

Article

Securing Olive Tree Data: Blockchain and InterPlanetary File System Integration for Unmanned Aerial Vehicles Operations

Jorge Cabañas¹ and Jesús Rodríguez-Molina^{2,*} ¹ Department of Audiovisual Engineering and Applications, Universidad Politécnica de Madrid, 28040 Madrid, Spain; jorge.cperetz@alumnos.upm.es² Department of Telematic Engineering and Electronics, Universidad Politécnica de Madrid, 28040 Madrid, Spain

* Correspondence: jesus.rodriquezm@upm.es

Abstract

The work presented in this document integrates blockchain, Unmanned Aerial Vehicles (UAVs) and Interplanetary File System (IPFS) to improve collection, storage and accessibility of aerial imagery transfer, when these technologies are applied to olive oil production. By using blockchain properties, we aim to provide a renewed perspective on aerial data transmission, ensuring security while optimizing operational efficiency. This manuscript describes the development of a transmission platform using blockchain to log each image captured by the UAV. It also aims to improve data distribution for applications like environmental monitoring and emergency response. This document outlines specific technological specifications, operational details, and performance requirements, emphasizing a structured approach supported by resources like the ARDrone 2.0 from Parrot, a Java-based blockchain implementation and an IPFS deployment. Each of these technologies are combined in an innovative manner so that they create a framework with enhanced security based on decentralization, redundancy and openness.

Keywords: Unmanned Aerial Vehicle (UAV); blockchain; InterPlanetary File System (IPFS)

Academic Editor: Sunan Huang

Received: 1 August 2025

Revised: 20 October 2025

Accepted: 29 October 2025

Published: 5 November 2025

Citation: Cabañas, J.; Rodríguez-Molina, J. Securing Olive Tree Data: Blockchain and InterPlanetary File System Integration for Unmanned Aerial Vehicles Operations. *Robotics* **2025**, *14*, 163. <https://doi.org/10.3390/robotics14110163>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Usage of technology for optimization in agricultural outputs is becoming more popular all the time, as it has been proven that investment in technology for this application domain is extremely useful [1]. The work presented here focuses on the integration in the usage of Unmanned Aerial Vehicles (UAVs) in olive oil production combined with distributed software technologies like blockchain [2] and Interplanetary File System (IPFS, [3]), to evaluate in detail how their integration and usage results in sharing information collected from the olive trees with enhanced security, redundancy and accountability.

1.1. Current Usage of UAVs in the Olive Oil Industry

Unmanned Aerial Vehicles (UAVs), or drones, are increasingly used in the olive oil industry for various applications. While their most typical usefulness lies on enabling farmers to monitor olive groves with high-resolution aerial imagery, the multimedia information collected by them can be used for several purposes. UAVs help in assessing tree health, identifying stress due to pests, diseases, or water deficiency. They also assist in managing irrigation more efficiently, mapping and surveying land and optimizing the layout of groves and planning harvests. Additionally, UAVs can be equipped with multispectral cameras to

analyze plant health and detect early signs of problems [4]. This technology improves yield prediction, enhances resource management, and reduces operational costs, contributing to more sustainable and efficient olive oil production.

1.2. Current Usage of Blockchain in the Olive Oil Industry

Blockchain technology is increasingly applied in the olive oil industry to improve transparency, traceability, and trust across the supply chain. It functions as a distributed ledger, recording every exchange with cryptographic safeguards that enhance security and auditability of stored data. By leveraging these features, producers can provide consumers with reliable information on origin, quality, and production processes, ensuring authenticity and reducing fraud such as mislabeling or adulteration. Smart contracts further automate transactions and secure fair payments to farmers, while blockchain-based traceability helps reduce counterfeiting and supports sustainability verification. However, despite these advantages, most existing efforts overlook the direct management of UAV-generated imagery, leaving a gap in secure aerial data transmission that this work addresses [5]. Indeed, as it has been reviewed in Section 2, most blockchain applications in the olive oil sector emphasize supply chain documentation rather than UAV data.

1.3. Current Usage of Distributed Systems Technologies in the Olive Oil Industry

Distributed systems technologies are increasingly used in the olive oil industry to enhance efficiency, traceability, and quality control. Blockchain provides transparency and authenticity across the supply chain, but its integration with complementary technologies has often been limited. IoT sensors already monitor soil moisture, weather, and crop health, yet their data are seldom combined with UAV imagery to create a unified, verifiable dataset. Similarly, while distributed storage systems like IPFS enable decentralized and tamper-resistant data management, their potential to store UAV images securely remains unexplored. Advanced analytics and machine learning add value in predicting yields and detecting diseases, but they rely on reliable, secure datasets. By linking UAV image capture with blockchain logging and IPFS distribution, our platform addresses this gap, offering an integrated approach that strengthens trust, efficiency, and resilience in olive oil production [6,7].

1.4. Paper Contributions

With all the previous considerations, it becomes evident that combining devices of significant mobility like UAVs, a blockchain framework to provide additional security, and distributed system able to replicate and further distribute the information provided is a desirable development. The work presented in this manuscript introduces several key innovations in the integration of blockchain technology with UAV operations and IPFS:

1. **Secure Transmission Platform (STP):** The development of a Secure Transmission Platform that leverages blockchain technology to encrypt and log each image captured and transmitted by the UAV ensures protection against unauthorized access and data tampering, enhancing data security in aerial operations.
2. **Efficiency Optimization:** By exploring innovative data compression techniques and optimizing blockchain transaction speeds, the developed work aims to improve data transmission efficiency. This optimization is crucial for supporting real-time or near-real-time image analysis and decision-making processes in applications such as environmental monitoring and emergency response.
3. **Versatile Application:** The project aims to demonstrate the practical applications of the integrated UAV-blockchain-IPFS STP in an application domain that can be replicated in other ones including environmental monitoring, urban planning, and disaster

management. This showcases the versatility of the technology and its potential to address real-world challenges in diverse industries.

4. **Interdisciplinary Collaboration:** By emphasizing the consideration of ethical and privacy aspects in UAV operations and data management, the presented research work promotes interdisciplinary collaboration between unmanned hardware, distributed systems and wireless communications. This holistic approach enriches the project with new perspectives and challenges, driving additional innovations at the intersection of UAV technology and blockchain.

These innovations collectively contribute to advancing the fields of UAV technology and blockchain integration, paving the way for enhanced security, efficiency, and transparency in aerial data transmission processes.

1.5. Paper Structure

This scientific paper is divided as follows: an introduction with the main topics that have been researched on has already been provided. Section 2 offers a comprehensive study on the State of the Art with regard to UAV usage in olive oil industry and how UAVs can be integrated with other technologies. Section 3 presents the solution design and the components that have been used to create the system that provides the information related to olive trees in a secured, decentralized manner. Section 4 presents the implementation works that have been carried out and shows the overall appearance of the tested system. Section 5 offers conclusions and advances on the future works that could be used to enhance the developed system. Author contributions and bibliographical references close the manuscript.

2. Related Works

Considering the amount of interest in the usage of UAVs and distributed Cyber Physical Systems in the field of agriculture, it comes as no surprise the fact that there is a significant amount of literature with regard to these technologies. However, the specific use case and general usability that is provided by our solution has yet to be matched in the context put forward in this manuscript.

2.1. Study of the State of the Art

Arena et al. [8] introduce BRUSCHETTA, an Internet of Things (IoT) blockchain-based framework developed to certify the Extra Virgin Olive Oil (EVOO) supply chain. By leveraging blockchain technology and IoT sensors, BRUSCHETTA provides innovative procedures for the certification process by providing a comprehensive traceability system that tracks the entire journey of EVOO production, from the initial plantation stages to the final retail distribution. The offered framework offers numerous advantages, including the ability to ensure fine-grained traceability of EVOO, enhance transparency for end-users by granting access to a tamper-proof history of the product via smartphones, maintain quality standards through IoT sensors dedicated to quality control, and meet the requirements set by European laws in the food industry. Despite these significant benefits, the implementation of BRUSCHETTA does not make use of images obtained from UAVs, so at its current state BRUSCHETTA is not able to use UAV to the advantage of the framework that is put forward.

Conti et al. [9] propose a comprehensive traceability system for EVOO using Near-Field Communication (NFC) technology, allowing consumers to access product information via smartphones. The methodology involves developing an application for each stage of the food chain, claiming to enable customization and easy system updates. Along with this easy customization, system updates, and integration with different supply chains are

provided too. The use of NFC technology enhances consumer confidence by providing detailed information on product quality, fostering loyalty. However, disadvantages may include the initial setup costs and the need for NFC-enabled smartphones. In addition to that, while NFC technology can provide significant amounts of information regardless of its short range, it typically is not able to provide imagery of the locations where NFC-based sensors are placed.

Gupta et al. [10] describe VAHAK, a Blockchain-based Outdoor Delivery Scheme using UAV for Healthcare 4.0 Services. VAHAK leverages Ethereum Smart Contracts and IPFS for secure and efficient delivery of medical supplies. It addresses challenges such as latency, network bandwidth, and storage costs by utilizing 5G-enabled TI for communication and storing hash keys on the Blockchain. VAHAK offers advantages like ultra-low latency, ultra-high reliability, and improved scalability compared to traditional approaches. It ensures real-time tracking of deliveries, secure data storage, and efficient communication among stakeholders. The usage of IPFS is in line with the work that has been performed in these research activities as it guarantees a further layer of decentralization that is available in the other proposals, as well as the utilization of UAVs as tools to collect data. However, the application domain where this proposal has been developed is vastly different from the one (olive tree farming and olive oil production) that is being put forward in this manuscript.

Kechagias et al. [11] discuss the practical application of an Ethereum-based distributed application for enhancing traceability in the food supply chain, focusing on a Greek table olives producer. They mention how using blockchain technology ensures immutable data recording, enhancing data integrity and authenticity, and describe how blockchain enables quick issue identification, promoting product safety and quality, crucial for maintaining producer reputation. The study highlights the benefits of blockchain, such as improved supply chain management and reduced food fraud, while also acknowledging challenges like implementation costs. Overall, the research demonstrates the potential of blockchain in enhancing traceability and transparency in the olive industry, emphasizing the need for further development to optimize blockchain-based systems and address limitations in implementation, particularly for small-scale producers and low-income countries. Unfortunately, the built system relies on packaging information rather than having direct information from the olives themselves or the olive trees. Imagery or UAVs are also not present in the works that have been carried out.

Bean Ayed et al. [12] discuss the integration of innovative technologies in the agri-food sector, focusing on DNA-based traceability of olives from fruit to oil. It highlights the importance of utilizing advanced technologies such as IoT, blockchain, big data, artificial intelligence and nanotechnologies to enhance productivity, ensure product quality, optimize trade markets, and promote sustainability in agriculture. The study emphasizes the significance of these technologies in meeting the challenges posed by factors like the global population growth and the impact of the COVID-19 crisis on the agri-food industry. The authors describe how these technologies offer numerous benefits such as increased productivity, cost savings, and market expansion. Overall, the paper underscores the transformative potential of integrating innovative technologies in the agri-food industry, but its scope differs from the one that is provided by our manuscript, in the sense that DNA-based traceability is mentioned as a critical topic, but there is no significant mention to, for example, UAVs.

Mercuri et al. [13] introduce Devoleum, which is an agri-food sector startup that delves into the potential of blockchain technology to enhance transparency and sustainability in global supply chains. By utilizing blockchain, Devoleum aims to improve traceability, security, and information integrity while reducing transaction costs and time associated with intermediaries. The study highlights how blockchain can drive the development

of sustainable business models by leveraging its decentralized and immutable nature. The advantages of blockchain technology in this context are clearly stated, and include increased transparency, trust-building with customers, and reduced transactional costs. However, limitations of the study lie in the operational status of Devoleum as a start-up and not yet fully incorporated. In addition to that, the study that is presented here does not mention at all the existence of UAVs in smart farming, and how they can be used in the context of olive oil production.

Bistarelli et al. [14] research work describes the utilization of the **-chain* framework for modeling and implementing supply chain management systems, using the olive oil supply chain as a case study. The framework integrates a graphical editor and a Domain-Specific Graphical Language (DSGL) to design supply chain models, which are then translated into smart contracts for blockchain implementation. The advantages such platform include its ability to represent complex supply chain processes, generate optimal solidity code, and provide a user-friendly interface for domain experts. However, there is no mention about how UAVs could be used to gather information from the olive trees or the olives themselves before being harvested. No overall mention is made about sensing equipment so that it can be understood how information can be collected.

Haque et al. [15] introduce a similar approach to enhancing the efficiency and security of the oil supply chain through the integration of blockchain technology and smart contracts. As described before in previous proposals, the authors leverage blockchain decentralized and immutable nature by creating a system that aims to address key issues such as trust, third-party interference, end-to-end monitoring and data security. Advantages of this approach include enhanced data security, real-time data updates, customer access to information and the elimination of intermediaries controlling oil prices. However, challenges may arise in terms of implementation complexity, scalability, regulatory compliance and the need for widespread adoption. As in previous cases, while sensors (and Wireless Sensor Networks) are mentioned as sources of data, no consideration is taken to the usage of UAVs to collect information.

Violino et al. [16] present a full Technological Traceability System (TTS) for EVOO production, focusing on the implementation of an electronic traceability prototype in a small Italian farm. The system utilizes blockchain technology and QR codes to track the production process from olive tree to EVOO bottle, ensuring product authenticity and consumer safety. Advantages of the TTS include enhanced transparency, improved product quality and increased consumer trust through secure information sharing. The system also offers economic benefits by potentially attracting more consumers willing to pay for traceable products. However, challenges such as initial implementation costs, technological complexity and lack of popularity or effective adoption may hinder its full-scale integration in the olive oil production chain. UAVs are mentioned as a tool only as a future work, so they have yet to be implemented under this system.

Abenavoli et al. [17] depict the implementation of a traceability system for olive oil production in Calabria, focusing on tracking and tracing products throughout the supply chain. The study emphasizes the use of web service-based technology, such as a web application on a cloud server, to centralize information and improve collaboration among stakeholders. Among other features, the described system includes real-time data access, notification messages for quality control and improved transparency in the production process. The developed software enables efficient monitoring of critical phases like harvesting, processing, and distribution, enhancing product quality and safety. However, potential disadvantages may include initial implementation costs, training requirements for users, and the need for continuous updates to maintain system effectiveness. In addition to the latter, while the software technology that is effectively used as middleware is described

with thorough detail, there is next to no information about what kind of hardware devices (sensing equipment, UAVs) would be used to collect information straight from olives or olive trees. Further decentralization performed at the upper layers of the system (i.e., using IPFS or a comparable technology) is not mentioned either.

Ktari et al. [18] present a case study on an Agricultural Lightweight Embedded Blockchain System applied to the olive oil industry. The system establishes a secure supply chain involving farmers, oil manufacturers, quality control companies, and transporters, utilizing private and public blockchain networks. By implementing the system on Raspberry Pi and Arduino boards, the researchers demonstrate the feasibility of an embedded blockchain system with low power consumption. The use of smart contracts ensures transparent and traceable transactions throughout the supply chain. The system enhances data security, confidentiality, and trust among stakeholders. However, the complexity of the system and the need for multiple platforms may pose challenges. Unlike other proposals that have been mentioned before, there is an explicit description of the sensors that are used to collect information. Unfortunately, UAVs are not mentioned as data sources, nor does an application layer distribution system like IPFS seem to have been used.

Ushasri Peddibhotla et al. [19] propose a blockchain-enabled framework to secure UAV communications in smart agriculture. It introduces a permissioned blockchain architecture combined with multi-stage authentication and key agreement protocols among UAVs, edge servers, and cloud servers. The system leverages session-based authentication, pseudo-identities, and cryptographic techniques such as Elliptic Curve Encryption and SHA-512 hashing to protect sensitive agricultural data. Proof-of-Authority (PoA) consensus ensures low latency and high throughput, while smart contracts enforce secure and transparent rules for data access. This system includes strong protection against impersonation, Man-in-the-Middle (MITM) and replay attacks, guaranteed data integrity and offers resilience against tampering and scalability through PoA consensus. However, it also has some drawbacks: it relies on a Trusted Authority (TA) for registration, which introduces centralization risks, it has a potential overhead from complex key exchanges, has challenges in practical deployment due to computational requirements on UAVs and presents limitations in handling large multimedia datasets. The proposed framework focuses primarily on secure communication rather than efficient storage or distribution of UAV imagery, and there is no data about decentralized or tamper-resistant storage.

