n'
412             ])
413             for object in video[1]:
414                 # Convert NumPy array to PIL Image
415                 pil_img = Image.fromarray(object[3])
416                 # from IPython.display import display
417                 # display(pil_img)
418                 """ Convert PIL Image to bytes, allows
    you to work with the image data in-memory without
    having to save it to a file on disk."""
419                 img_bytes = io.BytesIO()
420                 pil_img.save(img_bytes, format='PNG')
421                 img_bytes.seek(0)
422                 img = RLImage(img_bytes, width=(
    imgwidth/300)*inch, height=(imgheight/300)*inch)
423                 print(object)
424                 data_object.append(['Keyword: '+ str(
    object[0]) + '\nTimestamp: \nStart: '+ object[1]+ '
    - End: '+ object[2]+'\nConfidence: '+ str(object[4])
    ])
```

```
425                 data_object.append([img])
426     if configuration['ocr']:
427         data_ocr.append([Paragraph('Parsing Result of
    Optical Character Recognition', style=sub_title)])
428         for video in configuration['result_ocr']:
429             data_ocr.append([Paragraph('Video Path: '+
    video[0], style=video_path_style)])
430             file_md5 = md5(video[0])
431             # extracted video information
432             metadata = extract_video_metadata(video[0])
433             imgwidth = metadata.get('ImageWidth')
434             imgheight = metadata.get('ImageHeight')
435             FileName = metadata.get('FileName')
436             FileType = metadata.get('FileType')#
437             FileSize = metadata.get('FileSize')
438             FrameRate = metadata.get('VideoFrameRate')#
439             FrameSize = f"{metadata.get('ImageWidth')}x
    {metadata.get('ImageHeight')}"#
440             Duration = metadata.get('Duration')#
441             FileCreateDate = metadata.get('
    FileCreateDate')
442             FileAccessDate = metadata.get('
    FileAccessDate')
443             ModificationDate = metadata.get('ModifyDate
    ')#
444             DeviceMake = metadata.get('Make')#
445             DeviceModel = metadata.get('Model')#
446             # extracted GPS video information
447             GPSPosition = metadata.get('GPSPosition')
448             if GPSPosition is not None:
449                 latitude = metadata.get('GPSLatitude')
450                 longitude = metadata.get('GPSLongitude'
    )
451                 latitude_decimal =
    convert_coordinates_gps(latitude)
452                 longitude_decimal =
    convert_coordinates_gps(longitude)
```

```
453                 google_maps_link = f"\"https://www.
    google.com/maps?q={latitude_decimal},{
    longitude_decimal}\""
454             else:
455                 google_maps_link = "-"
456             data_ocr.append([Paragraph('Video
    Information', style=sub_title)])
457             data_ocr.append([
458                 '\nVideo Name: '+ str(FileName) +
459                 '\nFile Type: '+ str(FileType) + #
460                 '\nVideo File Size: '+ str(FileSize) +
461                 '\nMD5 Hash Value: '+ str(file_md5) +
462                 '\nFrame Rate: '+ str(FrameRate) + #
463                 '\nFrame Size: '+ str(FrameSize) + #
464                 '\nDuration: '+ str(Duration) + #
465                 '\nCreation Time: '+ str(FileCreateDate
    ) +
466                 '\nLast Access Time: '+ str(
    FileAccessDate) +
467                 '\nModification Date: '+ str(
    ModificationDate) + #
468                 '\nDevice Make: '+ str(DeviceMake) + #
469                 '\nDevice Model: '+ str(DeviceModel) +
    #
470                 '\nLocation: '+ str(GPSPosition) +
471                 '\nGoogle Map Link: ' + str(
    google_maps_link) +
472                 '\n'
473             ])
474             for ocr in video[1]:
475                 # Convert NumPy array to PIL Image
476                 pil_img = Image.fromarray(ocr[3])
477                 # from IPython.display import display
478                 # display(pil_img)
479                 """Convert PIL Image to bytes, allows
    you to work with the image data in-memory without
    having to save it to a file on disk."""
```

