ALFA BK UNIVERSITY
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY O F MATHEMATICS AND COMPUTER SCIENCE
International Scienti Conference

Alfa BK Univerzitet

UDK:004.6-022.59
DOI: 10.46793/AlfaTech1.2.39P
Original scientific paper

# Integrating OSINT with Hadoop Analytics for Rapid Person Identification in Smart-City Events

## Instructions for Authors

Nikola PETROVIĆ[1], Vojkan NIKOLIĆ[2].

**Abstract**: Apache Hadoop is a platform for storing, processing, and analyzing large amounts of data. In this paper, data are ingested into HDFS and queried using HiveQL, while MapReduce is applied for large-corpus text processing. For image analysis, convolutional neural networks (CNN) with OpenCV are used for object/face detection and matching. OSINT (Open-Source Intelligence) techniques collect images, videos, and text from publicly available sources and fuse them with camera streams to accelerate person identification in crowded events. We evaluate the system by measuring precision/recall, processing time, and overall throughput. We also note legal and ethical safeguards (public sources only, data minimization, audit logging). This article is an invited, extended version of our AlfaTech 2025 conference paper [1].
**Keywords**: Big Data, Apache Hadoop, OSINT, HDFS, HiveQL, OpenCV

## 1 INTRODUCTION

Smart-city initiatives seek to improve quality of life and public safety by leveraging modern sensing and analytics. Dense camera networks, however, generate heterogeneous, high-velocity data that exceed the capabilities of traditional systems. To address this, we explore how open-source big-data infrastructure and computer vision can be combined with open-source intelligence (OSINT) to support faster and more reliable person identification during crowded urban events.

In this work, we integrate the Apache Hadoop ecosystem—HDFS for storage, HiveQL for ad-hoc queries, and MapReduce for large-corpus text processing—with convolutional-neural-network (CNN)–based face and object analysis in OpenCV. Publicly available images, videos, and text collected via OSINT are fused with camera streams to expand candidate evidence while respecting legal and ethical constraints (public sources only, data minimization, audit logging).

**Contributions**. This invited journal extension of our AlfaTech 2025 paper [1] offers:

1. a formalized end-to-end workflow for smart-city incident triage (ingest → vision analysis → OSINT enrichment → Hadoop analytics);
2. practical guidance for data management and querying at scale (HDFS/HiveQL/MapReduce);
3. an evaluation plan covering precision/recall, processing time, and system throughput; and
4. an ethics/governance checklist for responsible use.

The remainder of the paper details related work, system architecture, the case scenario and data, evaluation methodology, ethical considerations, discussion and limitations, and concludes with directions for future research.

## 2 APACHE HADOOP PLATFORM AND BIG DATA PROCESSING SYSTEMS

"Big data" commonly refers to datasets whose volume, velocity, and variety (often extended with veracity and value) exceed the capabilities of traditional systems. Apache Hadoop is an open-source platform that enables cost-effective storage and distributed processing of such data on clusters of commodity machines.

Storage and access. The Hadoop Distributed File System (HDFS) provides fault-tolerant, distributed storage that accepts heterogeneous formats (text, images, video, sensor streams). Hadoop follows a schema-on-read paradigm: data are stored in their raw form and structured at query time. On top of HDFS, Hive offers a SQL-like layer (HiveQL) for analytical queries, while HBase is a column-oriented NoSQL store optimized for large tables and fast random reads/writes.

Processing model. Classic Hadoop relies on MapReduce, a programming model for parallel batch processing across nodes. The Map phase transforms input into key–value pairs; the Reduce phase aggregates results by key to produce final metrics or artifacts. Resource management is handled by YARN, allowing scalable, fault-tolerant execution of large jobs.

Near-real-time note. Although "real time" is often mentioned alongside big data, MapReduce itself is batch-oriented. Low-latency processing typically uses streaming frameworks (e.g., Kafka with Spark Structured Streaming or Flink) or interactive Hive variants. In this paper, Hadoop primarily supports HDFS storage and HiveQL/MapReduce analytics, while CNN/OpenCV components handle image/face detection and matching.

Why Hadoop here. This stack lets us retain and fuse OSINT artifacts (text/images/video) with computer-vision outputs in one place, issue ad-hoc queries over the combined corpus via Hive, and run wide batch computations with MapReduce when needed.

## 3 OSINT AND ARTIFICIAL INTELLIGENCE

Definition and scope. Open-Source Intelligence (OSINT) is the collection, analysis, and interpretation of information that is lawfully and publicly available. It is widely used in journalism, public safety, compliance, and research. While OSINT predates the web, its reach expanded dramatically with the growth of search engines, social media, and open data portals.