Shebl Soliman et al. [20] depict a blockchain-based smart agriculture architecture using Hyperledger Fabric deployed on Kubernetes to coordinate UAV swarm operations. The system leverages a multi-organization blockchain with peers, orderers, and Certificate Authorities to manage UAV registration, mission updates, and supply chain transactions. Chaincode-as-a-Service (CaaS) provides flexibility for updates without network disruption, while Kubernetes ensures scalability, fault tolerance, and persistence of ledger data. UAVs communicate through secure APIs, sending real-time telemetry and receiving mission updates, with TLS and HTTPS ensuring communication security. This system offers strong scalability, tamper-proof data handling and a modular chaincode tailored for agriculture and supply chain management, but it has some complexity in deployment (requiring extensive Kubernetes expertise besides blockchain), performance limitations under extreme transaction loads and does not provide information on real-world field validation, as the work remains primarily at the architectural and benchmarking level.

Sanjeev Kumar Dwivedi et al. [21] introduce a lightweight authentication and data storage scheme tailored for the Internet of Drones (IoD) in tactile Internet environments. It leverages Elliptic Curve Cryptography (ECC), one-way hash functions and blockchain to secure communication between users, ground station servers, and UAVs. The system

resists impersonation, UAV capture, session-key disclosure, desynchronization, and replay attacks, while providing anonymity, unlinkability, and untraceability. Formal verification with the Scyther tool and Real-Or-Random (ROR) model supports its robustness. The system offers reduced computational overhead (≈ 95 milliseconds), efficient communication cost (3.2 Kbits), and stronger security properties compared to many existing schemes. Unfortunately, its current validation is simulation-based rather than based on real-world deployments. Also, its focus remains more on authentication than on scalable UAV data management and does not directly address UAV multimedia data challenges.

Qianqian Zheng et al. [22] describe a blockchain and IoT-enabled service framework designed to improve plant protection operations for small-scale farmers. It develops a five-layer agricultural service platform integrating IoT sensing, blockchain security, operations research optimization, and smart contracts. The system collects real-time crop condition data, generates optimized drone scheduling plans using mixed-integer linear programming and deploys smart contracts on Ethereum to ensure service traceability, fraud prevention, and secure execution. It also includes a disruption recovery model to handle unexpected events such as new service orders. Experimental validation demonstrates the system ability to minimize costs, improve scheduling efficiency and maintain robustness during disruptions. This system can offer strong optimization for resource allocation, improved service transparency and automated enforcement via smart contracts. However, its scope (focused mainly on pesticide spraying) is limited and relies on cloud-based optimization rather than fully distributed mechanisms.

Lastly, Tanya Garg et al. [23] provide a comprehensive survey of UAV technologies, their integration with IoT and 5G networks, and the role of blockchain in enhancing security. It explores UAV components, deployment architectures (manual, autonomous, swarm-based, and WSN-enabled), communication protocols (WiFi, LTE, 5G, LoRaWAN), and ad hoc networking approaches. It also details smart city applications such as traffic management, waste monitoring, precision agriculture, and connected vehicle systems. This work offers holistic coverage of UAV hardware, software, and networking aspects, its integration of blockchain as a scalable security solution against cyberattacks and discusses the inclusion of AI/ML for UAV communications. However, since it is a survey, it does not provide a working prototype or experimental validation, leaving its proposals largely conceptual.

2.2. Open Issues

As can be seen, there are a significant number of solutions that have been put forward to integrate several key technologies into smart farming platforms. However, there are still significant open issues that demand additional work to be tackled in a successful manner. A summary of the open issues that have been found is displayed in Table 1.

It can be claimed by the authors of this manuscript that the solution we put forward addresses, to a significant extent, the common issue of the underuse of UAVs in smart farming, especially in the context of EVOO. Our framework demonstrates how UAVs, as data collection devices, and IPFS, as a decentralized distribution layer, can be effectively combined to enhance security and transparency. At the same time, it is important to recognize that UAV integration is not the only open challenge in this field. Existing blockchain agriculture solutions also struggle with scalability of transactions, latency in real-time data delivery, and the storage overhead of large multimedia datasets. These issues remain critical for any system that aims to support continuous UAV operations and high-frequency image capture. By highlighting both UAV underusage and these systemic limitations, we provide a more balanced perspective on the research gaps that motivate our proposed platform.

Table 1. Summary of the studied research works, their advantages and disadvantages.

Research Work	Advantages	Disadvantages
Arena et al. [8]	Access to a tamper-proof history of the product via smartphones, IoT sensors for quality control, and meet the requirements set by European laws.	Usage of UAVs or application layer distribution for redundancy (IPFS-like solution) is not considered.
Conti et al. [9]	NFC provides detailed information on product quality fostering loyalty.	No imagery or application layer distribution (IPFS-like solution) used.
Gupta et al. [10]	Ultra-low latency, ultra-high reliability, improved scalability. Usage of IPFS guarantees a further layer of decentralization.	Scope differs from the one that is provided by our manuscript.
Kechagias et al. [11]	Traceability and transparency, particularly for small-scale producers and low-income countries.	Relays on packaging information rather than olives trees.
Bean Ayed et al. [12]	Underscores the transformative potential of integrating innovative technologies in the agri-food industry.	Scope differs from the one that is provided by our manuscript.
Mercuri et al. [13]	Increased transparency, trust-building with customers, and reduced transactional costs.	Progress linked to the operational status of Devoleum Usage of UAVs or application layer distribution is not considered.
Bistarelli et al. [14]	Represents complex supply chain processes, generates solidity code, and provides a user-friendly interface.	Usage of UAVs or application layer distribution is not considered. Sensing information is missing.
Haque et al. [15]	Real-time data updates, customer access to information, and the elimination of intermediaries.	Usage of UAVs or application layer distribution (IPFS-like solution) is not considered.
Violino et al. [16]	Blockchain technology and QR codes to track the production process from olive tree to EVOO bottle.	Usage of UAVs is left for future works. Application layer distribution is not considered. Complexity and initial running costs.
Abenavoli et al. [17]	De facto middleware monitoring critical stages of olive oil production.	Usage of UAVs or application layer distribution (IPFS-like solution) is not considered. Lack of information on hardware devices.
Ktari et al. [18]	Blockchain throughout supply chain. Usage of Smart Contracts.	Usage of UAVs or application layer distribution (IPFS-like solution) is not considered.
Ushasri Peddibhotla et al. [19]	Security-related features. Offers resilience against tampering and scalability through PoA consensus.	Focuses primarily on secure communication. No information about decentralized or redundant storage.
Shebl Soliman et al. [20]	Scalability-related features, tamper-proof data handling, tailored chaincode.	Complexity in deployment, performance limitations, work remains primarily at the architectural and benchmarking level.
Sanjeev Kumar Dwivedi et al. [21]	Reduced computational overhead. Efficient communication cost. Significant security properties.	Current validation is simulation-based. Focus remains more on authentication than on scalable UAV data management.

Table 1. *Cont.*

Research Work	Advantages	Disadvantages
Qianqian Zheng et al. [22]	Optimization for resource allocation, improved service transparency, automated enforcement via smart contracts.	Scope is limited, relies on cloud-based optimization rather than distributed mechanisms.
Tanya Garg et al. [23]	Holistic coverage of UAV hardware, software, and networking aspects, integration of blockchain, discusses the inclusion of AI/ML.	It is a survey, so it does not provide a working prototype or experimental validation.

3. Solution Design

3.1. Methodology

The performed research activities were guided by the following central question: How can blockchain and IPFS be integrated with UAV operations to provide secure, efficient, and transparent transmission of aerial imagery in the context of olive oil production? To address this, a three-step methodology was followed:

1. **System Design:** A Secure Transmission Platform (STP) was conceptualized, integrating UAV image capture with blockchain for immutability and IPFS for decentralized storage. The design specified technological components (Parrot ARDrone 2.0, Java-based blockchain, IPFS client) and performance requirements.
2. **Implementation:** The platform was developed in Java, including blockchain classes for block creation, encoding and persistence, and an interface to IPFS for distributed storage. UAV imagery was encoded in Base64 and embedded into blockchain blocks to validate security and traceability.
3. **Testing and Evaluation:** Field experiments were carried out in olive groves in Castilla-La Mancha (Spain). A dataset of 282 UAV images divided in 3 datasets was collected, encoded, stored on the blockchain, and distributed through IPFS. The evaluation focused on system feasibility, storage overhead, and data integrity verification.

The workflow followed is depicted in Figure 1. It begins by defining the research question, followed by designing the Secure Transmission Platform (STP) that integrates UAVs, blockchain, and IPFS. The platform is then implemented in Java, including blockchain and IPFS modules. UAV imagery is collected in the field, encoded, stored, and distributed through the STP to ensure security and decentralization. Finally, system performance is evaluated in terms of data security, efficiency, and storage overhead, providing a structured process that links conceptual design and technical implementation.

This methodology ensures that the proposed contributions are reproducible and verifiable, and that they directly address the research question.

3.2. Solution Design Description

In this subsection, we aim to show the integration between UAVs, blockchain and IPFS in the application domain of olive trees and olive oil industry taking shape as the STP that embeds blockchain technology into the image transfer process of the UAV. The platform manages every image captured and transmitted, thereby safeguarding against unauthorized access and data tampering while also enabling a new level of auditability and transparency in UAV data management. To ensure the platform feasibility, it will encompass comprehensive planning around distribution technologies, blockchain architecture, and seamless integration with the UAVs existing communication systems. To underpin the development work and ensure its success, a detailed specification of tech-

nological, operational, and performance aspects has been outlined. This includes the hardware capabilities of the ARDrone 2.0 from the Parrot manufacturer [24], the usage of blockchain, and the implementation of a client able to connect and upload information to an IPFS-based network.

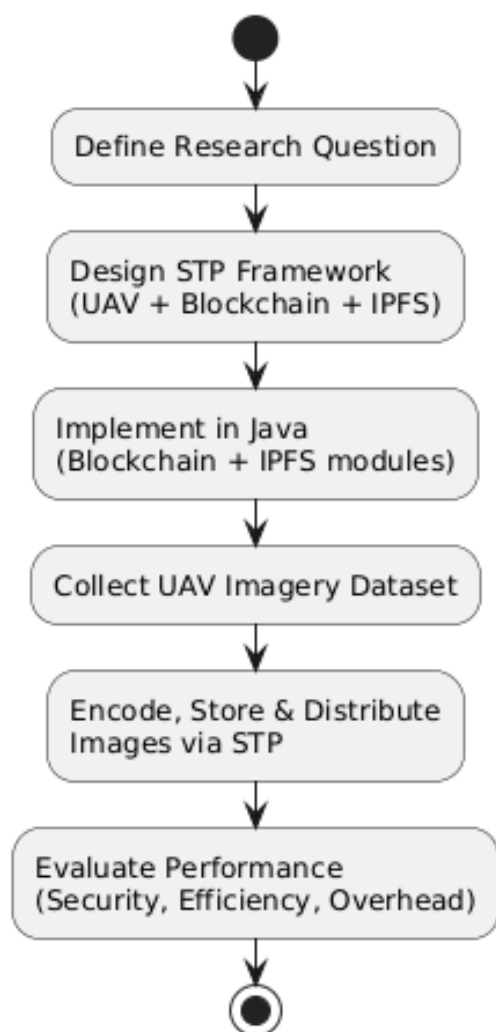


Figure 1. Workflow diagram of the methodology followed.

As can be seen in Figure 2, the STP has been created with three different prominent subsystems:

1. The UAV used to collect images from the olive trees being monitored (communications depicted as red arrows in Figure 2). A laptop used as the base station connected to the UAV via an 802.11 Wi-Fi interface provides the seamless transfer of the images collected by the UAV onto a hardware device (that is, the laptop itself). This hardware device acts effectively as the entry point to the blockchain network, ensuring that the data is securely and accurately logged. Additionally, the same hardware is shared with the IPFS network, providing a decentralized and distributed storage solution. This dual use of hardware not only enhances the efficiency of the system but also ensures the integrity and accessibility of the collected data, thereby optimizing the monitoring process and improving data management within the blockchain framework. The Parrot ARDrone 2.0 was selected as the UAV platform primarily due to its accessibility, affordability, and programmability. As a widely available consumer-grade drone, it provides a cost-effective option for prototyping without requiring the acquisition of specialized equipment, which often involves higher costs and stricter

regulatory restrictions. Its open API and compatibility with Java-based development allowed straightforward integration with our blockchain and IPFS modules. While professional UAVs could offer longer flight times and higher-resolution sensors, the ARDrone 2.0 provided sufficient imaging capabilities (1280 p × 720 p resolution) and stable flight performance to validate the feasibility of our proposed framework. This choice demonstrates that secure UAV–blockchain–IPFS integration can be implemented even with low-cost hardware, improving the accessibility and reproducibility of the platform for academic and small-scale agricultural contexts.

2. The blockchain network (green arrow communications in Figure 2), which is used to obtain UAV images from the olive trees, ensures the integrity and security of the collected data by adding them as timestamped entries in Base64 format onto a Java-developed blockchain. This process not only preserves the chronological order of the images but also provides an immutable record, enhancing the traceability and verification of the agricultural monitoring data. The use of Base64 encoding facilitates efficient storage and transmission of the images within the blockchain, while the Java-based implementation offers robust performance and compatibility with various systems and applications. Storing Base64-encoded images directly on the blockchain introduces inefficiency, as Base64 increases file size by approximately 33%. In our tests, an average UAV image of 250 KB expanded to roughly 333 KB when encoded, which in turn increased block size and transaction delays. However, this design choice was intentional at the prototyping stage, as it enabled us to embed raw visual data directly within the blockchain without requiring external file references. This approach ensured end-to-end immutability and verifiability of the images, a critical step in validating the security model. To mitigate scalability issues, our system complements on-chain storage with IPFS, where full-resolution images are distributed, while the blockchain stores either the Base64 hash or a reference CID. Thus, the Base64 experiment demonstrated feasibility and security guarantees, while the long-term design shifts heavy storage to IPFS to address size overhead and latency concern.
3. The IPFS network—represented as communications with blue arrows in Figure 2—enhances system decentralization and security. Each of the hardware devices that contains the blockchain node is equipped with an IPFS client, enabling a seamless switch to the IPFS network. This integration allows the interchange information from the blockchain to be formatted as a JSON array of objects written onto a file. This file is then distributed across the IPFS network, creating a second layer of decentralization. By leveraging IPFS, the system significantly improves data transparency and security. The decentralized nature of IPFS ensures that the data are not stored in a single location, reducing the risk of data loss or tampering. This dual-layer approach, combining blockchain and IPFS, provides robust data integrity, making the entire monitoring process more resilient and trustworthy. Additionally, the use of JSON arrays for data formatting enhances the readability and interoperability of the data, facilitating easier access and analysis by various stakeholders. This comprehensive system design underscores the importance of advanced technologies in creating secure and transparent data management solutions.

Each of the critical components of the subsystems that represent the overall integrated structure will be described in further detail in the following subsections of this manuscript.

