Building an annotation pipeline for atmospheric images: web infrastructure for low-cost cameras

Aleksandar Plamenac[1,4,*], Pavle Dakić[1,2,3,4], Terézia Hubáčková[2], Savo Tomović[1,4] and Ján Lang[1]

[1]Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Ilkovičova 2, 842 16 Bratislava, Slovakia
[2]Department of International Relations, Slovak University of Technology in Bratislava, Vazovova 5 81 243 Bratislava, Slovakia
[3]Faculty of Informatics and Computing, Singidunum University, Danijelova 32, 11 000 Belgrade, Serbia
[4]Faculty of Natural Sciences and Mathematics, University of Montenegro, Cetinjski Put bb, 81 000 Podgorica, Montenegro

Abstract

Cloud observation remains one element of a standard surface weather report that is not automated. While temperature, humidity, and pressure are routinely transmitted from automated weather stations, cloudiness and sky coverage are still visually assessed by human observers. This paper presents a low-cost infrastructure that bridges that gap for Montenegro's national hydrometeorological service. We describe the design and deployment of a system that captures sky images at meteorological stations, transfers them to a central server, supports their annotation by trained operators through a custom web interface, and assembles the result into a reproducible dataset through an ETL pipeline. During a 56-day pilot deployment, the system collected 2,522 sky images at a delivery rate of 93.8%, with each image labeled by 4.44 trained operators on average, providing more than 11,000 annotation records across seven SYNOP categories. The results show that the system loses random captures rather than full days and that annotator agreement varies systematically with the difficulty of the underlying meteorological judgment. The contribution is positioned as an end-to-end infrastructure account that the cloud classification literature, focused on algorithms and curated benchmarks, has so far left largely undocumented and as a replicable blueprint for other small national weather services facing similar operational constraints.

Keywords

cloud observation, automated weather stations, machine-readable dataset, inter-annotator agreement, ETL pipeline, reproducibility

CEUR-WS.org/Vol-4336/paper12.pdf

1. Introduction

Cloud observation is one of the few elements of standard Montenegrin surface weather reports that are still not automated. Although temperature, humidity, pressure, and wind are routinely measured and transmitted by sensors, cloud type and sky cover are still visually assessed and manually entered into SYNOP reports [1]. This creates a persistent bottleneck, especially in rural and remote stations with little or no human oversight, and it limits the temporal consistency of national cloud records.

Recent advances in ground-based cloud classification with deep learning [2, 3, 4] suggest that camera-equipped stations and trained models could bridge this gap. However, the existing literature focuses almost entirely on classification algorithms and curated benchmark datasets such as CCSN [2] and SWIMCAT [5]. The practical infrastructure needed to turn a camera network at operational stations into a usable, well-characterized resource—capture scheduling, data transfer, multi-operator annotation, and reproducible dataset assembly—is rarely documented, and its operational performance is seldom quantified. Consequently, a weather service that wants to deploy its own ground-based cloud

SQAMIA 2026: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 10–12, 2025, Niš, Serbia

*Corresponding author.
plamenac.a@ucg.ac.me (A. Plamenac); pavle.dakic@stuba.sk (P. Dakić); terezia.hubackova@stuba.sk (T. Hubáčková); savot@ucg.ac.me (S. Tomović); jan.lang@stuba.sk (J. Lang)
0000-0003-1925-9533 (A. Plamenac); 0000-0003-3538-6284 (P. Dakić); 0009-0005-2584-0983 (T. Hubáčková); 0000-0002-9388-7784 (S. Tomović); 0000-0002-3271-7271 (J. Lang)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-19

observation pipeline has little guidance on delivery reliability, annotation workflows, or the data-quality characteristics that downstream classifiers must address.

This paper addresses exactly that gap. We present a low-cost, end-to-end system that captures sky images at Montenegrin meteorological stations, transfers them to a central server, supports annotation by trained operators via a custom web application, and compiles labeled images into a reproducible dataset through an ETL pipeline. We characterize the system and its data with three research questions related to the proportion of delivered image captures, the agreement between annotators, and the balance of observed cloud classes. The contribution is positioned as an operational infrastructure account and a replicable blueprint for similarly resourced national weather services, rather than as a new classification algorithm.

2. Related Work and Background

This work sits at the intersection of ground-based cloud classification, cloud-cover estimation, and the software infrastructure needed to build a reproducible machine-learning dataset from operational observations.

The most widely used benchmark for ground-based cloud classification is the Cirrus–Cumulus–Stratus–Nimbus collection (CCSN) introduced by Zhang et al. [2]. It organizes 2,543 ground-based images into 11 categories aligned with World Meteorological Organization recommendations [6] and has been used as a reference for many classifier architectures. The smaller Singapore Whole-sky Imaging Categories (SWIMCAT) dataset of Dev et al. [5] plays a similar role for sky-condition recognition. Both collections have driven a steady stream of classifier improvements, including the lightweight Cloud-MobiNet architecture [3], the channel-attention variant of Zhu et al. [7], the transformer-based formulation of Li et al. [8], and the recent RepVGG-with-attention model of Liu et al. [4]. These works routinely report classification accuracy above 95%, which initially struck us as encouraging.

