Lightweight Pneumonia Detection Using Classical Descriptors on Edge AI Devices

Syed Muhammad Azeem Ul Hassan, Ali Hassan, Maria Hanif, M Uzair Wajeeh and Rizwan Ahmad
National University of Sciences and Technology (NUST), Islamabad, Pakistan
*Correspondence: rizwan.ahmad@seecs.edu.pk

Citation | Hassan. S. M. A. U, Hassan. A, Hanif. M, Wajeeh. M. U, Ahmad. R, "Lightweight Pneumonia Detection Using Classical Descriptors on Edge AI Devices", IJIST, Special Issue pp 583-594, May 2026
Received | April 02, 2026 **Revised |** May 10, 2026 **Accepted |** May 14, 2026 **Published |** May 17, 2026.

Pneumonia remains a pressing global health issue, and timely chest X-ray interpretation is central to effective patient care. This paper describes a lightweight pipeline for detecting pneumonia in chest radiographs using classical machine learning classifiers paired with handcrafted image features, aimed at hardware with tight processing budgets. Three widely used descriptors — HOG, SIFT, and ORB — were applied to the publicly available Kaggle chest X-ray dataset containing 5,863 labeled pediatric radiographs (4,273 Pneumonia, 1,590 Normal), pre-split into training (5,216), validation (16), and test (624) partitions. Four classifiers — Naive Bayes, SVM, Random Forest, and Logistic Regression — were trained with class-weight balancing to account for the roughly 2.7:1 class imbalance. Training and initial evaluation were carried out in Google Colab GPU-accelerated environment on a fixed held-out test set (approximately 89:11 train-to-test ratio), with 5-fold cross-validation on training data used to assess generalization. The strongest pipeline was subsequently ported to an NVIDIA Jetson Orin Nano Super for edge inference profiling. Random Forest with HOG features achieved the highest accuracy at 93.52% and an F1 score of 95.60%, while Logistic Regression with ORB features achieved the fastest inference speed at 0.187 ms per image (5,342.79 FPS). McNemar's test confirmed statistically significant gaps between the best and worst configurations ($p < 0.01$). Overall, the findings indicate that handcrafted features and traditional classifiers can produce clinically relevant performance within the power and memory envelopes of affordable edge hardware — a promising direction for pneumonia screening where specialist radiology support is scarce.

Keywords: Pneumonia Detection; Chest X-ray; Handcrafted Features; Edge AI; Machine Learning; HOG; SIFT; ORB; SVM; Random Forest; Logistic Regression.

Introduction:

Few infectious diseases kill on the scale that pneumonia does. According to the Global Burden of Disease study, roughly 2.56 million people died from pneumonia in 2018 [1]. The numbers have been inching downward since then — 2.55 million in 2019, 2.28 million in 2020, 2.18 million in 2021 — but they remain alarmingly high, particularly in low- and middle-income countries. Children are hit hardest: over 738,000 under-fives died of pneumonia in 2018 alone, and the toll still exceeded 500,000 in 2021. Much of this burden falls on a handful of nations — India, Nigeria, Pakistan, Ethiopia, and the Democratic Republic of the Congo [2].

Clinically, pneumonia is an acute inflammation of the lung parenchyma. A variety of organisms can trigger it — bacteria such as Streptococcus pneumoniae, Haemophilus influenzae, and Mycoplasma pneumoniae, viruses like Respiratory Syncytial Virus, and, less commonly, fungi or parasites. When the infection takes hold, the alveoli fill with fluid and inflammatory cells, impairing gas exchange. The typical presentation includes cough, chest pain, fever, dyspnoea, and tachypnoea. In the elderly or immunocompromised, pneumonia can rapidly become life-threatening [3][4].

Chest X-rays are the primary tool for diagnosing pneumonia in most clinical settings but reading them is far from straightforward. Cognitive biases, inter-observer variability, and reader fatigue all introduce error [5][6]. Pneumonia, pulmonary oedema, and bronchitis can look remarkably similar on film, and the degree to which different radiologists disagree on the same image is well documented [7]. Recent evidence shows that this inconsistency persists even among experienced radiologists — one study found disagreement rates as high as 22% for pediatric chest X-ray readings [8]. The situation is worse in rural areas, where there simply are not enough trained radiologists, and patients wait longer for a diagnosis [9][10]. A broad review of AI-assisted diagnostics noted another complication: collapsing three-dimensional anatomy onto a flat two-dimensional image creates tissue overlap that obscures early-stage disease and widens the gap between readers [11]. Automated tools could bring much-needed consistency to this process, but the deep learning systems that currently dominate the field tend to require computational resources well beyond what portable devices can supply.

