Blockchain-Based Traffic Surveillance Footage Authenticity Detection System

by

Tasnimul Bin Moshiur
18301014
Mohammad Zafar Ullah
18201153
Nahian Nawar
19241015
Tawsif Muhammed Tazwar
18301012
Rifah Nanjiba
19101522

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

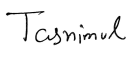
Department of Computer Science and Engineering
School of Data and Science
Brac University
January 2023

Declaration

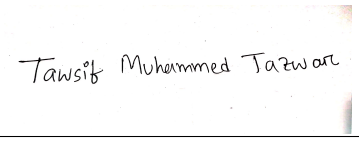
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Tasnimul Bin Moshiur
18301014



Tawsif Muhammed Tazwar
18301012

Nahian Nawar
19241015

Mohammad Zafar Ullah
18201153

Rifah Nanjiba
19101522

Approval

The thesis/project titled "Blockchain-Based Traffic Surveillance Footage Authenticity Detection System" submitted by

1. Tasnimul Bin Moshiur (18301014)
2. Tawsif Muhammed Tazwar (18301012)
3. Mohammad Zafar Ullah (18201153)
4. Nahian Nawar (19241015)
5. Rifah Nanjiba (19101522)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 19, 2023.

Examining Committee:

Supervisor:
(Member)

Dr.Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr.Golam Robiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr.Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

With the advancement in technology, fraudulent videos are becoming harder to detect and easier to produce. Surveillance footage can serve as circumstantial evidence when dealing with crimes, however when this footage is tampered with, there is a great loss in evidence and the footage loses its value. To combat this growing problem, in this paper, we aim to find a new system to determine authenticity in a video for security measures based on Blockchain & Deep Learning Tools. The importance of Blockchain in this era of time is gradually increasing due its decentralized features, fault-tolerance attribute, immutability. This paper is looking forward to introducing a system which protects the surveillance footage gathered from a camera in a faster and optimal approach so that the authenticity can be checked and protected. Our goal is to implement a system which would secure the importance of crucial footage as evidence.

Keywords: Blockchain; Surveillance Footage; Hashing; Cryptography; Deep Learning; CNN.

Acknowledgement

Firstly, all praise to the Almighty Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Muhammad Iqbal Hossain sir for his kind support and advice in our work. He guided us throughout our thesis work.
And finally to our parents, without their support we would not have been able to reach here.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blockchain, as the name suggests, is a chain of interconnected nodes holding data which is linked by encryption [13] in a decentralized and secured manner. The nodes are called blocks and to create a block two objects are required; them being the data and the hash of the data at the previous block. Any data input to the Blockchain is converted to a hash value which is then stored and linked to the previous block in the chain. Every node is created in this manner with the exception of the Genesis Block. This block being the first of the chain does not have any hash value of its previous block.
Video surveillance systems have evolved into an essential management tool for cities over the last few decades. Video surveillance systems are now imperative for security in banks, transit, and prisons. They are also widely utilized in residential areas, and shopping malls in urban regions, particularly in areas where people move around a lot. As crime and accident rates are typically higher in densely populated metropolitan regions, video surveillance systems serve a critical role in the areas of urban public safety, smart mobility, and crime prevention. Video monitoring systems are also important for supporting the growth of a peaceful community, environment and ensuring social stability. Basically, in a surveillance system, video surveillance devices and equipment are installed at a particular distance to ensure that target objects, persons or incidents of that area are captured. Video surveillance is used to a great degree in the traffic system to monitor vehicles, analyze any unprecedented events and also to detect other criminal incidents. Small vulnerabilities in this system can open up multitudes of exploitation possibilities for criminals with which they can get away easily. Hence, the maintenance of these surveillance footage must conform to the CIA triad - confidentiality, integrity and availability. It is of utmost importance to guarantee the clarity of video content and that its contents have not been doctored or distorted when examining it. The surveillance footage must be maintained carefully, but if an unprecedented event occurs in which the videos are leaked to unauthorized persons, who then watch and modify them, personal information is harmed, and the incident's transparency is harmed as well. Furthermore, the majority of video surveillance systems rely on the monitoring crew to watch the monitoring screen in real time and judge any unusual events that appear on the screen. In this scenario, manipulation of CCTV footage in attempts to cover up crimes and tampering with video evidence can be done by the internal manager of the video surveillance system. These types of malicious incidents are indeed happening in the legal, medical, banking field. So currently, such videos are

either not protected against tampering or even if they are, they are mainly protected using digital signatures such as watermark which is not difficult to forge with the advancement in technology.
To address this issue, we offer a method that detects illicit video content alteration, more specifically in traffic control systems. To store the authorized content, we recommend using the Blockchain Network. Within our proposal we want to store the video contents from the CCTV, IP surveillance cameras in the Blockchain. Furthermore, we want to make sure that the uploaded content in the Blockchain is untempered and there is no personnel or human intervention while the process is taking place. Therefore we want to use an automated procedure which will transfer the video data to the Blockchain with authenticity. So there remains no scope of the content being mutated when it is being uploaded to the Blockchain. Instead of storing the hash of the entire video in the Blockchain, we look for frames containing traffic accidents to be hashed in the Blockchain. When a particular video needs to be inspected then the hash of the data stored in the Blockchain network can be used as a reference to detect fraudulence in the video.

1.1 Motivation

Blockchain is becoming the driving force of the next industrial revolution with its wide range of use cases. But whenever we hear the word, nothing other than the picture of bitcoin crosses our mind. Mainly, this motivated us to do something different with the concept of Blockchain. Since most of the research is being made on developing the security and optimization of bitcoin transactions, our motivation was to do something unique which would unlock the potential to build a Blockchain-enabled nation. According to the guideline named National Blockchain Strategy: Bangladesh which was published by ICT Division back in March 2020, a strategic approach should be made to explore Blockchain opportunities to meet the new challenges. Our research purpose was to help build this idea coming to life. The motivation behind this mission was to integrate Blockchain with Deep Learning methods and to invent something beautiful. Our research goes along with the idea of deploying Blockchain based solutions in a Judiciary system where the digital evidences like CCTV footage are more secure and tamper proof. Not only that, this has the potential to be applied with Smart City Infrastructure and Protection of crucial digital assets. Through the proposed idea, we can optimize security, analyze surveillance systems, securely encrypt and store evidence data, while making data more accurate and tamper-proof in a way. Surveillance has come a long way, but there is still a long way to go. We would like to take this idea further and contribute to the privacy and security system.

1.2 Research Problem

Streaming video technology and media have advanced rapidly over the years, beginning with a simple 3GP format and progressing to MP4 and now H264/AVC and H265/HEVC formats. With the industry's rapid expansion, many additional forms of digital video are already in use, such as multimedia cloud, live streaming, and video applications supporting 8K Ultra High Definition (UHD) [23]. Along with

this, the risk of tampering with these media is also on the rise and it affects the legitimate owners, authorities and parties involved with the resources.
Security of digital media has become a burning question in recent years and one of the many ways that we are famously acknowledged with is video surveillance system with CCTV. These footage need special tools to protect data from unauthorized access. CCTV is being used in road safety, monitoring systems, access controls [16]. But with the uprising, the main concern of security still remains while the data is in the process of receiving, storing or transmitting.
According to [14], around 49% percent of police departments in the US take help of surveillance cameras to tackle crimes. Moreover, as the footage from these cameras are admissible to courts they play a crucial role in maintaining justice in the society. From here we can have a clear idea of the role of these surveillance cameras in our society and the importance of their footage in the fight against crimes. Thus the authenticity of these said footage is of utmost importance to not just solve a crime, and bring justice but to save countless lives.
Surveillance footage can be easily tampered by crooked people, and as we have said above this leads to injustice. Now, it is very hard to ensure that these videos are not altered by any third parties before going into the hands of authorities. Implementing Blockchain with hashing algorithms is a very effective way to do this, but now comes the question which hashing algorithm should we use. Various papers have used different algorithms according to the metrics they have focused on. Some algorithms are time efficient, some are much more accurate at detecting anomalies while others might have storage complexity.
In the modern era, the public figures are always a target with the publicly and readily obtainable images and videos that are highly used to train DeepFake Algorithms and Models. These are later used as a way to demean them, creating a serious menace to privacy and security with this falsification [18]. These footage can be accessed without any hassle from CCTV cameras of random parking lots, elevators, malls. The security in these places are very naive which later affects many lives in unimaginable ways.
Various professions which include video data transmission like cinematographer, video editor, graphics designer earn their income from selling arts in formats like video or images. But the main issue comes while we talk about copyrighting. Copyright infringement is defined as "making minor changes to a picture or video and receiving credit for the work without being the original creator of the same" [10]. Copyright infringement is present in text documents as well as covering the area of multimedia like images, videos, songs etc [4]. People are sharing videos, pictures of them on social media which are being blatantly doctored or edited in the name of content creation or making a mockery of people. These media are often unprotected and can harm different phases in a human life if it gets in the wrong hands [25].
In this paper we have tried to address this issue and come up with a Digital Video Anomaly Detection System based on Blockchain. Here digital video is strictly confined to that from the surveillance cameras.
There are many techniques that have been explored by others in ensuring the security of the video footage of CCTV cameras. Some have used Blocksee [7] while others have opted to implement IPFS [15]. Hence, we are trying to figure out a more effective way of detecting whether a video has been doctored, with the implementation of Blockchain. By implementing the system using various hashing algorithms

such as BLAKE, HMAC, MD6, SHA-1, SHA-2, Skein, we are trying to see which algorithm performs better at different metrics [3]. Finally, we aim to find an improved system which will outperform others on timing, quality and security metrics and can be deemed as a promising algorithm for Digital Video Anomaly Detection System.

1.3 Research Objectives

In this work, we attempted to address this issue by developing a Blockchain-based Digital Video Anomaly Detection System. Here, digital footage is strictly limited to surveillance cameras. The video footage from CCTV and IP surveillance cameras will be uploaded to a local server, and the hash of each frame will be added to blocks in the Blockchain network, which is an allowed medium. Through this research we aim to develop a method to detect if surveillance footage has been tampered with using Blockchain technology. The objectives of this research are:

1. To understand Blockchain and its working principles
2. To develop a framework that can detect anomalies in digital video
3. To know more about the smart contracts and proof of concept (PoC)
4. Understanding the encryption algorithm like SHA-256 or MD6
5. Developing an idea about how Blockchain and Internet works with multimedia quantities like video or images
6. To offer recommendations on improvement of the model

1.4 Thesis Structure

In our research paper, the first chapter has four parts which introduces the research topic, talks about the motivation along with the research problem behind choosing this domain, also discusses the research objectives and finishes off with thesis structure. The second chapter helps to give an overview about the existing research about our selected domain and what are the differences between theirs and ours. The third chapter goes on to discuss the background studies of Blockchain along with its different types, hashing mechanism and smart contracts technology. The fourth chapter shows the proposed architecture of our thesis along with different techniques of Deep Learning whereas the fifth chapter provides the implementation process along with results of our work. We bring our thesis report ending in the final chapter with Conclusion and Future Works.

Chapter 2

Literature Review

Blockchain has dominated the field of cryptocurrency for decades and over the years this technology has seeped into other fields as well. By 2030, there will be an estimated generation of 3.1 trillion dollars in new business evaluations, which was predicted by Gartner in the year of 2019 [8]. Research and development is underway to transform legal, real estate, banking, healthcare and video industries using Blockchain-based solutions. Since our working area narrows down to the traffic surveillance system rather than the deep sea of security domain, this will try to compare the works from a little different perspective.

This paper [7] suggested BlockSee, a Blockchain-based video surveillance system that ensures the integrity of surveillance videos and their configuration.

BlockSee advocated using segmentation and feature extraction to determine the camera configuration from an image frame. The surveillance footage is encrypted in the Blockchain, so not everyone can see it. The police are given access to the tape when it is requested by them. The timestamp, camera id, hash of video segments for video integrity, and features of the background model for camera settings integrity are all included in BlockSee transactions. Additionally two more bits are added that distinguish between the regular and secure frames.