Close inspection of these reported accuracies became relevant to our third research question, in a cautionary sense. CCSN images are curated, cropped in a square shape, and visually unambiguous, and the photographs were selected by domain experts precisely because they are recognizable instances of their class. On the other hand, our dataset is generated by fixed cameras pointed at whatever sky is above the station at the scheduled minute. The survey of Nie et al. [9], which inventories 72 open-source sky-image datasets, reinforces this concern by documenting wide variation in resolution, field of view, sampling rate, and label granularity across the literature, and by noting that comparability between datasets is constrained by exactly these factors. We took this as a methodological warning: producing yet another dataset is only justified if its acquisition context is documented as carefully as its content, which directly shaped how we report class distribution in Section 4 rather than reporting only an aggregate image count.

The closest precedent for our hardware setup is Richardson et al. [10], which combines a single-board computer and camera in a weatherproof housing as a low-cost alternative to commercial devices for total sky coverage. Although focused on solar irradiance forecasting rather than cloud type observation, it shares several design choices with our system: edge capture with deferred transmission, lightweight preprocessing on device, and an enclosure for unattended operation. A broader DIY ecosystem of all-sky cameras based on Raspberry Pi [11, 12, 13] further reduces costs and shows field reliability over months. We did not adopt these designs wholesale. Our stations use IP cameras with programmable triggers rather than embedded SBCs, but these accounts showed that per-station hardware costs in the low hundreds of euros are practically achievable. More importantly, they consistently highlight the same operational issue as our first research question: capture reliability under unattended deployment. None, however, reports a quantitative delivery rate against a known capture schedule, which is the gap we address.

The integration of such cameras into operational weather service workflows is less documented than the hardware or classification aspects. The WMO Manual on the Global Observing System [1] specifies requirements for automatic weather stations, but still treats cloud observation as a manually populated

SYNOP field. Practical reports on AWS deployment in resource-constrained settings, such as Provenzale et al.'s chapter on stations in Niger and Nepal [14], were more relevant to our work than algorithmic papers, highlighting that siting, power, connectivity, and local maintenance, rather than sensor choice, are the main constraints. In this description, we recognize our three pilot sites.

Our second research question, on inter-annotator agreement, engages a literature rarely cited in cloud classification. The key work for multilabel image annotation is Nowak and R"uger [15], whose main insight is that systematic disagreement among careful annotators is informative, which applies here directly, where meteorological operators sometimes assign different SYNOP cloud codes to the same image. We compute Cohen's κ [16] for the annotator pairs with the largest overlap and interpret the results using the standard Landis and Koch ranges [17]. For the three ordinal label types in our taxonomy (cloud-base height and the two coverage estimates calculated in eights), we also report linearly weighted kappa and ± 1-step tolerance accuracy, since adjacent ordinal values should be penalized less than categorically distinct ones, a methodological choice discussed in Section 4. Krippendorff's α [18] provides a complementary measure robust to small differences in annotator coverage.

Reproducible-computing guidelines also shaped the pipeline. Sandve et al.'s [19] rules—tracking every intermediate result, avoiding manual data manipulation, and preserving raw data—aligned with our choice to separate extraction, transformation, and acquisition into distinct, idempotent stages. Wilson et al.'s "good enough" practices [20] supported a notebook-based pipeline with explicit cells over a monolithic script, treating scientific software as a first-class research output. Samuel and Mehnert Spahn's view [21] that provenance matters as much as the final dataset motivated us to retain both the raw long-format annotation export and the pivoted version as separately version-controlled artifacts.

Taken together, the literature shows that cloud-classification algorithms and low-cost camera hardware are well studied, but the practical infrastructure that turns an operational camera network into a documented, reproducible, annotated dataset has received little attention. Existing benchmark datasets are curated and visually unambiguous, while operational datasets are not; low-cost camera reports focus on irradiance or coverage rather than on SYNOP-aligned class labels. We therefore ask: (i) whether an unattended low-cost capture schedule can deliver a high proportion of expected images, (ii) how much annotators agree on the resulting images, and (iii) how the class distribution reflects the winter observation window. The next section translates these questions into hypotheses and methodology.

3. Materials and Methods

We describe the proposed system in three parts: image acquisition, delivery to a centralized system suitable for expert annotation, and the offline module for dataset creation. Our design choices favor low-cost, easily reproducible components over specialized hardware or complex architectures, keeping operational constraints to a minimum.

3.1. Research Questions

The methodology was shaped by the following research questions:

1. What percentage of scheduled image captures are sent from the observatories to the central database?
2. How is the inter-annotator agreement distributed across the seven classes for cloud types?
3. What is the class distribution of the generated dataset and how does it differ from the expected balanced and uniform distribution?

