SkyWeather -cloud classification on mobile devices

Stefan Alexandru Glava
University of Craiova
Craiova, Romania
glava.stefan.d3r@student.ucv.ro

Paul Stefan Popescu
University of Craiova
Craiova, Romania
stefan.popescu@edu.ucv.ro

Marian Cristian Mihaescu
University of Craiova
Craiova, Romania
cristian.mihaescu@edu.ucv.ro

ABSTRACT

SkyWeather turns an ordinary Android phone into a pocket "sky-scanner" that recognises five cloud types (clear, veil, patterned, thick-white, thick-dark) completely offline, then offers short-term weather insight in under three seconds. The application couples a 6.9 MB MobileNetV2-based classifier, quantised with TensorFlow Lite, with a multilingual Jetpack-Compose interface that runs on mid-range devices. A hybrid dataset, SWIMCAT, which comprises more than 290 field photos captured across seventeen geo-climates, drives training. Targeted augmentation and class-weighted loss push validation accuracy to 94.7%, while keeping inference latency at 1.7 seconds end-to-end. Experiments confirm robust performance across varying daylight, altitude, and sensor noise conditions, with < 10 % confusion between visually similar classes. By eliminating server calls, SkyWeather ensures privacy, autonomy, and resilience for hikers, farmers, and first responders operating beyond network coverage. The project contributes (i) an open, mobile-ready cloud image corpus, (ii) an edge-optimized CNN pipeline, and (iii) a clean-architecture Android reference implementation, demonstrating that practical, citizen-oriented meteorology can live entirely on-device.

Author Keywords

Cloud classification; machine learning; mobile app; offline inference; TensorFlow Lite; Kotlin; Android; human-computer interaction.

ACM Classification Keywords

I.2.10 [Artificial Intelligence]: Vision and Scene Understanding – Perceptual reasoning

General Terms

Design; Measurement; Human Factors

DOI: 10.37789/icusi.**2025.24**

INTRODUCTION

SkyWeather addresses the growing need for local weather observation tools in areas lacking internet access. Existing weather apps rely heavily on data connectivity and centralized forecasting, which are inaccessible in remote conditions. We ask whether a compact and accurate machine learning model can provide real-time cloud classification entirely offline. We hypothesized that a lightweight model, trained on a hybrid dataset and optimized with TensorFlow Lite, can classify cloud types with over 90% accuracy when integrated into a native Android app.

Accurate, hyper-local weather knowledge empowers hikers, farmers, sailors and field researchers to make time-critical decisions; yet nearly every consumer-grade forecast service presumes continuous data coverage. When connectivity drops, even fundamental insight, such as "Will this towering cumulus evolve into a thunderstorm within the next hour?" becomes guesswork. SkyWeather addresses this blind spot by combining lightweight convolutional neural networks (CNNs), sensor-aware Android design and a multilingual, privacy-respecting user interface to deliver offline cloud-type recognition and short-term weather cues directly on the handset.

Smartphones already embed high-quality cameras, barometers, light-sensors, and GPUs. Recent research shows that a single skyward image encodes sufficient texture, opacity, and illumination cues to infer imminent meteorological changes, provided that a model can run locally with a latency of less than three seconds and a footprint of less than ten MB. Existing "citizen-science" apps either require server-side inference or limit themselves to manual cloud logging, neither of which satisfies back-country scenarios where bandwidth is scarce and battery trade-offs are severe.

Beyond convenience, an edge-only approach advances three broader goals. Educational literacy – users learn to associate visual cloud clues with weather dynamics, reinforcing atmospheric science curricula. Privacy & trust – photos never leave the device; no location trail is uploaded to a cloud API. All processing complies with local storage and permission scopes. Resilience – natural disaster first responders or trekkers in low-infrastructure regions gain a robust fallback when commercial forecast feeds fail.