Implementation on multichain to store information as multichain streams and advancement of BlockSee computer vision by revealing landmarks on screen and recognizing them on obscure frames are two other studies that might enhance this system. While this application has its advantages in keeping the camera configuration intact, it just sends the whole video data in the blockchain system to be fair which, in the long run, will cost a lot of memory along with money. Our research looks forward to integrating deep learning techniques like CNN to detect the useful frames in the video and extract that frame information to later save it in the blockchain network via IPFS. Not only will this pace the system but also will save a lot of memory and the blockchain network won't bloat with unnecessary data.

In this research [22], a system for detecting rules violations by security cameras and preventing video falsification was developed by constructing a trust chain of cameras. This method makes fabrication or falsification of digital video difficult if the target camera is always linked to either a trusted or secured camera in the chain. The motivation behind this proposal is to keep an area safe by rewarding any camera owner who shoots a violation and reports it to the authorities. Hence, more cameras will be set up in places where violations occur regularly. While the motivation sounds promising, it has its limitations. The trusted chain of cameras can

fail any moment. The camera configurations can be altered or fabricated. Also, the usage of blockchain technology in this research paper mostly includes rewarding the camera owner, not for integrating the authenticity of video frames. The fabrication of videos is possible if the target never moves and has no lighting to that target. Our research looks forward to using blockchain technology to detect the anomalies of the video frames as well as the camera configuration. Moreover, it will not depend on the cameras of civilians' personal cameras as they might alter the information for their own misconducts.

In this study [21], it was recommended that surveillance footage be encrypted and preserved in an IPFS node through a blockchain composed of reputable nodes. The decrypting key will be saved in the database of a given node that has group authentication authority. When someone requires viewing access to the footage, they must be approved by the network. However, it also has the possibility of bloating the blockchain due to storing every second of surveillance footage rather than the important one which our proposed technique does using Deep Learning methods.

This study [13] proposes encrypting and saving video obtained from the camera in an IPFS node connected to the Blockchain network, where encrypted films are saved. Instead of being put in the block, the video's cryptographic key is saved in the private database of the Blockchain node with special authorization and can only be handled by the chain code of the block chain. Because the interior manager does not know the video's decryption key, they cannot view or transfer the video without authorization. To export and decode video, one must verify the export of the video on the Blockchain network and be able to produce a permit in order to achieve the film's decryption key. This method is good in the sense that cryptography is properly used to encrypt and decrypt the video files and later asks for permission to get access to the archive. But, the process can take as long as it is pushing the entire stream coming from the camera in the blockchain connected by IPFS. The continuous stream of video data will take a long time to verify and the possibility to get the important information out of the footage in a shorter amount of time gets reduced. Our research works to provide a solution to this as we will store the important information, for example car accident, time of car crash, it will be easier to find the exact frames rather than whole video footage.

The goal of this work [15] was to evaluate aberrant behavior and provide early warning in order to fulfill the function of active monitoring and prevention without human involvement. The processed output data will be transferred to data centers with the use of edge computing since processed data, when contrasted to data sources, significantly reduces data connection bandwidth and improves real-time data processing speed. IPFS will be used to secure data storage on Blockchain. The enormous video data storage is achieved by use of the IPFS storage service. CNNs are used to implement real-time monitoring. Here, the usage of edge computing is appreciated as it is decreasing the time to send the data to the data center from the edge device. But it can be better if smart contracts are used which will improve data access control. If smart contracts are used in this, it can be implemented in a large-scale system. This exact solution is embedded in our works which is believed to better the system. Also, with the use of cryptography, files can't be accessed without permission in the edge devices which can be considered as a flaw to prevent data from forging and altering.

According to [19], the proposed framework divides each transaction into basic and

sensitive data where encryption is needed for the sensitive data so that it is stored in the HEVC video with the help of a steganography algorithm. This algorithm claims to protect the private transaction information of Blockchain and enhance the privacy data embedding capabilities. With a greater compressing rate and excellent quality of HEVC video, it shows promising results. But with the increment of Quantization Parameter values, there is an obvious reduction for the coded videos in terms of quality which is a drawback. But it doesn't really have the concept of surveillance footage authenticity. It talks about privacy of sensitive data in video with the term steganography. However, the authenticity of video from a surveillance camera can not be decided from this process. Our research is looking to increase privacy with the usage of IPFS and Blockchain which will hash and protect the video frame along with timestamp.

The Blockchain technology requires a timestamp server on a peer-to-peer basis which will need to be implied using hash functions. The bitcoin network is based on functions like this [2]. The hash functions are mainly built on various encryption algorithms. Different hash functions like dHash, pHash, wHash are being used widely. While the dHash reduces an image into 8x8 pixels in order to convert the image into a perceptual hash, the pHash reduces it to a 32x32 matrix for the same goal. Although wHash is very much similar to pHash, it uses Discrete Wavelet Transform (DWT) instead of DCT. Meanwhile, all the three functions turn an RGB image to grayscale while doing the perceptual hashing [25]. However, the difference between these hash and our research is that we upload the video frame extracted by CNN algorithm to the IPFS and save the output hash in the blockchain. It doesn't necessarily hash the image and turn it into something else. While all three hashing algorithms turn an image into a grayscale image, it can be hard to identify the colorful part in the actual frame which might be necessary in the context of criminal activity identification, car crash detection and smaller details. This possibility gets eliminated in our research.

Chapter 3

Background

3.1 Blockchain

A Blockchain is a decentralized database that keeps an ever-expanding collection of chronologically ordered entries known as blocks. Cryptography is used to bind these blocks together. Along with transaction information, each block includes a timestamp, a cryptographic hash of the one before it. A Blockchain is a decentralized, distributed, and public digital ledger that is used to record transactions across many machines so that the record cannot be changed retrospectively without affecting all following blocks and network consensus. The four pillars of Blockchain are Shared ledgers, Immutability, Smart Contracts and Consensus.

3.1.1 Types of Blockchain

Currently there are many types of Blockchain in the industry which different organizations use for different purposes.
First of all, Public Blockchains are permissionless, fully decentralized with no individual nodes having any control or more weight over the network. It is completely transparent to any nodes on the network, that is all data transfer history and all future transactions will be visible to any new node than joins. This visibility makes it difficult for any forging or hiding data [20]. A majority agreement is required to validate a new transaction and reach agreement on the present state of the ledger, which is done by a consensus algorithm that is run on every node. Secondly, Private Blockchains on the other hand is not decentralized as it is controlled by a single entity. Although it does work like public Blockchains utilizing P2P connections and decentralization [26]. These Blockchains operate the fastest due to the small size of the Blockchain network and consume less resources and power. Information is opaque to the individuals outside of this approved private network. However, due to it not being decentralized and instead hierarchical, it does not have the security benefits of a public Blockchain. If a third party manages to gain access to the central nodes, they can hijack the entire network. Finally, Consortium Blockchain is a hybrid of private and public Blockchains with characteristics of both. Because not everyone on the internet can participate in the Blockchain network, it is not truly decentralized. This partial decentralization enables users to control Blockchain access and address security and control issues that arise with public Blockchains. Consortium Blockchains continue to operate in the decentralized manner associated with

Blockchains. In contrast to every node in a public Blockchain, the consensus process is carried out by a small group of approved nodes. This speeds up the process of approving a transaction/adding new nodes while using significantly less computational resources and electricity. As the Blockchain is not open to all, the size is fairly small, making it more efficient and scalable [27]. It is still less secure than a public but more secure than a private Blockchain as there is no central authority.

3.1.2 Consensus Algorithm

The idea of consensus algorithms saw the light due to the existing problem of the Byzantine Generals Problem. The issue discusses how nodes in a network come to an agreement on doing a task. It is picturized as different Generals with battalions in a battlefield are waiting to attack a fortress. The success of the mission will depend on the synchronized attacks or defense. The main problem is, the communication process is not as advanced as current times and information can be altered or destroyed to mislead the overall order. Also, if one disloyal General decides to betray the others, what will happen then? How can every node in a system like the battlefield can come to a consensus? To solve this problem, comes the Byzantine Fault Tolerance protocol. As nodes can generate arbitrary data, it aims to solve the issue of reaching agreement, BFT works as a replication algorithm which ensures safety and security and liveness of the system i.e. protection from malicious nodes and continue working even if faulty nodes are discovered without pausing the system. It handles up to (n-1)÷3 malicious participants on the network where n = participating nodes. Over the evaluation of Blockchain, this Byzantine Fault Tolerance has given rise to various different consensus algorithms which are widely used. From Distributed Ledger to Proof of Stake, Blockchain has revolutionized the Consensus Mechanism. Some of them are listed below:

1. Proof of Work (PoW): It is another revolutionary protocol to authenticate newly entered nodes in the system. The PoW system searches for hashed values with hashes beginning with a number of zero bits. The hash is found by appending a nonce to the original value until the resulting hash begins with the required number of zero bits. Once this nonce has been discovered and the proof of work has been met, the block cannot be modified without redoing all subsequent blocks' work [6]. Although it is an energy consuming process as verification takes a lot of time and computational effort, it still works as attackers will not use huge amounts of energy just to get detected later on.

2. Distributed Ledger Technology: DLT is a technological architecture and protocols that allow for unchangeable simultaneous access, authentication, and data updates across a network that spans multiple entities or locations. It is a mechanism that enables the secure operation of a distributed digital database and reduces the need for a centralized authority to monitor for manipulation. If any change is there on any node, it will first verify the changes and update it in the ledger to redistribute it to the other participating nodes. It is the basic foundation of Blockchain.

3. Proof of Stake: PoS is an advancement from Proof of Work as it doesn't need every node in the network to verify the newly entered node. The system will

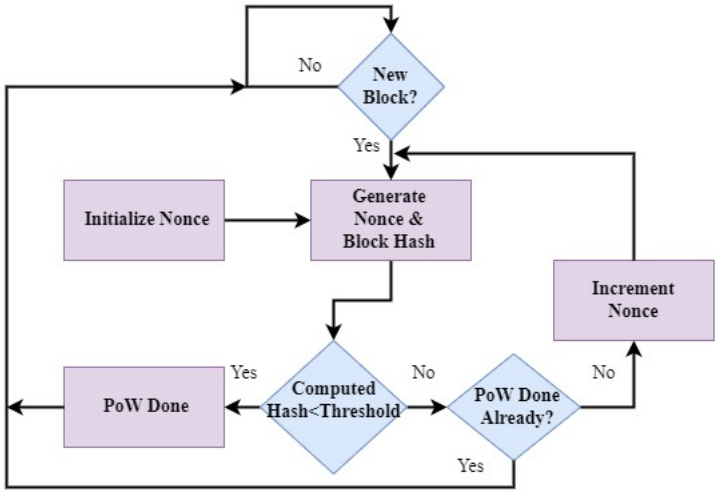


Figure 3.1: Proof of Work

select one among few authenticators based on the amount of reputation and tokens it has in the system. Its job is to verify the new node coming in the network after it has deposited a deposit. Failing to verify the new node or allowing a malicious node will result in losing the deposit and earning demerit points. The energy usage is less than PoW as well as the computing time is way faster.

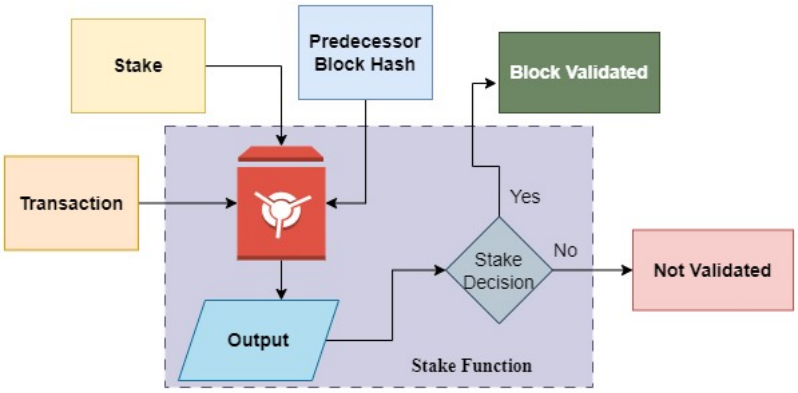


Figure 3.2: Proof of Stake

3.2 Hashing

A hash serves as the block's unique identification in the Blockchain system [5]. The hashed data are irreversible. Moreover they function as the basis of the Blockchain as they also connect the entire Blockchain network. So, the hashing process takes a transaction or data as an input and gives an output of specific length with the help of a hashing algorithm. So, in our methodology the hashing algorithm known as SHA-256 (Secure Hashing Algorithm 256) is being applied. One important point of hashing is that It is a one-way cryptographic process, not an "encryption," since we can't decrypt the hash to retain the original data. In the Blockchain system by using cryptocurrencies, transactions or data of various lengths is sent through the

hashing process, and each time it produces an output of a specified length. SHA-256 algorithm always produces a fixed-length final output, which is a 256-bit length or 32-bytes of hexadecimal data.