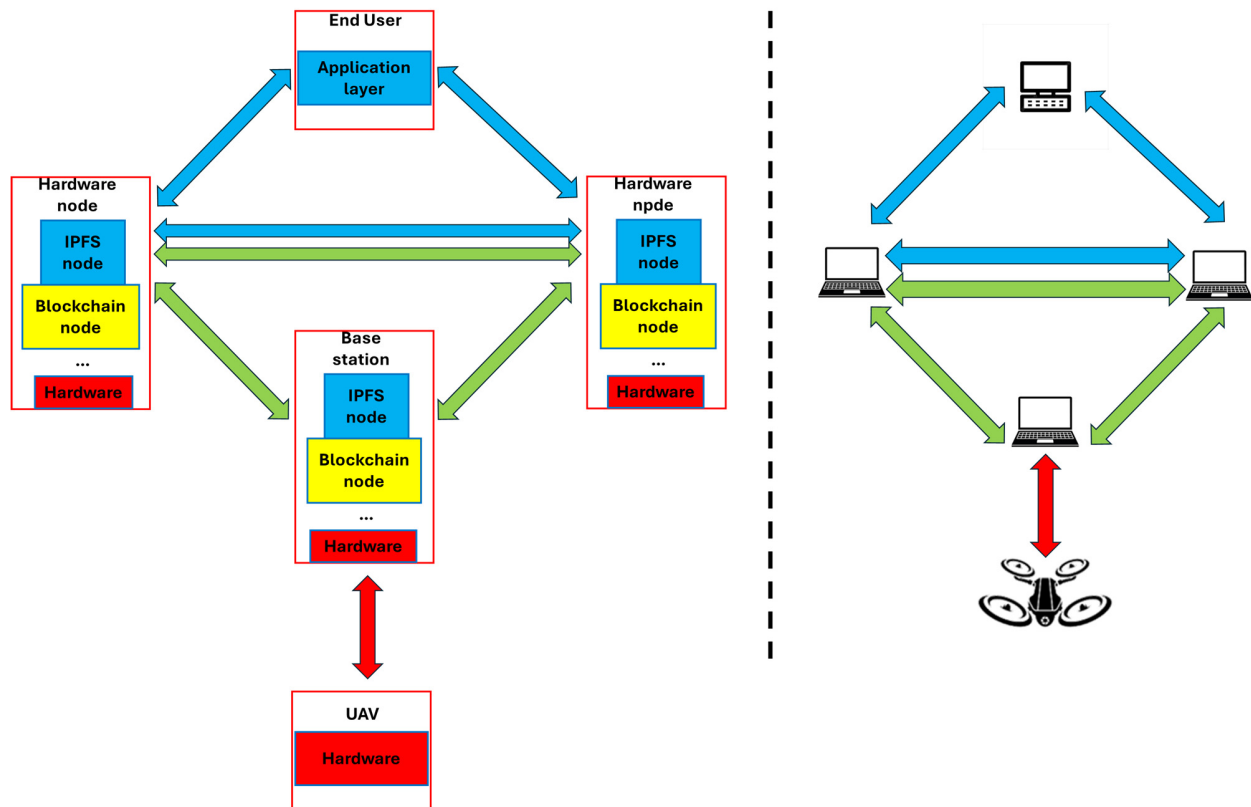


Figure 2. Deployed platform from the hardware and software points of view.

3.3. UAV Description

For the purpose of image collection, the ARDrone 2.0 from the UAV manufacturer Parrot has been chosen to perform the flight operations (Figure 3). The ARDrone 2.0 is equipped with a dual-camera system designed for image acquisition. The primary camera, mounted on the front of the UAV, can capture high-definition still images and record videos with a resolution of 720 p (1280 × 720 pixels) using the H.264 standard [25], which ensures efficient video compression and high-quality output.



Figure 3. ARDrone Parrot 2.0, as depicted in [24].

In addition to the frontal camera, the UAV features a secondary vertical camera located underneath the UAV. This secondary camera provides an alternative viewing angle, making it useful for tasks that require detailed ground observation and navigation assistance. Powering the ARDrone 2.0 is a 1000 mAh lithium-polymer battery, which provides a substantial

flight duration of up to twelve minutes on a single charge. This battery capacity allows for extended periods of operation, essential for capturing extensive image and video data during flight missions. The combination of these advanced imaging capabilities and the reliable power supply makes the ARDrone 2.0 a versatile tool for various applications, including environmental monitoring, agricultural assessments, and surveillance tasks. Its dual camera system and efficient battery life ensure that users can gather high quality visual data effectively, supporting a wide range of professional and research activities.

3.4. Blockchain Development and Description

In the context of this paper, the integration of blockchain technology plays a pivotal role in enhancing the security, efficiency, and transparency of aerial image transfer processes. It aims to leverage the unique properties of blockchain, such as decentralization, immutability, and transparency, to address critical challenges in data security, integrity, and management inherent in contemporary UAV operations. Its usage is of critical importance due to the following reasons:

1. **Secure transmission procedures:** One of the key applications of blockchain technology in these research activities is the development of secure transmission procedures that embed blockchain into the image transfer process. These procedures aim to collect and log every image captured and transmitted by the UAV, ensuring information redundancy and protection against data tampering. Thus, by utilizing blockchain decentralized and immutable nature, the security of data transmission becomes enhanced, making it resistant to unauthorized manipulation or interception.
2. **Data integrity and transparency:** Blockchain technology enables the creation of a tamper-proof and transparent record of all image transfers, ensuring data integrity throughout the process. Each transaction is securely recorded on the blockchain, providing an auditable trail of image transfers and ensuring that the data remains unchanged and verifiable. This level of transparency and auditability enhances trust in the data being transmitted and received, crucial for applications requiring accurate and reliable information.
3. **Efficiency and near Real-Time Analysis:** In addition to security and transparency benefits, blockchain integration also focuses on optimizing data transmission efficiency to support timely analysis. By exploring innovative data compression techniques, optimizing blockchain transaction speeds, and creating a blockchain-based application that supports high-throughput, low-latency operations, research works aim to enable real-time or near-real-time image data analysis—along with its integration in the aforementioned three subsystems- and decision-making. This emphasis on efficiency ensures that the UAV system can process and transmit data quickly and effectively, enhancing its operational capabilities in various sectors.
4. **Versatile Applications and End-User Engagement:** Through practical application and engagement with a real-world application domain, the performed research works aim to demonstrate the versatility and potential of the integrated UAV-blockchain-IPFS STP in addressing real-world challenges across different application domains that might or might not be strictly related to agriculture or olive tree farming. By showcasing the technology practical applications and engaging potential end-users to map out specific use cases, the project aims to highlight the system capabilities and opportunities for addressing diverse operational needs.

The design of the blockchain module was guided by specific requirements to ensure its suitability for UAV image transmission in olive oil production. These included (a) maintaining data integrity and immutability for every recorded image, (b) achieving sufficient efficiency to avoid delays in data logging, (c) enabling scalability to accommodate

larger UAV datasets, (d) ensuring seamless interoperability with IPFS and UAV control modules, and (e) providing transparency for auditing purposes. Storage optimization was also considered, given the overhead introduced by Base64 encoding. To evaluate these requirements, we defined concrete metrics, summarized in Table 2, which allowed us to measure performance in terms of security, efficiency, scalability, interoperability, auditability, and storage efficiency. Table 2 summarizes the overall described features:

Table 2. Comparison between design requirements, their description and their evaluation metric.

Design Requirement	Description	Evaluation Metric
Data integrity, immutability	Ensure UAV images cannot be altered once stored	Hash verification success rate per block
Efficiency	Maintain low overhead for encoding and storage	Avg. transaction delay (ms), block creation time
Scalability	Handle increasing UAV image datasets	Throughput (images/s), total blocks stored
Interoperability	Integrate with UAV control and IPFS storage seamlessly	Successful integration rate, API compatibility
Transparency, auditability	Enable full traceability of image transfers	Blockchain log completeness, audit trail validation
Storage optimization	Reduce costs and duplication by using off-chain distribution	Storage overhead (Base64 vs. IPFS)

As mentioned, the blockchain developed here is implemented in Java, which was chosen for its robustness, security, and portability. Java's cross-platform runtime (Java Virtual Machine, JVM) ensures code consistency across different operating systems, while its built-in memory management supports stable long-term execution. These features make Java a practical choice for integrating UAV control, blockchain management, and IPFS handling within a single framework, providing reliability without the overhead of platform-specific adaptations [26].

Java also stands out for its robust type of system and focuses on object-orientation, making it easier to structure code and maintain it in the long term. These features are essential to design a clear and modular code for controlling UAVs, where safety and clarity in status management are a priority. Object orientation makes it possible to encapsulate UAV behaviors and blockchain operations into separate objects, simplifying debugging, testing, and future expansion of code. In addition, the extensive developer community and extensive library of APIs and frameworks available for Java, such as Spring and Hibernate, offer invaluable resources to accelerate development and ensure the integration of complex and specialized functions. These libraries and tools facilitate everything from cryptography to network communication and file management, crucial components for the effective management of blockchain and IPFS [27]. Java is also preferred in enterprise environments for its commitment to security, which is of major importance when it comes to the transmission and storage of sensitive data over blockchain. Built-in security features and constant security updates help protect against external vulnerabilities and ensure the integrity of managed and transmitted data.

Finally, the use of Java in interaction with IPFS for file handling on the blockchain benefits from Java performance when handling large volumes of data and performing network operations efficiently. Java ability to integrate native systems and manage data-intensive processes ensures that files are handled efficiently, maintaining data integrity and accessibility in a globally distributed system. All these characteristics make Java a sound strategic and technical choice for the development of a system involving emerging

and critical technologies such as UAVs and blockchain, ensuring efficient, secure, and maintainable development in the long term.

3.5. IPFS Description

IPFS, or Interplanetary File System, is a technology that complements blockchain by offering a decentralized storage system. Its use in combination with blockchain is particularly advantageous because it allows large volumes of data to be handled efficiently and economically, addressing a significant challenge in traditional blockchain networks due to their full-replication nature and associated costs. In a conventional blockchain system, directly storing large amounts of data can be costly and impractical. However, using IPFS, the blockchain can simply store a reference to the data (usually, a cryptographic hash) while the entire data is stored in IPFS. Not only does this significantly reduce storage costs on the blockchain, but it also optimizes the speed and scalability of transactions. Another key benefit of IPFS in the context of blockchain is data persistence. Once a file is uploaded to IPFS, it is distributed over a peer-to-peer network, which means that as long as there are nodes on the network that maintain a copy of the file, it will remain accessible. This adds an extra layer of security and censorship resistance, as data is not dependent on a single server or location and is virtually immune to tampering. Additionally, IPFS enhances the reliability and availability of data, ensuring that it can be retrieved even if some nodes go offline.

Moreover, IPFS can help solve some of the scalability issues that current blockchains face when handling the loading of data off the main chain. By offloading data storage to IPFS, blockchain-based applications can operate more efficiently while maintaining the integrity and security of the network. This separation of data storage and transaction processing allows for more streamlined and faster blockchain operations, supporting more complex and data-intensive applications. Overall, the integration of IPFS with blockchain technology provides a robust solution for managing large datasets in a decentralized, secure, and cost-effective manner, paving the way for more advanced and scalable blockchain applications. As mentioned in the previous subsection, the implementation of IPFS has been carried out for this platform in two different ways: on the one hand, a client has been installed to connect to the IPFS global network so that images can be shared among all the members of such networks and chances of manipulation are further diminished. On the other hand, a Java development has been developed to connect the IPFS node with the images that are being formatted in the blockchain with the IPFS network itself.

3.6. Solution Development

The code has been structured in a series of major software components related with blockchain development. In order to clearly define the code, the relationships among classes, their attributes and methods, Figure 4 displays a class diagram with all the information.

The code that has been generated has been performed so partially based on what is described in [28]. In addition to that, further developments performed as part of the implementation of the research works have been uploaded onto GitHub [29]. The Java implementation adopts Proof-of-Work (PoW) as the consensus algorithm. Although PoW is often criticized for its energy intensity and latency in large-scale networks, its usage in our environment is justified by the limited computational and data requirements of the prototype. The blockchain is maintained on ground-based hardware rather than the UAV itself, meaning that the UAV energy consumption is unaffected by consensus. Furthermore, the number of images per flight session (282 in our tests) and the average encoded block size are modest, resulting in low transaction throughput that does not trigger prohibitive delays. In this constrained scenario, PoW provides a straightforward way to guarantee immutability and tamper resistance without introducing the complexity of setting up

validator committees or permissioned networks. The different Java classes implemented as represented in Figure 4 are further described in the following subsections.

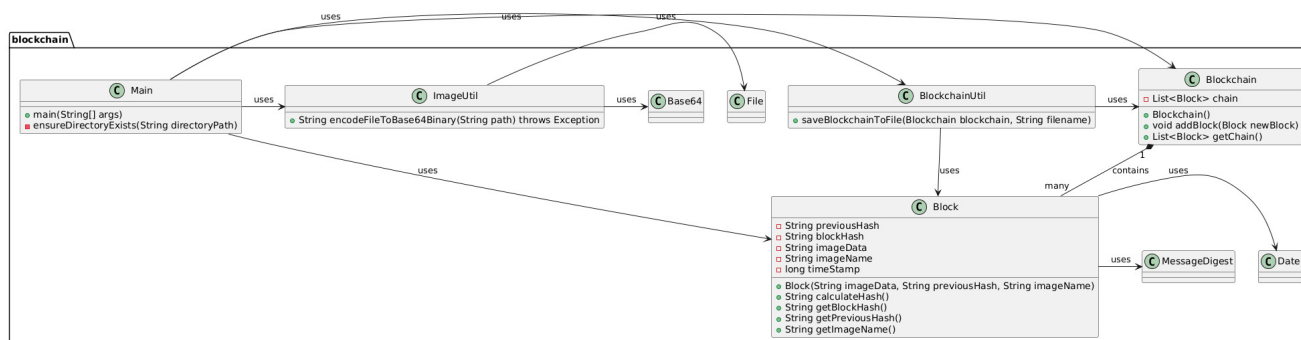


Figure 4. UML class diagram from the blockchain Java development.

3.6.1. Block Java Class

The Block Java class is a crucial component of the blockchain system designed to handle and secure image data. It models an individual block within a blockchain, where each block contains specific data from an image, plus cryptographic references to other blocks in the chain. This structure is essential to ensure the integrity and traceability of data throughout the entire chain. Its functionality could be summed up in two major steps.

1. **Initialization and Block Creation:** When a new block is created, the image information (data and name) is passed to it, along with the hash of the previous block. This ensures that each block is linked to the block that precedes it, thus forming a continuous and immutable chain. The timestamp captured during block creation helps record when the block was added to the chain, providing temporal context that is vital for auditing and verification operations.
2. **Hash Calculation:** The SHA-256 algorithm is used to generate a hash from the combination of the previous hash, the timestamp, and the image data. This hash acts as a digital signature of the block, ensuring that any alterations in the block's data are detectable. This is an essential part of the security provided by blockchain, as any change to a block will require the recalculation of all hashes of subsequent blocks, which is computationally prohibitive and serves as a strong deterrent against tampering.

By including the name and image data directly in the block, the design allows the blockchain to not only secure typical financial or transaction information but also handle multimedia data effectively. This is especially useful in applications where the authenticity and provenance of images are critical, such as in digital rights management or in documenting conditions in environmental monitoring applications.

3.6.2. Blockchain Java Class

The Blockchain Java class is used to manage a list of blocks, which are the fundamental structures where data is stored securely and sequentially in the blockchain system. This class plays a crucial role in the creation of the blockchain and its ongoing maintenance, ensuring that data is aggregated in an immutable and transparent manner. It works as follows:

1. **Blockchain initialization:** When the Blockchain class is instantiated, it is initialized with a genesis block. This is the first block on the chain and is automatically created with a previous dummy hash of "0" to mark the start of the chain. The genesis block is essential because it establishes the root of the blockchain from which all subsequent blocks will be linked.

2. Adding blocks to the chain: A method is provided to add new blocks to the blockchain. Every time a new block is created, for example, after the capture of a new image or transaction, this block is added to the end of the chain. The new block stores the hash of the last block in the chain, cryptographically linking them. This ensures that any modification to a previous block would invalidate all subsequent blocks, protecting the chain against tampering.
3. Chain access: The blockchain can be accessed using a method that returns the entire blockchain. This is useful for verifications, audits, and for applications that need to validate the integrity of the entire chain or extract historical data.

This class reflects how blockchain uses principles of immutability and sequentially to ensure the integrity of stored data. Each new block depends on the previous block, creating a permanent and verifiable historical record of all data added to the chain.

3.6.3. BlockchainUtil Java Class

In this BlockchainUtil java class, a method called *saveBlockchainToFile* has been implemented, which allows us to store the blockchain data in a file. This method is crucial to ensure that blockchain information can be preserved, audited, and reviewed beyond the current execution environment. This class consists of:

1. Blockchain data storage: Data persistence management is facilitated using the *saveBlockchainToFile* method. In this method, you receive as parameters a Blockchain instance, which contains all the blocks, and a filename, which specifies the path of the file where you want to save the blockchain.
2. Data writing process: When invoked, the method opens a BufferedWriter linked to the specified file. It is iterated over each block within the provided chain. For each block, the name of the image, the hash of the block, and the hash of the previous block are written to the file. Each block is visually separated by dashed lines to facilitate the readability of the resulting file.

This class provides a means to ensure the persistence of blockchain data, facilitating backup and recovery operations. Additionally, the generated file can serve as a form of auditing and verification of the blockchain status at any given time.