Data-driven approaches bring objectivity and repeatability to medical image analysis, qualities that help temper subjective reading errors. Classical ML models — Logistic Regression, Naive Bayes, SVM, Random Forest — have a credible track record in identifying radiographic signs of pneumonia [12][13]. For example, [14] demonstrated that SVM, Logistic Regression, and Random Forest applied to radiomic chest X-ray features can reliably separate pneumonia from COVID-19. Similarly, [15] showed that SVM and Random Forest using handcrafted descriptors (LBP and CNN-derived features) exceeded 97% accuracy in multi-class chest X-ray classification, underscoring the continued clinical relevance of these methods. Careful feature engineering is what makes these classical pipelines work [16][17]. Deep CNNs have, of course, achieved strong results for pneumonia and COVID-19 detection [18][19][20], but their computational appetite makes them impractical on portable hardware [21], a shortcoming that multiple recent pneumonia detection surveys have flagged [22][23], and their black-box nature undermines clinical trust [24]. This highlights a key practical challenge. Clinicians require tools that are both accurate and fast enough to run on the hardware they actually have — small single-board computers, not data-center GPUs. Yet despite years of work on both classical and deep learning approaches to chest X-ray analysis, remarkably few studies have asked a simple question: can traditional image descriptors, running directly on modern edge AI boards, deliver acceptable pneumonia screening performance? That is exactly what this study sets out to answer, and the question maps directly onto the paper's core aim — lightweight pneumonia detection through classical descriptors on edge AI devices.

Novelty of this Work:

Previous work has applied classical ML to chest X-ray classification, but this study differs from that body of work in three concrete ways: (i) we carry out a controlled, head-to-head comparison of twelve classifier–feature combinations under identical experimental conditions, something that earlier studies did piecemeal; (ii) we physically deploy and benchmark the top-performing pipelines on a commercially available edge AI board (the NVIDIA Jetson Orin Nano Super), reporting actual latency, throughput, and model size numbers rather than theoretical estimates; and (iii) we include formal statistical significance testing via McNemar's test to verify that observed performance differences are not due to chance — a step that comparable lightweight studies have skipped. Together, these three elements give practitioners concrete data to guide their choice of classifier–feature pairing for real clinical edge deployment.

Research Objectives:

Four specific objectives guided this study:

O1: Compare classification accuracy, precision, recall, and F1 score across all twelve classifier–feature combinations using a standardized chest X-ray dataset.

O2: Measure inference latency, frames per second, and model size for each pipeline when running on the NVIDIA Jetson Orin Nano Super.

O3: Identify which classifier–feature combinations offer the strongest trade-off between diagnostic accuracy and computational speed for edge deployment.

O4: Test whether the observed performance differences across configurations are statistically significant.

In addition to these objectives, the study offers the following technical contributions:

A lean pneumonia detection pipeline combining HOG, SIFT, and ORB feature extraction with four classical ML classifiers.

Actual deployment on the NVIDIA Jetson Orin Nano Super, with measured latency, model size, and throughput figures.

An evaluation that covers both classification quality (accuracy, precision, recall, F1) and runtime cost (inference time, FPS, model size), backed by statistical significance testing.

Why these four classifiers? They cover the spectrum of classical ML complexity — from simple probabilistic models (Naive Bayes, Logistic Regression) to a kernel method (SVM) and an ensemble approach (Random Forest). All four have established performance in medical image classification [25][26], and all fit comfortably in the memory and compute budgets typical of edge devices.

The paper is organized as follows. Section II describes the methodology. Section III defines the evaluation metrics. Section IV presents and discusses the results. Section V considers broader implications. Section VI sets out recommendations. Section VII concludes.

Methodology:

Figure 1 shows the end-to-end pipeline used in this study. It has five stages: (i) data acquisition and splitting, (ii) preprocessing, (iii) feature extraction with HOG, SIFT, or ORB, (iv) classifier training and testing using Naive Bayes, SVM, Random Forest, and Logistic Regression, and (v) deployment on the NVIDIA Jetson Orin Nano Super. Each stage corresponds to one or more of the research objectives (O1–O4) listed in Section I-B.