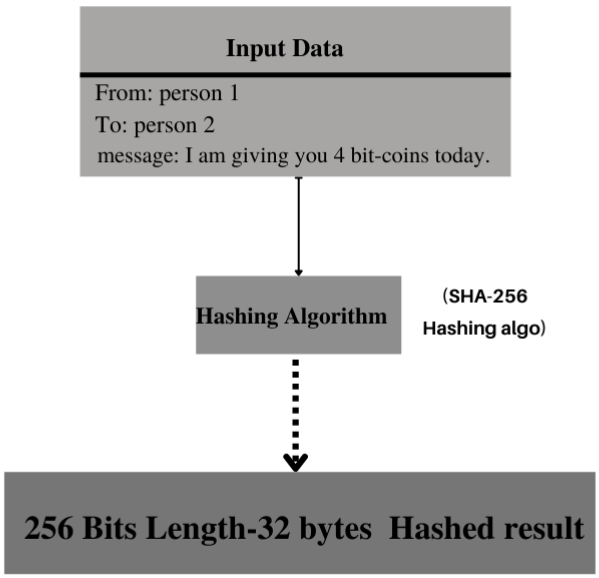


Figure 3.3: Use of hashing function to create unique identity

For example, in 3.3 the hashed value of both input Data segment 1 and Data segment 2 is of equal length regardless of the input size. A transaction happens between Person 1 and 2 in a single block and that record is saved by encrypting it with a hash function. This 32-byte unique hash will change if any information in this particular data is altered.
3.4 illustrates a simple overview of the hashing mechanism for the algorithm. To make a block secure in the Blockchain there needs to be a difficulty level in calculating the hash, so that the hashes are both non-deterministic and unique. The datasets that are used to construct these hashes are (a)Timestamp, (b)Prior Block's Hash, (c)Nonce value, (d)Stored Data in The Block The timestamp, also referred to as a sequence number, is a little amount of data recorded as a unique serial number in each block, whose principal purpose is to detect the precise time when the block was formed and confirmed by the block-chain network [9]. This date is then used to produce a specific hash for a single block in the block-chain, which improves its integrity. Then there is the previous block's hash which is a 64-bit SHA256 hash that is immutable, implying it can never be tampered with and is also utilized generating hash of the subsequent block. A nonce is a one-time-only random value used in cryptographic encryption. They are frequently composed of pseudo-random or random numbers. In proof-of-work systems, nonces are used to alter the input to a cryptographic hash function such that a hash for a given input matches certain arbitrary requirements. As a result, creating a 'desirable hash becomes significantly more difficult than validating it.

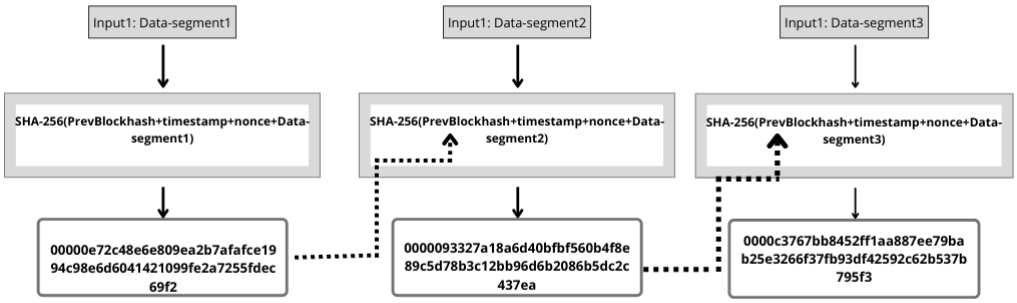


Figure 3.4: SHA-256 Algorithm Process

3.3 Chaining the Blocks

Blockchain is a chain of blocks, as its name indicates, and we'll look at how this system works. The data, its unique hash, and the prior block's hash are the three basic components. The initial block is known as the Genesis block, as it contains no hash of the preceding block since none exists. In 3.5 we can see how the blocks in a Blockchain network are connected in an immutable way. The hash values for blocks n-1, n, and n+1 are 64 bit SHA-256 hash. Now, if someone modifies data in Block n-1, the hash of that block will change, and the new hash will not be preserved in Block n. As shown in 3.6, (Block n-1 and Block n) are no longer connected. To keep a single change in the chain, all blocks must change their hash, which cannot go unnoticed.

3.4 Cryptography

Cryptography is a way of securing messages to protect the data being read by unauthorized parties. It involves encryption of data with a key and decryption of the encrypted data using the same or a different key [12].
Whereas Asymmetric encryption is the process of encrypting and decrypting data using two separate keys - a public and private key pair. Data is encrypted using the public key. Anyone in possession of someone's public key can encrypt the data but to decrypt it the undisclosed private key is required. The sender acquires the receiver's public key and encrypts the message they want to send. At the receiver end the message is decrypted to show the message using the private key. As the private key is never

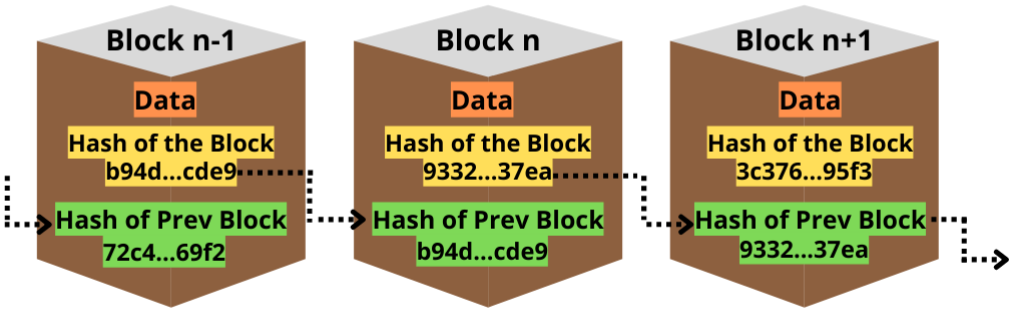


Figure 3.5: Chaining Blocks in Blockchain

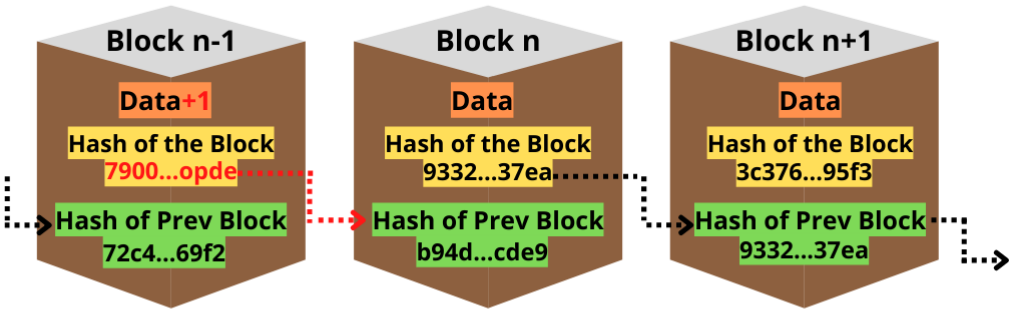


Figure 3.6: Blockchain diagram showing the effect of data alteration

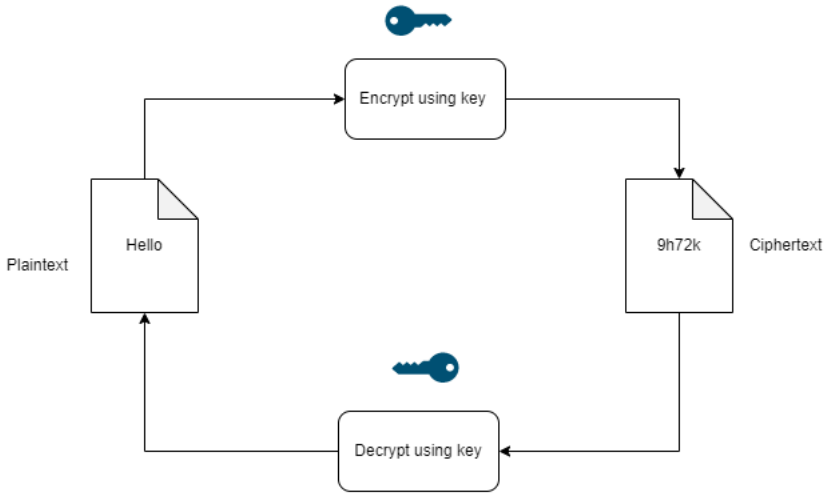


Figure 3.7: Visualization of encryption-decryption process

3.5 Smart Contracts

Smart contracts are programs that run automatically when the pre-established terms are met. These terms are written into the code in "if.../then..." form and when these conditions are true the smart contract program begins execution to produce predictable outcomes. These work are similar to legal, real-life contracts that record the agreement of the parties transferring/exchanging goods. Each contract statement's execution is recorded as an immutable transaction in the Blockchain. Smart contracts ensure that proper access control and contract requisites are in place. Developers, in particular, can grant access permission to each function in the contract. There are four consecutive phases of Smart Contracts [24] -

1. 1. Creation - Lawyers or counselors will assist the parties with preparation of the initial contract and if the contracts are not as formal or legally binding, the parties setting up the system come to an agreement as to what the terms should be. Software programmers turn the natural language agreement into a smart contract encoded in computer languages such as declarative languages and logic-based rule languages. The process of converting smart contracts is identical to that of software development in that it comprises design, validation and implementation.

2. 2. Deployment - Saved contracts cannot be modified due to the immutable nature of Blockchains. Small change requires the making of a brand new agreement. When smart contracts are implemented, it can be accessed by all parties on blockchains. Parties involved under the smart contract are sealed by freezing the associated digital wallet by using Blockchain & Digital Assets.

3. 3. Execution - The contractual provisions have been monitored and evaluated following the deployment of smart contracts. The contractual procedures or functions are automatically carried out when the contractual conditions are met. The corresponding statement is executed automatically, when a condition is true, resulting in a transaction being executed and validated by miners in the

Blockchains as it is important to comprehend that smart contracts are nothing but a program made of declarative statements with logical connections.

4. 4. Completion - All involved parties' states are updated after concluding smart contracts. The transactions occurring during the execution of smart contracts and updated states are stored in Blockchains. In the meantime, parties finish transferring digital assets from one to another. By then, digital assets of the parties involved have been unlocked and the entire life cycle is completed.

Smart Contracts is such a versatile accessory that comes with many advantages, making it easy for us to use it [28].

1. Security - Because the distributed ledger is impenetrable and impervious to variations.

2. Disintermediation - Allows parties to enter into agreements with less reliance on middlemen.

3. Near real-time execution - Once the necessary criteria are met, it occurs almost simultaneously for all parties, across participating computers.

4. Transparency - Because the logic and information in the contract are visible to all participants in the Blockchain network, it creates a trusting environment.

Despite its numerous benefits, it is not without drawbacks like-

1. 1. Creation Challenges -The critical part in putting smart contracts into action is "Contract Creation". Parties involved must write their own contracts and then deploy them across multiple Blockchain platforms. Because Blockchains are essentially immutable, Blockchain-based smart contracts cannot be changed after they have been deployed.

2. 2. Readability - The majority of smart contract projects are written in languages like Solidity, Kotlin, Go etc. After that, the source code will be compiled and executed. As a result, programs have different types of codes at different times. Making programs understandable within every type remains a major challenge.

We can achieve this through encryption, and we can encrypt the videos and store the hash in the Blockchain. Furthermore, all cameras will be assigned a smart contract so that we can identify the camera's origin, and the encryption/decryption key must be signed by the Smart Contract, an overall system design.

3.6 IPFS

IPFS is a file sharing platform that is decentralized and distinguishes files by their content rather than their location. It is dependent on a distributed hash table (DHT) that keeps track of the locations of the files and node linkage information [11]. It works by breaking the data into small blocks which are identified uniquely by a cryptographic hash.