3.6.4. ImageUtil Class

In this class, a static method called *encodeFileToBase64Binary* has been implemented, which is responsible for converting image files into a text string in Base64 format. This method is necessary for the manipulation of images within our blockchain, especially when they need to be securely integrated into the blocks of the chain. To encode these files the steps are:

1. Read Bytes: The method *Files.readAllBytes* allows for reading all bytes of a file located at a specified path. This is achieved by using the method *Files.readAllBytes*, which takes a Path object as an argument, representing the converted path.
2. Base64 encoding: These bytes are then encoded into a Base64 string using the *encodeToString* method of the *Base64.getEncoder()* encoder. Base64 is an encoding method commonly used to convert binary data into ASCII text strings.

The *encodeFileToBase64Binary* function is essential in the context of our blockchain, as it allows images to be securely integrated within blocks, without altering the binary nature of the data.

3.6.5. Main Java Class

The principal coordination for image processing and integration with the blockchain takes place in the Main class. The operations that are being carried out are as follows:

1. Directory Configuration: At the start of the main method, the base paths for images and results are set. The existence of the results folder is verified and, if it is not present, it is created.
2. Dataset Processing: For each dataset specified in an array, it is processed. The start of this process is notified in the console for each dataset.
3. Image Management and Blockchain: Within each dataset, image files are listed and processed. Each image is converted to Base64 format using a specific method of the ImageUtil class. With the image data already processed, a new block is created on the blockchain, which includes the image information along with the hash of the previous block in the chain. This newly created block is added to the blockchain.
4. Blockchain Persistence: Once all images in a dataset have been processed and recorded on the blockchain, the entire state of the blockchain is saved in a file within the results folder.

This main class acts as the operational core of the blockchain Java project, handling both the transformation of data and the integration and persistence of these in the blockchain. Proper directory management and careful exception handling are essential to the robustness and reliability of the system. This class ensures that data are not only processed securely but also stored in a way that allows for future retrieval and verification.

3.6.6. Results

After executing the developed code, data are stored in different blocks with their own hash output, as shown in the following results. Note that the previous hash displayed in each of the blocks is matching the hash than the block that came before had. Results containing the Base64 encoded images have been obtained and provided in [29] as well.

```
Image Name:  Genesis

Block Hash: cfb9172c4676f997c01dd6590be0c30dca57b8f5c9a9a3b2a55bbf614ae7e74a

Previous Hash:  0

-----

Image Name:  estadoActual0.jpg

Block Hash: e0b227b75c8ba467c26e5517b2edad30f777059628b46160438b2e9738c4e0ec

Previous Hash: cfb9172c4676f997c01dd6590be0c30dca57b8f5c9a9a3b2a55bbf614ae7e74a

-----

Image Name:  estadoActual1.jpg

Block Hash: f9ab5a39223d0bef0e0cbc895bb5d3cae57fd0f6d4bb20b64d055df6fa9fa9c9

Previous Hash: e0b227b75c8ba467c26e5517b2edad30f777059628b46160438b2e9738c4e0ec

-----

Image Name:  estadoActual10.jpg
```

```

Block Hash: 01d3d7005be2df4cf826f276552f84049fb4a0ee7b4658c2819bc167c2c92096

Previous Hash: f9ab5a39223d0bef0e0cbc895bb5d3cae57fd0f6d4bb20b64d055df6fa9fa9c9

-----

Image Name: estadoActual11.jpg

Block Hash: ee23741083b683cb07149c4996eae384c0d44d8eb7c851adfd01b1ca4f7d9bb5

Previous Hash: 01d3d7005be2df4cf826f276552f84049fb4a0ee7b4658c2819bc167c2c92096

-----

Image Name: estadoActual12.jpg

Block Hash: ee76e4d651eedfea2d6f14e4f6317fef272ea63840866612fb55243a25475a51

Previous Hash: ee23741083b683cb07149c4996eae384c0d44d8eb7c851adfd01b1ca4f7d9bb5

-----

Image Name: estadoActual13.jpg

Block Hash: 308746d54e587536f4fcb3a92a235c732f7e4d8423a66ddb7aaacfdccf88232

Previous Hash: ee76e4d651eedfea2d6f14e4f6317fef272ea63840866612fb55243a25475a51

-----

Image Name: estadoActual14.jpg

Block Hash: cacd3d4ea7ed62c8b1f1c4f7b0ed036dd33f1f003dd323e3d6395a6fb3cdec8

Previous Hash: 308746d54e587536f4fcb3a92a235c732f7e4d8423a66ddb7aaacfdccf88232

-----

Image Name: estadoActual15.jpg

Block Hash: 31e533a365514129502b2b4d63839c51247efed5abff063ed636750a19e9aa2a

Previous Hash: cacd3d4ea7ed62c8b1f1c4f7b0ed036dd33f1f003dd323e3d6395a6fb3cdec8

-----

[The blockchain follows further]

```

4. Solution Testing

In this section, field tests for collecting images using the UAV are discussed in detail. These tests have been conducted to gain an understanding of the dynamics of image capture and transmission in real-world environments. This understanding has been critical for

refining the consensus algorithm and the structure of the blockchain system designed to secure these images. Various factors such as the efficiency of the UAV camera settings, the stability of image capture under different flight conditions and the effectiveness of data transmission to a secure storage solution have been evaluated. Insights gained from these tests were crucial in developing a blockchain that ensures the integrity and privacy of images. Following the tests, the programming and implementation of the blockchain were carried out. A secure and efficient blockchain architecture was developed, capable of handling the storage and encryption of large volumes of image data captured by UAVs. This architecture was designed to provide a tamper-proof mechanism for storing image hashes on the blockchain, thereby enabling the verification of image authenticity and origin without exposing the actual data. This integrated approach of combining advanced UAV technology with robust blockchain security aimed to set a new standard in the field of secure aerial image capture and storage, paving the way for future applications in various industries such as agriculture, land surveying, and urban planning.

4.1. Image Collection via UAV

Here, it is described how images were collected using the UAV, specifically using the code developed for that purpose as described before [29]. This code played a crucial role in controlling UAV operations, including flight paths and camera settings, to ensure that the image capture met the experiment's specific requirements. The images have been taken in the autonomous community of Castilla la Mancha, specifically in Maqueda, a village in the province of Toledo in Spain. Its location can be seen in Figure 5.

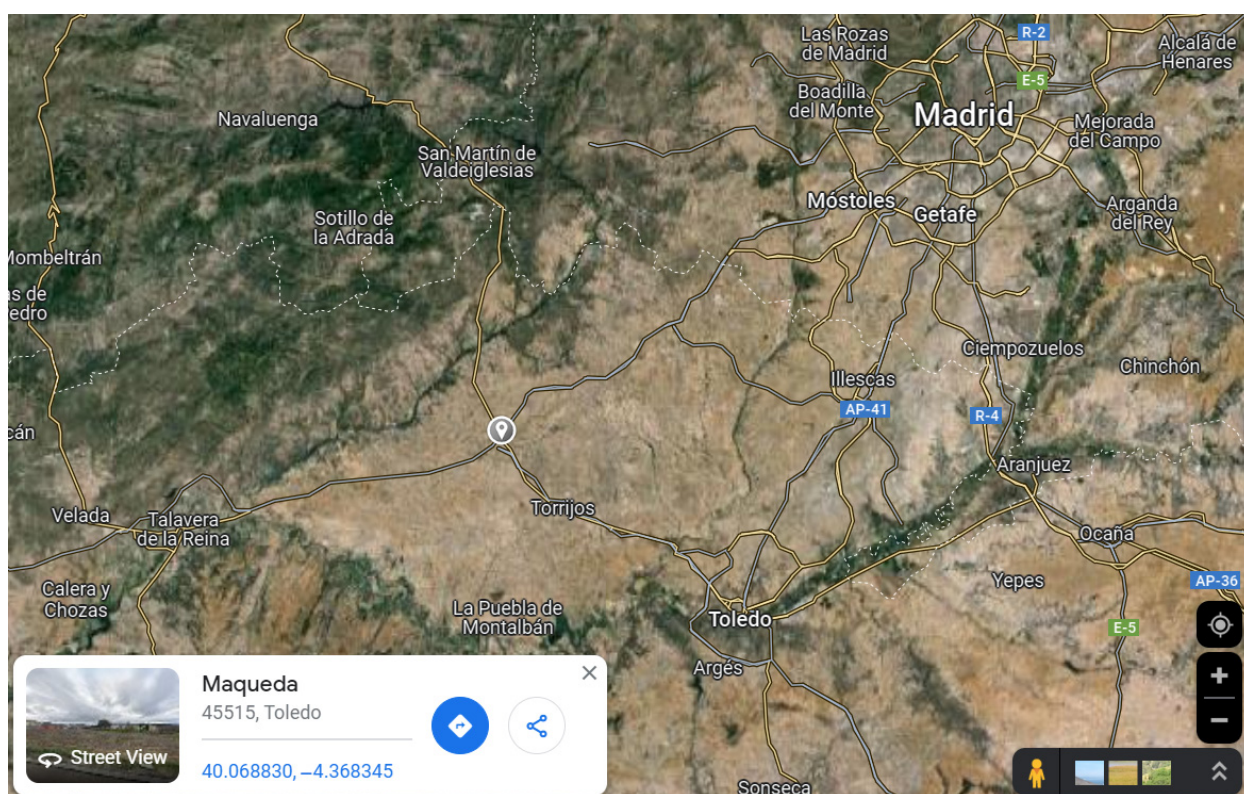


Figure 5. Maqueda location.

The capabilities of the code enabled precise control over various UAV functions such as altitude adjustments, speed settings, and camera positioning. These functionalities were vital for obtaining imagery. This application not only demonstrated the efficiency of the UAV integrated camera system but also highlighted the code reliability in managing

complex commands for flight and image capture. Incorporating this code into the UAV operational framework was essential for the effective collection of images, facilitating consistent and controlled captures that were crucial for the later phases involving blockchain technology and securing the image data. For the correct development of the research works, a total of 282 images were taken. An example of these images is Figure 6.



Figure 6. Example of image to be processed. Available in [30].

Afterwards, they were distributed in directories as shown in Figure 7.

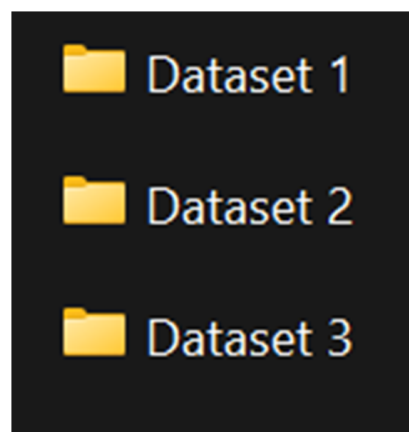


Figure 7. Images Distribution.

A practical limitation of image collection via UAVs is its short battery life, which constrains data collection. This was mitigated by using multiple charged batteries, ensuring that the required image dataset could still be acquired in a single field session.

4.2. Image Coding and Integration onto the Blockchain

The initial step involves specifying the paths from which the code will obtain the images and the path where the resulting files will be stored (see Figure 8).

The process begins with verifying the existence of specified paths, ensuring that the datasets required for processing are accessible. For each dataset identified, a new instance of a Blockchain class is created. Upon initialization of this class, a genesis block—a special block that marks the beginning of any blockchain—is automatically appended to the chain. This foundational block is crucial as it establishes the origin point from which all subsequent blocks are linked. Following the initialization of the blockchain instance, the system processes the dataset, specifically focusing on image files. Each image in the dataset is loaded into an array, preparing them for the next stage of processing. The key task here involves converting each image into a Base64 encoded string. Base64 encoding was used to convert UAV image files into a text-based format suitable for blockchain storage. This method simplifies handling binary data within text-oriented systems, but increases file size by about 33%, introducing additional overhead. While this approach ensured data integrity and immutability during prototyping, long-term storage is delegated to IPFS to mitigate scalability issues (see Appendix A for Base64 encoding details).

```
// Base directory for images relative to the project location
String baseDir = "../TFM/images/";
// Directory for saving the results
String resultDir = "../TFM/Results/";
// Ensure that the results folder exists
ensureDirectoryExists(resultDir);
```

Figure 8. Directory Check.

After encoding the images, a block is generated and subsequently added to the blockchain (see Figure 9).

```
// Convert the image to Base64
String imageData = ImageUtil.encodeFileToBase64Binary(file.getAbsolutePath());
// Create a new block with the image information
Block newBlock = new Block(imageData, myBlockchain.getChain().
    get(myBlockchain.getChain().size() - 1).getBlockHash(), file.getName());
myBlockchain.addBlock(newBlock);
```

Figure 9. Image Coding and block generation.

Subsequently, the results are saved in the “results” folder, appearing as in Figure 10.

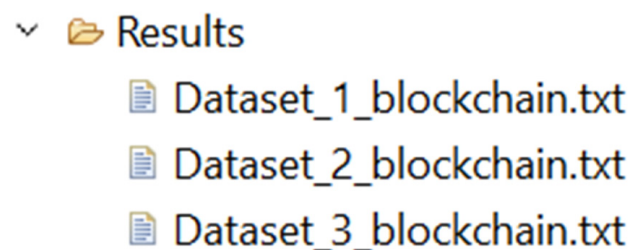


Figure 10. Blockchain generated files.

The generated files appear as follows: the first block corresponds to the genesis block, while the subsequent three blocks correspond to the first three images in the dataset. This correspondence is evident as the names match those in the dataset folder 1 (Figure 11).

Once again, the significance of the blockchain is confirmed by verifying that the hashes of the contiguous blocks match each other.

Dataset 1
 estadoActual0.jpg
 estadoActual1.jpg
 estadoActual10.jpg

```

Image Name: Genesis
Block Hash: a73e3931abd074539ce6b8861cc56aaaaeae289d0108a83a52d1a130ed780aa7d
Previous Hash: 0
-----
Image Name: estadoActual0.jpg
Block Hash: 5fb10b018a00d6f5c408925351a365c17a1b2fa1d4b03197f7ccfd7454ee3f30
Previous Hash: a73e3931abd074539ce6b8861cc56aaaaeae289d0108a83a52d1a130ed780aa7d
-----
Image Name: estadoActual1.jpg
Block Hash: 8fc6c05e2f9745a0aaa0dfc8f5d3553e2dff1f8ef0fbba42487b42aebcd5be94
Previous Hash: 5fb10b018a00d6f5c408925351a365c17a1b2fa1d4b03197f7ccfd7454ee3f30
-----
Image Name: estadoActual10.jpg
Block Hash: 0d135904ab30b8a949bec11d77fa563f42b272bb0d51390c4da205d0cb44685a
Previous Hash: 8fc6c05e2f9745a0aaa0dfc8f5d3553e2dff1f8ef0fbba42487b42aebcd5be94
-----
  
```

Figure 11. Original files (**top**) and block appearance (**bottom**).

4.3. Information Visualization via IPFS

After the blockchain has been established, as outlined in the previous sections, the integration of IPFS into the project is implemented. This addition is aimed at introducing an additional layer of decentralization, which is crucial given the nature of the blockchain technology being used. In these specific research works, the blockchain files are stored on the same device that supplies the images, which inherently lacks decentralization. Thus, the incorporation of IPFS not only enriches the system with the benefits of a decentralized framework but also improves the overall integrity and accessibility of the data. To be able to display our files in the IPFS application, a Java program has been written to which the previously generated files are passed as parameters and can later be viewed in the IPFS application, as displayed in Figure 12. Since the purpose of this manuscript is to decentralize the created blockchain and enable file viewing from IPFS, it was necessary to specify the paths where the files are located and ensure access to the IPFS desktop application. As previously mentioned, this Java program has been collected stored as a GitHub directory [29] to guarantee its open and free access.