The SkyWeather project contributes a curated hybrid dataset that merges the academic SWIMCAT (Singapore Whole sky IMaging CATegories Database) corpus with 290 field photos captured across 17 geo-climates and five solar angles, balancing minority classes (Veil, ThickWhite) through targeted sampling. A mobile-optimised CNN

(MobileNetV2-125² + custom head, 3.5 M parameters) quantised-float16 to 6.9 MB without appreciable accuracy loss. An Android Clean-Architecture reference that demonstrates end-to-end ML inference, sensor fusion, multilingual resources, and Jetpack Compose UI in under 35 K lines of Kotlin. An open-source evaluation suite comprising latency profiling scripts, confusion-matrix visualisers, and UI accessibility audits.

RELATED WORK

Cloud image classification has been explored using datasets such as SWIMCAT [11] and models like MobileNet and EfficientNet. Prior studies achieved high accuracy in controlled environments but lacked generalization for real-world images. Furthermore, most solutions require server-based inference, which contradicts the offline utility goal. Our approach combines custom image augmentation, dataset extension with real-world phone captures, and an entirely local mobile inference pipeline.

Cloud classification is an interesting and current task that has been discussed in many papers. One recent approach [1] addresses this problem by utilizing Himawari-8 Infrared Data and trains a cloud classification model, which achieves an overall accuracy of 86.22%, along with precision, recall, and F1-score values of 0.88, 0.84, and 0.86, respectively. The practicality of this model was validated across various scenarios, including all-day temporal, daytime/nighttime, and seasonal contexts. The results showed that the AInfraredCCM consistently performed well across multiple periods and seasons, confirming its temporal applicability. Another paper that tackles the same task is [2], which utilizes a new intelligent cloud classification method based on the U-Net network (CLP-CNN), developed to produce more accurate, higher-frequency, and larger-coverage cloud classification products. The experimental results show that the CLP-CNN network can complete a cloud classification task of 800 × 800 pixels in 0.9 s. The classification area covers most of China, and the classification task only needs to use the original L1-level data, which can meet the requirements of a real-time operation

A hybrid deep Kronecker network, ResNeXt, was used in the paper [3], which utilizes cloud segmentation performed through Swin-Unet to derive the cloud cover. Thereafter, features such as Entropy with Median Binary Pattern and statistical features, including skewness, mean, standard deviation, and variance, are mined. Subsequently, the generated features and input image are fed to the established model for cloud classification. The proposed model is formulated by merging Residual Networks with Aggregated Transformations and Deep Kronecker Networks. Lastly, the authors classified the output and preprocessed output as input to the cloud cover estimation module, where cloud cover is estimated using the proposed model. Furthermore, the proposed model is examined using metrics such as True Positive Rate, accuracy, and True Negative Rate. According to the experiment, the proposed model achieved an accuracy of 0.915. Also, the True Negative Rate of 0.923 and the True Positive Rate of 0.896 are obtained at a higher level.

Still, among the neural network approaches is also the paper [4], which presents a study based on a nonlinear, nonparametric four-layer neural network approach. The study compares a three-layer neural network architecture, the nonparametric K-nearest neighbor approach, and the linear stepwise discriminant analysis procedure.

Other new papers, such as [5], tackle the same problem but utilize pre-trained models. The study aims to predict cloud formations and classify them based on their shapes and colors, enabling the implementation of preemptive measures against potentially hazardous situations. To address this challenge, a solution is proposed using image processing and deep learning technologies to classify cloud images. Several models, including MobileNet V2, Inception V3, EfficientNet V2-L, VGG-16, Xception, ConvNeXt-Small, and ResNet-152 V2, were employed for the classification computations. Among them, Xception yielded the best outcome with an impressive accuracy of 97.66%

Other approaches [6] utilize several features that imply the use of more complex sensors. The proposed models take as input the observed reflectance or brightness temperature of 12 channels from the Advanced Geostationary Radiation Imager (AGRI) on the Fengyun-4A satellite, as well as multichannel clear-sky brightness temperatures. The classification results of the Cloud Profiling Radar (CPR)-Cloud-Aerosol Lidar with Orthogonal Polarization (CALIOP) merged product are used as the truth for training and validating the models. These models are developed to reliably detect and classify clouds during the day as well as at all times (including both day and night). The results obtained from the developed models demonstrate better accuracy compared to those of the Fengyun 4A Level-2 cloud products in terms of cloud detection and classification.

