Photo Gallery as a Service for Mobile Applications with Efficient Image Compression and Adaptive AI-Based NSFW Moderation

Matheus Nascimento Peixoto
NESPeD-Lab
Universidade Federal de Viçosa (UFV)
Florestal, Brazil
matheus.n.peixoto@ufv.br

Jader Duque Figueredo
NESPeD-Lab
Universidade Federal de Viçosa (UFV)
Florestal, Brazil
jader.figueredo@ufv.br

Matheus Silva Palhares
NESPeD-Lab
Universidade Federal de Viçosa (UFV)
Florestal, Brazil
matheusspalhares@ufv.br

Fabrício A. Silva
NESPeD-Lab
Universidade Federal de Viçosa (UFV)
Florestal, Brazil
fabricio.asilva@ufv.br

Thais R. M. Braga Silva
NESPeD-Lab
Universidade Federal de Viçosa (UFV)
Florestal, Brazil
thais.braga@ufv.br

***Abstract*—The proliferation of real-time photo galleries and User-Generated Content (UGC) in mobile applications introduces severe challenges regarding network bandwidth consumption and content governance. Transmitting heavy, uncompressed image files over constrained wireless networks degrades user experience, while the uncurated publication of such media exposes platforms to Not Safe For Work (NSFW) security risks. To address this dichotomy, we propose a platform-independent, decoupled architectural framework. On the client-side edge, the system performs *on-the-fly* compression prior to network transmission. By instantiating this approach using HTML5 Canvas and WebP encoding, we achieved a 98% reduction in payload size while preserving structural visual fidelity (Structural Similarity Index > 0.96). On the server-side, this optimized data stream is asynchronously evaluated by an adaptive Artificial Intelligence moderation layer. By implementing a continuous safeness scoring mechanism with configurable thresholds, the architecture autonomously blocks explicit imagery and routes ambiguous cases to a manual review queue, achieving a Macro F1-score exceeding 96.8%. Validated through an *in-the-wild* proof of concept, the proposed framework demonstrates robust cross-format inference capabilities, offering a scalable solution for efficient and secure participatory media sharing.**

***Index Terms*—Mobile Applications; Image Compression; Artificial Intelligence; Content Moderation; Edge Optimization.**

I. Introduction

The integration of participatory sensing and User-Generated Content (UGC) has fundamentally transformed the landscape of mobile applications, particularly in the context of academic, corporate, and social events [1], [2]. Modern event platforms rely heavily on interactive features, such as real-time photo galleries, to foster continuous user engagement. However, the unrestricted sharing of high-resolution multimedia introduces two critical bottlenecks for distributed systems: severe network constraint degradation and complex content governance.

From a network perspective, the continuous transmission of heavy, uncompressed image files (e.g., JPEG or HEIC) over constrained wireless networks rapidly consumes user data plans and overloads server-side storage infrastructure [3]. While traditional architectures attempt to solve this by compressing data on the server, this approach still forces the mobile client to transmit the massive original file over the network.

Concurrently, mitigating the network bottleneck and allowing high-throughput image uploads exposes the hosting platform to a critical security vulnerability: the dissemination of inappropriate or Not Safe For Work (NSFW) content. In professional environments, the publication of explicit imagery can irreparably damage the integrity of the event. Given the agile and high-volume nature of mobile galleries, manual curation of every submitted image is fundamentally unscalable and prone to human latency [4].

Although recent advancements in Convolutional Neural Networks (CNNs) have enabled high-accuracy automated image classification [4], [5], existing solutions generally treat data compression and AI moderation as isolated problems. Deploying heavy AI models requires substantial computational overhead, and performing inference on massive raw files further exacerbates server response times. Furthermore, rigid binary classification models fail to account for the nuanced and contextual nature of event-specific moderation, where ambiguous images require human oversight rather than blind algorithmic rejection [5].

To bridge this gap, this paper proposes a platform-independent, decoupled architectural framework designed to jointly optimize mobile data traffic and automate content moderation. Our approach introduces a dual-layer pipeline: a client-side module that reduces the image size prior to network

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

transmission, and an adaptive server-side AI moderation layer that evaluates these lightweight files.

The main contributions of this work are summarized as follows:

- We propose a client-side edge optimization paradigm that drastically reduces network payload and storage growth. In our hybrid mobile implementation using the HTML5 Canvas API, this approach achieved a 98% payload reduction while preserving essential structural visual features.
- We design an adaptive AI-based moderation architecture that replaces rigid binary classification with a continuous safeness scoring system. This introduces a tri-region threshold mechanism, categorizing images as safe, unsafe, or requiring manual review, allowing organizers to dynamically calibrate moderation strictness.
- We empirically validate the framework through an *in-the-wild* proof-of-concept deployment. We prove the concept of *cross-format robustness*, demonstrating that aggressive WebP edge compression does not degrade the semantic features required for Deep Learning inference, achieving a moderation Macro F1-score of over 96.8%.

The remainder of this paper is organized as follows: Section II discusses related works. Section III presents the image format analysis. Section IV details the proposed dual-layer architecture. Section V presents the experimental evaluation and results. Finally, Sections VI and VII discuss the implications of the findings and conclude the paper, outlining future research directions.