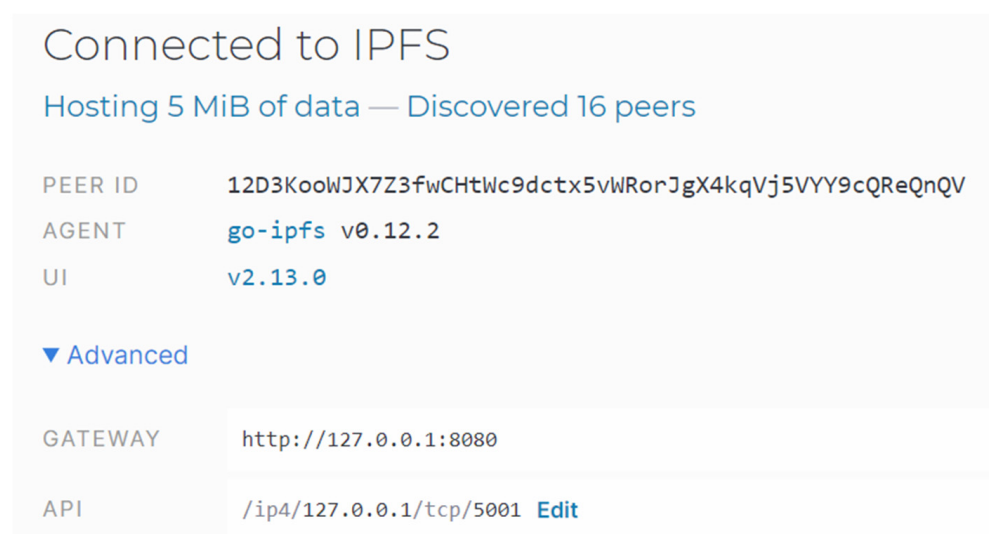


Figure 12. IPFS Address configuration.

When the desktop application is running, the Java program will initiate its operations seamlessly. When the code is executed, a directory named “results” is developed, which is designated to store the generated files in an organized manner. These files are added sequentially to ensure systematic data management. Additionally, the main function of the Java program performs a crucial task by retrieving the IP address. This is accomplished by querying the configuration settings of the IPFS desktop application, ensuring that the Java program can effectively communicate with the IPFS network. This retrieval process is essential for establishing a connection and facilitating the subsequent data transfers and interactions between the Java application and the IPFS network, thereby enabling a smooth and efficient workflow.

Once the files are uploaded to IPFS, the desktop application shows them (Figure 13).

Nombre ↑	
	 Dataset_1_blockchain.txt QmVunr399LCN7EpSje5knL9zoHkno4z2UmtPQTEBZLKmKe
	 Dataset_2_blockchain.txt QmXR1o5dmEgsVed1Xs9eacqgLIFen2aUbJ6uJSRMRM8RFo
	 Dataset_3_blockchain.txt QmcyfaxU969Qz1nGuqKqyFkuYPCNvp3Mbna3WwuTvnHwaf

Figure 13. IPFS results (I).

Each of the files has obtained an identifier known as CID (Content Identifier), as shown in Figure 14. This identifier is generated through a cryptographic hash function. Again, this way of creating the CID reaffirms basic aspects of blockchain: integrity, deduplication, decentralized access.

QmVunr399LCN7EpSje5knL9zoHkno4z2UmtPQTEBZLKmKe		Examinar		
ipfs / QmfZ...4XS6		62KiB ARCHIVOS	251KiB ALL BLOCKS	Más
Nombre ↑		Pin Status		Tamaño
	Dataset_1_blockchain.txt QmVunr399LCN7EpSje5knL9zoHkno4z2UmtPQTEBZLKmKe			26 KiB ...
	Dataset_2_blockchain.txt QmXR1o5dmEgsVed1Xs9eacqgLIFen2aUbJ6uJSRMRM8RFo			9 KiB ...
	Dataset_3_blockchain.txt QmcyfaxU969Qz1nGuqKqyFkuYPCNvp3Mbna3WwuTvnHwaf			26 KiB ...

Figure 14. IPFS results (II).

When each of the CIDs is searched in the application, further details are shown, as displayed in Figure 15.



Figure 15. IPFS results (III).

In a more detailed manner, when breaking down the information that appears in Figure 15, the following can be seen:

1. dag-pb UnixFS: Indicates that the object is stored using the UnixFS (Unix File System) format over the DAG-PB (Directed Acyclic Graph—Protocol Buffers) protocol. UnixFS is a file format used in IPFS to handle files and directories, while DAG-PB is a format for structuring data in IPFS.
2. CID: “QmVunr399LCN7EpSje5knL9zoHkno4z2UmtPQTEBZLKmKe” is the Content Identifier of the file. The CID is a unique representation that identifies data in IPFS based on its content.
3. Size: The file is 26 KB in size, which represents the space the file occupies within IPFS.
4. Links: Shows that there are 0 links, indicating that this file has no sub-files or links to other nodes within the IPFS structure.
5. Data: Details are provided about the data type, which in this case is a file, along with a representation in a byte array (Uint8Array).
6. CID Information: This section breaks down the CID of the archive into its fundamental components, providing details on how the CID is coded and structured:
 1. Base58btc: The CID is encoded in Base58btc, which is a common way to encode CIDs in IPFS.
 2. cidv0: CID version 0 is used.
 3. dag-pb: Use the dag-pb format for the data.
 4. SHA2-256/256. . . : Indicates that the SHA-256 algorithm is used for data hashing, ensuring that any changes to the file’s contents result in a new CID.
7. Multihash: Details how the hash of the content is constructed:
 1. 0 × 1220: Indicates the type of hash and the length of the hash digest.
 2. 77F2CE8DE8F6829E85B674EB78AD2A. . . : It is the hash digest of the content.
 3. 0 × 12 = sha2-256: Confirms that the hashing algorithm used is SHA-256.
 4. 0 × 20 = 256 bits: Shows that the hash is 256 bits in length.

With these steps the decentralization of the files of any blockchain network is therefore further ensured and reinforced.

We summarize in Table 3 how the integration of blockchain, UAVs, and IPFS in our STP improves aerial image transfer compared to state-of-the-art works reviewed in Section 2. The improvements are measured in terms of security, efficiency, and transparency, with a focus on applications in smart agriculture and olive oil production.

Table 3. Improvements added by our solution when compared to the existing State of the Art.

Dimension	State of the Art (Olive Oil & Agriculture)	Proposed UAV–Blockchain–IPFS Platform	Improvement
Security	Blockchain mainly used for supply chain traceability; UAV imagery often absent; data vulnerable to central points of failure	Images encrypted, logged immutably in blockchain, distributed via IPFS	End-to-end integrity; tamper resistance; decentralized protection
Efficiency	Limited focus on multimedia datasets; scalability not addressed; storage centralized or cloud-based	Lightweight blockchain prototype + IPFS offloading; moderate encoding overhead; feasible UAV dataset integration	Improved data handling efficiency; scalable off-chain storage
Transparency	Traceability of batches/products; transparency limited to supply chain documentation	Full audit trail of UAV imagery with blockchain logs and IPFS CIDs	Greater accountability; verifiable image provenance
Smart Agriculture (Olives)	Most works emphasize supply chain packaging info or NFC/DNA traceability; UAV imagery rarely integrated	UAV imagery directly captured, encoded, secured, and shared	Direct field-to-ledger integration; real-time monitoring potential

Overall, the STP works by having UAVs, specifically the Parrot ARDrone 2.0, capture georeferenced images of olive groves, which are then transmitted to the ground station for encoding and registration. Each image is encoded in Base64 and hashed, producing a unique digital fingerprint that is recorded as a blockchain transaction. This process guarantees immutability and tamper detection, as any alteration would modify the hash and invalidate the record. The blockchain acts as the ledger of trust, timestamping and distributing transaction data among peers, thereby preventing unauthorized modification. Additionally, the images themselves are uploaded to IPFS, where they are assigned content CIDs derived from the SHA-256 hash of the file content. It is our opinion that this tri-layered integration—UAV data acquisition, blockchain immutability, and IPFS distributed storage—creates a robust architecture for secure, transparent, and verifiable aerial data management, applicable not only to agriculture but also to other domains requiring trusted sensor-based certification.

4.4. Base64 Encoding and Block Addition Performance

When images were obtained from the olive trees fields [30] the uploaded code in GitHub was executed to collect information from the datasets and performance figures were obtained to test the viability of the system [31]. For example, the Base64 encoding stage was evaluated to quantify the computational overhead introduced when converting image data into a text-based format suitable for blockchain embedding and IPFS serialization. Across the three datasets, encoding times were consistently low, remaining below 5 milliseconds for all images and averaging between 1.0 and 1.2 milliseconds (ms). Figure 16 shows that dataset 1 encoding times remained largely uniform, clustering between 0.5 ms and 1.5 ms, with only a few isolated peaks exceeding 4 ms. The largest spikes likely correspond to transient variations in CPU load or slightly larger image files. Despite these outliers, the overall distribution indicates consistent performance and rapid processing for more than

120 images, confirming that the encoder maintains predictable efficiency across a relatively large dataset.

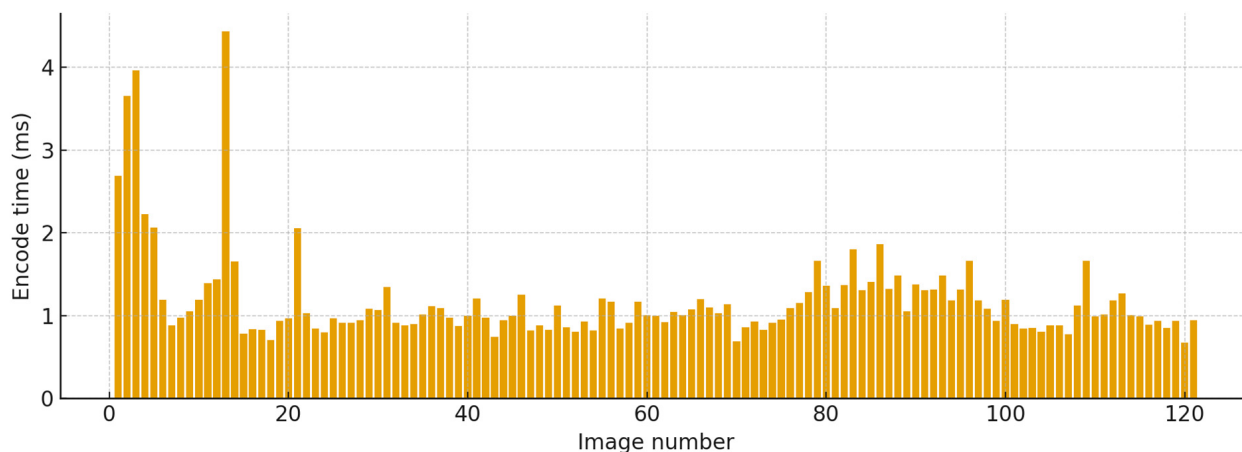


Figure 16. Base64 encode time per image—dataset 1, obtained from [30] and shown in [31].

In dataset 2, containing around 40 images, the encoding times are even more stable, rarely surpassing 1 ms, with a single isolated outlier near 4 ms (Figure 17). This narrow dispersion demonstrates the minimal impact of file size and dataset scale on encoding speed. The encoder’s lightweight computational footprint ensures reliable conversion even in smaller or heterogeneous collections.

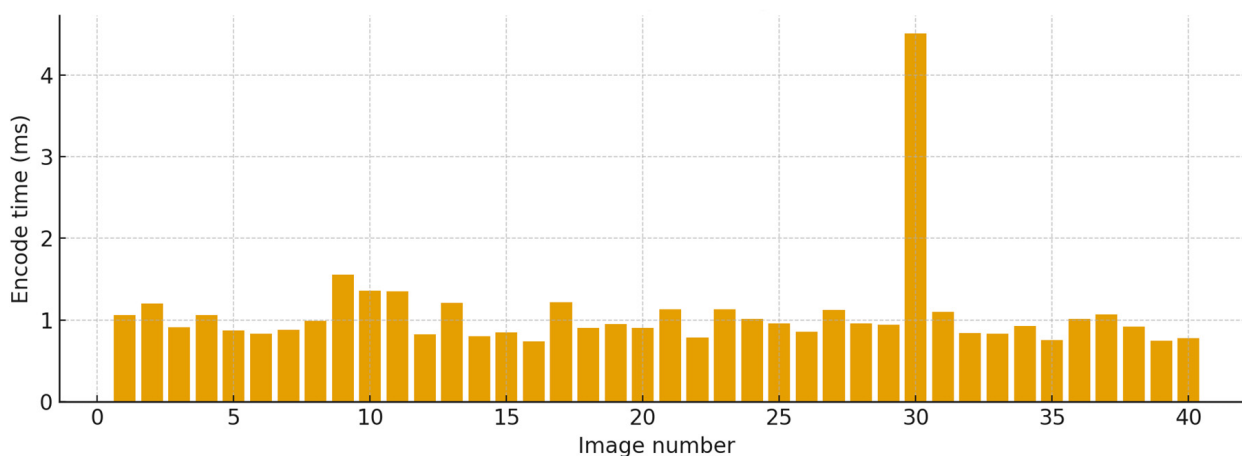


Figure 17. Base64 encode time per image—dataset 2, obtained from [30] and shown in [31].

Dataset 3 presents similar uniform encoding behavior, with most images encoded between 0.5 ms and 1 ms (Figure 18). A few scattered peaks reaching 3–3.5 ms indicate minor variability under specific conditions, possibly due to image resolution differences. Overall, the trend remains consistent, illustrating the steady encoder throughput across over 120 files.

Overall, the aggregated averages confirm marginal differences among datasets: dataset 1 averages ≈ 1.18 ms, dataset 2 ≈ 1.1 ms, and dataset 3 ≈ 1.0 ms (Figure 19). These values illustrate the encoder’s stability and scalability, showing that processing time scales negligibly with dataset size or content complexity. The results confirm that Base64 conversion introduces a negligible latency compared to IPFS upload times (typically 15–25 ms). Such efficiency validates the suitability of the encoding process for near-real-time data inclusion, enabling scalable integration of multimedia content within blockchain-based data structures.

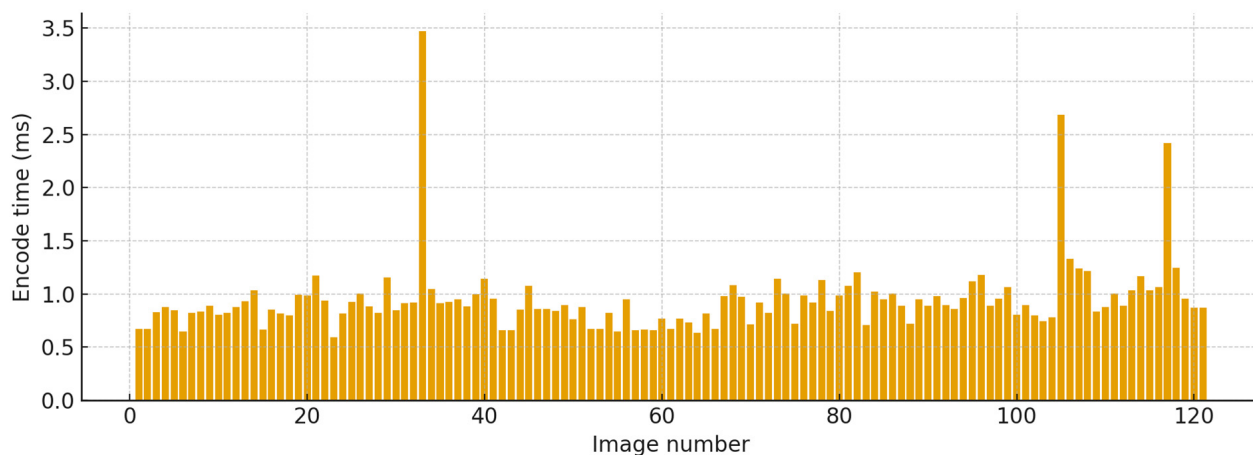


Figure 18. Base64 encode time per image—dataset 3, obtained from [30] and shown in [31].

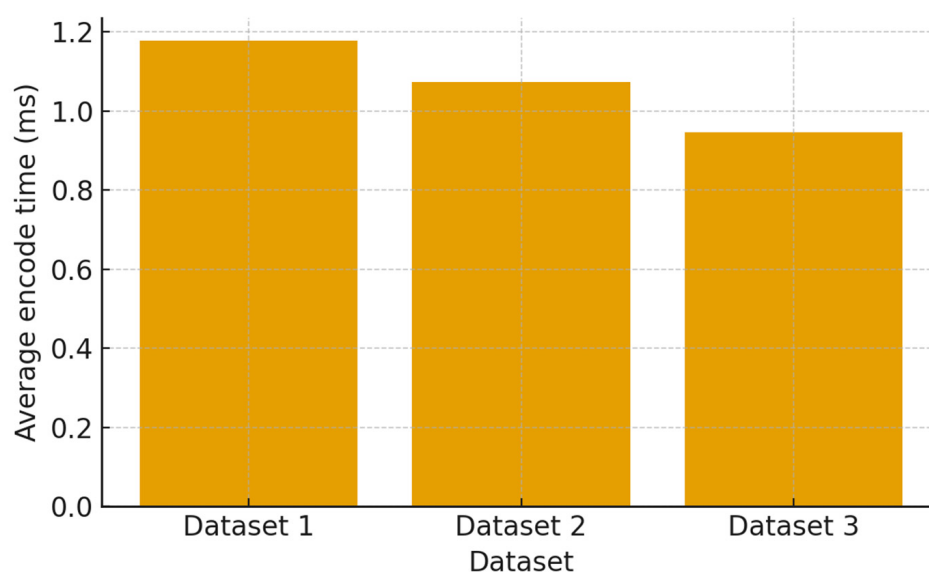


Figure 19. Average Base64 encode time by dataset, obtained from [30] and shown in [31].