Every block of data is given a unique identifier - the hash of the data - and is stored in the DHT as a key-value pair where the key is the identifier and the value is the data [1]. When a file needs to be retrieved, it looks the file up through the hash and communicates with a node that is near the key. That node then replies with the value if it contains it, else sends back information of the node that is even nearer.

3.7 Merkle Tree

Merkle trees are built bottom-to-up by hashing blocks of data in pairs (if binary trees; else can have more than two child nodes) until a single root hash is found. A leaf node is the data block and a non-leaf node is the hash of its child nodes. This is an efficient method of checking the integrity of transactions in a Blockchain. Merkle trees greatly reduce the amount of data that need to be transferred through a network and make Blockchain nodes light-weight.

Chapter 4

Proposed Model

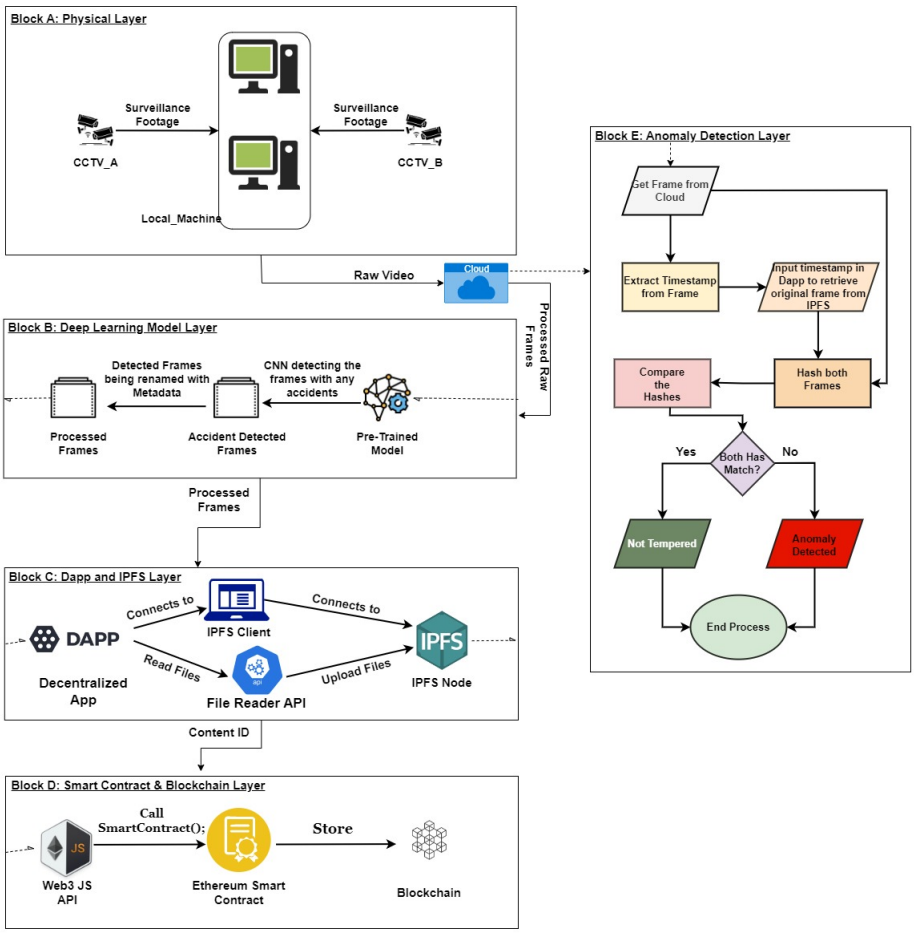


Figure 4.1: Dataflow within the Architecture

Our thesis paper revolves around the research of a holistic system to detect anomalies in videos. Thus the model includes data extraction, processing, communication and storing. In this section, we have talked about the architecture at length and how each module is designed. To make it easier to understand, we have divided this section into three subsections.
Firstly, in the physical layer the video footage from different CCTV cameras is being accumulated in the local server. The raw video is then sent to the cloud server and a data processing algorithm is extracting frames. Next, the Data Processing section

talks about how the live video feedback is extracted and processed to make it suitable for storing in Blockchain. In the Deep Learning Model layer the raw extracted frames are fed to a pre-trained CNN model that detects the accident occurrence frames and renames those frames with metadata. Thirdly, the IPFS section explains how this processed data is stored on IPFS. Here the decentralized application is using the IPFS client to connect to an IPFS node and using the Filereader API to upload files to an IPFS node. Moving on, the Blockchain and Smart Contract section highlights how the data is stored on Blockchain, explains the interaction between IPFS and Blockchain and how they are connected. With the aid of web3 JS API the smart contract is being called to store the content ID of frames to the Blockchain. Lastly, on the anomaly detection part we deduce, whether a suspicious image frame is doctored or not. Based on a certain timestamp, a frame from the local server and from Blockchain is retrieved. After that their hashes are compared to detect any ingenuity in the content of the questioned frame.

4.1 Data Processing

Our aim is to store video frames from CCTV cameras on Blockchain to ensure the original footage is untempered which can also be used to crosscheck the footage in question. Now, as we have discussed previously we stored only selected frames from the live video feed. This would reduce the storage complexity immensely, keeping all the necessary video frames in the Blockchain.
The data processing part is further divided into three modules, firstly processing the live raw video feed from the CCTV camera, secondly filtering the frames with pre-trained Deep Learning model, lastly ensuring a smooth data flow throughout the architecture.
As we know video is composed of many continuous still images known as frames. The video quality, or how smoothly each transition in the video is taking place is measured by frame rate (fps). This frame rate depends on the camera used, so it is very important to use a camera with better frame rate. Thus, even if the industry standard is 15 fps cameras, we used 20fps cameras for detecting each action clearly. As seen in the figure 4.2 every CCTV camera is connected with a local machine where an algorithm is running for processing the live video feed. This algorithm captures the live video from the camera connected to the machine. It captures a frame each second and before saving the frame the algorithm puts a timestamp on them. The frames are then renamed containing the time and the camera number of its capture, which plays an important part in the anomaly detection in the latter part. The algorithm has a connection with a cloud server where the frames are then saved.
On the cloud server the processed frames from every camera are stored and are fed to a Deep Learning model. We have used the CNN model to detect the frames of car accidents, which we want to store in Blockchain. We have used ‘car accident’ as an example of activities that authority wants to monitor. So, our model takes ‘car accident’ as a criteria and only works on it. The pretrained CNN model is kept on the cloud server. It takes the continuous frames of each camera as input and predicts whether the frame is of car accident or not. The model outputs frames classified into two classes labeled ‘accident’ and ‘not accident’.
Video frames have a lot of parameters which is very expensive for fully connected

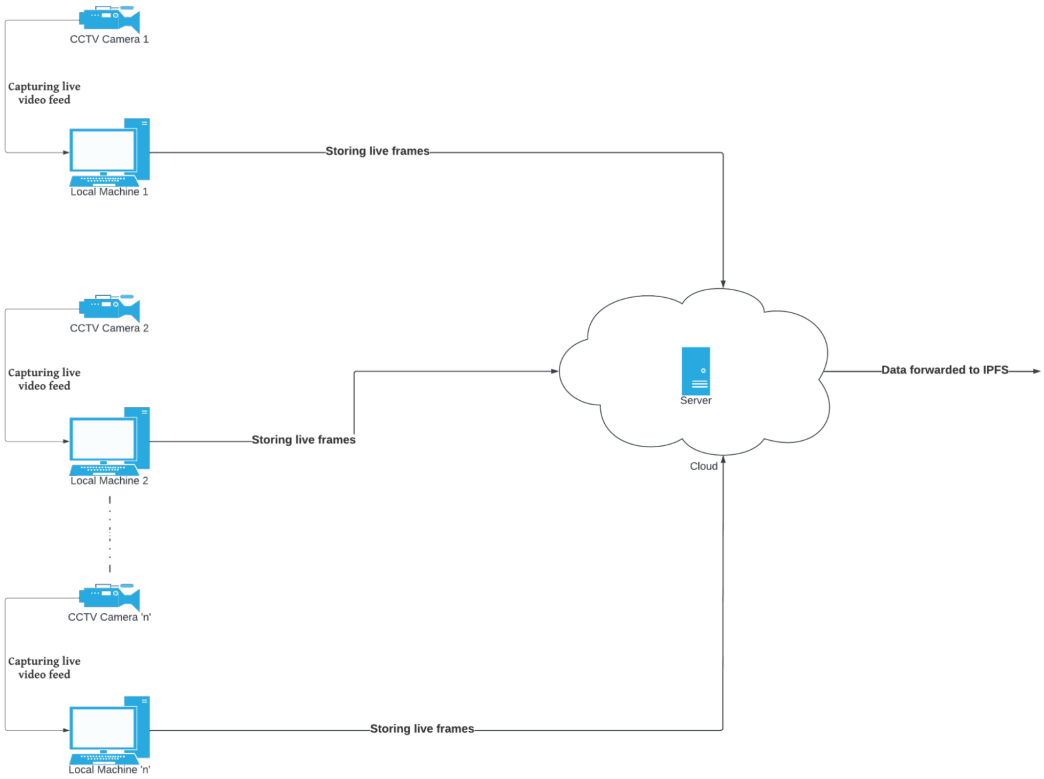


Figure 4.2: Dataflow of frame processing

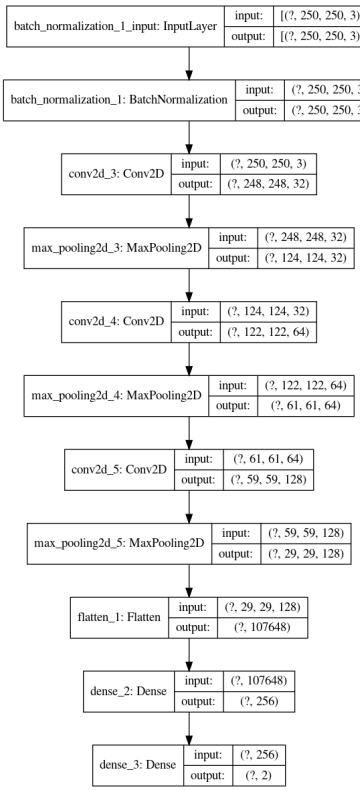


Figure 4.3: Dimension of frame at each stage of CNN model

neural networks to train on them. This is where CNN comes, it uses shared weight edges to reduce the number of edges between neurons. CNN is composed of convolutional, max pooling and flattening layers before the fully connected hidden layers and output layer. Convolution layer filters a smaller portion of a frame from the previous layer. It performs a dot matrix with the filter and a small portion of the frame we want to predict on. Max pooling layer grids the output matrix after convolution layer and finds the maximum from each grid. This halves the dimension of frames reducing parameters needed to characterize the frame. Lastly, the flattening layer converts the stacked up reduced output which we got after max pooling layer into a layer of individual neurons. The output of the flattening layer is then fed to fully connected hidden layers which perform computations to classify the frame and give output to the output layer.
The frames labeled as 'accident' class are then saved in the cloud server. These frames along with some metadata of the frame like the timestamp, camera number are forwarded to IPFS. There is a connection between the cloud server and the IPFS through which the processed frames are automatically uploaded to the IPFS.