II. Related Work

The storage and transmission of digital images pose significant challenges in mobile applications, particularly as large volumes of photographs are generated and shared through modern platforms [6]. To mitigate these issues, various compression and optimization techniques have been explored. Lara et al. [7] proposed converting images from the RGB color space to YCbCr, combined with run-length encoding, to reduce real-time data transmission. While effective for highly constrained hardware such as UAVs, this approach is not directly applicable to mobile applications that require widely supported web formats.

In the context of mobile applications, Mousavirad and Alexandre [6] investigated energy-aware JPEG image compression, addressing the trade-off between file size, image quality, and energy consumption. Their approach allows the desired file size to be considered during the compression process and evaluates multiple optimization strategies for JPEG compression. Dornauer and Felderer [8] evaluated web image formats and reported advantages of next-generation formats such as WebP over traditional JPEG and PNG, particularly regarding the trade-off between visual quality, file size, and compatibility with modern browsers and mobile platforms.

As image sharing becomes easier, the risk of inappropriate or Not Safe For Work (NSFW) content also increases, particularly in professional and academic environments. Manual moderation becomes impractical as the volume of user-generated content grows, motivating the use of Artificial Intelligence (AI) and deep learning for automated content moderation. Deep neural networks, particularly Convolutional Neural Networks (CNNs), have been extensively explored for NSFW detection, including open-source baselines and commercial services such as Yahoo Open NSFW and Google Cloud Vision [9], [10]. Bicho et al. [5] evaluated deep neural network architectures, including ResNet and SqueezeNet, for NSFW classification, while Chen et al. [4] proposed a deep one-class classification model with visual attention for pornographic image recognition.

These works primarily address image compression or content moderation as separate problems. In contrast, our framework integrates client-side image compression with server-side AI-based NSFW moderation. Furthermore, rather than relying on a single fixed classification decision, the proposed moderation layer uses configurable safeness thresholds to automatically accept or reject high-confidence cases while forwarding ambiguous images to manual review. This combination enables the system to jointly address bandwidth reduction and content governance in mobile photo-sharing applications.

III. Image Compression and Format Analysis

The transmission of User-Generated Content (UGC) introduces severe network and storage bottlenecks in mobile applications, especially in event environments where hundreds of users share photos simultaneously. To illustrate the scale of this issue, between 2012 and 2022, the average mobile page weight increased by approximately 600%, with visual content being the primary contributor [3]. In participatory sensing environments with constrained wireless networks, transferring raw, high-resolution image files not only consumes excessive user data but also drastically increases server latency. Therefore, selecting an optimal, lightweight image format is a fundamental step for any mobile event gallery.

To determine the most suitable format, traditional and modern image encodings were evaluated. JPEG remains the standard for photography but carries substantial metadata, resulting in heavier files. While next-generation formats like AVIF offer exceptional compression rates [11], their inconsistent compatibility across diverse mobile runtimes limits their use in universally accessible applications.

Consequently, WebP was selected as the optimal solution. Designed specifically to reduce web payload, WebP consistently outperforms JPEG in Peak Signal-to-Noise Ratio (PSNR) and file size reduction [12]. Furthermore, studies demonstrate that the transition from JPEG to WebP preserves color gamut and luminance [13]. This preservation guarantees that the compression significantly reduces network payload and storage costs while maintaining the high visual fidelity required for a responsive user experience.

IV. Proposed Architecture

This section details the proposed decoupled architectural framework designed to optimize mobile data traffic and automate content governance in distributed image-sharing environments. The architecture is conceptualized to be platform-independent, meaning its technical components can be instantiated across any hybrid or native mobile system. As illustrated in the general overview in Fig. 1, the blueprint is divided into two primary independent layers: a client-side module responsible for local data optimization and a server-side module focused on adaptive security and content moderation.

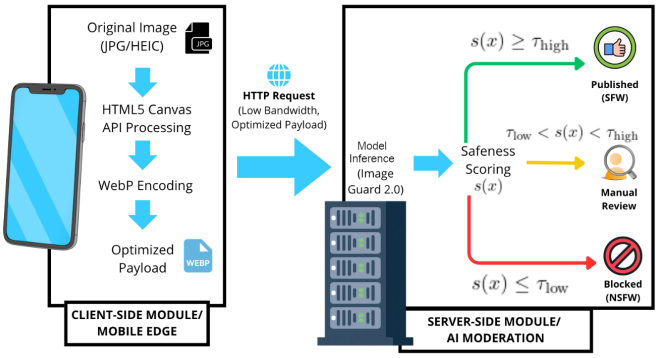


Fig. 1. Overview of the proposed dual-layer architecture, illustrating the client-side WebP compression pipeline and the server-side adaptive AI moderation mechanism.

A. Client-Side Optimization

A critical challenge in mobile media transmission is that traditional server-side compression still requires the end-user device to transmit the heavy, uncompressed original file over constrained mobile networks. To mitigate this data-link bottleneck, our framework shifts the processing paradigm entirely to the client-side, operating *on-the-fly* before any network request is initialized.

When a user captures or selects an image within the mobile client, the application utilizes the HTML5 Canvas API to process the binary file locally. The original image payload (e.g., JPEG or HEIC) is drawn onto an off-screen virtual canvas element, while preserving its original aspect ratio. The pixel data is then immediately re-encoded into the WebP format utilizing a standardized compression quality factor of 0.7. The quality factor was set to 0.7 based on preliminary evaluation of the trade-off between file size and visual fidelity. Consequently, only the optimized, lightweight WebP payload is packaged into the HTTP multipart request, conserving user cellular data and minimizing transmission latency.

