

Scalable and Accessible AR Education: Integrating Cloud-Optimized CNNs and Dynamic Multilingual Synthesis in Wildlife Visualization

Neha Rahul Patil¹, Srushti Pradeep Sawant², Aaliya Amjad Patel³,
Avneet Kaur Amar Singh Panesar⁴, and Purva Ghanasham Pardeshi⁵,
K. R. Kava⁶, and S. K. Mandlik⁷

^{1,2,3,4,5}Second Year Students of Artificial Intelligence and Machine Learning Department
Guru Gobind Singh Polytechnic, Nashik

^{6,7}Faculty, Guru Gobind Singh Polytechnic, Nashik

Abstract: *This paper proposes a scalable, multilingual extension of the 3D Wildlife AR concept. Cloud GPU training is separated from server-side animal classification, local augmented-reality rendering, and asynchronous translation and text-to-speech. Versioned releases, confidence-based abstention, reviewed terminology, and content-aware caching support reliable delivery without making visualization wait for narration. A controlled evaluation protocol addresses training efficiency, recognition quality, interaction latency, language accuracy, cost, and learning outcomes. Analytical latency models distinguish visual readiness from audio availability. No implementation measurements or participant data are available; the contribution is an architecture and validation framework, not demonstrated performance or educational improvement.*

Keywords: augmented reality, wildlife education, cloud training, convolutional neural networks, multilingual text-to-speech

I. INTRODUCTION AND RELATED WORK

Wildlife AR connects animal images with interactive three-dimensional models and explanations of taxonomy, habitat, and behavior. However, workstation-bound training constrains model iteration, while English-only narration restricts linguistic access. This proposal extends the supplied 3D Wildlife AR concept through independently provisioned training, recognition, and localization services. No prior publication or implementation of that project was supplied; neither a novel CNN nor an unprecedented combination of technologies is claimed.

A synthesis of 64 studies found a medium overall AR learning effect [1]; subsequent work emphasizes pedagogical dependence [2]. These findings motivate, but do not validate, wildlife learning. Cloud-edge separation motivates keeping latency-sensitive rendering local [3]. MobileNetV2 offers an efficient reference backbone [4], while iNaturalist documents fine-grained ambiguity and class imbalance [5]. Translation glossaries and voice discovery support localization [6], [7], not guaranteed scientific accuracy.

The contributions are controlled cloud model releases, independently delivered multilingual narration, and a factorized evaluation. Three questions guide validation: does cloud training improve time-to-quality and cost; does localization preserve visual responsiveness while providing timely audio; and does reviewed preferred-language narration improve comprehension and usability?



II. ARCHITECTURE AND RECOGNITION METHOD

A. Training, Release, and Classification

Cloud object storage holds images and versioned manifests containing image identifiers, labels, acquisition groups, permissions, and split assignments. Containerized single-GPU jobs record code/data versions, stage data near compute, prefetch inputs, and persist checkpoints before releasing resources. “Cloud-optimized” denotes resource allocation and pipeline design, not proven acceleration or multi-GPU optimization. Training and inference use separate reserved resources.

A release bundles weights, preprocessing, ordered class labels, calibration, and validation results. Quality and compatibility checks precede warm-up and an atomic version switch with rollback. Stable species identifiers connect predictions, educational records, and AR assets, preventing class-map mismatches.

The Unity client sends an image, locale, and request identifier to a Python inference API. The task is dominant-animal classification, not multi-object detection; ambiguous framing requires recapture. Responses contain species/confidence, model/content versions, asset identifier, and localization status. Classification selects the asset; local plane detection anchors it. Inference remains server-side, so cloud training alone cannot remove upload latency or enable offline recognition.

B. Dataset and CNN Configuration

Dataset source, size, and class count K remain to be finalized. Proposed 70/15/15% training/validation/test splits keep observations, sequences, and near-duplicates together; photographer/location separation is added where feasible. Report per-class counts and split deviations. Unfamiliar species, empty scenes, and unsuitable images form a separate challenge set. Only training images receive augmentation.

TABLE I
REFERENCE TRAINING PROTOCOL

Parameter	Proposed setting
Schedule	5 head-only epochs at 10–3 learning rate; 25 fine-tuning epochs at 10–4
Fine-tuning	Unfreeze convolution weights; freeze batch-normalization statistics
Hardware control	Batch 32, full precision; common smaller batch if required
Augmentation	Horizontal flips, modest rotation, brightness variation
Selection/repeats	Best validation macro-F1; five paired seeds with matched initialization/data order

The reference model uses ImageNet-pretrained MobileNetV2 (width 1.0), 224×224 RGB input with matching

$$\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i=1}^B \sum_{k=1}^K y_{ik} \log p_{\theta}(k | x_i), \quad (1)$$

preprocessing, global average pooling, and a K -class softmax head. Adam

[8] minimizes categorical cross-entropy,

where B is batch size, y_{ik} the one-hot target, and p_{θ} the predicted probability. Table I specifies proposed settings, not executed experiments.