```
480                 img_bytes = io.BytesIO()
481                 pil_img.save(img_bytes, format='PNG')
482                 img_bytes.seek(0)
483                 img = RLImage(img_bytes, width=(
    imgwidth/300)*inch, height=(imgheight/300)*inch)
484                 data_ocr.append(['Keyword: '+ ocr[0] +
    '\nTimestamp: \nStart:'+ ocr[1]+' - End: '+ ocr[2] +
     '\nConfidence: '+ str(ocr[4])])
485                 data_ocr.append([img])
486  if configuration['speech']:
487      data_speech.append([Paragraph('Parsing Result
    of Speech Recognition', style=sub_title)])
488      for video in configuration['result_speech']:
489          data_speech.append([Paragraph('Video Path:
    '+ video[0], style=video_path_style)]) #
    ########################################
490          file_md5 = md5(video[0])
491          # extracted video information
492          metadata = extract_video_metadata(video[0])
493          FileName = metadata.get('FileName')
494          FileType = metadata.get('FileType')#
495          FileSize = metadata.get('FileSize')
496          FrameRate = metadata.get('VideoFrameRate')#
497          FrameSize = f"{metadata.get('ImageWidth')}x
    {metadata.get('ImageHeight')}"#
498          Duration = metadata.get('Duration')#
499          FileCreateDate = metadata.get('
    FileCreateDate')
500          FileAccessDate = metadata.get('
    FileAccessDate')
501          ModificationDate = metadata.get('ModifyDate
    ')#
502          DeviceMake = metadata.get('Make')#
503          DeviceModel = metadata.get('Model')#
504          # extracted GPS video information
505          GPSPosition = metadata.get('GPSPosition')
506          if GPSPosition is not None:
```

```
507                 latitude = metadata.get('GPSLatitude')
508                 longitude = metadata.get('GPSLongitude'
    )
509                 latitude_decimal =
    convert_coordinates_gps(latitude)
510                 longitude_decimal =
    convert_coordinates_gps(longitude)
511                 google_maps_link = f"\"https://www.
    google.com/maps?q={latitude_decimal},{
    longitude_decimal}\""
512             else:
513                 google_maps_link = "-"
514             data_speech.append([Paragraph('Video
    Information', style=sub_title)])
515             data_speech.append([
516                 '\nVideo Name: '+ str(FileName) +
517                 '\nFile Type: '+ str(FileType) + #
518                 '\nVideo File Size: '+ str(FileSize) +
519                 '\nMD5 Hash Value: '+ str(file_md5) +
520                 '\nFrame Rate: '+ str(FrameRate) + #
521                 '\nFrame Size: '+ str(FrameSize) + #
522                 '\nDuration: '+ str(Duration) + #
523                 '\nCreation Time: '+ str(FileCreateDate
    ) +
524                 '\nLast Access Time: '+ str(
    FileAccessDate) +
525                 '\nModification Date: '+ str(
    ModificationDate) + #
526                 '\nDevice Make: '+ str(DeviceMake) + #
527                 '\nDevice Model: '+ str(DeviceModel) +
    #
528                 '\nLocation: '+ str(GPSPosition) +
529                 '\nGoogle Map Link: ' + str(
    google_maps_link) +
530                 '\n'
531             ])
532             for target in video[1].keys():
```

```
533                 data_speech.append(['keyword: '+ target
    ])
534                 for speech in video[1][target]:
535                     data_speech.append(['Timestamp: \
    nStart: '+ speech[1] + ' - End: '+ speech[2] + '\
    nConfidence: '+ str(speech[3]) + '\nThe Context: '+
    speech[4] ])
536     # Specify column widths
537     colWidths = [6.5*inch]
538     # Generate rowHeights to match the number of rows
539     row_height_for_header = 0.5 * inch
540     row_height_for_other_rows = 1.4 * inch
541     rowHeights = [row_height_for_header] + [
    row_height_for_other_rows] * (len(data) - 1)  #
    Adjust row heights
542     # Create the table with specified dimensions
543     table = Table(data)
544     table.setStyle(table_style)
545     # Building the PDF
546     elements = [title_paragraph, spacer, table]
547     if configuration['object']:
548         table_object = Table(data_object)
549         table_object.setStyle(table_style_result)
550         elements.append(spacer)
551         elements.append(table_object)
552     if configuration['ocr']:
553         table_ocr = Table(data_ocr)
554         table_ocr.setStyle(table_style_result)
555         elements.append(spacer)
556         elements.append(table_ocr)
557     if configuration['speech']:
558         table_speech = Table(data_speech)
559         table_speech.setStyle(table_style_result)
560         elements.append(spacer)
561         elements.append(table_speech)
562     doc.build(elements)
563     print("PDF generated successfully!")
```