Public sources. Publicly accessible information includes both online and offline media: websites, search-engine results, social platforms, forums, video/photo sharing sites, broadcast and print media, public databases and registries, and open Web-GIS layers. In this work, OSINT artifacts encompass images, videos, text posts/comments, and basic profile metadata obtained without bypassing access controls.

Information types and uses. OSINT can surface personal, geographic, social, organizational, political, technical, and cultural signals. Common use cases include situational awareness, event verification, and lead generation. Our focus is security-relevant triage during crowded events, where public signals augment camera observations.

AI components. We apply convolutional neural networks (CNNs) for visual tasks and OpenCV for detection and pre-/post-processing. Specifically:

Face and object detection. Detectors localize faces and security-relevant objects in frames; face crops are aligned and normalized.

Embeddings and matching. A CNN produces feature embeddings that are compared (e.g., cosine similarity) to identify visually similar faces across images. Proper thresholds balance true- and false-positive rates under varying pose, illumination, and occlusion.

Object recognition. OpenCV/CNN pipelines rapidly classify or flag entities (e.g., people, vehicles, potential weapons) to prioritize review.

Integration with OSINT. Camera detections seed OSINT searches; publicly available imagery and text provide additional candidate views and contextual cues. These heterogeneous artifacts are then persisted and analyzed at scale to improve coverage and reduce time-to-identify compared with a camera-only baseline.

Ethical note. Only public sources are queried; no circumvention of access controls is performed. Data minimization, retention limits, and operator audit logging are applied, and human verification is required before any operational action.

## 4 DATA AND EXPERIMENTAL SETUP

4.1 Data sources. The study uses publicly available OSINT artifacts (images, videos, text/comments) and smart-city camera streams captured during a public event. In total, the corpus contains 1,221 images, 213 videos, and ~3,000 text files.

4.2 Ground truth and task. The goal is rapid person identification/triage. Ground truth is established by human review of matched faces and event logs.

4.3 Pre-processing. Frames are sampled at $s$ s fps; faces are detected, aligned, and normalized (e.g., 112×112). Text is normalized (lowercasing, tokenization, stop-word removal).

4.4 Models and tools. Object/face detection and CNN embeddings in OpenCV; storage in HDFS; querying with HiveQL; large-corpus counting via MapReduce.

4.5 Hardware/cluster. Hadoop cluster with $N$N nodes; key parameters (RAM/CPU, HDFS replication factor, Hive execution engine).

4.6 Metrics. Precision/recall, processing time (latency) and overall throughput.

4.7 Ethics. Only public sources; data minimization; audit logging; human-in-the-loop verification.

## 5. ANALYSIS OF DATA COLLECTED FROM CAMERAS IN A SMART CITY USING BIG DATA PROCESSING SYSTEMS, ARTIFICIAL INTELLIGENCE, AND OSINT

For the purposes of the research, a scenario was designed that represents a possible method for identifying a suspicious person in large gatherings in a smart city.

The scenario is as follows:

In a large crowd at Republic Square during a concert taking place there, an object detection system on camera footage sends an alert that a weapon has been detected in the possession of a person. The concert security notices the alert, and the goal is to identify the person as quickly as possible.

1. The security officer immediately reviews all available cameras. The camera search program, which has the ability to search both in real-time and retroactively, and which uses software to search for specific objects in the images, including weapons, sets the criteria for the search. The criteria are to separate a middle-aged man who is carrying a weapon as an object.


**FIGURE 1.0**: Camera review with search criteria before the search.

The program, with the help of smart cameras, almost in real time, separates a few photos that meet the criteria, on which the sought-after man can clearly be seen. However, the live camera feed shows that the last time the criteria were met was ten minutes ago, meaning that the sought-after person is currently not visible on any of the cameras.

**FIGURE 1.1**: Display of the person recognized by the system.

3. The search of existing data reveals that the system has not been able to find a match or identify the sought-after person in the current databases. Therefore, in this step, techniques for searching and comparing photos on the internet will be used. The goal now is to

search the internet and social media to find a match for the person from the camera. We once again use the Python OpenCV library, which is now used for comparing the same facial images, and alongside it, we use OSINT, which helps us search available sources on the internet with predefined functions for optimal searching.

4. A match has been found, and through visual inspection, it can be confirmed that it is indeed the same person. An account was found on the social media platform Instagram, which contains multiple posts with photos featuring the sought-after person.