3.2. Hypotheses

Based on the research questions, we formulate the following hypotheses.

- **H1:** At least 90% of scheduled image captures are stored in the central database and ready for annotation.

- **H2:** For at least 80% of the images, we expect either that the annotators have given the same answer or that we have a strong majority.
- **H3:** Considering that the image capture process is performed only in November and December, we cannot expect all classes to be evenly distributed. The resulting dataset will probably contain fewer instances of zero cloudiness and of clouds usually seen in spring.

3.3. Methodology

Image acquisition was scheduled at 15-minute intervals, with the exception of night. From 18:00 until 06:00 on the following day, the cameras remained inactive and did not capture any images. Following acquisition, each image was transmitted over the network to an annotator's workstation as a backup. The same image was subsequently transferred to the central database, and an additional copy was stored on the main server. As the acquisition module itself has three components (camera, annotator workstation, and central server), we define the delivery rate as the ratio between the expected number of photos and the number of photos stored at the central server. We can calculate the expected number of photos per day from the active window T (in seconds) and the capture interval Δt (in seconds):

$$N = \frac{T}{\Delta t}. \tag{1}$$

In our case, $\Delta t = 900$ s (15 min) and $T = 43{,}200$ s (12 h), giving $N = 48$ images per day.

3.4. Annotation procedure

The annotation workflow lets trained meteorological observers label images with the same categories they use for conventional SYNOP reports. Annotators were recruited from the staff of Montenegro's Hydrometeorological and Seismological Service (ZHMS); all had operational experience with manual cloud observation and were familiar with WMO cloud-code conventions. No external crowd-workers were used, so the labels reflect institutional observing practice. Images were captured automatically during daylight hours and appeared in the web interface shortly after transfer; annotations were therefore made after acquisition but were based on the same visual information an observer at the station would have had at the scheduled time. Each image was independently classified by every annotator who logged in, with no upper limit on the number of annotators per image (hence the mean of 4.44). All values were retained; there was no forced consensus or adjudication, so the released dataset preserves the full distribution of annotator opinions for each image and supports either majority labels or soft target distributions. Section 4 reports pairwise agreement over the subset of images annotated by all three most active annotators.

Table 1
Label types used to annotate each cloud image, aligned with the SYNOP form fields used by operators at ZHMS.

Label type	Description	# values
Altitude grouping	Dominant cloud altitude class	6
Base height	Lowest cloud base, discretized	11
Total coverage	Total sky coverage (in eights)	11
Low coverage	Cover of low clouds (in eights)	11
CL	Low cloud type (WMO codes 0–9, /)	11
CM	Middle cloud type (WMO codes 0–9, /)	11
CH	High cloud type (WMO codes 0–9, /)	11

Table 1 summarizes the seven label types used for each image, all aligned with the SYNOP cloud fields that ZHMS observers already report operationally. The coverage labels and base height are ordinal variables expressed in discrete steps; total and low coverage are reported in eighths of sky cover (0/8 to 8/8), while base height is encoded as a WMO code. The CL, CM, and CH labels are categorical WMO cloud-type codes, including a value for "none of this genus" and a missing-observation symbol. For

the ordinal coverage and height fields, we treat values within ±1 step as effectively equivalent when measuring annotator agreement.

3.5. System architecture

The complete system consists of four logical and physical components. The entire dataset is stored on a central server that hosts all images together with the associated database. Cameras are installed at the weather stations and each expert annotator at these stations uses a laptop to access and operate the web-based annotation application. All collected data are subsequently aggregated on a research workstation, where machine learning modules can be executed.