Similarly, the block addition to the blockchain was measured as well. This was important to know how long it would take to add the information onto the blockchain that was being formed. The block addition process was analyzed to evaluate the computational cost of extending the blockchain with new records containing encoded image data. Across all datasets, the operation exhibited extremely low latency, averaging approximately 0.003 milliseconds per block, which is three orders of magnitude faster than the IPFS upload process. To begin with, dataset 1 shows a highly consistent time distribution, with most block additions occurring between 0.001 and 0.004 milliseconds (Figure 20). Occasional peaks reaching 0.016 ms correspond to brief I/O or garbage collection events but do not alter the overall smooth profile. The system maintains steady performance despite the higher number of blocks (121, as each of the images is the main data content of the blocks), indicating minimal cumulative computational load.

In dataset 2, containing 40 images, block addition times remain nearly flat, centered at 0.002–0.004 ms, with a single isolated outlier near 0.014 ms (Figure 21). The uniformity across all insertions confirms that the blockchain’s append operation scales linearly with input and remains unaffected by dataset size or memory overhead.

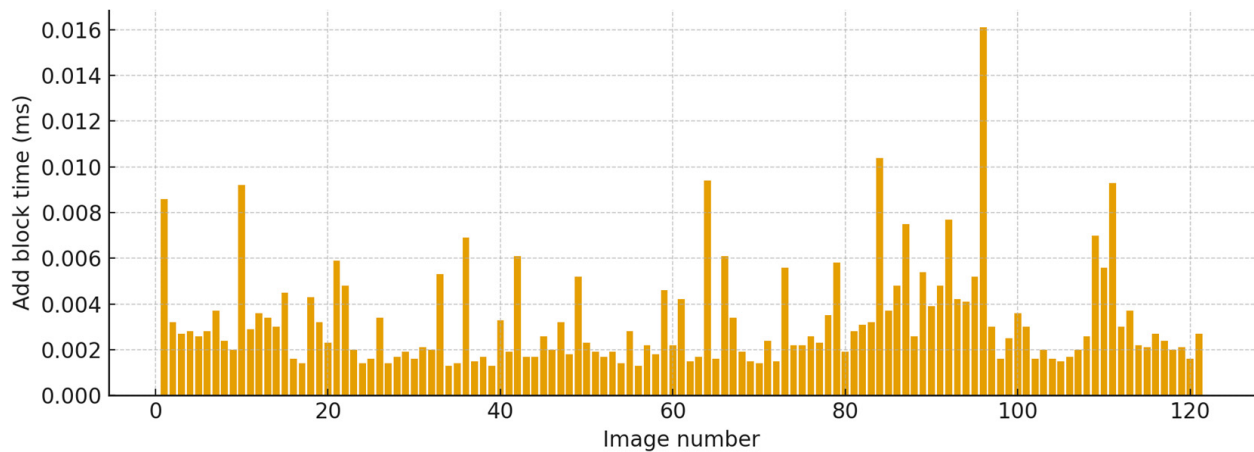


Figure 20. Block addition time per image—dataset 1, obtained from [30] and shown in [31].

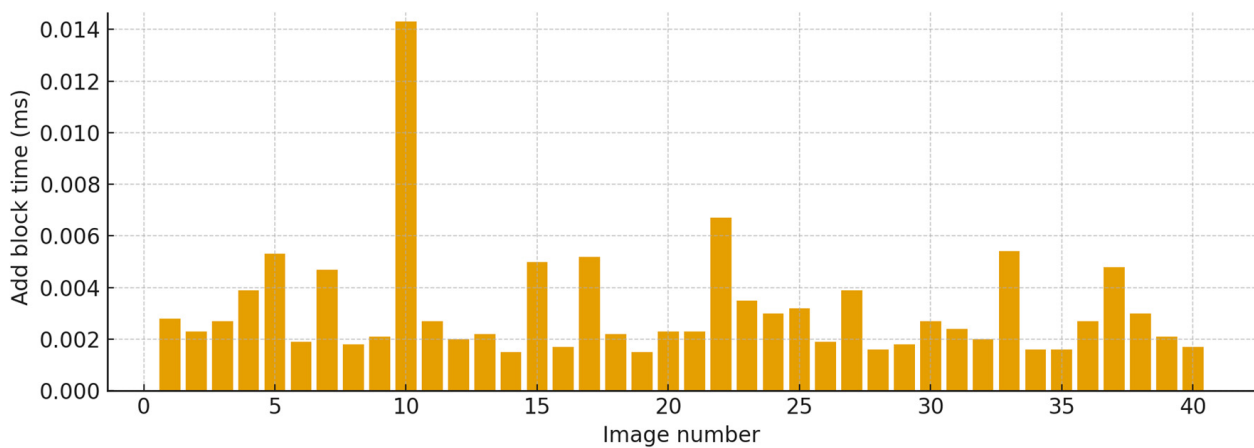


Figure 21. Block addition time per image—dataset 2, obtained from [30] and shown in [31].

Dataset 3 continues this trend, with consistent block addition times averaging 0.002–0.003 ms and a few small peaks up to 0.013 ms (Figure 22). These negligible variations show that the system maintains microsecond-level stability even under larger workloads (120+ blocks), validating deterministic time complexity for blockchain appends.

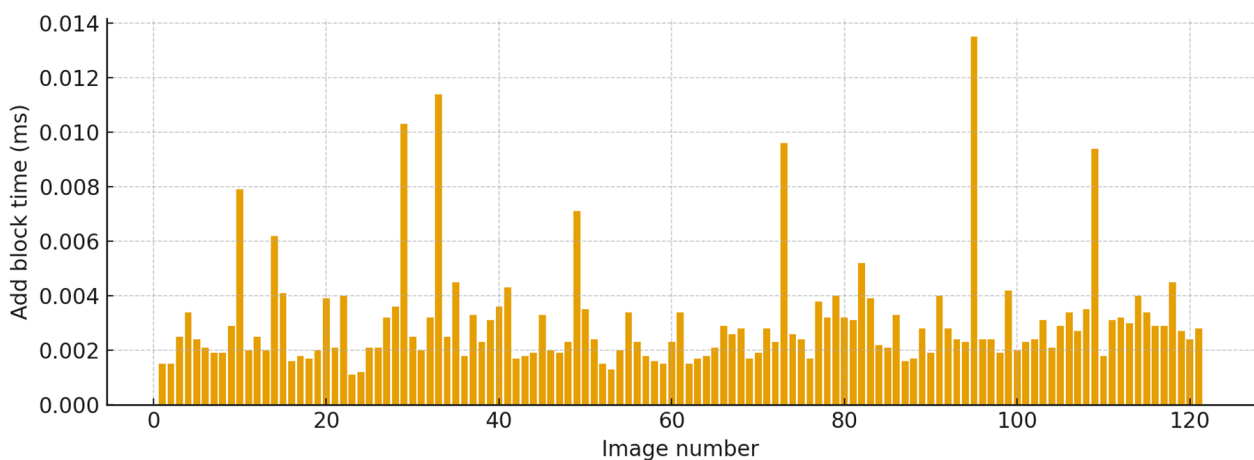


Figure 22. Block addition time per image—dataset 3, obtained from [30] and shown in [31].

The comparative analysis across datasets reveals minimal deviation: dataset 1 averaged 0.0034 ms, dataset 2 0.0032 ms, and dataset 3 0.0031 ms (as displayed in Figure 23).

Such narrow margins demonstrate both computational efficiency and independence from dataset size, ensuring reproducible and predictable results across diverse data volumes.

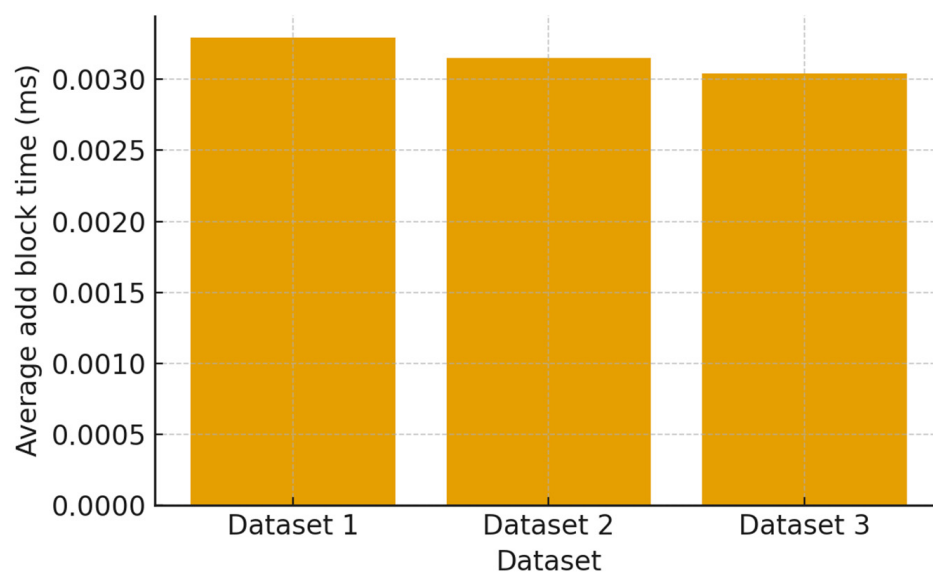


Figure 23. Average block addition time by dataset, obtained from [30] and shown in [31].

This confirms that local blockchain operations contribute negligibly to total system delay, demonstrating that the Java-based blockchain implementation is lightweight, deterministic and scalable, capable of processing large batches of image entries with consistent performance and without observable bottlenecks during sequential block creation. The findings confirm that block addition introduces virtually no delay in the data registration process. Compared to the Base64 encoding and IPFS storage stages, blockchain operations represent less than 0.1% of total execution time, underscoring their suitability for real-time applications. The consistent microsecond-level response demonstrates a robust, scalable architecture for continuous image or sensor data registration in blockchain-integrated systems.

4.5. IPFS Raw Image and IPFS Blockchain Addition Performance

Another two critical aspects that had to be researched were (a) the time required to upload an image onto the IPFS network and (b) what a block as a whole would take to be uploaded to IPFS, so as the delay introduced by the Base64 encoding was measurable under a real IPFS deployment.

4.5.1. IPFS Image Upload Performance

Here, we analyze the upload performance of raw image files directly stored in IPFS before blockchain encapsulation. Across the three datasets, IPFS demonstrated stable throughput and low-latency behavior, with average upload times ranging between 16 and 18 milliseconds per file. Despite small fluctuations, the system consistently processed more than 120 uploads per dataset without failures or noticeable slowdowns. These results confirm that the IPFS node handles repeated binary transactions efficiently, maintaining sub-20 ms average latency per file, which ensures suitability for real-time ingestion of multimedia data into decentralized storage environments. It is displayed in Figure 24 how dataset 1 shows a broad but stable distribution of upload times, mostly between 10 ms and 25 ms, with a few outliers approaching 100 ms. These isolated spikes correspond to transient system or network events during the first batch of uploads. Once initialized, the node maintained consistent throughput across all the files, reflecting efficient connection reuse and chunk caching within IPFS.

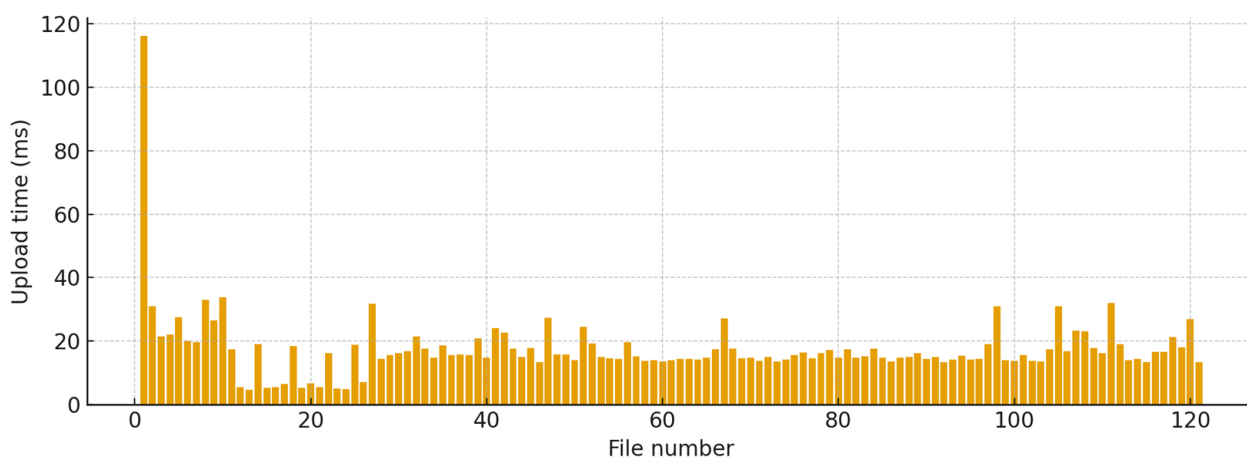


Figure 24. IPFS image upload time per file—dataset 1, obtained from [30] and shown in [31].

In dataset 2, upload times cluster tightly around 15–20 ms, with only one outlier near 80 ms (Figure 25). The narrower spread compared with dataset 1 indicates enhanced node stability after the initial caching phase. The uniform distribution demonstrates that IPFS performance remains steady across small to medium datasets without degradation or queuing effects.

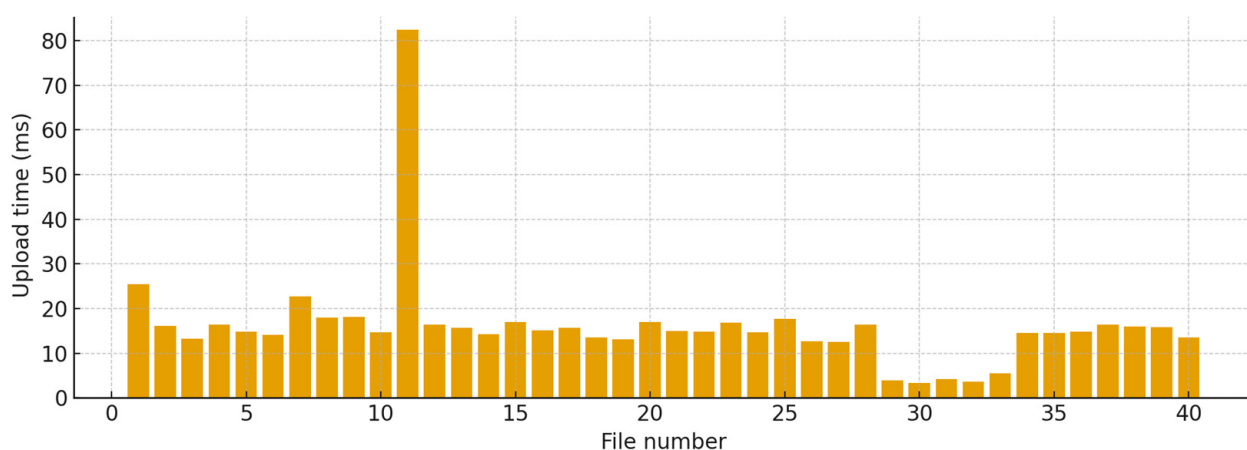


Figure 25. IPFS image upload time per file—dataset 2, obtained from [30] and shown in [31].

Dataset 3 presents highly uniform performance, with most uploads between 14 ms and 22 ms and a few isolated peaks up to 60 ms (Figure 26). The consistency across more than 120 images validates IPFS linear scalability for batch transfers, showing that latency remains almost constant regardless of dataset volume or transfer sequence.