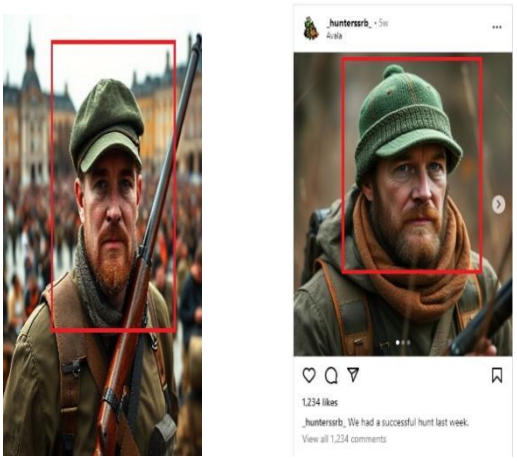


**FIGURE 1.2**: Display of the match between the search sample photo and the photo from the Instagram social media account.

5. It has been established that this is the profile of the hunting association *_hunterssrb_*.

6. In this step, the entire list of accounts, as well as all photos from the account that follow the *_hunterssrb_* profile and all the accounts that *_hunterssrb_* follows, are downloaded. Specialized OSINT techniques are used to download the content from the Instagram social media platform.


**FIGURE 1.3**: Display of the command for downloading the list of all followers of the profile.

7. The downloaded material is stored in the big data processing system's storage. A total of 1,221 photos, 213 videos, and 3,000 text files have been downloaded.
8. All photos from the *_hunterssrb_* profile that feature the person we are looking for are downloaded.
9. The downloaded photos from the *_hunterssrb_* profile are now compared using artificial intelligence techniques with the photos and videos downloaded from the profiles that *_hunterssrb_* follows and those who follow him. The assumption is that the sought-after person will be identified in these images, as the likelihood of following a profile in which the person is tagged is high.
10. After the download, no match was found, so now all comments on the posts from the *_hunterssrb_* profile featuring the person to be identified are being downloaded.


**FIGURE 1.4**: Display of the command for downloading comments from the profile on the Instagram social media platform.

11. After the download, similar comments are separated using HiveQL queries in the big data processing systems. It is established that the word "president" is mentioned in almost all posts.
12. Now, using MapReduce code to search for similar words in all downloaded files, the mentioned word is searched for and counted.

```
from mrjob.job import MRJob
import re

class MRPredictWord(MRJob):

    # Mapper function: processes each line from multiple files
    def mapper(self, _, line):
        # Convert line to lowercase and check if the word "president" is present
        words = re.findall(r'\b\w+\b', line.lower())
        for word in words:
            if word == "president":  # Only look for the word "president"
                # Emit the word 'president' as the key and 1 as the value
                yield (word, 1)

    # Reducer function: sums up the occurrences of the word "president"
    def reducer(self, word, counts):
        # Sum all the occurrences of the word
        yield (word, sum(counts))

if __name__ == '__main__':
    MRPredictWord.run()
```

**FIGURE 1.5**: Display of MapReduce code for searching similar words.

13. The assumption is that the person to be identified is the president of this hunting association, so a search is now conducted in

the Agency for Business Registers to determine the name of the association's president and his identification number.

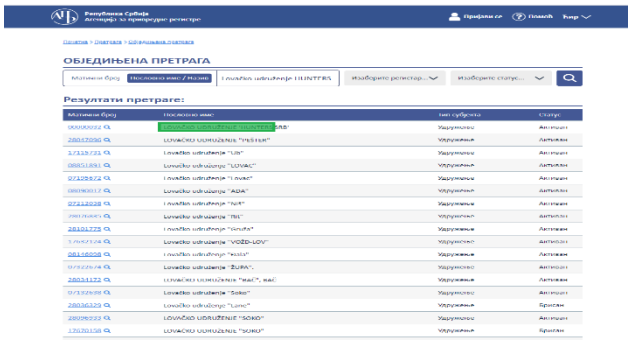


**FIGURE 1.6**: Display of the search in the Agency for Business Registers.

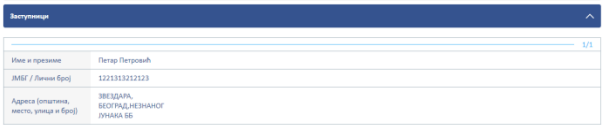

**FIGURE 1.7**: Display of representative data from the Agency for Business Registers.

14. The data obtained from the Agency for Business Registers, after further verification, confirmed that it is indeed the person that needed to be identified. Security personnel on the field, after receiving the information, can proceed with their work.