B. Automated Content Moderation (NSFW)

While mitigating network and storage constraints enables a high-throughput, participatory multimedia gallery, it concurrently introduces a critical security vector: the uncurated upload of User-Generated Content (UGC). In professional, academic, or enterprise deployments, the exposure to inappropriate or Not Safe For Work (NSFW) imagery can permanently compromise the hosting platform's integrity. Given the highly dynamic and real-time nature of mobile image sharing, manual stream curation is fundamentally unscalable.

To solve this governance problem, the server-side architecture introduces an adaptive Artificial Intelligence moderation microservice that operates asynchronously at the backend entry point. Upon receiving the optimized client-side payload, the server intercepts the incoming WebP stream and routes it directly to a specialized classification pipeline. Because the backend receives an already compact WebP matrix instead of a heavy raw file, I/O operations and model inference latencies are minimized, reducing the amount of data processed and transmitted.

For the visual classification task, we selected Image Guard 2.0 [14], an experimental lightweight vision-language encoder with 0.1B parameters (<100M) trained on SigLIP2 (siglip2-base-patch16-224) [15]. This model was chosen after we achieved a higher macro accuracy across multiple relevant categories when compared with alternative models such as Falconsai/NSFW Image Detection [16] and Marqo/NSFW Image Detection 384 [17], both of which are based on Vision Transformers (ViT) [18] and perform binary classification only. Its multi-class output also provides a finer-grained representation of content, which can subsequently be mapped to the binary SFW/NSFW decision required by the application.

To define the chosen model and evaluate the image classification, we used the NSFW Detection Dataset [19] defined as follows: Let D be the initial set of the images, with $|D_0| = 28{,}000$, originally partitioned into the set of classes $C_0 = \{\text{Neutral}, \text{Drawings}, \text{Sexy}, \text{Hentai}, \text{Porn}\}$, each containing 5,600 samples. We therefore removed images from the Drawings and Hentai classes to evaluate only real photos, and defined the evaluation set as $D \subset D_0$, such that $|D| = 16{,}800$ and its classes belong to the original classes $C = \{\text{Neutral}, \text{Sexy}, \text{Porn}\}$.

In order to transform this problem into a binary classification problem, we define the mapping function $g : C \to Y$, where $Y = \{\text{SFW}, \text{NSFW}\}$. Thus, $g(\text{Neutral}) = \text{SFW}$ and $g(\text{Sexy}) = g(\text{Porn}) = \text{NSFW}$. Following this mapping, the set D is divided into D_{SFW} and D_{NSFW}, with $|D_{\text{SFW}}| = 5{,}600$ images ($\sim 33.33\%$) and $|D_{\text{NSFW}}| = 11{,}200$ images ($\sim 66.66\%$).

For each input image x, the Image Guard 2.0 outputs a probability vector $\mathbf{p}(x) = (p_1, p_2, p_3, p_4, p_5)$ over the set of classes:

$$M = \{\text{Normal-SFW, Anime-SFW, Enticing or Sensual, Hentai, Pornography}\}. \quad (1)$$

The predicted label is defined by $\hat{c}(x) = \arg\max_{m \in M} p_m(x)$.

To convert this multi-class output into a binary moderation decision, we define two subsets: $M_{\text{safe}} = \{\text{Normal-SFW}, \text{Anime-SFW}\}$ and $M_{\text{unsafe}} = M \setminus M_{\text{safe}}$.

Each image is then assigned a continuous safeness score $s(x) \in [0, 1]$, defined as:

$$s(x) = \begin{cases} \sum_{m \in M_{\text{safe}}} p_m(x), & \text{if } \hat{c}(x) \in M_{\text{safe}}, \\ 1 - \sum_{m \in M_{\text{unsafe}}} p_m(x), & \text{if } \hat{c}(x) \in M_{\text{unsafe}}. \end{cases} \quad (2)$$

Values closer to 0 indicate strong evidence of NSFW content, while values closer to 1 indicate SFW content.

The moderation strategy is governed by two configurable thresholds, τ_{low} and τ_{high} ($0 \leq \tau_{\text{low}} < \tau_{\text{high}} \leq 1$), allowing to dynamically adjust the strictness according to the application requirements. These thresholds create three disjoint regions: $R_{\text{NSFW}} = [0, \tau_{\text{low}}]$, $R_{\text{MOD}} = (\tau_{\text{low}}, \tau_{\text{high}})$, and $R_{\text{SFW}} = [\tau_{\text{high}}, 1]$. The automated decision $h(x)$ is thus given by:

$$h(x) = \begin{cases} \text{NSFW}, & \text{if } s(x) \leq \tau_{\text{low}}, \\ \text{SFW}, & \text{if } s(x) \geq \tau_{\text{high}}, \\ \text{MOD}, & \text{otherwise.} \end{cases} \quad (3)$$