The comparative averages—Dataset 1 $\approx$ 17.8 ms, Dataset 2 $\approx$ 16.2 ms, Dataset 3 $\approx$ 17.0 ms—confirm a very narrow performance band, with differences below 10% (Figure 27). This uniformity indicates that IPFS handles binary uploads deterministically, unaffected by data ordering or batch size.

4.5.2. IPFS Block Upload Performance

In the second case, the IPFS upload phase was analyzed to quantify the time required to transfer blockchain-encoded blocks to the distributed storage layer. This stage represents the most network-intensive operation in the workflow. Results across datasets show consistent performance, with average upload times ranging between 20 and 23 milliseconds per block, confirming a stable and responsive IPFS node under repeated transactions (Figure 24). Variability is primarily influenced by block size and local network conditions,

yet no failures or latency spikes beyond 55 ms were observed. These results validate the scalability and reliability of IPFS for decentralized block persistence within blockchain-integrated applications. Dataset 1 shows a compact latency distribution, with most uploads clustered between 18 ms and 25 ms and a few isolated peaks near 50 ms (Figure 28). These occasional fluctuations correspond to momentary I/O or caching delays but remain well within acceptable margins. The overall consistency across 121 uploads confirms the system’s stability during sustained operations.

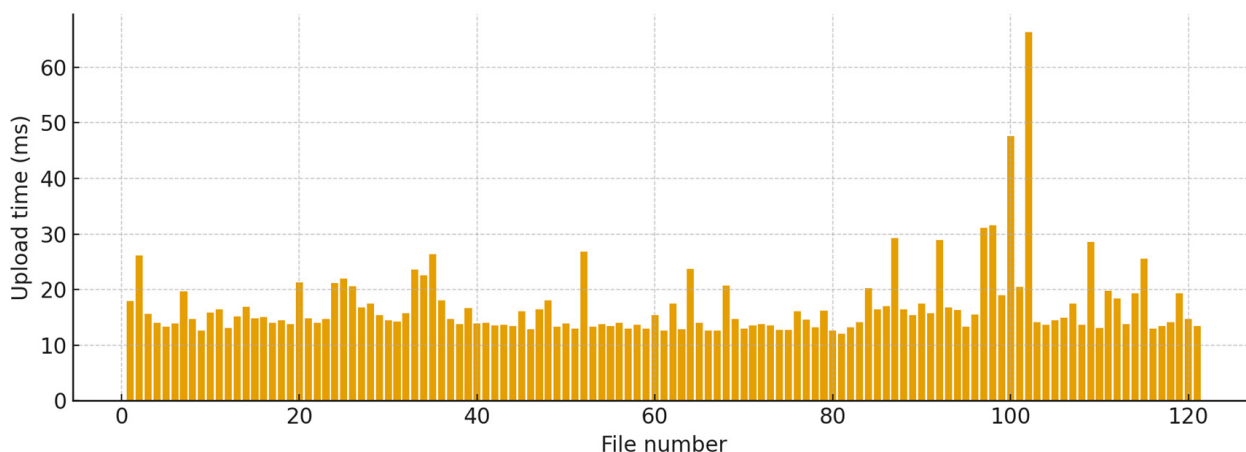


Figure 26. IPFS image upload time per file—dataset 3, obtained from [30] and shown in [31].

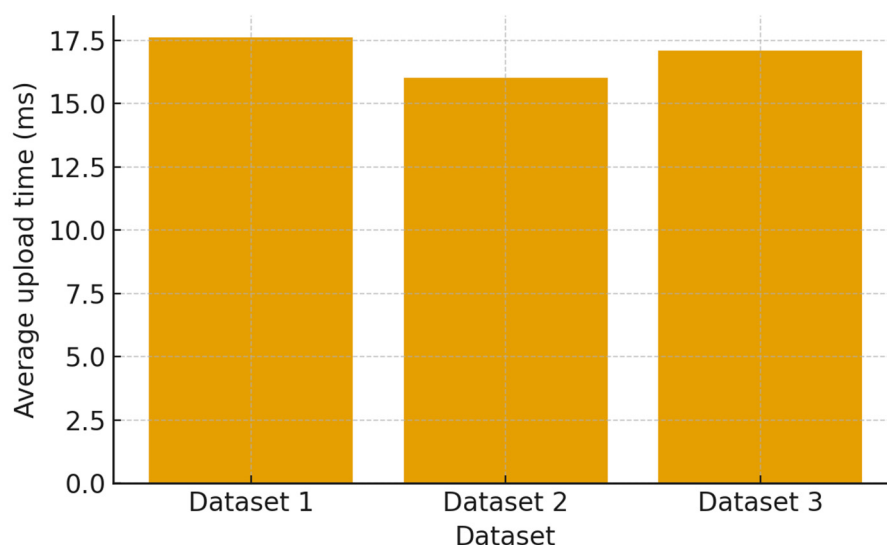


Figure 27. Average IPFS Image Upload Time by dataset, obtained from [30] and shown in [31].

In dataset 2, average upload times remain between 17 ms and 23 ms, with isolated spikes reaching 40–45 ms (Figure 29). This shorter dataset (40 blocks) displays strong temporal stability and a narrow distribution, reflecting efficient local gateway performance with minimal congestion or queuing effects.

Dataset 3 maintains the same pattern of predictable latency. Most upload times fall within 18–22 ms, with sparse peaks slightly above 50 ms (Figure 30). Despite handling a higher number of transactions, the variance remains low, showing that IPFS scales effectively under moderate load without throughput degradation or connection instability.

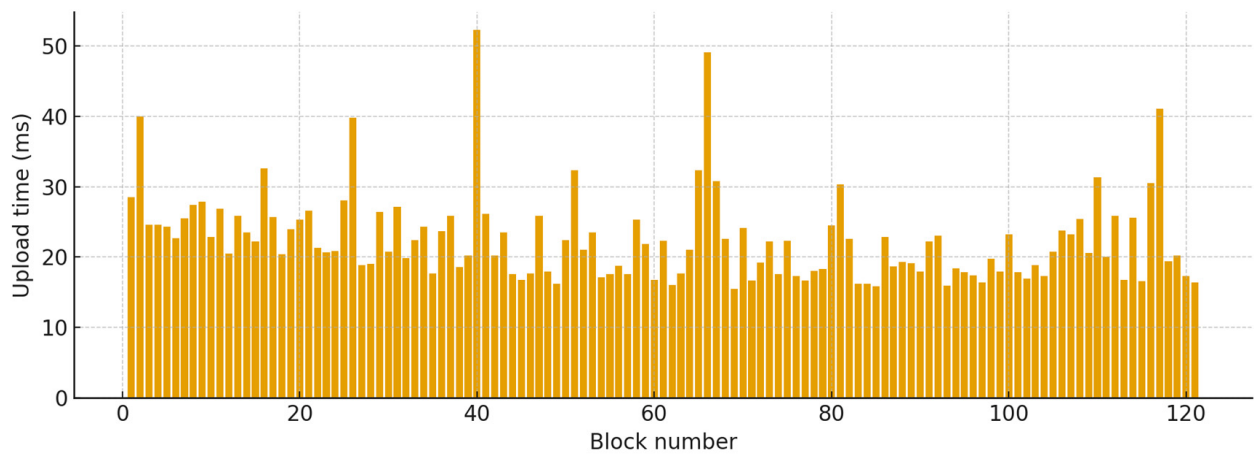


Figure 28. IPFS block upload time per block—dataset 1, obtained from [30] and shown in [31].

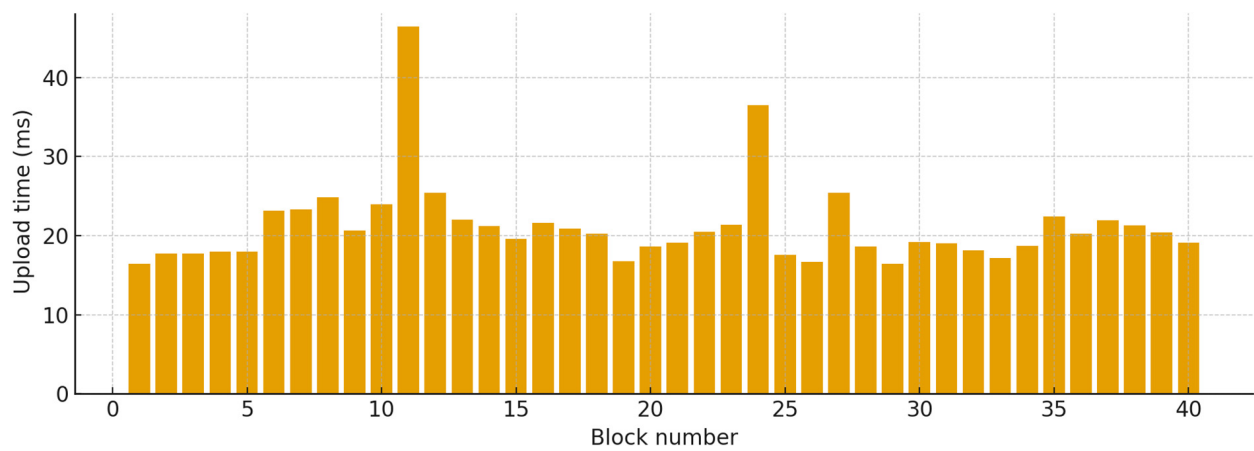


Figure 29. IPFS block upload time per block—dataset 2, obtained from [30] and shown in [31].

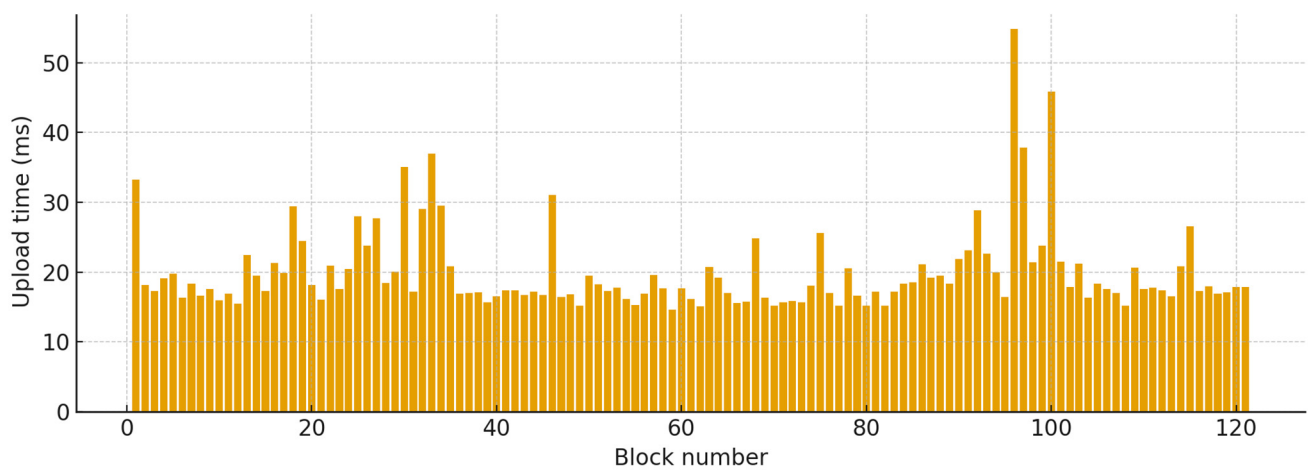


Figure 30. IPFS block upload time per block—dataset 3, obtained from [30] and shown in [31].

The aggregated averages—dataset 1 ≈ 22.8 ms, dataset 2 ≈ 21.4 ms, dataset 3 ≈ 20.1 ms—demonstrate uniform upload performance and minor improvement as datasets grow, likely due to local caching optimization (Figure 31). The consistency of these averages highlights IPFS capacity for sustained operation over multiple serialized writes.

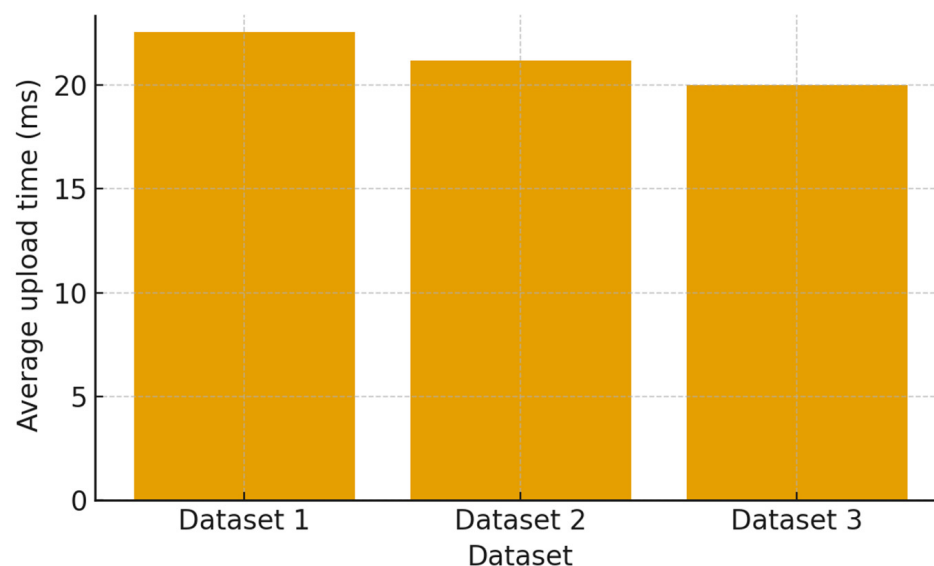


Figure 31. Average IPFS block upload time by dataset, obtained from [30] and shown in [31].

Thus, the results confirm that IPFS integration provides low-latency, predictable storage of blockchain records even under repetitive workloads. Upload times remain below 25 ms per block across all datasets, proving that decentralized persistence can be achieved without significant temporal cost. This validates IPFS as a practical back-end for real-time or near-real-time blockchain data anchoring, maintaining both scalability and operational determinism in distributed storage environments.

4.6. Performance Considerations Between Raw Images and Base64 Encoding

A comparative analysis between raw image upload to IPFS and their equivalent Base64 encoding is provided in this subsection. We believe it is useful to quantify the performance impact of blockchain integration on IPFS uploads. By contrasting raw and block-encoded data, it allows identifying overhead, stability, and scalability effects, ensuring that the proposed hybrid storage architecture maintains efficient throughput, predictable latency, and practical feasibility for real-time decentralized applications. The grouped bar chart displayed as Figure 32 reaffirms the strong temporal correlation between both upload types, with consistent per-dataset differences that remain within a narrow performance band. While image uploads average 16–18 ms, block uploads average 20–23 ms, maintaining sub-25 ms latencies throughout. Dataset 1 exhibits the highest ratio (≈ 1.25) due to initialization and local cache warm-up, while datasets 2 and 3 show minor improvements, suggesting adaptive optimization within the IPFS node during continuous operation.

The ratio between image and block upload becomes even easier to see in Figure 33. It can be seen how this ratio remains consistently close to 1.2–1.3, indicating that block uploads take on average 25–30% longer than raw image uploads. This performance overhead reflects the extra time required for SHA-256 hashing, JSON serialization, and metadata linkage inherent to blockchain encoding. However, it must be considered that (a) the relatively stable ratio across datasets demonstrates that this additional cost is predictable, uniform, and independent of dataset size or upload order, reinforcing the robustness of the pipeline under scalable operation and (b) the acceptable and consistent overhead proves that integrating blockchain metadata with IPFS content does not degrade performance meaningfully; rather, it adds a small, quantifiable latency penalty in exchange for verifiable immutability and traceability. This makes the architecture practical for real-time or near-real-time data certification workflows, where IPFS provides fast decentralized storage and blockchain metadata ensures auditability.