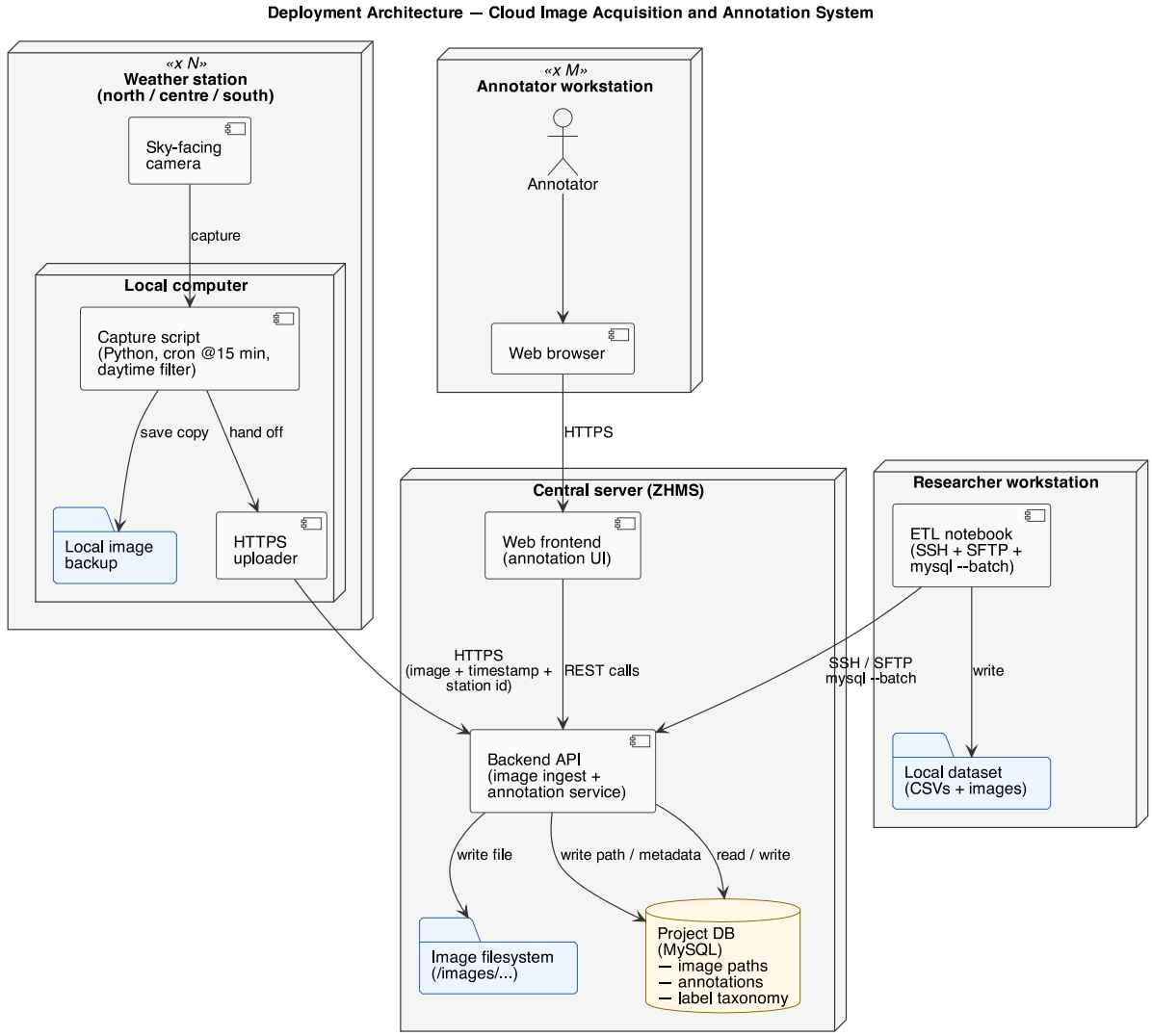


Figure 1: Deployment architecture of the cloud image acquisition system.

An overview of the system is presented in Figure 1. Low-cost cameras were installed at two observation stations in Podgorica and one in Bar, each oriented toward a different cardinal direction. One camera was directed south, one north, and one west. The western orientation was selected based on information provided by station experts, indicating that heavy cloud cover and precipitation that affect the Podgorica municipality most frequently approach from that direction.

The cameras used in the system architecture were DAHUA IPC-HFW1430DS-SAW-0280B units. This model was selected due to its support for programmability, which enables remote trigger of image acquisition via Python scripts. Each device is equipped with an Ethernet port and a Wi-Fi antenna,

allowing flexible connectivity; depending on the deployment context, the cameras are connected through a wireless network or through a wired Ethernet link. A Python script runs on each observer workstation, triggers the camera, and sends the captured image to the centralized server.

Listing 1: Function for capturing the snapshot from the camera

```
def capture_snapshot(retries=3, backoff=2):
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S_%f")[:-3]
    iso_timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    today = datetime.now().strftime("%Y-%m-%d")
    day_dir = SAVE_DIR / today
    day_dir.mkdir(parents=True, exist_ok=True)
    filename = day_dir / f"snapshot_{timestamp}.jpg"
    url = f"http://{CAMERA_IP}/cgi-bin/snapshot.cgi"

    attempt = 0
    while attempt <= retries:
        try:
            response = requests.get(url, auth=HTTPDigestAuth(USERNAME, PASSWORD),
                timeout=10)
            if response.status_code == 200:
                if len(response.content) < 1000:
                    raise ValueError("Snapshot may be corrupted (too small)")
                with open(filename, 'wb') as f:
                    f.write(response.content)
                logger.info(f"Snapshot saved: {filename}, {CAMERA_NAME},
                    {CAMERA_IP}, {timestamp}")
                return filename, timestamp, response.content, iso_timestamp
            else:
                logger.warning(f"Failed to capture snapshot: HTTP
                    {response.status_code}")
                return None, None, None, None

        except Exception as e:
            logger.error(f"Attempt {attempt + 1}: Error capturing snapshot: {e}")
            time.sleep(backoff ** attempt)
            attempt += 1

    logger.error("Exceeded maximum retry attempts. Snapshot not saved.")
```

The code in Listing 1 shows how the snapshot is captured. After generating the timestamp and creating the destination folder, the script sends the HTTP request to the camera to trigger the image. The image content and filename are then returned.