## 6 SUMMARY OF EVALUATION, ETHICS, DISCUSSION, LIMITATIONS, AND FUTURE WORK

We evaluate the system by measuring precision/recall, processing time, and overall throughput, comparing a camera-only baseline with the camera+OSINT variant and using HiveQL/MapReduce analytics over the corpus (~1,221 images, 213 videos, ~3,000 text files). Only publicly available sources are used, with no circumvention of access controls, and we enforce data minimization, limited retention, and access auditing; every candidate is confirmed by human verification. Results indicate higher recall and shorter time-to-identify compared with traditional procedures, while acknowledging risks from noise/duplicates and dependence on camera quality and data availability. Limitations include focus on a single event and the batch nature of MapReduce/Hive; streaming is needed for lower latency. Future work includes integrating Kafka+Spark/Flink streaming, improving re-identification models, advancing deduplication, and broader comparisons on public benchmark datasets.

## 7 CONCLUSION

This paper demonstrated how combining AI-based object and face analysis with Hadoop-scale analytics and OSINT can accelerate person triage at large public events in smart-city settings. By fusing camera detections with publicly available imagery and text, and storing/analyzing the resulting corpus in HDFS with HiveQL/MapReduce, the approach reduces time-to-identify compared with traditional, manual workflows. We evaluated the system through precision/recall, processing time, and overall throughput, and embedded safeguards—public-sources-only collection, data minimization, audit logging, and human verification before any action—to support responsible use.

While the scenario illustrates practical benefits, it also highlights limits: performance depends on camera quality, data availability, and threshold choices; OSINT correlations remain hypotheses until officially confirmed. Future work includes lower-latency streaming (e.g., Kafka + Spark), improved deduplication and re-identification across viewpoints, and benchmarking on public datasets to complement the case study. Overall, the proposed integration offers a scalable, ethically grounded pathway to faster, more reliable incident response in smart cities.

## 8 REFERENCES

[1] N. Petrović and V. Nikolić, “Analysis of Security and Intelligence Data Obtained through OSINT Techniques Using the Apache Hadoop Big Data Platform,” in AlfaTech 2025: International Scientific Conference Proceedings, Alfa BK University, Belgrade, 2025, pp. 188–191, doi: 10.46793/ALFATECHproc25.188P.

[2] Koops, BP., Hoepman, JP. & Leenes, R. 2013. Open Source Intelligence and Privacy By Design. Journal of Computer Law & Security Review, Elsevier, Tilburg University, The Netherlands

[3] Wiil, U. K. 2011 Counter Terrorism and Open Source Intelligence: Lecture Notes In Social Networks. Vol 2. 15 – 28. Denmark.

[4] Pradeepa, A., &Thanamani, A. S. (2013). Hadoop file system and fundamental concept of MapReduce interior and closure rough set approximations. International Journal of Advanced Research in Computer and Communication Engineering, 2(10), 5865-5868

[5] J. Ekanayake, S. Pallickara, G. Fox, MapReduce for data intensive scientific analyses, in: Proceedings of Fourth IEEE International Conference on eScience, Indianapolis, Indiana, USA, 2008, pp. 277–284

[6] Verma, A., Mansuri, A. H., & Jain, N. (2016, March). Big data management processing with HadoopMapReduce and spark technology: A comparison. In 2016 Symposium on Colossal Data Analysis and Networking (CDAN) (pp. 1-4). IEEE.

[7] Grinev. M., Grineva. M., et ai, "Analytics for the real-time web", In PVLDB, 4(12):1391-1394, September 2011.

[8] Helmi, R., Yusuf, S., & Jamal, A. (2019). Face recognition automatic class attendance system (FRACAS). In *IEEE international conference on automatic control and intelligent systems (I2CACIS 2019)*, Selangor, Malaysia, June 29, 2019.

[9] Zhao, Z.-Q., Zheng, P., Xu, S., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems,30*(11), 3212–3232.

**Contact information:**

**Nikola Petrović 1**, Ministry of Internal Affairs, Republic of Serbia,11000 Belgrade,Serbia
nikola.spetrovic@mup.gov.rs
https://orcid.org/0009-0004-2325-2822

**Vojkan Nikolić 2**,
Department for Informatics and Computing, University of Criminal Investigation and Police Studies, 11000 Belgrade, Serbia
Vojkan.nikolic@kpu.edu.rs