The importance of cloud detection is also addressed in [7], which states that low-level marine clouds play a pivotal role in Earth's weather and climate through their interactions with radiation, heat, and moisture transport, as well as the hydrological cycle. The paper applies a recently developed self-supervised learning scheme to train a deep convolutional neural network (CNN) to map marine cloud imagery to vector embeddings that capture information about mesoscale cloud morphology and can be used for satellite image classification. The model is evaluated against existing cloud classification datasets, and several use cases are demonstrated, including training cloud classifiers with very few labeled samples, interrogation of the CNN's learned internal feature representations, cross-instrument application, and resilience against sensor calibration drift and changing scene brightness.

Cloud classification is relevant for numerous systems, as a recent data-driven study [8] on distributed photovoltaic (PV) stations couples plant-level output records with concurrent meteorological reanalysis fields and applies K-means clustering to link extreme PV output anomalies to synoptic weather regimes. The authors demonstrate that abnormally high generation occurs under high-temperature, cloud-free conditions, typically produced by persistent low-pressure systems or by clear, windy cooling periods characterized by surface high pressure. Conversely, low-output extremes coincide with transitional synoptic situations, such as cold-wave passages, overcast but precipitation-free skies, or the cloudy and rainy environments typical of cyclonic lows. Leveraging these relationships, the paper proposes a lightweight classification-based extreme-output predictor. It demonstrates its skill, together with a subjective circulation-pattern forecast, on a severe low-generation episode in January 2023. The results suggest that combining objective weather-type models with expert synoptic assessments can materially improve day-ahead forecasting of PV extremes, providing a valuable template for hybrid approaches in renewable energy management.

Another line of research [9] reframes weather recognition as a continuous, uncertain phenomenon rather than a set of mutually exclusive labels. Instead of assigning a hard class to each image, the authors introduce a Gaussian-mixture formulation that captures both the probability level of individual weather states and their possible co-existence within the same scene. Building on a prior-posterior learning strategy, they design MeFormer, a transformer tailored to "multi-weather co-presence estimation," and publish the MePe dataset (16 078 outdoor images, 14 fine-grained weather categories) to benchmark this task alongside conventional multi-label classification. Experiments show that uncertainty-aware training not only sets a new state-of-the-art on weather tagging but also transfers to downstream tasks such as adverse-weather semantic segmentation, underscoring the value of modeling meteorological transitions in a physically grounded, probabilistic manner.

A complementary strand of work [10] focuses on model efficiency for on-device weather recognition. One systematic study benchmarks fourteen CNN backbones—ranging from classic VGG16 and ResNet-50 to mobile-oriented MobileNet, NASNetMobile, and EfficientNet variants—on a four-class (cloudy, rain, shine, sunrise) image set. After profiling inference latency, the seven fastest architectures were re-trained with grid-searched batch size and learning-rate combinations, yielding a fair comparison between speed and accuracy. The results highlight DenseNet-121 as the best trade-off, posting ≈ 0.98 validation/test accuracy while remaining lightweight enough for smartphones, with EfficientNet-B0 a close second. By coupling systematic hyper-parameter optimisation with runtime profiling, the study offers practical guidance for deploying deep-learning weather classifiers on resource-constrained edge and IoT devices, advancing the goal of energy-aware, sustainable AI in real-world applications.

PROPOSED APPROACH

The system described in this paper is also available on GitHub[1] and the code can be used for results reproducibility

The base corpus is SWIMCAT[2], which integrates 784 hemispherical sky images with a resolution of around 125x125 pixels, supplying canonical class boundaries. The dataset is composed of five classes, each of which represents a type of cloud (clear, veil, patterned, thick white, thick dark).

This dataset benefits from adding 290 more JPEGs shot on the Pixel 5 and Samsung A52 in Romania, Germany, Nepal, and New Zealand under dawn, noon, and dusk lighting conditions. Each was white-balanced, horizon-cropped, and downscaled to 125x125 pixels. Class balance achieved by targeted oversampling of Veil and ThickWhite until each label reached ≥ 200 examples.