4.2 Storing frames in IPFS

We propose to make sure the transparency of data in our recommended architecture. In our case the data is the image itself, the metadata of the image etc. We decided to use IPFS as our storage facility because IPFS provides a distributed system for storing and accessing files, websites, applications, and data. IPFS is built on three main attributes: decentralization, content addressing and participation. The decentralized property ensures that the contents are supported by a resilient internet. The contents on IPFS are not situated in a central webserver, so the risk of hacking or getting attacked is eliminated as contents are distributed across IPFS nodes. Secondly, decentralization makes it harder to censor content. Because files on IPFS can come from many places, it's harder for anyone whether they are states, corporations, or someone else to block content. The second property is content addressing. Normally content on the internet is identified by where it's located – what computer it's on and where on that computer's hard drive it is. Whereas in IPFS addresses a file by what's in it, or by its content. The content identifier is a cryptographic hash of the content of the file itself. The hash is unique to the content that it came from. Here to emphasize content on IPFS the contents can point to many different types of data like the metadata of a single piece of a file, a whole file, a directory, a whole website, or any other kind of content. Lastly, IPFS is based on the ideas of possession and participation, where many people possess each other's files and participate in making them available. Every computer participating in IPFS is a node. By connecting one node to another node or peer and downloading, storing resources and keeping those resources available for other nodes, IPFS conforms to possession and participation.
Here, our data is processed frames that are images. To store image data on IPFS for convenience we convert the image into a readable array of bytes. This is done by the FileReader API. The FileReader converts the content of the file that is an individual image to an ArrayBuffer representing the file's data. Now to process the data in IPFS the first step is to connect to an IPFS node. Nodes are an IPFS program that runs on local computers to store files and connect to the IPFS network. So,

after initializing our local IPFS node in our local computer the contents are ready to be inserted on the IPFS. After the contents are stored, IPFS returns a content identifier. In our case a single content identifier maps to a single image. Now let's look at the anatomy of this content identifier that is returned by IPFS: CIDS are based on the contents cryptographic hash. Hashes are a one way data encrypting function and any change Inside the content will produce a different hash of that file. There are two versions of CID and those CIDS can take different forms based on encoding versions. We are incorporating the v0 in our project. The following is an example of a CID v0 format:

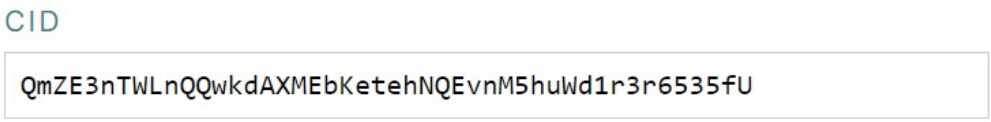


Figure 4.4: Received Content ID From IPFS

The CID v0 is 46 characters long starting with fixed prefix "Qm". And by default, IPFS uses SHA-256 hashing algorithm to encrypt the content. Then the hash is encoded in base58btc which is a multibase encoding system and results in the full format. After this CID is perceived by a node, each node follows a data structure called DAGs (Directed Acyclic Graphs) to split a content. Each node arranges its content to build a Merkle DAG representation. IPFS splits DAG representation into blocks. Splitting it into blocks means that different parts of the file can come from different sources and be authenticated quickly. Here is a Representation of a content in IPFS nodes arranged in DAG structure:

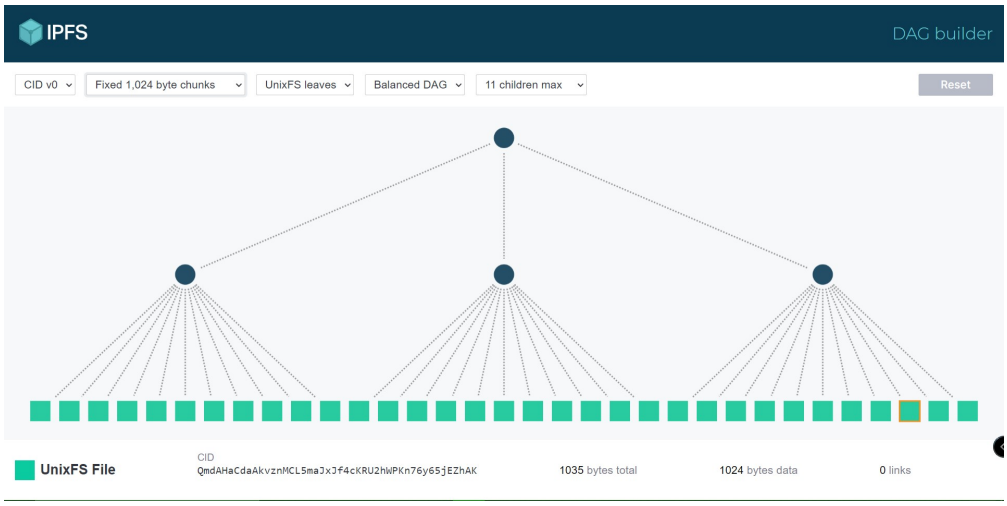


Figure 4.5: DAG Structure of Content in IPFS

This is a Merkle DAG representation of a file and the DAG is balanced because each node is linked to the same amount of Blocks . Here a single file is chunked according to a fixed block size, the block size here is represented in bytes. Every block containing the chunked data has their own CID and a set of blocks is the children of a parent node. And every node also has a CID. So the actual content ID of a file that is returned from a node is the CID of the top node.

4.3 Arranging CID's in Merkle Trees

After IPFS has returned the CID of frames a method needs to be incorporated to significantly reduce the memory needed to verify that data has maintained its integrity and hasn't been altered before storing it to the blockchain as well as make space and computationally efficient for strong in Blockchain. We have proposed to use the Data structure called Merkle Tree. Merkle trees are efficients in quickly determining whether a single data is a part of a bigger set of data as if the data is maintaining integrity or not. They are frequently employed in peer-to-peer networks, where effective proofs of this kind will help the network become more scalable [29]. So in the binary Merkle Tree will consist of multiple hashes or CID's and they will be reduced to a single hash.

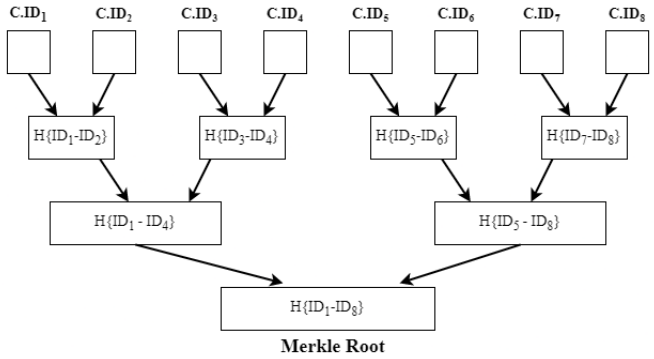


Figure 4.6: Merkle Tree Representation of Content IDs

In 4.6 there are eight leaf hashes. From there pairs of hashes are taken as input to hash again and result in a single hash corresponding to a pair. This is how we get the hash of the root node. Arranging the data with this tree structure will give the advantage of verifying a single data computably efficiently.

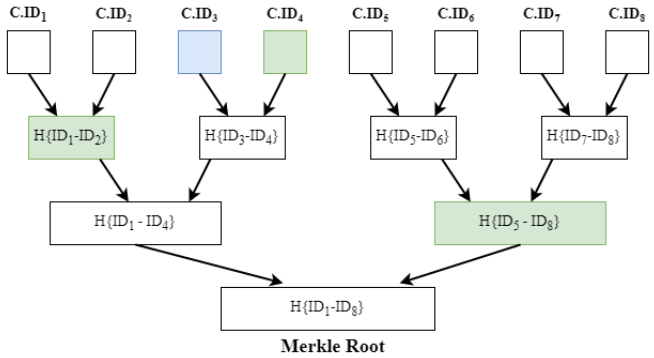


Figure 4.7: Verification of Data Integrity

For example in 4.7 if a hash is required to be verified then we only need less than half the nodes to validate that, not all data is required. This increase of the proof element is logarithmic in time. So, the higher the input or the nodes (n) are, the proof element will be log(n). Suppose if we need to check the integrity of C.ID3, we would only need to know the Hash of C.ID4, Hash of C.ID1 & C.ID2 combined and lastly the combined hash of C.ID5 to C.ID8. Then, it would easily do the validation.

Moreover this data structure will also give an advantage of efficiently using space in the blockchain as well as good for scalability. Instead of storing a single CID per block here a large number of CID's is being scaled down to a single CID then that is being secured. Therefore there could be thousands of images being reduced to one single hash and that hash is going to be stored in the block. Ultimately it will have an impact on the cost and scalability of storing in the blockchain. Here is a graphical analysis of how Merkle tree is better than different complexity:

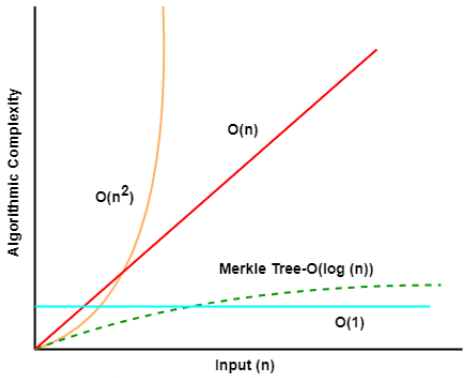


Figure 4.8: Comparison between Time Complexities

4.4 Data Flow into Blockchain via IPFS

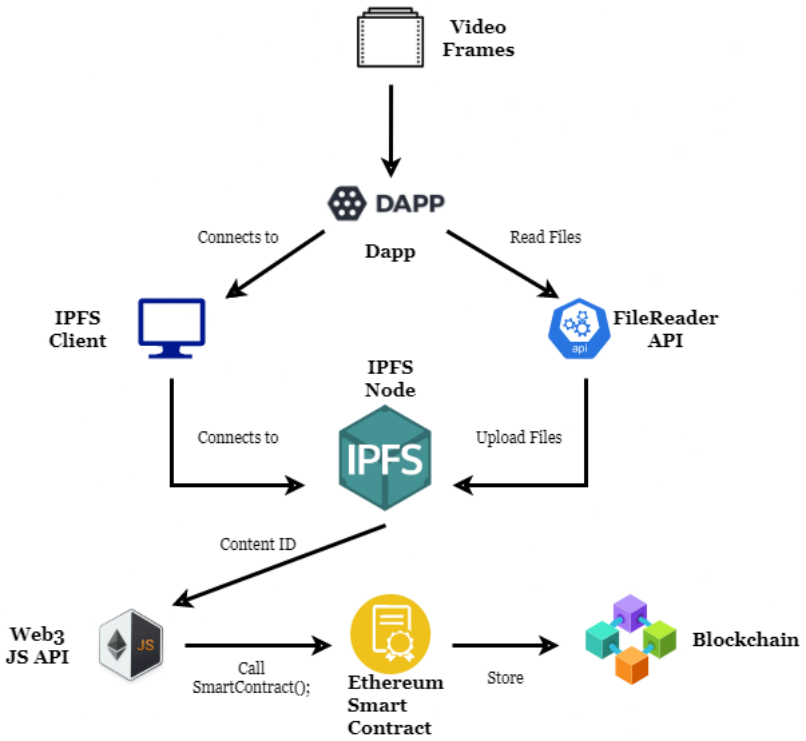


Figure 4.9: Dataflow From Dapp via IPFS to Blockchain

After the frames are stored in IPFS now the CID of those frames are to be secured in the Blockchain. It is not so convenient just to store the image CID. We need another

parameter to relate with the CID, so that those can be retrieved for later purpose. To incorporate that we took the advantage of keeping the exact date and time data as a pointer to that specific time image frame. When each frame's CID is uploaded to a block, the timestamp related to that frame is also uploaded. Therefore, each frame's CID is mapped to the time it was created. Rather than storing the exact timestamp as a string , for convenience the timestamp is converted to an epoch Unix timestamp which decodes the time to milliseconds.

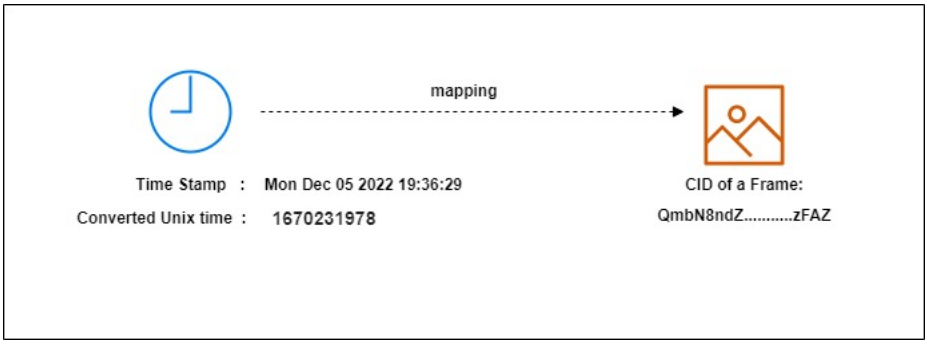


Figure 4.10: Mapping Each Frame to its CID

Now that the data pair (timestamp, CID) is ready to be stored in the Blockchain here comes the intermediary to do that is smart contract. The smart contract has a specific function that receives two parameters: one is the time an integer value and second a string of content ID. By calling that function the Content ID values of the intended frames are securely stored in the Blockchain.