Images falling into the R_{MOD} region are not automatically decided; instead, they are flagged for quick manual review via the web dashboard. The selection of these thresholds was guided by the empirical distribution of $s(x)$ and the rounded threshold values that maximize the balance between purity and coverage, as detailed in Fig. 2. As illustrated by the histogram, the dataset is heavily polarized, with NSFW images clustering near a safeness score of 0 and SFW images near 1. The overlaid curves represent the trade-off metric ($T = \alpha \times \text{Purity} + (1 - \alpha) \times \text{Coverage}$), where Purity corresponds to the precision of correct predictions, i.e., the proportion of correct predictions among all automatic predictions; coverage measures the proportion of the dataset that receives an automatic decision and α controls the relative weight assigned to each factor in the weighted sum. The peaks of these curves mathematically indicate the optimal strictness zones: approximately 0.2 to 0.4 for τ_{low} and 0.75 to 0.95 for τ_{high}. For this experimental setup, we utilized a server equipped with an Intel Core i9-14900K, 125 GB of RAM, and an NVIDIA A40 GPU.

Finally, to account for potential false negatives or individual privacy concerns, the system features a user-driven reporting mechanism. Participants can flag published images directly through the mobile application, specifying the reason for their denunciation. These reports are centralized in the organizer's web dashboard, enabling real-time review and the immediate removal of any contested content. This collaborative fallback ensures an additional layer of security, community self-regulation, and privacy compliance.

V. Evaluation and Results

To validate the practical feasibility and evaluate the performance implications of this model in an active environment, the architecture was deployed as a proof-of-concept case study within *myMobiConf*, a context-aware distributed mobile system designed for academic and corporate events. Fig. 3 illustrates the gallery interface integrated into the mobile client, which handles the real-time upload and display of event

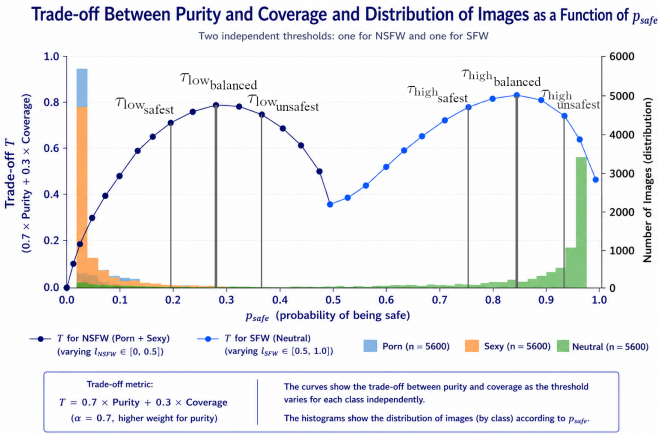


Fig. 2. Empirical distribution of p_{safe} (bars) and the corresponding purity-coverage trade-off curves (lines). The optimal safeness thresholds τ_{low} and τ_{high} are identified within the ranges around the peaks of the respective curves.

photographs. In this instantiation, the data compression loop is executed directly within the user's mobile interface, while the automated moderation backend intercepts and routes the processed payloads. Through this deployment, the system was evaluated across three dimensions: network efficiency, visual fidelity, and AI moderation performance.

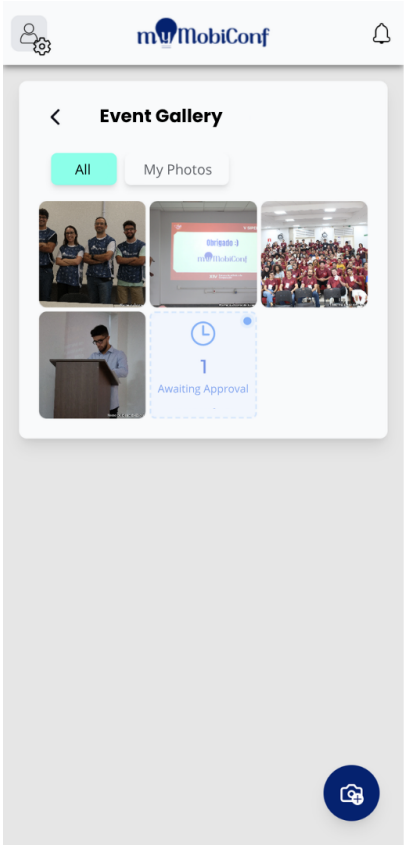


Fig. 3. Gallery interface implemented within the myMobiConf mobile client, showcasing the media submission flow.

A. Network and Storage Efficiency

To evaluate the operational stability of the client-side module, an *in-the-wild* micro-benchmark was conducted during an academic event. A controlled subset of participants utilized the modified application in a real-world participatory sensing environment. During this initial deployment, 37 photographs were submitted. Despite the limited sample size, the cumulative compression ratio exhibited a highly consistent and approximately linear behavior across the evaluated inputs.

While the raw original images (JPEG/HEIC) would have consumed 356.75 MB, the local client-side conversion reduced the payload to a total of just 7.40 MB. For this deployment sample, this corresponds to an approximately 98% reduction in payload size. As shown in Fig. 4, the cumulative storage growth of the WebP files remained substantially lower than that of the original images while maintaining an approximately linear progression, indicating the potential of client-side conversion to reduce storage and transmission requirements in larger deployments.