Listing 2: Function for sending the snapshot to the server

```
def send_snapshot_to_server():
    try:
        filename, timestamp, content, iso_timestamp = capture_snapshot()
        if filename and timestamp and content and iso_timestamp:

            log_image_base64_preview(content)
            image_io = BytesIO(content)
            image_io.name = filename.name # mimic a file object with name
```

```
9
10          files = {
11             "file": (filename.name, image_io, "image/jpg")
12          }
13          data = {
14             "device_id": CAMERA_NAME,
15             "timestamp": iso_timestamp
16          }
17          response = requests.post(FASTAPI_BACKEND_URL, data=data, files=files)
18
19          if response.status_code == 200:
20             logger.info(f"Snapshot sent successfully: {filename.name},
                   {CAMERA_NAME}, {CAMERA_IP}, {timestamp}")
21          else:
22             logger.warning(f"Snapshot failed to send: HTTP
                   {response.status_code}, {response.text}, {filename.name},
                   {data}, {files}")
23       else:
24          logger.warning("Snapshot capture failed or returned empty values.")
25
26    except Exception as e:
27       logger.error(f"Error sending snapshot to server: {e}")
```

Listing 2 shows the sending function. The script calls the capture function, retrieves the image content and filename, and sends them to the backend, which stores the file and metadata in a MySQL database. The web application uses Angular on the frontend and FastAPI on the backend. Angular's component model lets us reuse and validate the same annotation form across the seven SYNOP label types, reducing field-entry errors in a multi-user operational setting, while FastAPI and MySQL were chosen because the rest of the pipeline—camera scripts, ETL notebooks, and planned downstream classifiers—is written in Python and SQL, simplifying maintenance by meteorological IT staff. The full acquisition, backend, and ETL code will be released in a public repository (URL to be inserted after anonymization review).

3.6. Dataset construction pipeline

The offline ETL pipeline transforms raw uploads and annotation exports into the four relational CSV files summarized in Table 2. The design separates stable reference data from volatile annotations so that the dataset can be versioned and reproduced without re-exporting the database.

Table 2
The four CSV tables that form the released dataset.

File	Granularity	Key columns
`label_types.csv`	1 row per category	`id`
`labels.csv`	1 row per value	`id`, `label_type_id`
`annotation_items.csv`	1 row per image	`id`, `name`
`annotations.csv`	1 row per (image, annotator)	`annotation_item_id`, `user_id`

`label_types.csv` stores the seven classification tasks and their display metadata. `labels.csv` enumerates every admissible value for each task, including ordinal codes 0–8 and WMO cloud-type codes. Keeping labels in a separate file prevents duplicate strings and makes it easy to add new label sets without restructuring the main annotation table.

`annotation_items.csv` is the image inventory: one row per captured JPEG with a stable identi-

fier and the relative path to the image file. `annotations.csv` is the core observation matrix, with one row per annotator–image–label combination; its key columns are `annotation_item_id` (foreign key to the image), `user_id` (anonymous annotator identifier), `label_type_id`, and `label_id` (the selected value). Splitting annotations into long format preserves every annotator's original decision, supports reproducible pivoting to a majority-label or soft-target matrix, and avoids manual-spreadsheet corruption.

4. Results

The results are organized around the three research questions: delivery reliability, inter-annotator agreement, and class distribution. We first report the overall dataset summary and delivery rate, then examine pairwise agreement among the most active annotators, and finally characterize the concentration and seasonality of the observed labels.

Table 3 shows that the pilot produced 2,522 usable images over 56 active days, a delivery rate of 93.8%. Annotation coverage is 100% because every uploaded image was labeled, with 4.44 independent annotations per image on average—substantially more redundancy than is common in curated cloud-classification datasets, where a single expert label is typical.

Table 3
The summary of the created dataset.

Metric	Value
Observation period (days)	56
Scheduled captures (theoretical)	2 688
Successful uploads	2 522
Delivery rate (%)	93.8
Annotated images	2 522
Annotation coverage (%)	100.0
Annotators (total)	9
Annotators contributing $\geq$ 1 000 labels	5
Annotations (item, user) pairs	11 191
Mean annotators per image	4.44
Median annotators per image	5
Maximum annotators per single image	9
Label fields per annotation	7
Total label values written	78 214
Mean field-completion rate (%)	99.8

Figure 2 reports Cohen's κ for the three annotators who labeled every image (Annotator 1, Annotator 2, and Annotator 3). Agreement is highest for total sky coverage, reaching 0.89 for one pair—the variable most directly comparable to conventional SYNOP reports. Agreement is below the conventional moderate threshold ($\kappa < 0.6$) for all other label types, with the lowest value, 0.09, for base-height annotations, indicating that estimating cloud-base height from a single fixed camera is the most subjective task.