4.5 Mining

Mining the data in the Blockchain is one of the most crucial parts for successful verification of data in the block. In public Blockchain the mining process is much more complex as there are thousands of nodes competing with each other to validate a transaction in a block. But here as we are using a consortium Blockchain, the process of mining is much more simpler and automated. After the smart contract function is called, a certain amount of transactions is queued. In this case ten transactions are queued to be added in the Blockchain. Then in order to sign these transactions a private key is needed. We have provided a private key to an automated component of the system that will sign these transactions. The Blockchain mining system will also sort out invalid transactions and reject them. After ten transactions have been validated the transaction data will be added inside a block, and that block will be added to the Blockchain. Therefore a new block will be mined.

4.6 Application based system for judicial officials

In the event a footage is required as evidence and needs to be reviewed in court, an application based system will be used by judicial officials to perform a real-time analysis to check the integrity of the footage. The officials can perform this by retrieving the hash from the Blockchain through the timestamp and the application will compare the hash of the frames at that timestamp. Additionally, the image

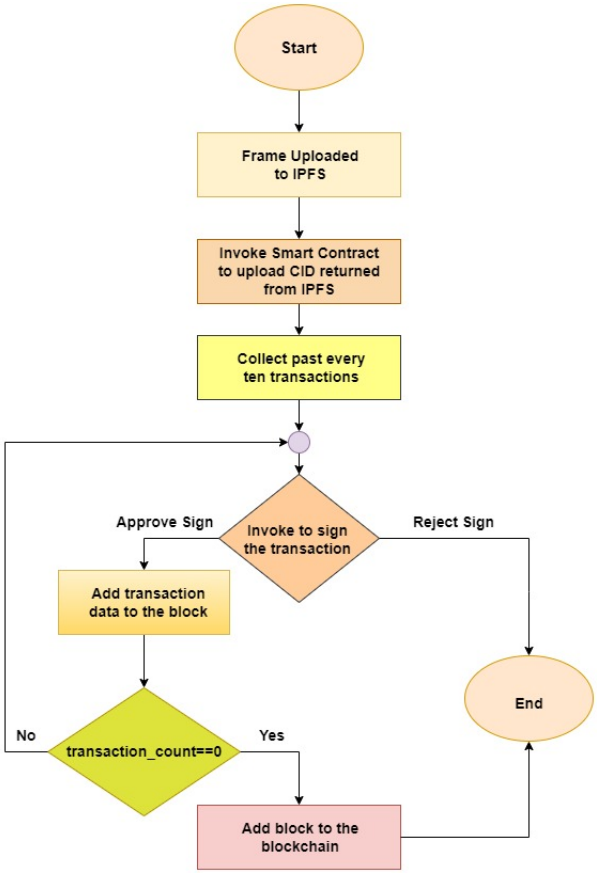


Figure 4.11: Mining Process of the Blocks

retrieved from the IPFS can serve as evidence if it was found that the footage that was tested had been tampered with. The application is designed to be usable without having any knowledge of Blockchain, hashing, IPFS, or other technicalities.

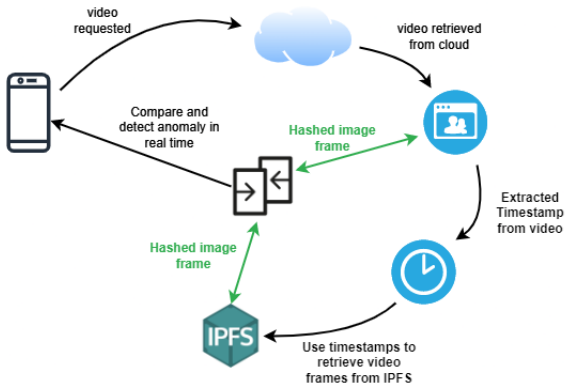


Figure 4.12: Mobile Based Application for Judiciary Use

Chapter 5

Implementation

5.1 Video Processing

On each of the machines connected to the cameras there is an algorithm running to capture the live video feedback. We have used the OpenCV library in python's environment to process the video file. OpenCV is a widely used open-source library for computer vision, machine learning, and image processing. It is essential in modern systems for real-time operations. With OpenCV, it is possible to analyze images and videos to detect objects, faces, and even handwriting. Initially, we capture video by running the camera. Each frame is extracted and a timestamp is put on them. These timestamps along with the frame number they belong to are stored in a list. Next, the frames are named in the convention 'CameraNo_FrameNo_date_time.jpg'. Lastly, the frames are saved in the cloud.

```
# i variable is used to give unique name to images
i = 1

# List to contain all the timestamp
timelist = list()

# Open the camera
video = captureVideo()

Loop:
        # will extract and return the frame
        img = video.extractFrame()

        # Put current timestamp on each frame
        time = timestamp()
        putText(img, times)

        #saving file
        filename = 'Camera1_Frame'+str(i)+"_"+time+'.jpg'
        save(filename, img)

        #Add timestamps with frame number to the list
        timelist.append('Frame_'+str(i)+' '+time)

        i = i+1

#Storing the timestamps with frame number in a text file
open('times.txt'):
    for each timestamp in timelist:
        write(timestamp)
save(times.txt)
```

Figure 5.1: Algorithm for Pre-processing Video Frames

In the cloud also lies a pre-trained CNN machine learning model and an algorithm to further process the frames. The CNN model filters the relevant frames, in our case the frames of car accidents. The model is trained on the dataset [17] containing

frames captured from youtube videos involving accidents (Figure 5.3). The dataset is divided into three splits - training, testing and validation data, and each of these splits contain frames under two classes - 'accident' and 'non-accident' (Figure 5.2).

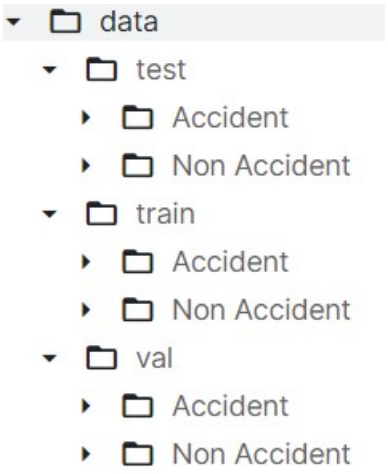


Figure 5.2: Directory of Dataset

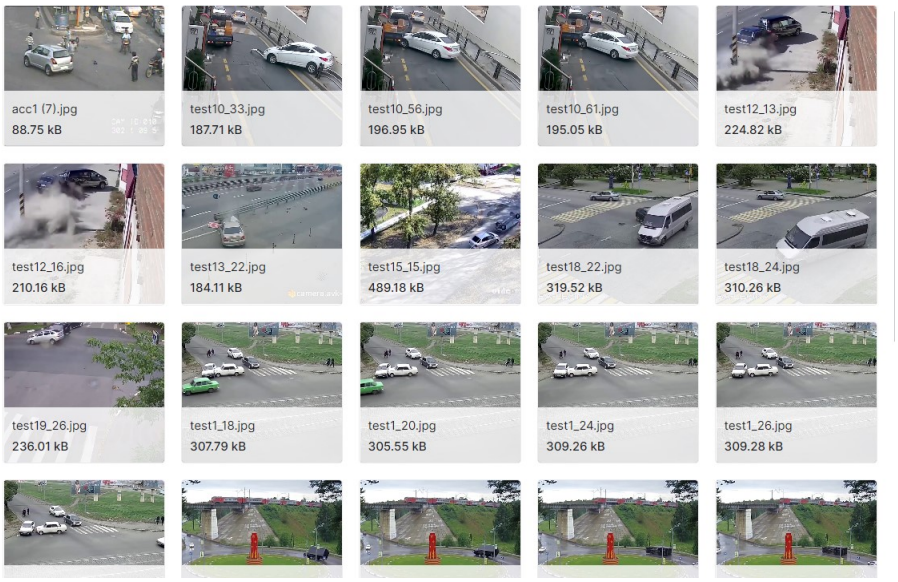


Figure 5.3: Snippet of Dataset

It is trained using tensorflow and keras libraries in a python environment. The input frame is defined with image height and width be 250. First, the CNN is defined with consecutive convolution and max pooling layers with 'relu activation' (Figure 4.3). Next, it is trained and tested on the above dataset. The model is then stored on the cloud. The algorithm on the cloud uses this model to predict the frames of car accidents from the dataset. Then, the selected frames are renamed in the convention 'CameraNo_savedImageNo_date_time.jpeg' and are stored on the cloud, ready to be uploaded on the IPFS.

5.2 Configuring Implementation Model

For the implementation in Blockchain and of our proposed model, we have used a local Blockchain platform named Ganache. Ganache is an Ethereum based Dapp building platform that enables develop, deploy, and test Dapps in a safe and deterministic environment. Next, we have used Truffle which is a development environment, testing framework and asset pipeline for Blockchains using the Ethereum Virtual Machine (EVM). Truffle provides built-in smart contract compilation, linking, deployment to Blockchain, contract testing before deployment etc. Then we have used the MetaMask cryptocurrency wallet to interact with the Ethereum Blockchain. MetaMask wallet provides to manage Ethereum accounts and keys, to integrate smart contract functionalities to the Dapp, signing transactions to the Blockchain etc. Lastly for writing smart contracts which actually talks with the Blockchain we used Solidity. Solidity is an object-oriented, high-level, statically-typed programming language for implementing smart contracts on Ethereum Blockchain. Solidity is designed to run on the Ethereum Virtual Machine (EVM), which is a runtime environment that works like a virtual computer to run software programs. The programs that the EVM runs are called smart contracts, and smart contracts are written in Solidity.
Firstly, a specific command is run to create an empty Ethereum project. As we are using the truffle framework, we run the following command in our project directory:

```
PS C:\Users\Mahi\OneDrive\Desktop\NODE js\BChain\StoreImgHashToBC> truffle init

√ Preparing to download
√ Downloading
√ Cleaning up temporary files
√ Setting up box

Unbox successful. Sweet!

Commands:

  Compile:        truffle compile
  Migrate:        truffle migrate
  Test contracts: truffle test
```

Figure 5.4: Initializing an Empty Ethereum Project

After running this command an environment for running our decentralized application is created in the directory. Now we mine the genesis block in our local Blockchain and as well as deploy our smart contract. Ganache local Blockchain comes with ten Ethereum accounts and with their private keys. Each account has been given 100 ether to perform the operations such as handling gas fees. Apparently these ethers are not real. Ganache will automatically use one of those Ethereum accounts to mine the genesis blocks. So, we run this specific truffle command for mining the genesis block and to deploy our smart contract :

```
PS C:\Users\Mahi\OneDrive\Desktop\NODE js\BChain\StoreImgHashToBC> truffle migrate --reset
Compiling .\contracts\Migrations.sol...
Writing artifacts to .\build\contracts
```

Figure 5.5: Command to Mine

```
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0xf773c57acbd0ca9767d1721e52e2d4d07090b1e9f18bec96537db0c3bb20f1b8
   > Blocks: 0            Seconds: 0
   > contract address:    0x95Bb5e33A215bD2cc4985D82D0a50D49945a5542
   > account:             0x3dA03725d339B8cBC4fA7AF6139e9880185AA33C
   > balance:             99.99640562
   > gas used:            179719
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00359438 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00359438 ETH


2_deploy_contracts.js
=====================

   Deploying 'storeHash'
   ---------------------
   > Total cost:          0.00548814 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.00908252 ETH
```

Figure 5.6: Mining Genesis Block and Deploying Smart Contract

So, after running this command Block 0 has been mined. The mining process is done by an Ethereum account and requires some gas cost to deploy the contracts. As well as two smart contracts have also been deployed to the Blockchain.