Serialize augmentation bounds, optimizer parameters, software versions, and determinism settings. Fixed-epoch runs complete all 30 epochs; time-to-quality runs use a preregistered target. Mixed precision, larger batches, weighted loss, stronger augmentation, and alternative backbones are separate ablations. Temperature scaling [9] and abstention thresholds are fitted on validation data against a declared accepted-error tolerance; report coverage. Low-confidence predictions prompt recapture or manual selection instead of narration. Softmax confidence alone does not guarantee unknown-species rejection.



C. Localization and Client Interaction

Reviewed English records contain scientific name, explanation, learning objective, and content version. Locale mapping (e.g., hi-IN) requires translation, voice, font, and reviewed-content support. The worker checks reviewed translations and cached audio, otherwise applies glossary-controlled translation and TTS. Scientific identifiers remain unchanged. Classroom delivery requires review; unreviewed generation is tested only in a technical sandbox.

Text and audio caches are separate. Audio keys include species, source version, locale, translation/review and glossary revisions, provider configuration, voice, rate, and encoding. Record exposed provider revisions or local configuration revision and generation time; coalesce identical concurrent jobs. Recognition metadata returns with a narration handle without waiting for language services. Bounded polling or notification retrieves downloaded audio independently of asset loading; streaming synthesis is not assumed. Reviewed captions can precede narration.

Vuforia Ground Plane requires compatible devices [10]; others receive a non-AR viewer. Unity coroutines are not threads [11], so encoding, decoding, and asset preparation need profiling and background-safe or incremental processing. Local placement, rotation, and scaling target 60 FPS (16.7 ms/frame) on capable devices, not guaranteed performance. Request identifiers cancel stale responses/audio after animal or language changes. Timeouts, bounded retries, and rate limits accompany captions or explicitly selected fallback languages; language-service failures need not block visualization. Offline use is limited to downloaded content and manual selection. Replay, speech-rate and text-size controls support accessibility without claiming certification.

III. EVALUATION AND ANALYTICAL MODEL

A. Controlled Comparisons

Cross local/cloud training with English/multilingual narration to obtain four conditions. Hold inference resources, client, content, and assets constant; localization comparisons use identical weights. Test cold caches, text-only hits, audio hits,

and blocking versus independent delivery. Match English TTS/cache behavior to isolate translation overhead; bundled or pregenerated reviewed audio is a separate product baseline.

Report CPU/GPU, RAM/accelerator memory, host, storage, region, and software. CPU-to-cloud-GPU comparisons con-found hardware with deployment; a comparable local GPU control is preferable. Separate provisioning, staging, optimization, validation, and export. For matched total durations, speedup is $S = T_{\text{local}}/T_{\text{cloud}}$ and reduction is $100(1 - 1/S)\%$; also report compute-only speedup. Nested training subsets retain classes and fixed validation/test sets. Measure epoch time, throughput, peak memory, time-to-quality, and cost; larger-dataset claims require an explicit resource/turnaround budget.

Evaluate accuracy, macro-F1, class recall, confusion matrices, calibration, accepted accuracy/coverage, and challenge-set false acceptance/rejection. Report five-seed results and paired uncertainty intervals; bootstrap acquisition groups rather than correlated images. Freeze models and thresholds before inspecting test outcomes.

B. Runtime and Cost

Use client monotonic clocks to measure delays from capture to metadata, stable placement, and audio playback; synchronize clocks before cross-machine subtraction. Trace validation, queues, preprocessing, inference, lookup, translation, synthesis, and retrieval. Test validated English, Hindi, and Spanish configurations across device tiers and controlled round-trip delay (20/80/200 ms), bandwidth (1/5/20 Mbps), packet loss, and pilot-fixed load levels.

Each cell uses at least 200 requests in each of five independently started sessions, increasing samples if tail estimates remain uncertain. Report median/p95 latency, session-clustered intervals, timeouts, and failures; distinguish service cold starts from content misses. Sustained tests record frame times, drops, memory, thermal behavior, and battery use. Report cost per validated model and per 1,000 completed interactions, including compute, storage, transfer, inference, translation, and TTS; disclose local depreciation/electricity accounting.