Table 4 reports the concentration of labels for each of the seven tasks. The Gini coefficient quantifies how uniformly label values are used: near 0 means roughly equal use, near 1 means a few classes dominate. Total coverage has the lowest Gini (0.316), reflecting broad use of the 0/8–8/8 scale, while CH has the highest (0.726) because the "none" code alone accounts for 57.2% of annotations; CM and CL are similarly dominated by "none." This pattern is expected for a November–December dataset in Montenegro, where high and middle clouds are often absent or obscured.

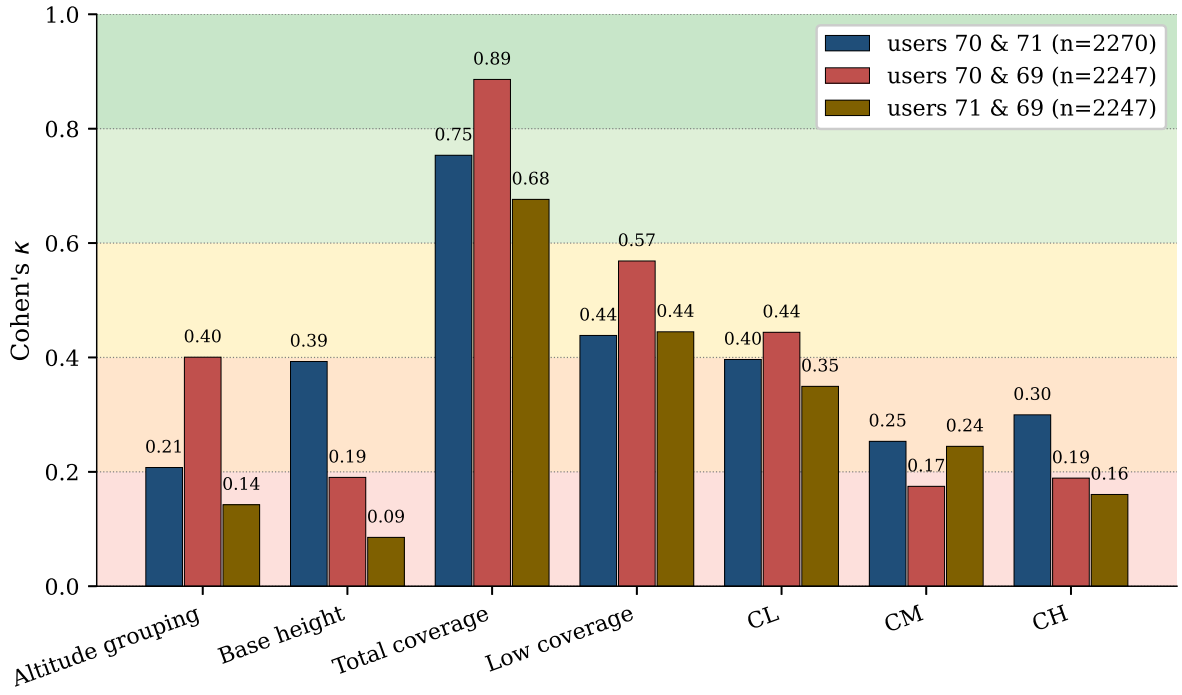


Figure 2: Pairwise inter-annotator agreement (Cohen's κ) among the three most active annotators, labeled Annotator 1, Annotator 2, and Annotator 3. Each panel corresponds to one SYNOP label type; the height of a bar shows agreement for one annotator pair.

Table 4
Class distribution statistics for the seven SYNOP label types. Gini coefficients close to 0 indicate near-uniform usage of available label values; values close to 1 indicate strong concentration in a few classes.

Label type	n	Most freq. class	%	Gini
Altitude grouping	11,190	low clouds	43.8	0.336
Base height	11,191	code 6	32.5	0.461
Total coverage	11,191	8/8 (overcast)	23.3	0.316
Low coverage	11,186	0/8 (clear)	27.3	0.402
CL (low clouds)	11,182	code 0 (none)	30.0	0.572
CM (middle clouds)	11,132	code 0 (none)	51.8	0.625
CH (high clouds)	11,135	code 0 (none)	57.2	0.726

5. Discussion

The active observation window, defined as days with at least one recorded image, was 56 days. Applying Equation 1 gives the theoretical maximum number of captures:

$$N_{\max} = D \cdot \frac{T}{\Delta t} = 56 \cdot \frac{43\,200}{900} = 56 \cdot 48 = 2{,}688 \text{ images.} \tag{2}$$

The number of successful uploads was 2,522, corresponding to a delivery rate of 93.8%. This result supports Research Question 1, which concerned the proportion of received captures. On the basis of this outcome, we consider hypothesis H1 to be supported.