```
564 ## The Main Function that Process VCAT Requests
565 # from google.colab.patches import cv2_imshow
566 from datetime import datetime
567 import time
568
569 def main(configuration):
570     # record start time-------------------------
571     start_time = time.time()
572     start_date = datetime.now().strftime("%H:%M:%S %d/%
    m/%Y")
573     ocr = None
574     dino_model = None
575     transform = None
576     wsr = None
577     option = None
578     if configuration['object']:
579         # load Dino model
580         dino_model = load_model(CONFIG_PATH,
    WEIGHTS_PATH, 'GPU') ##### comment GPU if CPU
581         transform = T.Compose(
582             [
583                 T.RandomResize([800], max_size=1333),
584                 T.ToTensor(),
585                 T.Normalize([0.485, 0.456, 0.406],
    [0.229, 0.224, 0.225])
586             ]
587         )
588     if configuration['ocr']:
589         # Initialize PaddleOCR
590         ocr = PaddleOCR(use_angle_cls=True, lang='en')#
    , use_gpu=True)  # Assuming CPU usage
591         # ocr = easyocr.Reader(['en'], gpu=False)  #
    Assuming CPU usage # this line for easyosr
592     if configuration['speech']:
593         # Initialize whisper
594         wsr = whisper.load_model(
595                 name='base.en',
```

```
596                 # device= 'cpu', # comment it if GPU
597                 in_memory= True
598             )
599         option = whisper.DecodingOptions(language='en',
     fp16=False)
600         videos_paths = find_videos(configuration['
    root_directory'])
601         result_ocr = []
602         result_object = []
603         result_speech = []
604         print(videos_paths)
605         for video_path in videos_paths:
606             output_ocr = []
607             output_object = []
608             output_speech = []
609             metadata = extract_video_metadata(video_path)
610             frame_rate = metadata.get('VideoFrameRate')
611             if configuration['object']:
612                 object_data = get_object(dino_model,
    transform, predict, annotate, video_path,
    configuration['object_val'])
613                 output_object = process_frames(object_data,
     configuration['frame_threshold'], frame_rate)
614                 if len(output_object) > 0:
615                     result_object.append([video_path,
    output_object])
616                     print('result_object', result_object)
617             if configuration['ocr']:
618                 ocr_data = get_ocr(ocr, video_path,
    configuration['ocr_val'])
619                 output_ocr = process_frames(ocr_data,
    configuration['frame_threshold'], frame_rate)
620                 if len(output_ocr) > 0:
621                     result_ocr.append([video_path,
    output_ocr])
622                     print('result_ocr', result_ocr)
623             if configuration['speech']:
```

```
624             output_speech = process_speech(wsr, option,
    video_path, configuration['speech_val'])
625             if len(output_speech) > 0:
626                 result_speech.append([video_path,
    output_speech])
627                 print('result_speech', result_speech)
628     # take end time----------------------
629     end_time = time.time()
630     end_date = datetime.now().strftime("%H:%M:%S %d/%m
    /%Y")
631     # Calculate the execution time
632     execution_time = end_time - start_time
633     print("-------------------------------------
    Execution time:", execution_time, "seconds")
634     # write the pdf file
635     if configuration['report']:
636         investigation_info = configuration['
    investigation_info']
637         investigation_info['start_date'] = start_date
638         investigation_info['end_date'] = end_date
639         investigation_info['duration'] = execution_time
640         pdf_configuration = {
641             'investigation_info': investigation_info,
642             'ocr' : configuration['ocr'],
643             'object' : configuration['object'],
644             'speech' : configuration['speech'],
645             'result_ocr' : result_ocr,
646             'result_object' : result_object,
647             'result_speech' : result_speech,
648         }
649         make_pdf(pdf_configuration)
650 ## The Starting Point for Proposed Digital Forensics
    Video Contant Analysis Tool (VCAT)
651 #@markdown ###Digital Forensic Video Contant Analysis (
    VCAT)
652 #@markdown ---
653 #@markdown ####1. Video Analyst Information:
```

```
654
655 analyst_name = "Ruwa' Abu Hweidi" #@param {type:"string
    "}
656 analyst_organization = "PTUK" #@param {type:"string"}
657 analyst_department = 'Cybercrime_PTUK' #@param {type:"
    string"}
658 #@markdown ---
659
660 #@markdown ####2. Case Information:
661 case_number = '7102024' #@param {type:"string"}
662 report_id = "1948" #@param {type:"string"}
663 #@markdown ---
664
665 # # This step valid if the forensic image is extracted
    in the given path
666 # #@markdown ####Forensic Image Information:
667 # Forensic_Image_Path = "/content/drive/MyDrive/VCAT/
    project/data" #@param {type:"string"}
668 # #@markdown ---
669
670 #@markdown ####3. Input Path:
671 root_directory = '/content/drive/MyDrive/VCAT/project/
    data' #@param {type:"string"}
672 #@markdown ---
673
674 #@markdown ####4. Outpu Path:
675 report_path = "/content/drive/MyDrive/VCAT/project/
    result" #@param {type:"string"}
676 #@markdown ---
677
678 #@markdown #### 5. Keyword/Prompt:
```