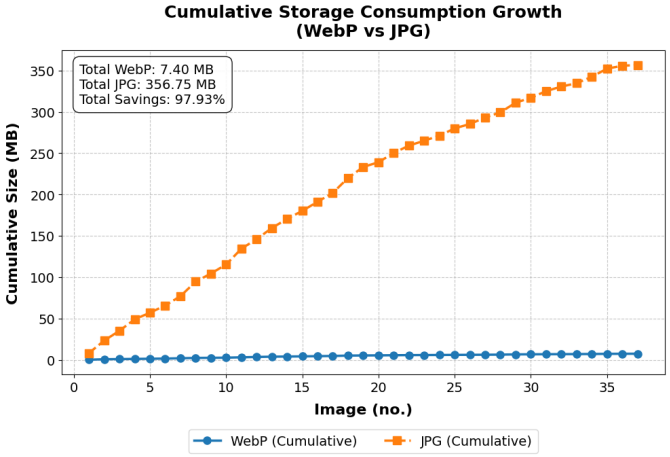


Fig. 4. Cumulative storage consumption: WebP vs. Original (JPG)

To further evaluate the client-side overhead of local compression, we profiled the application's current consumption during real-world usage. Capturing a photo, performing the WebP conversion via HTML5 Canvas, and transmitting the optimized payload resulted in a measured current of approximately 974 μA. For comparison, a standard in-app interaction measured approximately 750 μA. The relatively small difference between the two interactions indicates that the compression pipeline introduced limited additional current consumption on the tested device.

B. Visual Quality Assessment

To ensure the client-side compression did not degrade the visual features required for the subsequent NSFW classification model, structural metrics were rigorously analyzed. As shown in Fig. 5 and Fig. 6, the Peak Signal-to-Noise Ratio (PSNR) achieved a median above 40 dB. In image processing literature, PSNR values exceeding 40 dB are widely considered indicative of excellent reconstruction quality, where the compressed image is virtually indistinguishable from the source [12].

Furthermore, the Structural Similarity Index (SSIM) remained above 0.96. Since SSIM values are normalized between 0 and 1, where a value closer to 1 denotes higher structural preservation, a threshold of 0.96 confirms that structural integrity, borders, and luminance were strongly preserved. These results indicate that the compression preserves relevant low-level structural characteristics of the images. This supports the feasibility of using the compressed images for subsequent AI-based analysis.

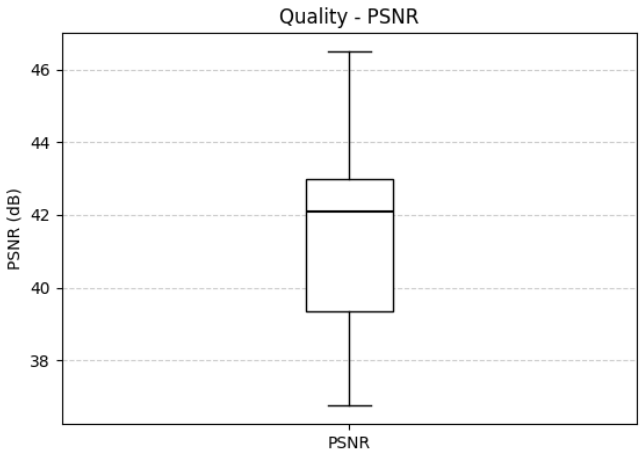


Fig. 5. PSNR distribution comparing original and WebP.

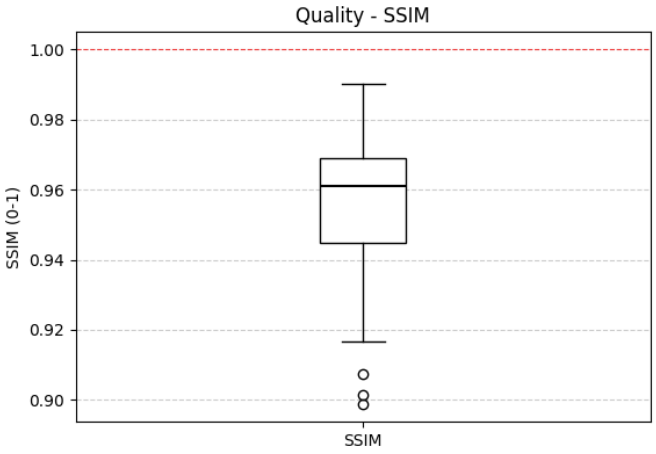


Fig. 6. SSIM distribution comparing original and WebP.

C. NSFW Moderation Performance

We evaluated the NSFW moderation model on two datasets. The first is a publicly available dataset [19] comprising 16,800 images labeled according to the categories Neutral, Sexy, and Porn. This dataset allowed us to assess how well our approach classifies images in a large and established benchmark. The second is a real-world dataset comprising 37 images obtained during the use of the deployed myMobiConf application. All

images in this dataset were labeled as neutral by our team, and the evaluation results allowed us to investigate false-positive and false-negative rates in a real-world scenario.

Given the high structural preservation (SSIM > 0.96) observed in the previous subsection, we further evaluated the moderation model on the large-scale dataset D comprising 16,800 images to assess its classification performance.