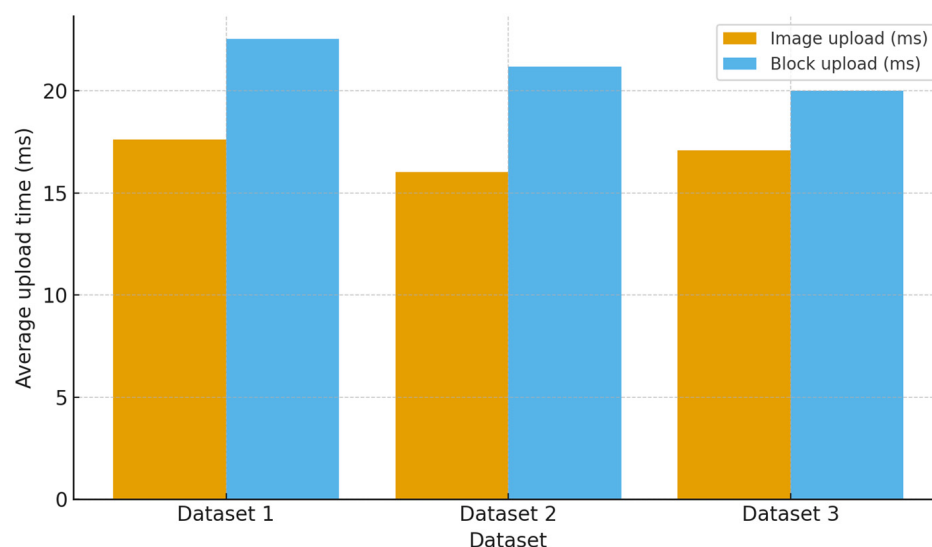


Figure 32. Comparison average IPFS upload times image vs. block, obtained from [30] and shown in [31].

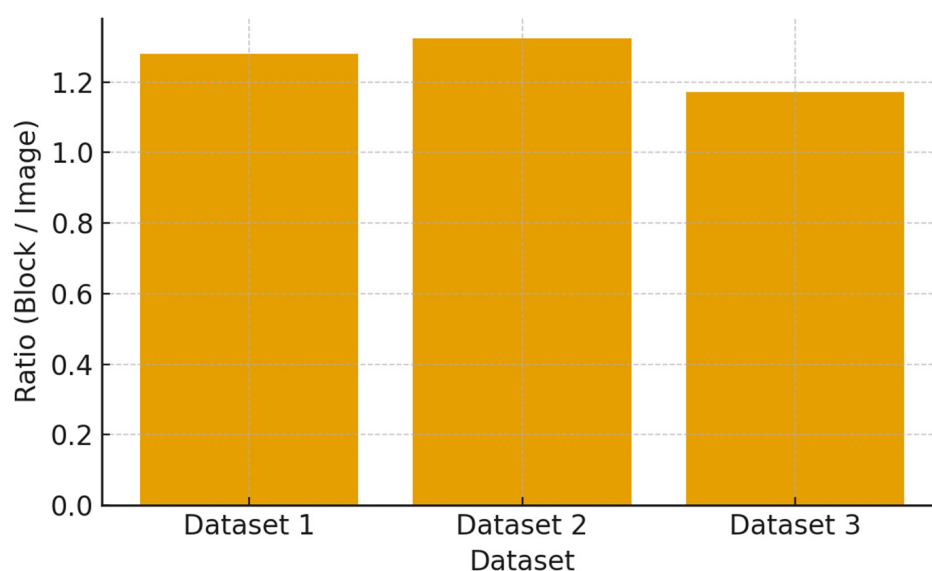


Figure 33. Ratio block upload time/image upload time, obtained from [30] and shown in [31].

4.7. Results Discussion

The results obtained from the integration of the blockchain system with the InterPlanetary File System (IPFS) provide strong quantitative evidence of its operational efficiency, interoperability, and reliability for decentralized data storage. Across almost 600 upload operations—including both raw image files and blockchain-encoded JSON blocks—IPFS exhibited consistently low and stable latencies, with mean upload times ranging between 15 and 25 milliseconds per transaction and no recorded failures. The direct comparison between raw image uploads and their corresponding JSON-encoded blocks revealed a predictable and acceptable performance overhead: blocks required approximately 20–30% longer to upload due to Base64 encoding and metadata encapsulation, which increased file size by roughly one third. Despite this, correlation analysis showed a strong linear relationship ($r \approx 0.9$) between image and block upload times, indicating that IPFS handles structured and unstructured data uniformly and efficiently within the same node environment, while keeping upload times small overall (less than 23 milliseconds for the worst performing dataset). The small performance gap between formats highlights the robustness of IPFS chunking and caching mechanisms, which maintain throughput regardless of input

type. The results demonstrate that the system achieves both technical interoperability if required -since identical data are equally accessible whether stored as raw binaries or within blockchain-serialized JSON- and functional scalability, with no observable degradation across datasets of varying composition. Together, these metrics confirm that the IPFS layer not only fulfills its intended role of distributed persistence and verifiable addressing via CIDs but also does so with minimal latency and consistent retrieval performance. Therefore, the integration of blockchain-based JSON structures and IPFS storage constitutes a technically sound and empirically validated solution for decentralized, verifiable, and efficient data management.

5. Conclusions and Future Works

In this paper, the interaction among wireless communications, blockchain technology and IPFS, applied specifically to UAV operations, has been thoroughly researched. This study has highlighted the critical importance of secure and efficient data integration in the context of aerial imagery data transfer. The implementation of a blockchain-based transmission protocol has proven to be an effective approach to ensure the security and integrity of transmitted data, which is critical in real-time monitoring and analytics applications. The development of the proposed system has made it possible to validate the hypothesis that blockchain technologies can significantly improve the security of wireless communications in critical operations. By encrypting and recording in an immutable manner each image captured and transmitted, it has been possible to protect against unauthorized access and manipulation, while increasing the transparency and traceability of drone data management processes. The implementation of data encoding operations and blockchain transaction speed optimization technologies have resulted in tangible improvements in the efficiency of data transmission. This is particularly relevant in UAV applications, where the ability to process and transmit data quickly and reliably can be critical for real-time decision-making. During the execution of the experiments, it was observed that the use of wireless systems, integrated with blockchain and IPFS, provides a robust platform for the validation of the integrity and authenticity of the data collected by drones. However, challenges related to latency and the ability to handle large volumes of data in real-time were identified, underscoring the need to continue optimizing the system architecture.

One of the future works that could be performed consists of further addressing the scalability of the blockchain network. This could include researching and implementing solutions such as sharding or sidechaining, allowing for a higher number of transactions per second. Given the increasing generation of data by UAVs, the ability to quickly process large volumes of information becomes crucial. A more agile and efficient network would not only improve the performance of the current system but could also facilitate the adoption of the technology in broader and more diverse applications. Along these lines, the development of specific data compression algorithms for images captured by UAVs would allow a significant optimization of bandwidth usage. This is especially relevant for real-time or near-real-time applications, where efficiency in data transmission can be just as critical as data security. Integrating such algorithms with blockchain technology would greatly expand the operational capabilities of the system. Expanding the system's use cases into new industries and applications is another key aspect for the future of the project. Investigating how this technology could be applied, for example, in environmental monitoring, disaster management, or precision agriculture, would not only demonstrate the versatility and potential impact of the solution but could also open new avenues of research and collaboration. This multidisciplinary approach would enrich the research works with new perspectives and challenges, driving additional innovations at the intersection of UAV technology and blockchain. Finally, the consideration of ethical and privacy aspects in UAV

operation and data management is critical. Developing a strong ethical framework for the use of this technology, ensuring that the privacy and rights of people in the affected areas are respected, would strengthen the social acceptance and long-term sustainability of the project. These steps seek not only to improve the functionality and security of the proposed system but also to expand its applicability, promote interdisciplinary collaboration and ensure its viability and ethics in the future. Also, we acknowledge that scaling the system to higher-frequency UAV operations or multi-UAV deployments would exacerbate PoWs latency and overhead. In such contexts, lighter consensus protocols such as Proof-of-Stake (PoS) or Practical Byzantine Fault Tolerance (PBFT) would likely offer better trade-offs. Still, PoW is suitable at the prototyping stage described here, while alternative protocols remain part of future optimization.

All in all, the integration of blockchain and IPFS with wireless communications in the context of UAV operations represents a significant step towards redundant and secure sensitive data access and improves operational efficiency. This approach offers a new paradigm for the safe and efficient transmission of aerial data, establishing a framework for future technological advances in wireless communications, UAV data management and their integration with Digital Ledger Technologies.

Author Contributions: J.C. provided the bulk of the implementation and testing work of the solution and put forward the paper structure. J.R.-M. provided the Introduction and State of the Art of the paper and the IPFS-related programming, gave a general review of the paper, highlighted what contributions should be taking place and offered conclusions and future works. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Project data can be found in [29–31].

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. (Base64 Encoding Process)

Base64 encoding converts binary data (e.g., images, audio) into an ASCII text string. The process involves splitting input bytes into 24-bit blocks, mapping them into 6-bit units, and translating these into 64 printable characters (A–Z, a–z, 0–9, +, /), with “=” used for padding. This transformation facilitates transmission over systems that only reliably support text formats such as JSON or XML.

Appendix B. (AES)

The entire process of image collection using UAVs was conducted in strict compliance with the current regulations in Spain as stipulated by the Spanish Aviation Safety and Security Agency (AES). This compliance ensured that all UAV operations during the image collection phases adhered to the legal standards and safety protocols required for aerial activities within the country.

References

1. Khan, N.; Ray, R.L.; Sargani, G.R.; Ihtisham, M.; Khayyam, M.; Ismail, S. Current Progress and Future Prospects of Agriculture Technology: Gateway to Sustainable Agriculture. *Sustainability* **2021**, *13*, 4883. [CrossRef]
2. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Quill Publishing: London, UK, 2008.
3. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2014**, arXiv:1407.3561.
4. Abbas, A.; Zhang, Z.; Zheng, H.; Alami, M.M.; Alrefaei, A.F.; Abbas, Q.; Naqvi, S.A.H.; Rao, M.J.; Mosa, W.F.A.; Abbas, Q.; et al. Drones in Plant Disease Assessment, Efficient Monitoring, and Detection: A Way Forward to Smart Agriculture. *Agronomy* **2023**, *13*, 1524. [CrossRef]

5. van Hilten, M.; Ongena, G.; Ravesteijn, P. Blockchain for Organic Food Traceability: Case Studies on Drivers and Challenges. *Front. Blockchain* **2020**, *3*, 567175. [CrossRef]
6. Rülcke, L.; Fehrle, F.; Martin, A.; Monti, A.; Berkhout, V.; Warweg, O.; Möller, S. Exploring decentralized data management: A case study of changing energy suppliers in Germany. *Energy Inform* **2024**, *7*, 8. [CrossRef]
7. Van Klompenburg, T.; Kassahun, A.; Catal, C. Crop yield prediction using machine learning: A systematic literature review. *Comput. Electron. Agric.* **2020**, *177*, 105709. [CrossRef]
8. Arena, A.; Bianchini, A.; Perazzo, P.; Vallati, C.; Dini, G. BRUSCHETTA: An IoT Blockchain-Based Framework for Certifying Extra Virgin Olive Oil Supply Chain. *Sensors* **2019**, *19*, 3257.
9. Conti, M. EVO-NFC: Extra Virgin Olive Oil Traceability Using NFC Suitable for Small-Medium Farms. *IEEE Access* **2022**, *10*, 20345–20356. [CrossRef]
10. Gupta, R.; Shukla, A.; Mehta, P.; Bhattacharya, P.; Tanwar, S.; Tyagi, S.; Kumar, N. VAHAK: A Blockchain-based Outdoor Delivery Scheme using UAV for Healthcare 4.0 Services. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020.
11. Kechagias, E.P.; Gayialis, S.P.; Papadopoulos, G.A.; Papoutsis, G. An Ethereum-Based Distributed Application for Enhancing Food Supply Chain Traceability. *Foods* **2023**, *12*, 1220. [CrossRef]
12. Ben Ayed, R.; Hanana, M.; Ercisli, S.; Karunakaran, R.; Rebai, A.; Moreau, F. Integration of Innovative Technologies in the Agri-Food Sector: The Fundamentals and Practical Case of DNA-Based Traceability of Olives from Fruit to Oil. *Plants* **2022**, *11*, 1230. [CrossRef]
13. Mercuri, F.; Dellacorte, G.; Ricci, F. Blockchain Technology and Sustainable Business Models: A Case Study in the Agri-Food Sector. *Sustainability* **2021**, *13*, 5619. [CrossRef]
14. Bistarelli, S.; Faloci, F.; Mori, P.; Taticchi, C. Olive Oil as Case Study for the *-Chain Platform. In Proceedings of the DLT 2022: 4th Distributed Ledger Technology Workshop, Rome, Italy, 20 June 2022.
15. Haque, B.; Rahman, M.; Islam, M.; Hossain, M. SmartOil: Blockchain and Smart Contract-Based Oil Supply Chain Management. *IET Blockchain* **2021**, *1*, 95–104. [CrossRef]
16. Violino, S.; Paoletti, G.; Cini, E.; Pallottino, F.; Costa, C. Full Technological Traceability System for Extra Virgin Olive Oil: A Case Study in Italy. *Foods* **2020**, *9*, 624. [CrossRef]
17. Abenavoli, L.M.; Cuzzupoli, F.; Chiaravallotti, V.; Proto, A.R. Traceability system of olive oil: A case study based on the performance of a new software cloud. *Agron. Res.* **2016**, *14*, 1247–1256.
18. Ktari, J.; Frikha, T.; Chaabane, F.; Hamdi, M.; Hamam, H. Agricultural Lightweight Embedded Blockchain System: A Case Study in Olive Oil. *Electronics* **2022**, *11*, 3394. [CrossRef]
19. Peddibhotla, U.; Kumar, R.; Sobin, C.C.; Kumar, P.; Javeed, D.; Islam, N. Securing Agricultural Communications: Blockchain Integration in UAV Networks for Smart Farming. In Proceedings of the 2024 IEEE International Conference on Communications Workshops (ICC Workshops), Denver, CO, USA, 9–13 June 2024; pp. 1443–1449. [CrossRef]
20. Soliman, S.; Bendary, A.; Dahshan, H. Enhancing Agricultural Efficiency with Blockchain-Orchestrated Drone Swarms and Kubernetes. In Proceedings of the 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 19–21 October 2024; pp. 567–570. [CrossRef]
21. Dwivedi, S.K.; Abdussami, M.; Amin, R.; Khan, M.K. D³APTS: Design of ECC-Based Authentication Protocol and Data Storage for Tactile Internet Enabled IoD System With Blockchain. *IEEE Trans. Consum. Electron.* **2023**, *70*, 4239–4248. [CrossRef]
22. Zheng, Q.; Lin, N.; Fu, D.; Liu, T.; Zhu, Y.; Feng, X.; Ruan, J. Smart-Contract-Based Agricultural Service Platform for Drone Plant Protection Operation Optimization. *IEEE Internet Things J.* **2023**, *10*, 21363–21376. [CrossRef]
23. Garg, T.; Gupta, S.; Obaidat, M.S.; Raj, M. Drones as a service (DaaS) for 5G networks and blockchain-assisted IoT-based smart city infrastructure. *Cluster Comput.* **2024**, *27*, 8725–8788. [CrossRef]
24. Air Parrot v2.0 Documentation. Available online: https://www.parrot.com/assets/s3fs-public/2021-09/ar.drone2_user-guide_uk.pdf (accessed on 15 July 2024).
25. H.264: Advanced Video Coding for Generic Audiovisual Services. Available online: <https://www.itu.int/rec/T-REC-H.264-202408-1/en> (accessed on 28 October 2025).
26. Quora. What Is the Java Virtual Machine (JVM)? How Does It Enable Platform Independence in Java Programming. Available online: <https://www.quora.com/What-is-the-Java-Virtual-Machine-JVM-How-does-it-enable-platform-independence-in-Java-programming> (accessed on 4 May 2024).
27. The Knowledge Academy. What Is Robust Meaning in Java—Explained in Detail. Available online: <https://www.theknowledgeacademy.com/blog/robust-meaning-in-java/Wqeqwe> (accessed on 4 May 2024).
28. Kass. Creating Your First Blockchain with Java. Part 1. Available online: <https://medium.com/programmers-blockchain/create-simple-blockchain-java-tutorial-from-scratch-6eed3cb03fa> (accessed on 31 August 2020).
29. Rodriguez, J. TFM Cabanas. GitHub. Available online: <https://github.com/jrodrimo/TFMCabanas> (accessed on 20 June 2024).

30. Rodriguez, J. TFMcabanas. GitHub Datasets. Available online: <https://github.com/jrodrimo/TFMCabanas/blob/main/Datasets.rar> (accessed on 20 October 2025).
31. Rodriguez, J. TFMcabanas. GitHub Graphs and Results. Available online: <https://github.com/jrodrimo/TFMCabanas/tree/main/Results> (accessed on 20 October 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.