Regarding annotation agreement, five annotators contributed at least 1,000 labels each, but the analysis in Figure 2 focuses on the three annotators who labeled all 2,522 images, because only they provide complete pairwise overlap. Cohen's κ is sensitive to category prevalence and to the exact set of images shared by a pair, so the reported values describe agreement among the most active operators

rather than the full pool of nine. The highest agreement is for total sky coverage (κ up to 0.89), the variable most directly comparable to conventional SYNOP reports. Agreement is below the conventional moderate threshold for all other label types, with base height reaching 0.09. This pattern is consistent with the meteorological difficulty of the judgments: coverage is visually salient, while cloud-base height from a single fixed camera requires inference about distance and perspective.

Our initial H2 hypothesis expected either exact agreement or a strong majority on at least 80% of the images, but this formulation was too broad: it did not distinguish label types, account for the ordinal nature of coverage and height codes, or specify the agreement metric. Only total coverage meets a strict 80% pairwise κ threshold. Interpreted as "the most visually salient and operationally relevant label reaches strong agreement," H2 is supported; interpreted literally across all seven label types, it is not. We therefore treat H2 as a prompt for future, more targeted hypotheses, and will report both strict agreement and ± 1-step tolerance for ordinal fields, comparing the majority-vote label with the full annotator distribution.

H3 was confirmed in direction but not in every detail. Clear-sky total coverage accounts for only 14.8% of images, while 8/8 overcast is the dominant class at 23.3%. Low-cloud types typical of warmer months (CL codes 1 and 2) are rare, and the most common CL value is "none" (code 0), followed by mixed layers (code 8). The dominant category is therefore often absence of a cloud genus rather than a winter-specific type. This distinction matters for classifier training: seasonality affects both which classes are rare and which are absent, and these phenomena have different downstream implications.

5.1. Limitations

Several limitations should be kept in mind. The 56-day window covers only November and December, so the dataset does not capture summer cloud regimes, convective afternoon development, or the full annual cycle of cloud-base heights. The three cameras in and around Podgorica and Bar may not generalize to coastal, mountain, or continental stations with different cloud climatologies. Camera positioning was fixed and horizon-inclusive rather than zenith-facing, so the field of view differs from all-sky cameras used in several benchmark datasets. Pairwise Cohen's κ was measured over the subset of images labeled by all three most active annotators; this is informative but does not capture the full variability among nine annotators or the uncertainty introduced by different overlap patterns. Finally, the study demonstrates feasibility but does not yet evaluate a downstream classifier; those experiments will follow once a full annual dataset is available.

5.2. Low-cost justification

The "low-cost" claim in the title is a quantitative comparison. Commercial total-sky imagers typically cost several thousand euros per unit and require specialized maintenance, whereas the cameras used here (DAHUA IPC-HFW1430DS-SAW-0280B) retail for under 200 EUR. Comparable DIY all-sky systems built around Raspberry Pi cameras also report per-station hardware costs in the low hundreds of euros[11, 12, 13]. The main savings are operational: the cameras use existing station power and network infrastructure, the acquisition script is a small Python program maintainable by in-house IT staff, and the FastAPI/MySQL backend runs on existing university servers. A full cost breakdown and the open-source release of the acquisition and ETL code will accompany the dataset publication.

6. Conclusion

This paper presented a low-cost, end-to-end infrastructure for acquiring a ground-based cloud image dataset integrated into the operational information system of Montenegro's hydrometeorological service. Rather than proposing a new classifier, the contribution lies in documenting and quantifying the complete link between a distributed camera network and a research-grade dataset: capture scheduling, data transmission, multi-operator annotation, and a reproducible ETL pipeline. Three research questions on delivery reliability, inter-annotator agreement, and class distribution structured the evaluation.

During a 56-day pilot, the system acquired 2,522 sky images with a 93.8% delivery rate; each image was annotated by an average of 4.44 trained operators, yielding more than 11,000 annotation records and 78,214 individual labeling decisions.

The most consequential software decisions were architectural: separating acquisition, annotation, and dataset construction into distinct phases with well-specified interfaces, treating every intermediate artifact as a first-class output, and using a repeatable pipeline. The low-cost claim is supported by off-the-shelf IP cameras (under 200 EUR per unit), the use of existing station power and network infrastructure, and a maintenance stack familiar to meteorological IT staff.

Future work will extend the deployment to a complete annual cycle, use the multi-annotator labels as soft target distributions for downstream classifier training, and publish the dataset and pipeline code under an open license to enable similar deployments at other small national meteorological services.

Acknowledgments

The authors acknowledge that work was supported by Erasmus+ ICM 2023 No. 2023-1-SK01-KA171-HED-000148295, the Cultural and Educational Grant Agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic (KEGA) - Model-based explication support for personalized education (Podpora personalizovaného vzdelávania explikovaná modelom) - KEGA (014STU-4/2024), the Slovenian Research and Innovation Agency (ARIS) (Research Core Funding No. P2-0057) and the Slovak Research and Development Agency under Contract no. APVV-23-0408. The creation of the original system architecture with low-cost cameras was supported by the Montenegrin Innovation Fund, through the Proof of Concept program, project number: POC-008-24.