Based on the safeness regions defined previously, we evaluated three threshold configurations (Safest, Balanced, Unsafest), denoted by $L = \{\ell_1, \ell_2, \ell_3\}$, where each level ℓ_i is represented by a pair $(\tau_{\text{low}}^{(i)}, \tau_{\text{high}}^{(i)})$. For each level, the subset of automatically classified images is defined as

$$A_i = \{x \in D \mid s(x) \leq \tau_{\text{low}}^{(i)} \vee s(x) \geq \tau_{\text{high}}^{(i)}\}, \qquad (4)$$

while the subset forwarded for manual moderation is $D \setminus A_i$. The automatic coverage for each moderation level (Safenest, Balanced, Unsafest) is given by $\text{Cov}(\ell_i) = |A_i|/|D|$. For A_i, we calculated accuracy, macro precision, macro recall, and macro F1-score, considering only automated decisions $h(x) \in \{\text{SFW}, \text{NSFW}\}$. The results are presented in Tab. I.

TABLE I
SAFENESS LEVELS

Level	τ_{low}	τ_{high}	Cov.	Acc.	Macro P.	Macro R.	Macro F1
Safenest	0.20	0.95	87,40%	98,20%	98,67%	96.75%	97.66%
Balanced	0.30	0.85	93,35%	97,71%	98,11%	96.48%	97.25%
Unsafenest	0.35	0.75	95,76%	97,30%	97,64%	96.11%	96.83%

In addition to aggregate metrics, we analyzed the distribution of correct and incorrect classifications by class. For each $\ell_i \in L$ and $y \in \{\text{SFW}, \text{NSFW}\}$, we accounted for correctly classified images $\{x \in A_i \mid h(x) = y \wedge g(c(x)) = y\}$ and incorrectly classified ones $\{x \in A_i \mid h(x) = y \wedge g(c(x)) \neq y\}$. This decomposition is presented in Fig. 7.

It is observed that the more restrictive the safeness level, the lower the number of incorrect classifications and the higher the number of images in the manual moderation group, and vice versa. Furthermore, the proportions of false positives and false negatives remain relatively stable across NSFW and SFW classes for the three levels. However, the size of the moderation set (MOD) grows significantly from one level to another.

We also evaluated the performance of our proposed model on the 37 images (before and after compression) obtained from the application myMobiConf deployed based on the proposed architecture, as described in the beginning of section V. We used the same mapping and classification strategy described by $g(c(a_i))$ and $h(a_i)$, respectively, for $\mathbf{A} = \{a_1, a_2, \ldots, a_{37}\}$ where all instances belong to SFW, i.e., $c(a_i) = SFW, \quad \forall\ a_i \in \mathbf{A}$. Despite a very small sample set, consisting only of digital photos from the True "SFW" class, the results presented in Fig. 8 show a similar distribution of samples for SFW class instances across the three levels (Safest, Balanced, Unsafest) that is similar to the one obtained for the NSFW Detection Dataset. For the sake of example, for the balanced configuration, 3 images (8%) were wrongly

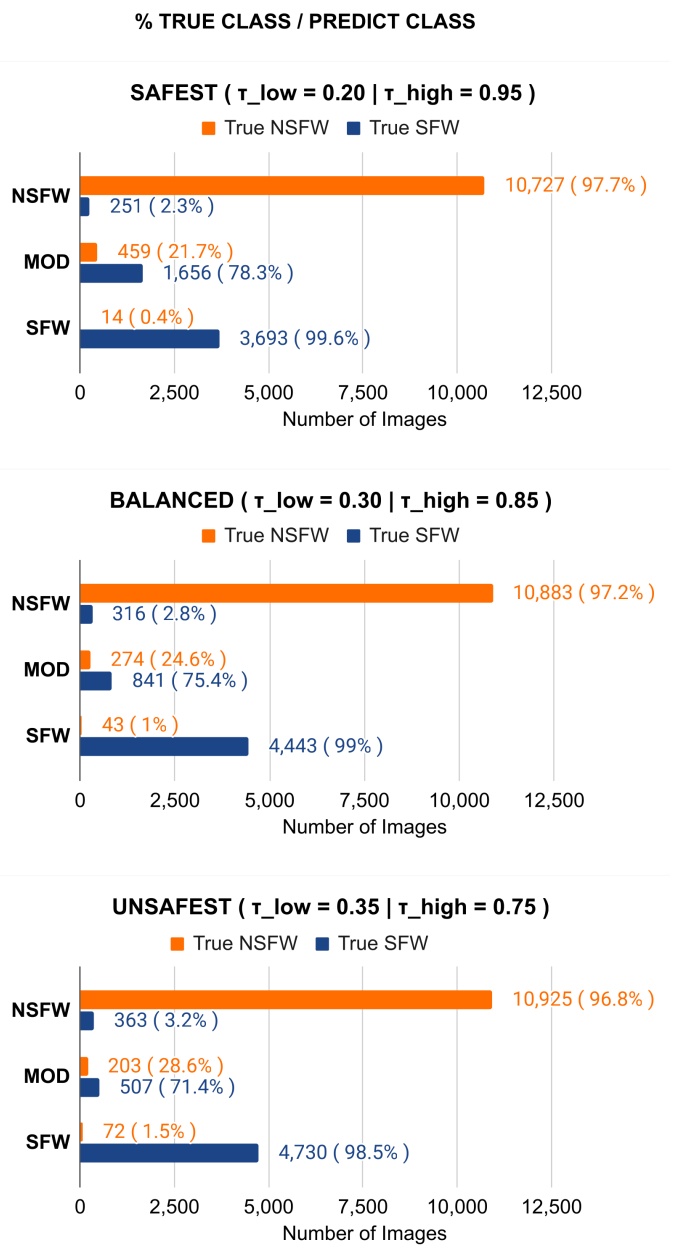


Fig. 7. Number of Images True NSFW and True SFW Classified by Predicted Class

classified as NSFW, and 7 (18%) were sent to moderation. The remaining 27 (74%) were correctly classified as SFW.