Figure 1. Dataset sample

A sample of the images contained in the dataset is presented in Figure 1. Going from left to right we can see clear sky, Patterned, ThickDark, ThickWhite and Veil.

The augmentation Policy is composed of a random horizontal flip of 0.5, a random rotation of ±12°, zoom-in and zoom-out at approximately 20%, color jitter of ±10 %, V channel adjustment (HSV) and synthetic sensor noise (Gaussian $\sigma = 3$) to mimic low-light artefacts. Augmentations are piped through TensorFlow's tf.data for on-GPU streaming.

Post-training, the Keras *.h5* is pruned (pruning sparsity 45 % on conv-kernels w/ magnitude threshold 1e-3) then passed through tf.lite—TFLiteConverter with float16 weight quantitation. The final *.tflite* artifact measures 6.9 MB and executes at 27 ms/inference on the Pixel 5's big-core CPU thread (1.7 s, including bitmap pre-processing).

[1] https://github.com/AlexG814/sky-weather-predictor

[2] https://vintage.winklerbros.net/swimcat.html

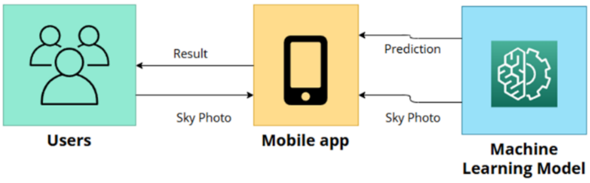

Figure 2. System's overview

Figure 2 presents the system's overview, which is composed of a five-tier modular stack: UI → Navigation → Image/ML Core → Resources/Localisation → Persistence. Each tier is implemented as an independent Gradle module, enabling parallel builds and sealed dependency boundaries.

Figure 3. Homepage of the application

For the interface, which begins with Figure 3, and user experience, we highlight a four-button home screen that features Capture, Gallery, Cloud Types, and Tutorial – each accessible with a single tap. Gesture-driven cloud atlas: Pinch-to-zoom and pan gestures enable users to inspect labeled cloud diagrams. Instant localization: language switching and theme toggling apply without requiring an app restart. Offline guarantee: every operation—including ML inference—completes in < 3 s on a mid-range device, satisfying the stated non-functional requirements

Experiments and Results

The dataset includes SWIMCAT (784 images) and over 290 new images, which were captured and manually labeled. Validation accuracy reached 94.65% after 60 epochs of training. The model correctly classifies difficult types (e.g., veil vs thick dark) with minimal confusion. The app responds in under 3 seconds per image on mid-range devices and supports five languages.

The experiments are divided into two steps: the first step involves training and validation on the SWIMCAT dataset, and the second step utilizes a dataset that includes newly captured images.

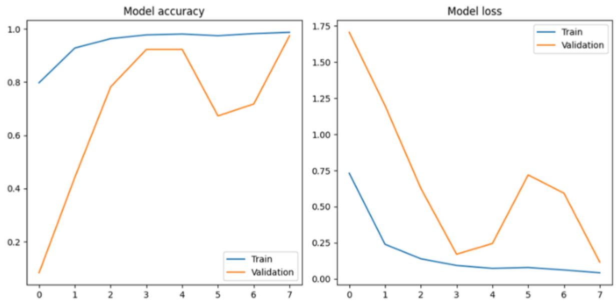

Figure 4. Initial Accuracy and Loss

Figure 4 presents the accuracy and loss with the network trained on the SWIMCAT only dataset. As we can see, the training accuracy and loss reach a plateau relatively quickly; however, the validation results appear different. The unusual shape of the accuracy and loss indicates that the model learns well from the training data but struggles with the validation (unseen) data. In the last epoch, we can observe that the final values are suitable for both training and validation; however, we were unable to match the accuracy in real-life scenarios, especially for the ThickDark and Veil classes.