5.3 Storing in Blockchain

To store the hash in Blockchain there are two phases. Firstly, on the client side the application will connect to an IPFS node and do required steps to successfully submit the frame in IPFS. Then in the second part the application will interact with the smart contract to store the returned hashes of those frames from IPFS. So, in Figure 5.7 pseudocode, the first step is to create a connection to IPFS by connecting to a node. Then the processFile() function processes the image file by extracting the file name along with the timestamp, as well as converts the image to readable buffer data for the convenience of storing the content of file. Next, the handleSubmit() function receives the processed file and submits it to IPFS. This function also returns the hash. Lastly, in the first phase the storeToBlockchain() function is called with two arguments to initiate the process of reserving data inside the Blockchain.
In the second phase the first step is to connect the application with our local Blockchain network by fetching network info, Ethereum accounts address and connecting with the smart contract. The main medium of the data flow to the Blockchain is the smart contract. Here the map data structure is used so it will be suitable for storing the data pair (timestamp,hash). The map data structure associates each image hash with its timestamp. Time is converted to unix timestamp and repre-

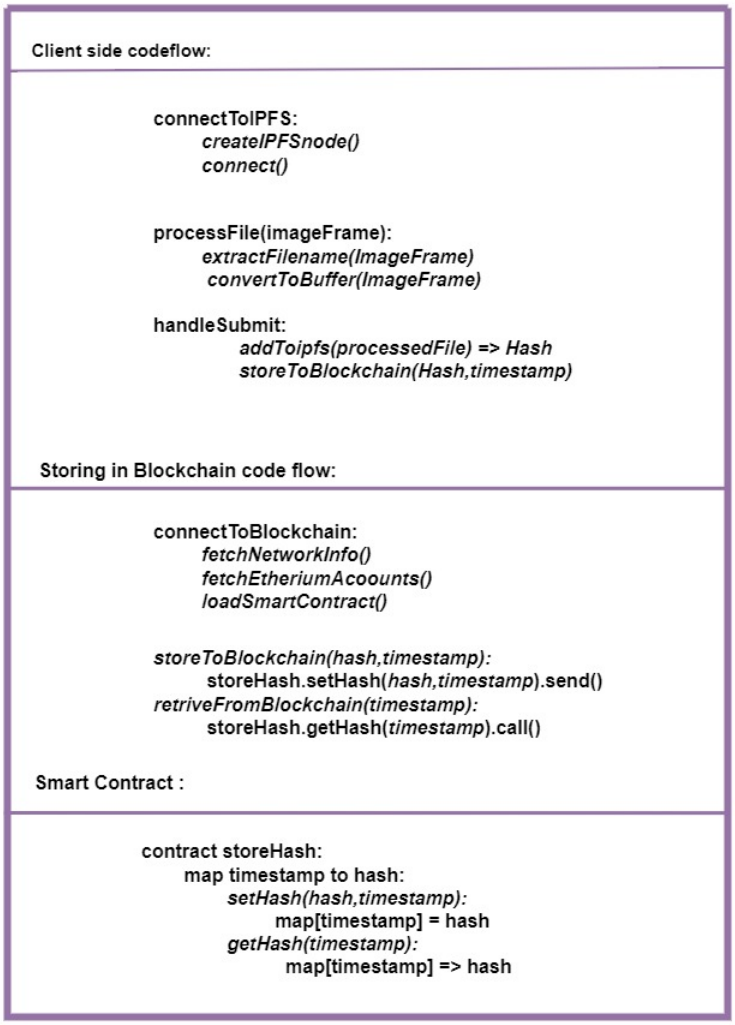


Figure 5.7: Storing Data to the Blockchain

sented by an unsigned integer and the hash is a string. Next, storeToBlockchain() function calls the smart contracts sethash() method to store the data in a block. Moreover, the send() method defines that something new is going to be added in the Blockchain and it will cost some gas fee. So this concludes the data storing flow. Finally, the last function of the application is the hash retrieving function retrive-FromBlockchain() which takes a timestamp as an argument and sends a query to the Blockchain to retrieve that specific hash.

So, in the implementation UI firstly, there is an Account attribute. This attribute's value is an ethereum address that is required to sign the data transaction to the Blockchain. As we mentioned earlier this account's private key will be provided to an automated system of the application that will handle signing the transactions. Next, the frames are automatically selected and submitted to the IPFS node After the images are successfully stored in the IPFS node and the Content ID of the images are returned. After this the application will trigger the process of storing the CID in the Blockchain. Consequently, creating a new block and putting the transaction data in the block. In order to validate that new block a signing is required. A Metamask window will pop up to confirm the transaction to the Blockchain.

So, in this transaction confirmation window it is showing that the setHash function

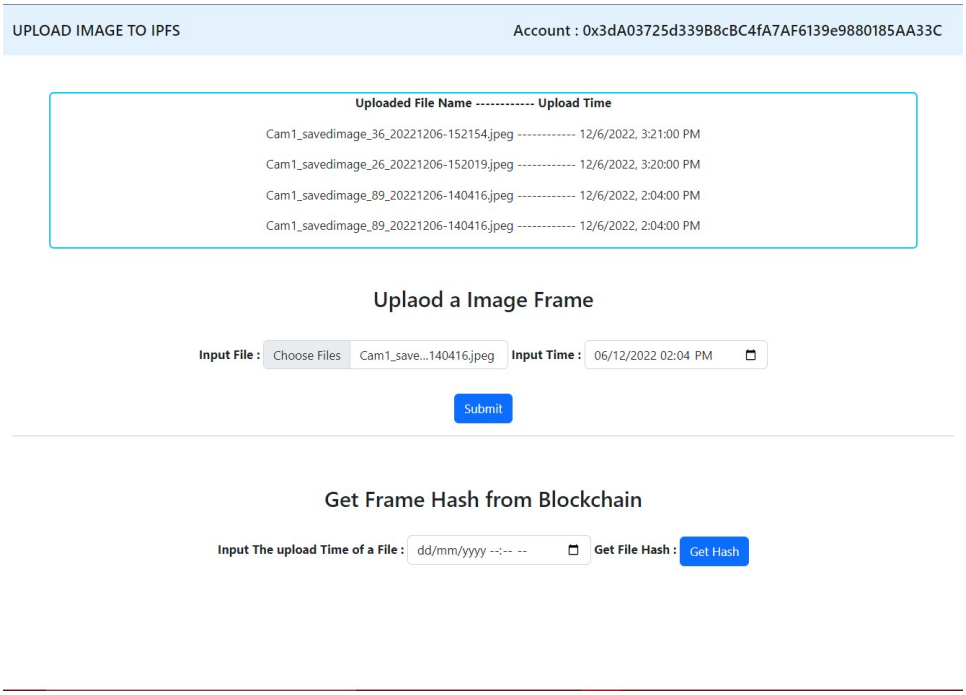


Figure 5.8: User Interface for Uploading and Retrieving Image Frames

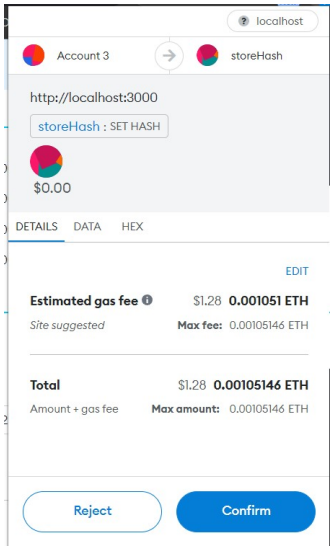


Figure 5.9: Transaction Data Authentication to Blockchain

is going to modify something in the Blockchain. Also, changing or updating the state of any data in the Blockchain costs gas fee which is also shown here. After clicking the confirm button the data is successfully stored in a block inside the Blockchain. Finally, the uploaded image frames where the file name consists of the camera number and image number are mapped to their individual timestamp is shown in the UI.

5.4 Result

The objective of this thesis paper, as discussed earlier, is to propose and implement an effective architecture for detecting anomalies in videos from security cameras. Moreover, for the implementation purpose we have worked with the security camera videos containing car accidents; here, the 'car accidents' are considered as incidents deemed to be tampered maliciously. So, in order to address this problem in this thesis paper we have proposed the idea of storing the original video frames in Blockchain from the cameras. These frames, later on, are used as reference to detect the authenticity of the frames in question.
Usually in this architecture, all the security video frames are available on the cloud for various authority personnel, whereas the frames on Blockchain will have a limited accessibility and will only be used to check the ingenuity of the frames. So when a frame is needed by authority they will download from the cloud. Next, they will send the frame to check for its authenticity. There, the timestamp of the frames will be checked and this timestamp will be used to retrieve the original frame from the Blockchain. These two frames are from the same camera at the exact same time. Lastly, these two frames then will be hashed and then the hashed will be checked to see whether the image on the centralized cloud is tampered or not. We have used the SHA-256 hashing algorithm for our system, which provides a reliable, irreversible cryptographic security.

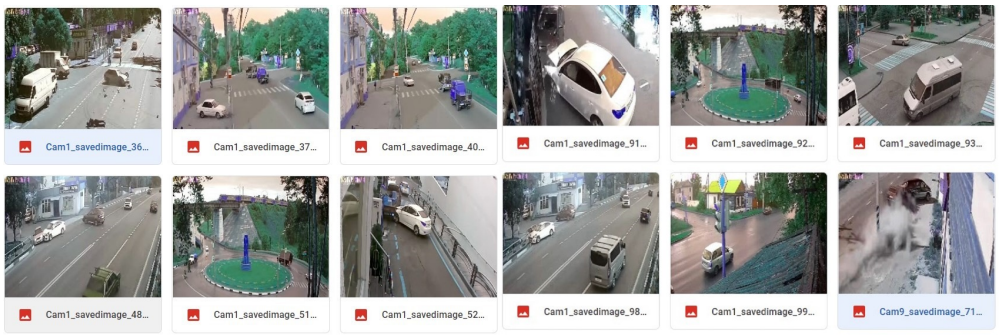


Figure 5.10: Frames Selected From the Cloud Server

To demonstrate this, we have downloaded two frames from the cloud storage, highlighted in skyblue (Figure 5.10 & 5.11). The timestamp of the frames are extracted and used to retrieve the original frames from the Blockchain. The UI for retrieving frames from the Blockchain has a search option where we can input the frame's timestamp (Figure 5.12). The UI will return a link through which we can access the frames.
In Figure 5.13, we have downloaded the original frames from the Blockchain using their timestamps. For the demonstration purpose we have intentionally tampered

Figure 5.11: Downloaded Frames from the Cloud

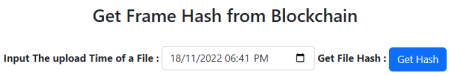


Figure 5.12: Retrieving Frames from Blockchain

with one of the frames on the cloud storage. The frame from camera 1 is tampered with and the frame from camera 9 is kept as it is.

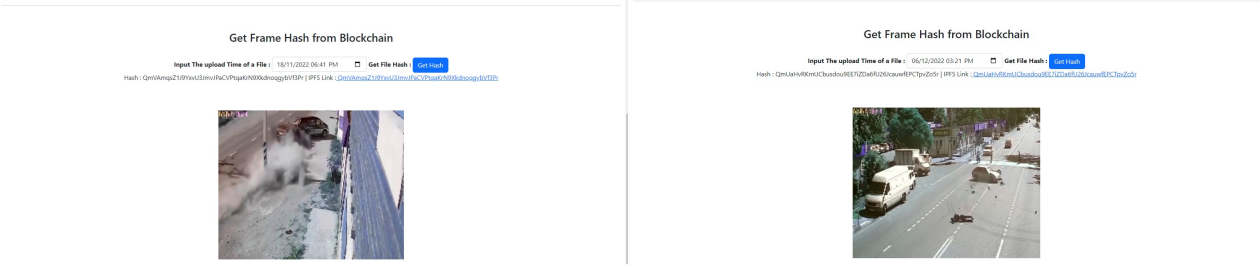


Figure 5.13: Retrieved Original Frames from Blockchain

Now, we will hash each set of the frames and compare them. We can see that the hash generated from the frames of camera 1 of exact time is different, thus we can conclude that the frame from the cloud has been tampered (Figure 5.14). But, the frames from camera 9 are the same, which indicates that the frame from the cloud has not been doctored (Figure 5.15).

5.5 Comparative Result Analysis

In our works, our goal was to come up with a new architecture which will use the latest Blockchain technology to detect authenticity in Traffic Surveillance Footage, specifically in Car Crash. Our paper proposes the architecture which is using Blockchain to store the hashed Content ID that we receive from IPFS after we upload it in an IPFS node. It also uses CNN Model to detect important frames. All of these result in having better Decentralization, Large Scaling Potentiality, better storage utilization along with deep learning model and obviously authentication.