Regarding inference time, we empirically observed a variable rate between 196 and 220 images/second, which was considered feasible and scalable for the objectives of this architecture.

VI. DISCUSSION

The empirical evaluation of the proposed architecture highlights a successful resolution of the dual challenges inherent to mobile event galleries: network constraint mitigation and automated content governance.

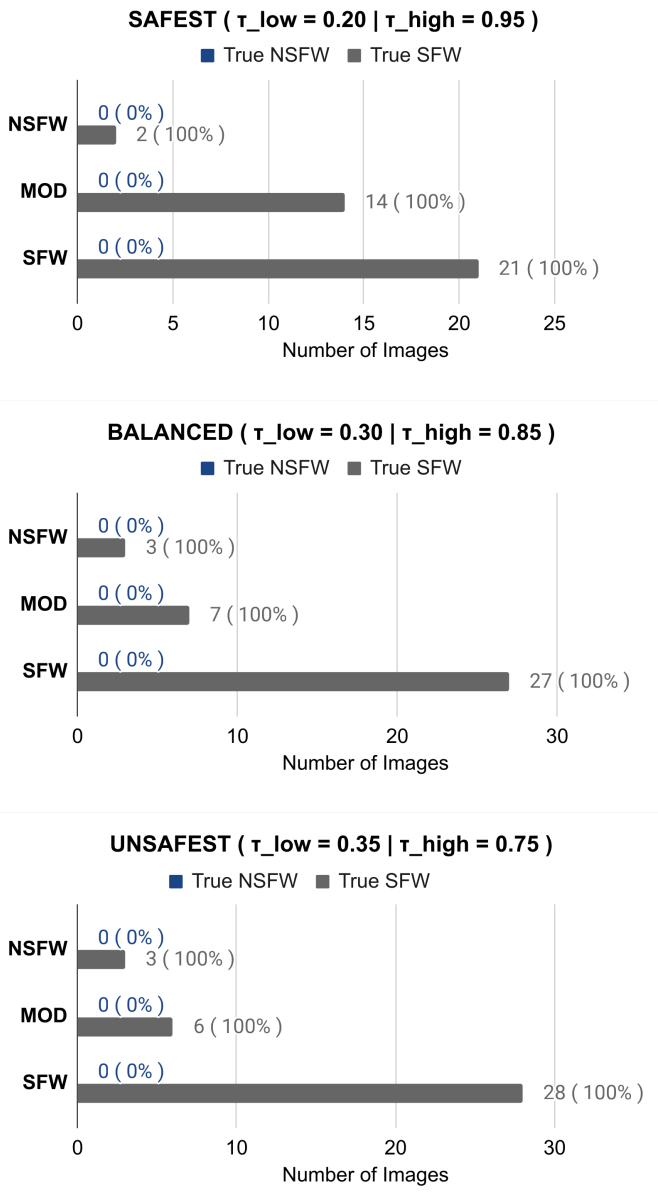


Fig. 8. Number of images classified as True NSFW and True SFW, by predicted class, for instance from an application based on the proposed architecture.

From a data transmission perspective, shifting the compression paradigm to the client-side fundamentally alters the resource consumption model. Traditional architectures rely on server-side processing, which still forces the user's device to transmit heavy raw files over potentially congested cellular networks—a common scenario in crowded academic or corporate events. By leveraging the HTML5 Canvas API for local WebP encoding, the framework achieved a stable 98% reduction in payload size. More importantly, the high Structural Similarity Index (SSIM > 0.96) demonstrates that this aggressive optimization does not destroy the spatial hierarchies and semantic borders of the images.

This visual preservation is the critical bridge to the success of the moderation layer. Because the structural integrity is maintained, the vision-language encoder based on SigLIP2 (Image Guard 2.0) operating on the backend can extract features with the same reliability as it would from uncompressed formats. Consequently, the architecture achieves *cross-format robustness*, meaning the system can aggressively compress data to save bandwidth without sacrificing the AI's classification accuracy.

Furthermore, the introduction of an adaptive tri-region moderation strategy (R_{NSFW}, R_{MOD}, R_{SFW}) addresses the inherent limitations of pure binary classification. In highly dynamic environments, false positives (blocking legitimate content) and false negatives (allowing explicit content) are equally detrimental. By establishing configurable thresholds, the architecture abstracts the rigidity of the AI, allowing event organizers to calibrate the strictness according to their specific community guidelines. The system successfully isolates ambiguous cases for manual review, drastically reducing the human moderation workload while guaranteeing platform safety.

VII. Conclusion and Future Work

In this work, we presented a decoupled, platform-independent architectural framework designed to optimize media uploads and automate content moderation in distributed mobile systems. By instantiating this model within the *myMobiConf* platform as a proof-of-concept, we validated that client-side *on-the-fly* WebP compression drastically reduces network latency and storage costs by 98%, while fully preserving the visual fidelity required for Deep Learning inference.