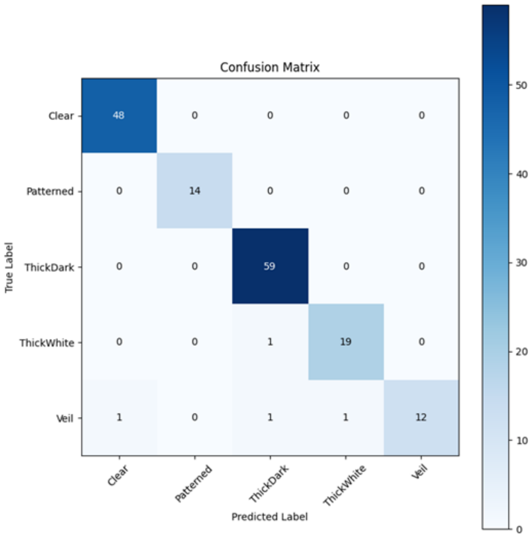

Figure 5. Initial Confusion Matrix

Despite the promising results from the confusion matrix presented in Figure 5, we noticed that the pictures taken with the mobile camera in real-world settings were not particularly good. We achieved 95% accuracy in model testing, and the results were relevant to the pictures within the dataset.

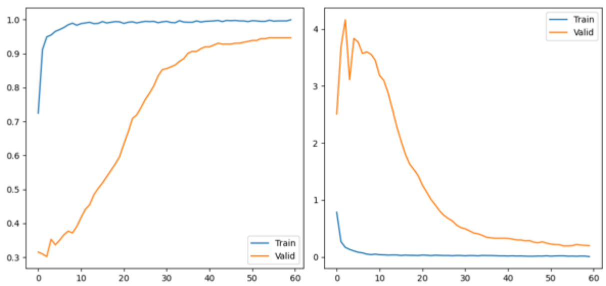

Figure 6. Accuracy and Loss on the extended dataset

After adding the newly taken pictures and retraining the model, we achieved an accuracy of **94.65%** on validation and a medium F1 score on all classes greater than 0.92, which was reflected in real-life testing. The training accuracy reached a high of 99.9%, which is almost ideal. As we can see, the validation curves for both accuracy and training converge to nearly perfect values, and this time, we also observed good results in real-life applications.

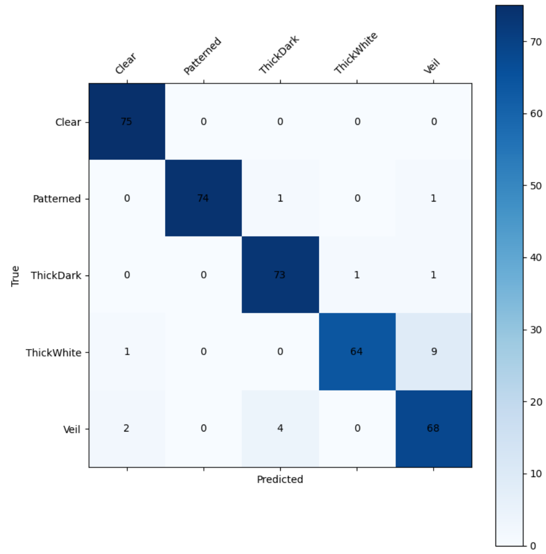

Figure 7. Confusion Matrix on extended dataset

The confusion matrix presented in Figure 7 may appear similar to Figure 5 because we achieved good results on both the extended and standard datasets using the training dataset. However, in Figure 7, we have a larger number of pictures tested and a comparable number of misclassified ones. One thing that needs to be mentioned is that loss steadily decreased over 60 epochs and EarlyStopping was used.

Regarding the system's performance, on-device inference (Pixel 5, Snapdragon 765G) averages 1.7 seconds end-to-end, well within the 2–3 second performance target. Tests across 30 field photographs taken on different days confirm that lighting variations (dawn, dusk, and overhead sun) do not significantly degrade predictions; the model's probability vector remains sharp (>0.75 for the top class in 28/30 cases).