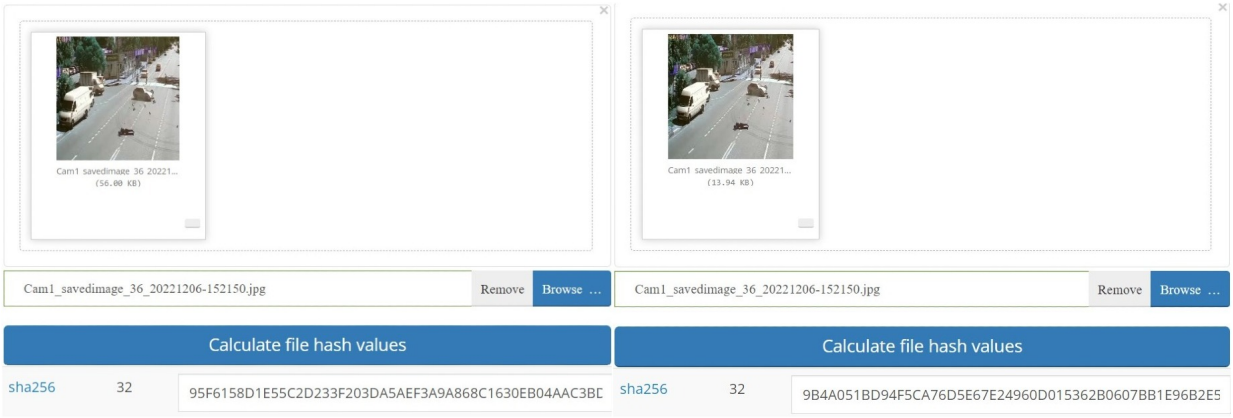


Figure 5.14: Comparing Hash Values of Camera 1 Frames

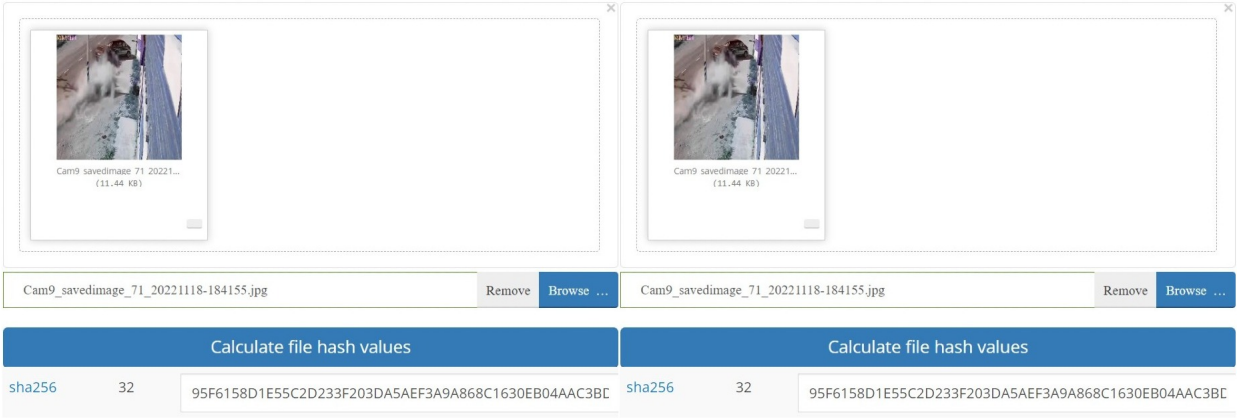


Figure 5.15: Comparing Hash Values of Camera 9 Frames

In the works that we have reviewed in Chapter 2, a major concern was that the Blockchain would grow at a very fast pace and it would become increasingly time and resource consuming to add new blocks. As we only create blocks in the event of a traffic accident, the amount of data that needs to be stored is vastly reduced. Compared to the huge quantity of data generated each second by multitudes of cameras, the data our system is dealing with is considerably less as traffic accidents do not occur at the same frequency.
We are also reducing the storage load on IPFS nodes by only storing images from the footage where the previous works store the entire videos on IPFS. This additionally reduces the bandwidth requirement of the network as smaller quantities of data are being transferred to and from IPFS and Blockchain. Overall the time required to secure the images, cost, bandwidth and storage requirements are vastly reduced with our proposed solution. Here is a tabular comparison of what we have in our model and what the other researches are potentially missing:

Researches	**Decentralization**	**Large Scale Potential**	**Storage Utilization**	**Deep Learning Model**	**Authenticity Detection**
[13]	Yes	No	No	No	Not fully
[7]	Yes	Yes	No	Yes	No
[22]	No	Yes	No	No	Yes
[21]	Yes	No	No	No	No
[15]	Yes	No	Yes	Yes	No
[19]	No	No	No	No	Yes
This Paper	Yes	Yes	Yes	Yes	Yes

Table 5.1: Comparison Analysis Between Existing Research and Ours

Chapter 6

Conclusion and Future Works

Surveillance cameras serve an important role in improving security, safety, and evidence in our society. Surveillance footage from cameras, at least the ones in an average home, are not usually protected. Anyone with access to the videos can alter them and there would be no evidence to prove that it is not the original, which is why it is important that we look into ways to protect the integrity of surveillance footage. Protecting the integrity of surveillance footage is crucial to public safety and judicial evidence. And with this research, we aim to contribute to the research on securing CCTV footage. As there are several different technologies attached with Blockchain, it is important to ensure each of them are specified properly in order to operate correctly. The thesis paper highlights how the data is processed at different phases and fed to the Blockchain so that when the videos are checked, anomalies can be detected. This paper has demonstrated the idea of comparing hash values of the questioned video frames with the hash value of untempered, original video frames in the Blockchain. Thus, this paper has proposed a complete architecture for ensuring the authenticity of any video file. We hope this research helps to ensure that authentic video footage is used as evidence which is currently done by digital forensics.

Our traffic accident detecting model has a scope for further improvement. Given that there are multiple cameras shooting the same incident from different angles, our model will not come to the same conclusion for all of the footage. Although an accident might not be recognizable from a certain angle, all camera footage capturing that view needs to be secured. A feature that is capable of such can be integrated into our system.

Even though our research paper has proposed the idea of a fully automated system. Theoretically, we have shown that each local system connected to the CCTV cameras upload raw frames to the cloud server but we have not been able to implement it yet. Moreover, the output frames of the CNN model on the cloud are supposed to be automatically uploaded to the IPFS. But currently human interference is required to upload data to the IPFS. Here a pipeline architecture could solve the problem. This would eliminate the human interaction between data transfer from source to terminal. Lastly, we could not fully implement the automated retrieval of hash value of each frame uploaded to the IPFS, which was theorized to be automatically stored on blocks of Blockchain.

We have considered the traffic surveillance system from the perspective of its security and designed our model accordingly. In conclusion, we have proposed and

implemented a functional system for securing surveillance videos and also demonstrated that integrating Blockchain, artificial intelligence, and machine learning has the ability to revolutionize the protection of surveillance footage.

Bibliography

[1] S. Götz, S. Rieche, and K. Wehrle, *Selected dht algorithms, volume 3485 of lecture notes in computer science, chapter 8*, 2005.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21 260, 2008.

[3] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

[4] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1336–1349, 2015.

[5] J. Lindman, V. K. Tuunainen, and M. Rossi, "Opportunities and risks of blockchain technologies–a research agenda," 2017.

[6] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Ieee, 2018, pp. 1545–1550.

[7] P. Gallo, S. Pongnumkul, and U. Q. Nguyen, "Blocksee: Blockchain for iot video surveillance in smart cities," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, IEEE, 2018, pp. 1–6.

[8] R. Kandaswamy, D. Furlonger, and A. Stevens, *Digital disruption profile: Blockchain's radical promise spans business and society*, 2018. [Online]. Available: https://www.gartner.com/en/doc/3855708-digital-disruption-profile-blockchains-radical-promise-spans-business-and-society.

[9] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2018.

[10] Z. Meng, T. Morizumi, S. Miyata, and H. Kinoshita, "Design scheme of copyright management system based on digital watermarking and blockchain," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, vol. 2, 2018, pp. 359–364.

[11] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for ipfs," in *2018 Ieee international conference on internet of things (iThings) and ieee green computing and communications (GreenCom) and ieee cyber, physical and social computing (CPSCom) and ieee smart data (SmartData)*, IEEE, 2018, pp. 1499–1506.

[12] Q. AM and N. Varol, "A review paper on cryptography," in *7th International Symposium on Digital Forensics and Security (ISDFS 2019)*, 2019, pp. 1–6.

[13] Y. Jeong, D. Hwang, and K.-H. Kim, "Blockchain-based management of video surveillance systems," in *2019 International Conference on Information Networking (ICOIN)*, IEEE, 2019, pp. 465–468.

[14] E. L. Piza, B. C. Welsh, D. P. Farrington, and A. L. Thomas, "Cctv surveillance for crime prevention: A 40-year systematic review with meta-analysis," *Criminology & Public Policy*, vol. 18, no. 1, pp. 135–159, 2019.

[15] R. Wang, W.-T. Tsai, J. He, C. Liu, Q. Li, and E. Deng, "A video surveillance system based on permissioned blockchains and edge computing," in *2019 IEEE international conference on big data and smart computing (BigComp)*, IEEE, 2019, pp. 1–6.

[16] V. Yatskiv, N. Yatskiv, and O. Bandrivskyi, "Proof of video integrity based on blockchain," in *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, IEEE, 2019, pp. 431–434.

[17] C. K. C, *Accident detection from cctv footage*, 2020. DOI: 10.34740/KAGGLE/DSV/1379553. [Online]. Available: https://www.kaggle.com/dsv/1379553.

[18] C. C. K. Chan, V. Kumar, S. Delaney, and M. Gochoo, "Combating deepfakes: Multi-lstm and blockchain as proof of authenticity for digital media," in *2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G)*, IEEE, 2020, pp. 55–62.

[19] S. Liu, Y. Liu, C. Feng, H. Zhao, and Y. Huang, "Blockchain privacy data protection method based on hevc video steganography," in *2020 3rd International Conference on Smart BlockChain (SmartBlock)*, IEEE, 2020, pp. 1–6.

[20] M. M. Milovanova, T. S. Markova, V. Mushrub, M. E. Ordynskaya, and J. V. Plaksa, "Business education: Training in the use of blockchain technology for business development," *Revista Inclusiones*, pp. 408–420, 2020.

[21] H. Prajval and S. Sandhya, "Block chain based efficient management of iot smart video surveillance system.," *International Journal of Advanced Research in Computer Science*, vol. 11, no. 3, 2020.

[22] R. Uda, "Data protection method with blockchain against fabrication of video by surveillance cameras," in *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology*, 2020, pp. 29–33.

[23] H. Zhao, Y. Liu, Y. Wang, and Y. Huang, "Hiding data into blockchain-based digital video for security protection," in *2020 3rd International Conference on Smart BlockChain (SmartBlock)*, IEEE, 2020, pp. 23–28.

[24] Z. Zheng, S. Xie, H.-N. Dai, *et al.*, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[25] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T. R. Gadekallu, and N. N. Xiong, "A secured distributed detection system based on ipfs and blockchain for industrial image and video data security," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 128–143, 2021.

[26] C. Parizo, *What are the 4 different types of blockchain technology?* May 2021. [Online]. Available: https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology.

[27] P. Paul, P. Aithal, and R. Saavedra, "Blockchain technology and its types—a short review," *International Journal of Applied Science and Engineering (IJASE)*, vol. 9, no. 2, pp. 189–200, 2021.

[28] M. Kukkuru, *Smart contracts: Introducing a transparent way to do business*, 2022. [Online]. Available: https://www.infosys.com/insights/digital-future/smart-contracts.html?fbclid=IwAR2WWTmUsRBteKrHQPY6pEjCyMe-Iltspiw7MX9GumVau6nW3WjwkrX0-A4..

[29] A. Luken, *Merkle trees in blockchains*, Nov. 2022. [Online]. Available: https://university.alchemy.com/course/ethereum/md/merkle-trees.

Smart Contract for Storing Data to and Retrieving Data from Blockchain

pragma solidity >= 0.4.22 < 0.9.0;
contract storeHash mapping(uint=> string) public hashes;
function setHash(uint time, string memory hash)public hashes[time]=hash; function get(uint time) public view returns(string memory) return hashes[time];