```
# target_objects = 'Ruwa, MD-59, Post, Match direction of car, try it today, man, from, dear, user, Washington Bivd, 033-YCS, 59.legoland, list2023, step, parking, Final Shot, Original Plate, EACH PAGE, DOWNLOAD, depend on user restrictions, final mobile, mobile edit, mobile, data, hash, telegram, to continue with results, extraction, salamalicom, trim, function, important, soccer, ball, thats right, focus, fuck, tough guy, mother, mother fucker,man shadow, bag, walking man shoes, hat, yellow bike, wheel of black car, right buildings, umbrella' #@param {type:"string"}
target_objects = "solider, killer, kitchen, fight, Freaking awesome with that, I'm going to chop off your goddamn head, dull knife, gangster, Shit, tough stuff, give you five bucks, goalie, best of the best, soccer player, What's your name, soccer, ball, thats right, focus, fuck, tough guy, mother fucker" #@param {type:"string"}
target_objects = target_objects.split(',')
#@markdown ---
#@markdown #### 6. Search Choices:
Optical_Character_Recognition = False # @param {type:"boolean"}
Object_Detection = False # @param {type:"boolean"}
Speech_Recognition = False  # @param {type:"boolean"}
Genarate_Report = True  # @param {type:"boolean"}
#@markdown ---
#@markdown #### Analizing & Genarating the Report ...
# @title Click 'Show code' in the code cell. { display-mode: "form" }

investigation_info = {
    # Video Analyst Information
    'analyst_name': analyst_name,
    'analyst_organization': analyst_organization,
    'analyst_department': analyst_department,
```

```
    # Case Information
    'case_number': case_number,
    'report_id': report_id,
    'sys_date': datetime.now().strftime("%d/%m/%Y"),
    # Work Time
    # 'start_date': datetime.now().strftime("%d/%m/%Y")
    ,
    # 'end_date': datetime.now().strftime("%d/%m/%Y"),
    # Forensic Image Information
    'Forensic_Image_Path': root_directory,
    'Forensic_Image_size': '390.6GB',
    'MD5_Hash_Value': '7D1D5D54E9FE1E94CEEE88120BEC4A56
    '
}
project_configuration={
    'investigation_info': investigation_info,
    "root_directory":root_directory,
    'object': Object_Detection,
    'object_val': target_objects,
    'ocr': Optical_Character_Recognition,
    'ocr_val': target_objects,
    'speech': Speech_Recognition,
    'speech_val': target_objects,
    'report': Genarate_Report,
    'report_path': report_path,
    'frame_threshold': 15, # one frame in each 15 frame
    without any differance
    # 'conf_threshold': 0.5
    }
main(project_configuration)
```

Listing 1: Python code of the experiment VCAT

Appendix B

Speech recognition models sample results of spoken text in Whisper AI, Google Recognizer, and Wav2vec Meta respectively.

1. Spoken text for Whisper, with each word probability, are:


```
{ 'start': 39.48, 'end': 55.76, 'text': ' To continue with
   results, the successful extraction of videos, file and
   images, depends on user', 'tokens': [51065, 1675, 2555,
   351, 2482, 11, 262, 4388, 22236, 286, 5861, 11, 2393,
   290, 4263, 11, 8338, 319, 2836, 51441], 'temperature':
   0.0, 'avg_logprob': -0.3541427894874855, '
   compression_ratio': 1.4378698224852071, 'no_speech_prob':
    0.532045304775238, 'words': [{'word': ' To', 'start':
   39.48, 'end': 39.84, 'probability': 0.9744758605957031},
   {'word': ' continue', 'start': 39.84, 'end': 40.3, '
   probability': 0.9928157329559326}, {'word': ' with', '
   start': 40.3, 'end': 40.54, 'probability':
   0.9916259050369263}, {'word': ' results,', 'start':
   40.54, 'end': 40.86, 'probability': 0.7976126074790955},
   {'word': ' the', 'start': 41.4, 'end': 41.82, '
   probability': 0.8641127943992615}, {'word': ' successful'
   , 'start': 41.82, 'end': 42.26, 'probability':
   0.9893069863319397}, {'word': ' extraction', 'start':
   42.26, 'end': 43.12, 'probability': 0.9964053630828857},
   {'word': ' of', 'start': 43.12, 'end': 43.74, '
   probability': 0.9945244789123535}, {'word': ' videos,', '
   start': 43.74, 'end': 44.2, 'probability':
   0.9796793460845947}, {'word': ' file', 'start': 44.7, '
   end': 44.88, 'probability': 0.7820780277252197}, {'word':
    ' and', 'start': 44.88, 'end': 45.26, 'probability':
   0.9132970571517944}, {'word': ' images,', 'start': 45.26,
    'end': 45.68, 'probability': 0.9954653382301331}, {'word
   ': ' depends', 'start': 45.68, 'end': 46.18, 'probability
   ': 0.9552022218704224}, {'word': ' on', 'start': 46.18, '
   end': 55.28, 'probability': 0.9981535077095032}, {'word':
    ' user', 'start': 55.28, 'end': 55.76, 'probability':
   0.9896223545074463}]}
```


Listing 2: Whisper AI

2. Spoken text for Google Recognizer, with a probability of the text as one piece is:

```
('continue with the results the successful extraction of
    videos file and images depends on user restrictions or
    per',
     0.84709144
)
```

Listing 3: Google Recognizer

3. Spoken text for Wav2vec Met, without probability, is:

```
{
'text':
     'TINU WITH RESULTS THE SUCCESSFUL EXTRACTION OF FEDIOS
    FILE AND IMAGES DEBENTS ON USE OUR RESTRICTIONS OR
    PENSIONS'
}
```

Listing 4: Wav2vec Meta

أداة رقمية جنائية مقترحة لتحليل المحتوى المرئي في عملية التحقيق: التغلب على العقبات وتحسين الفاعلية

A Proposed Digital Forensic Tool for Video Content Analysis in the Investigation Process: Overcoming Challenges and Improving Efficiency

إعداد

رواء فايق سليمان أبوهويدي

إشراف

د. إيمان دراغمة

قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير
في علم الجرائم الالكترونية وتحليل الادلة الرقمية

كليّة الدراسات العليا

جامعة فلسطين التقنية – خضوري

2024

الملخص

حفز الاعتماد المتزايد على الوسائط الرقمية في التحقيقات الجنائية على تطوير أداة تحليل محتوى الفيديو ،(VCAT) وهي حل جنائي رقمي متطور مصمم لتعزيز تحليل بيانات الفيديو. تدمج أداة VCAT نماذج رائدة في مجال التعلم الآلي بما في ذلك Ground-ingDINO للكشف عن الكائنات، و للتعرفPaddleOCR البصري على الأحرف، و AI Wisper للتعرف على الكلام، مما يسهل اتباع نهج شامل في الطب الشرعي. تُمكِّن هذه التقنيات VCAT من معالجة محتوى الفيديو من مصادر متنوعة مثل كاميرات المراقبة ووسائل التواصل الاجتماعي بفعالية، مما يحقق دقة وكفاءة عالية. بالإضافة إلى ذلك، تسمح وظيفة البحث المتقدم المستند إلى النصوص في VCAT بإجراء تحقيقات مستهدفة باستخدام كلمات رئيسية أو مطالبات محددة. وتسلط التقييمات التجريبية الصارمة التي أجريت باستخدام Collaboratory Google الضوء على أداء VCAT المتفوق على أدوات الطب الشرعي القياسية في الكشف عن عناصر الفيديو المتنوعة. ومن خلال الالتزام الصارم بمعايير المعهد الوطني للمعايير والتكنولوجيا (NIST) والحفاظ على سلامة سلسلة الحفظ، يضمن VCAT أن تكون مخرجاته مقبولة قانونيًا. يتم تنسيق التقارير النهائية التي يتم إنشاؤها بواسطة VCAT بدقة عبر ReportLab لتلبية المعايير القانونية. وعلى الرغم من قدراته القوية، فقد تم تحديد تحديات مثل كفاءة المعالجة والتعامل مع بيانات الفيديو منخفضة الجودة، والتي تحتاج لتحسينات مستقبلية. يمثل VCAT تقدمًا كبيرًا في تكنولوجيا الطب الشرعي الرقمي، حيث يوفر أداة قابلة للتطوير وفعالة ومتوافقة تعزز بشكل كبير قدرة المحللين الجنائيين على استخراج بيانات الأدلة وتحليلها بسرعة ودقة.

الكلمات المفتاحية :الطب الشرعي الرقمي، والمقبولية القانونية، والتعلم الآلي، وNIST، والكشف عن الكائنات، والتعرف الضوئي على الحروف، والتعرف على الكلام، والبحث القائم على النص، وتحليل الفيديو.