C. Latency Decomposition

Let T_{rec} include capture, encoding, upload, queueing, pre-processing, inference, and lookup; T_{meta} and T_{asset} denote metadata delivery and asset loading. Then

$$\begin{aligned} T_{ready} &= T_{rec} + T_{meta} + T_{asset}, \\ T_{visual} &= \max(T_{ready}, T_{plane}) + T_{place}, \end{aligned} \quad (2)$$

where plane readiness is measured from capture and placement excludes separately reported user deliberation. Already available planes/assets contribute no remaining delay. Localization is absent from this path, although shared-resource contention can affect it.

$$\begin{aligned} T_{audio,miss} &= T_{rec} + T_{dispatch} + T_{cache} \\ &\quad + T_{translation} + T_{synthesis} \\ &\quad + T_{delivery} + T_{decode}. \end{aligned} \quad (3)$$

For an audio miss without reusable translation,

T_{cache} denotes cache lookup; dispatch includes worker queues, and delivery includes status discovery, transfer, and buffering. Human review is excluded and reported separately. Full hits substitute cached retrieval for generation; text-only hits still require synthesis. For audio-hit probability h ,

$$\begin{aligned} \mathbb{E}[T_{audio}] &= h\mathbb{E}[T_{audio} | \text{hit}] \\ &\quad + (1 - h)\mathbb{E}[T_{audio} | \text{miss}]. \end{aligned} \quad (4)$$

This conditional expectation is not a measured prediction: improved mean latency need not improve cold-start tails, and no universal acceptable overhead is assumed.

D. Language Quality and Learning

Bilingual reviewers assess scientific accuracy, terminology, fluency, pronunciation, and intelligibility, including ambiguous common/scientific names; report agreement and adjudication. Automated metrics are supplementary. Randomize English versus reviewed preferred-language narration while matching content, interaction, exposure, and assessment objectives; block by proficiency and account for classroom clustering if applicable. Review assessment translations for equivalence. Pretests, immediate comprehension tests, and delayed retention tests separate learning from engagement. The primary endpoint is delayed score adjusted for pretest; usability, task completion, and comprehensibility are secondary. Determine sample size prospectively from pilot variance and a meaningful effect. Ethics approval, guardian permission, and age-appropriate assent precede child participation.

IV. LIMITATIONS AND CONCLUSION

All performance and learning outcomes remain unmeasured. Cloud GPUs may shorten optimization while staging/provisioning dominates small jobs; faster training need not cost less. Caching trades generation delay against storage, and pregenerated narration may suit stable catalogs. Inference queues, API quotas, connectivity, client limits, domain shift, and rare-class imbalance remain bottlenecks. Voice availability does not establish scientific accuracy, appropriate literacy/culture, or accessibility; preferred-language benefit depends on proficiency and pedagogy.

Process camera images transiently without default retention or image-content logs. Keep credentials server-side, restrict dataset access, and integrity-check models/assets. Voluntary contributions require separate consent, review, and a new manifest; uploads never enter training automatically. The application is educational, not safety-critical wildlife identification.

The design separates model development, visual interaction, and narration for independent validation. Future work includes quantized on-device inference (accuracy, memory, energy, offline testing), curriculum-based audio prefetching, and conversational explanations grounded in reviewed records, with separate factuality, age-appropriateness, privacy, and learning evaluations. No code, datasets, logs, or participant records accompany this



proposal; future releases should provide permitted manifests, configurations, versions, evaluation scripts, and anonymized aggregate results.

REFERENCES

1. J. Garzón and J. Acevedo, "Meta-analysis of the impact of Augmented Reality on students' learning gains," *Educational Research Review*, vol. 27, pp. 244–260, 2019.
2. J. Garzón, Kinshuk, S. Baldiris, J. Gutiérrez, and J. Pavón, "How do pedagogical approaches affect the impact of augmented reality on education? A meta-analysis and research synthesis," *Educational Research Review*, vol. 31, p. 100334, 2020.
3. M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
4. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, arXiv:1801.04381.
5. G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The iNaturalist species classification and detection dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8769–8778.
6. Google Cloud, "Creating and using glossaries (Advanced)," Cloud Translation documentation, n.d., official product documentation. Accessed September 10, 2026.
7. "Method: voices.list," Cloud Text-to-Speech REST API v1 reference, n.d., official product documentation. Accessed September 10, 2026.
8. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015, arXiv:1412.6980.
9. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. *Proceedings of Machine Learning Research*, vol. 70. PMLR, 2017, pp. 1321–1330.
10. PTC, "Ground plane," Vuforia Engine Library, n.d., official product documentation. Accessed September 10, 2026.
11. Unity Technologies, "Coroutines," Unity User Manual, version 2022.3, n.d., official product documentation. Accessed September 10, 2026.