Declaration on Generative AI

During the preparation of this paper, the author(s) used Grammarly to do: Grammar and spelling check. While the authors did not use any generative AI tools to generate content, text, or ideas in this paper.

References

[1] World Meteorological Organization, Manual on the Global Observing System, WMO-no. 544, volume i ed., WMO, Geneva, Switzerland, 2013.

[2] J. Zhang, P. Liu, F. Zhang, Q. Song, CloudNet: Ground-based cloud classification with deep convolutional neural network, Geophysical Research Letters 45 (2018) 8665–8672. doi:10.1029/2018gl077787.

[3] E. K. Gyasi, P. Swarnalatha, Cloud-MobiNet: An abridged mobile-net convolutional neural network model for ground-based cloud classification, Atmosphere 14 (2023) 280. doi:10.3390/atmos14020280.

[4] C. Liu, S. Yang, D. Di, Y. Yang, C. Zhou, X. Hu, B.-J. Sohn, Improved RepVGG ground-based cloud image classification with attention convolution, Atmospheric Measurement Techniques 17 (2024) 979–997. doi:10.5194/amt-17-979-2024.

[5] S. Dev, Y. H. Lee, S. Winkler, Categorization of cloud image patches using an improved texton-based approach, in: 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 422–426. doi:10.1109/icip.2015.7350833.

[6] World Meteorological Organization, International Cloud Atlas: Manual on the Observation of Clouds and Other Meteors, WMO-no. 407 ed., WMO, Geneva, Switzerland, 2017. URL: https://cloudatlas.wmo.int/.

[7] W. Zhu, T. Chen, B. Hou, C. Bian, A. Yu, L. Chen, M. Tang, Y. Zhu, Classification of ground-based cloud images by improved combined convolutional network, Applied Sciences 12 (2022) 1570. doi:10.3390/app12031570.

[8] X. Li, B. Qiu, G. Cao, C. Wu, L. Zhang, A novel method for ground-based cloud image classification using transformer, Remote Sensing 14 (2022) 3978. doi:10.3390/rs14163978.

[9] Y. Nie, X. Li, A. Scott, Y. Sun, V. Venugopal, A. Brandt, Open-source ground-based sky image datasets for very short-term solar forecasting, cloud analysis and modeling: A comprehensive survey, 2022. doi:10.48550/arXiv.2211.14709. arXiv:2211.14709.

[10] W. Richardson, H. Krishnaswami, R. Vega, M. Cervantes, A low cost, edge computing, all-sky imager for cloud tracking and intra-hour irradiance forecasting, Sustainability 9 (2017) 482. doi:10.3390/su9040482.

[11] Jippo, Cheap all sky camera project with Raspberry Pi, Hackaday.io project 175041, 2020. URL: https://hackaday.io/project/175041-cheap-all-sky-camera-project-with-raspberry-pi.

[12] A. Felong, An inexpensive and compact allsky camera, andyfelong.com, 2023. URL: https://andyfelong.com/2023/02/an-inexpensive-compact-allsky-camera/.

[13] J. Bowling, Weatherproof Pi looks up so you don't have to, Hackaday, 2017. URL: https://hackaday.com/2017/11/09/weatherproof-pi-looks-up-so-you-dont-have-to/.

[14] M. Provenzale, S. Tani, A. Antonini, B. Gozzini, A. Ortolani, F. Caparrini, S. Melani, Setting up and managing automatic weather stations for remote sites monitoring: From niger to nepal, in: M. Tiepolo, A. Pezzoli, V. Tarchiani (Eds.), Renewing Local Planning to Face Climate Change in the Tropics, Springer International Publishing, 2018, pp. 21–39. doi:10.1007/978-3-319-59096-7_2.

[15] S. Nowak, S. Rüger, How reliable are annotations via crowdsourcing? a study about inter-annotator agreement for multi-label image annotation, in: Proceedings of the International Conference on Multimedia Information Retrieval (MIR '10), ACM, New York, NY, USA, 2010, pp. 557–566. doi:10.1145/1743384.1743478.

[16] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1960) 37–46. doi:10.1177/001316446002000104.

[17] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174. doi:10.2307/2529310.

[18] K. Krippendorff, Content Analysis: An Introduction to Its Methodology, 4th ed., SAGE Publications, Thousand Oaks, CA, 2018.

[19] G. K. Sandve, A. Nekrutenko, J. Taylor, E. Hovig, Ten simple rules for reproducible computational research, PLOS Computational Biology 9 (2013) e1003285. doi:10.1371/journal.pcbi.1003285.

[20] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, T. K. Teal, Good enough practices in scientific computing, PLOS Computational Biology 13 (2017) e1005510. doi:10.1371/journal.pcbi.1005510.

[21] S. Samuel, D. Mehnert-Spahn, Machine learning pipelines: Provenance, reproducibility and FAIR data principles, 2020. doi:10.48550/arXiv.2006.12117. arXiv:2006.12117.