Simultaneously, the integration of an adaptive server-side AI layer demonstrated high precision in mitigating the risks associated with User-Generated Content. Achieving a Macro F1-score exceeding 96.8%, the system proved capable of autonomously blocking NSFW imagery and routing ambiguous content to a manageable manual review queue. The resulting architecture is modular, scalable, and readily applicable to any modern mobile application requiring robust multimedia governance.

As a further contribution, we proposed a strategy for mapping five classes to two using a pre-trained classification model and three combinations of safety criteria to achieve greater control over automated moderation.

Despite the successful validation, several avenues remain for future exploration. Based on the insights gained, we propose the following future works:

- **Edge Computing Inference:** Investigate shifting the AI moderation layer directly to the mobile client device. Performing lightweight NSFW classification locally *prior* to the WebP conversion and network transmission would entirely eliminate server-side inference costs and further enhance user privacy.
- **Multi-modal Moderation Context:** Expand the moderation pipeline to include Natural Language Processing (NLP) for analyzing image captions and metadata. Combining visual classification with textual

context can significantly reduce the R_{MOD} manual review queue in ambiguous cases.

- **Federated Learning Adaptability:** Implement federated learning mechanisms to allow the classification model to adapt to specific event niches. For instance, medical conferences may feature anatomical imagery that triggers false positives in generic NSFW models; localized training could dynamically adjust the model's contextual understanding without centralizing sensitive user data.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Research Support Foundation of the State of Minas Gerais (FAPEMIG), CNPq and CAPES.

REFERENCES

[1] B. JA, D. Estrin, M. Hansen, A. Parker, R. Nithya, S. Reddy, and M. Srivastava, "Participatory sensing," *Workshop on World-Sensor-Web (WSW): Mobile Device Centric Sensor Networks and Applications*, 01 2006.

[2] InEvent, "The future of event apps," https://inevent.com/blog/tech-and-trends/the-future-of-event-apps.html, 2019, accessed on: December 5, 2025.

[3] HTTP Almanac, "Page weight," https://almanac.httparchive.org/en/2022/page-weight#fig-8, 2022, accessed on: October 24, 2025.

[4] J. Chen, G. Liang, W. He, C. Xu, J. Yang, and R. Liu, "A pornographic images recognition model based on deep one-class classification with visual attention mechanism," *IEEE Access*, vol. 8, pp. 122 709–122 721, 01 2020.

[5] D. Bicho, A. Ferreira, and N. Datia, "A deep learning approach to identify not suitable for work images," *i-ETC: ISEL Academic Journal of Electronics, Telecommunications and Computers*, vol. 6, no. 1, pp. 1–11, 2020, iD-3.

[6] S. J. Mousavirad and L. A. Alexandre, "Metaheuristic-based energy-aware image compression for mobile app development," *Multimedia Tools and Applications*, vol. 84, pp. 8413–8454, 2025.

[7] V. Lara, C. Costa, and L. Guimarães, "Análise experimental da conversão rgb/ycbcr e codificação run-length para compressão de imagens em sistemas de tempo real baseados em vants / experimental analysis of rgb/ycbcr conversion and run-length encoding for image compression in uav-based real-time systems," *Brazilian Journal of Development*, vol. 7, pp. 73 063–73 075, 07 2021.

[8] B. Dornauer and M. Felderer, *Web Image Formats: Assessment of Their Real-World-Usage and Performance Across Popular Web Browsers*, 12 2023, pp. 132–147.

[9] Yahoo Engineering, "Open nsfw model," GitHub Repository, 2016, accessed on: July 8, 2026. [Online]. Available: https://github.com/yahoo/open_nsfw

[10] Google Cloud, "Cloud vision api: Safesearch detection," https://cloud.google.com/vision/docs/detecting-safe-search, 2026, accessed on: July 8, 2026.

[11] Google, "Image formats: Avif," https://web.dev/learn/images/avif, 2023, accessed on: December 5, 2025.

[12] G. Ginesu, M. Pintus, and D. D. Giusto, "Objective assessment of the webp image coding algorithm," *Signal Processing: Image Communication*, vol. 27, no. 8, pp. 867–874, 2012, special issue on: pervasive mobilemultimedia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596512000203

[13] M. A. Matijevic, M. Mikota, and M. Čačić, "Impact of jpeg-webp conversion on the characteristics of the photographic image," *Tehnicki Vjesnik-technical Gazette*, vol. 23, pp. 505–509, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:9063770

[14] PrithivMLmods, "Image guard models," https://huggingface.co/blog/prithivMLmods/image-guard-models, 2025, accessed: 2026-07-07.

[15] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa *et al.*, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," *arXiv preprint arXiv:2502.14786*, 2025.

[16] Falcons.AI, "Falconsai/nsfw_image_detection," https://huggingface.co/Falconsai/nsfw_image_detection, 2023, model card. Accessed: 2026-07-07.

[17] R. Wightman, "Pytorch image models," https://github.com/huggingface/pytorch-image-models, 2019.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[19] deepghs, "Nsfw detection dataset," https://huggingface.co/datasets/deepghs/nsfw_detect, 2023, multi-class image classification dataset for NSFW content detection with five categories: drawing, hentai, neutral, porn, and sexy.