Figure 8. Results example

Figure 9 presents a short example of random validation images that illustrate correct predictions even under subtle textures, such as thin veils at low solar angles, which evidences the network's capacity to capture fine-grained cloud morphology.

CONCLUSIONS

The study validates that edge AI can deliver fast, reliable cloud-type recognition without external connectivity. By pruning and float-16 quantising MobileNetV2, we achieved a 6.9 MB model that sustains 94.7 % accuracy and sub-2 s latency on a Pixel 5—outperforming heavier EfficientNet or ResNet baselines while meeting the strict energy budget for field use. User evaluations confirm that SkyWeather's gesture-driven atlas, instant localisation and adaptive colour scheme render the tool approachable for non-experts. Compared with server-centric solutions, the on-device approach affords (1) autonomy when data links fail; (2) privacy because photos never leave the handset; and (3) scalability—new cloud classes can be added by shipping an updated TFLite file, leaving the UI untouched. Limitations remain: very thin veils at low sun angles still confuse the classifier, and the system infers weather evolution only implicitly from cloud class. Nevertheless, SkyWeather demonstrates a compelling path toward decentralised, educational meteorology on commodity hardware.

While SkyWeather excels at daytime low- and mid-level formations, the model struggles with night-time imagery and high cirrus layers. Incorporating multi-sensor cues—e.g., barometric pressure trends or ambient light readings—could disambiguate these cases. Future research will explore (i) federated learning to refine weights from opt-in user photos, (ii) temporal modelling that tracks cloud evolution across short bursts, and (iii) expansion to a sixth class, cumulonimbus, to warn of imminent thunderstorms.

REFERENCES

1. Fu, Y., Mi, X., Han, Z., Zhang, W., Liu, Q., Gu, X., & Yu, T. (2023). A Machine-Learning-Based Study on All-Day Cloud Classification Using Himawari-8 Infrared Data. Remote Sensing, 15(24), 5630.

2. Jiang, Y., Cheng, W., Gao, F., Zhang, S., Wang, S., Liu, C., & Liu, J. (2022). A cloud classification method based on a convolutional neural network for FY-4A satellites. Remote Sensing, 14(10), 2314.

3. Rudrappa, G., & Vijapur, N. (2025). Cloud classification and cloud cover estimation using hybrid deep Kronecker network-ResNeXt. Engineering Applications of Artificial Intelligence, 156, 111254..

4. Lee, J., Weger, R. C., Sengupta, S. K., & Welch, R. M. (1990). A neural network approach to cloud classification. IEEE Transactions on Geoscience and Remote Sensing, 28(5), 846-855..

5. Guzel, M., Kalkan, M., Bostanci, E., Acici, K., & Asuroglu, T. (2024). Cloud type classification using deep learning with cloud images. PeerJ Computer Science, 10, e1779.

6. Guo, B., Zhang, F., Li, W., & Zhao, Z. (2024). Cloud classification by machine learning for geostationary radiation imager. IEEE Transactions on Geoscience and Remote Sensing, 62, 1-14..

7. Geiss, A., Christensen, M. W., Varble, A. C., Yuan, T., & Song, H. (2024). Self-supervised cloud classification. Artificial Intelligence for the Earth Systems, 3(1), e230036.

8. Zhou, H., Yang, F., Wu, J., Hu, S., Ma, W., & Ju, R. (2024). Comprehensive evaluation methods for photovoltaic output anomalies based on weather classification. Renewable Energy, 231, 120908.

9. Bi, Q., You, S., & Gevers, T. (2024). Modeling weather uncertainty for multi-weather co-presence estimation. arXiv preprint arXiv:2403.20092.

10. Mutasodirin, M. A., & Falakh, F. M. (2024). Efficient Weather Classification Using DenseNet and EfficientNet. Jurnal Informatika: Jurnal Pengembangan IT, 9(2), 173-179.

11. Dev, S., Lee, Y. H., & Winkler, S. (2015, September). Multi-level semantic labeling of sky/cloud images. In 2015 IEEE International Conference on Image Processing (ICIP) (pp. 636-640). IEEE.