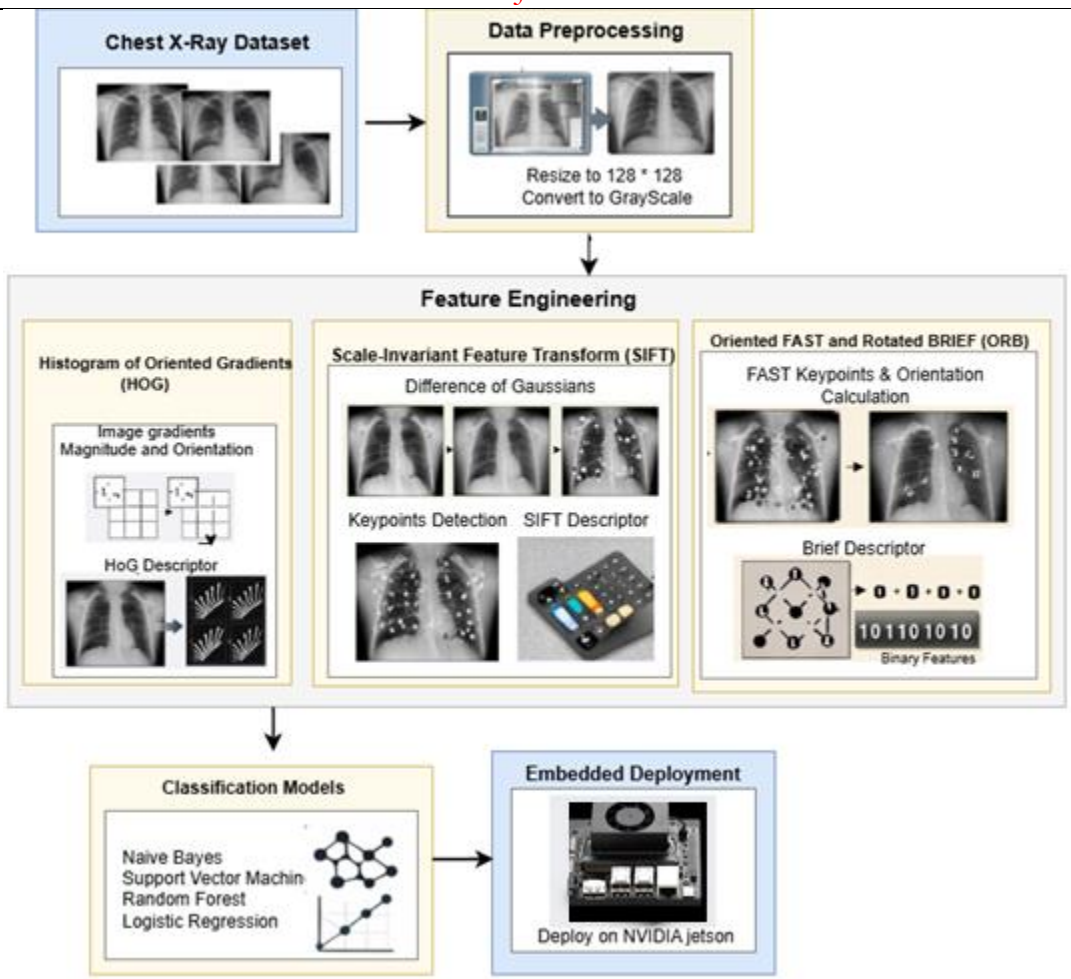

Figure 1. Pipeline for pneumonia detection using classical feature extraction and machine learning classifiers.

Dataset:

We used the publicly available Kaggle chest X-ray dataset, which contains 5,863 labeled pediatric radiographs split into two classes: Pneumonia (n = 4,273; 72.9%) and Normal (n = 1,583; 27.0%). The dataset comes with fixed splits — 5,216 training images, 16 validation images, and 624 test images. Given the roughly 2.7:1 class imbalance, we set class_weight='balanced' in every classifier to give the minority class greater weight during training. We intentionally left out oversampling and augmentation to keep the original distribution intact — a choice that better reflects what a real clinical dataset looks like. We included only frontal-view pediatric chest radiographs with confirmed labels; lateral views and unlabeled images were excluded.

Data Preprocessing:

All images were resized to 128 × 128 pixels for uniform input dimensions and to keep compute costs down. We converted every image to grayscale and normalized pixel intensities to the [0, 1] range before extracting features. This removes unnecessary color variation from the images.

Feature Extraction:

Three handcrafted feature extraction methods were used:

Histogram of Oriented Gradients (HOG) [22]: HOG works by computing image gradients using derivative kernels:

$$D_x = I * K_x, \qquad D_y = I * K_y$$

And then calculating per-pixel gradient magnitude

$$M(x,y) = \sqrt{\{D_x^2 + D_y^2\}}$$

And orientation θ(x,y). The image is divided into small cells, and orientation histograms from each cell are assembled and normalized over larger blocks. This makes the descriptor robust to lighting changes and produces a dense vector encoding the local edge structure of the image.

Scale-Invariant Feature Transform (SIFT) [23]: SIFT locates keypoints that remain stable across changes in scale and rotation by computing Difference-of-Gaussians:

$$\Delta I(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)\,.$$

Each keypoint receives an orientation from the local gradient distribution and gets a 128-dimensional descriptor f = [h_1, h_2, h_{128}]. Since the number of keypoints varies from image to image, we used a Bag-of-Visual-Words scheme with k = 500 visual words to produce fixed-length feature vectors.

Oriented FAST and Rotated BRIEF (ORB) [24]: ORB pairs the FAST keypoint detector with a rotation-aware version of the BRIEF binary descriptor. Keypoint orientation $\theta = \tan^{-1}\left(\frac{m_{01}}{m_{10}}\right)$ is calculated from image moments. The output is a compact binary string based on intensity comparisons at fixed point pairs — a format that lends itself naturally to fast inference on low-power hardware.

Classifiers and Hyperparameter Settings:

We trained four classifiers on each feature type. Table II lists the full hyperparameter settings so that every experiment reported here can be reproduced (supporting O1).

The choice of classifiers was guided by three considerations: (i) all four have been shown to work well on medical image classification tasks [12]; (ii) they are lightweight enough in memory and inference time to run within edge hardware constraints; and (iii) their decision boundaries are fundamentally different — linear (Logistic Regression), probabilistic (Naive Bayes), kernel-based (SVM), and ensemble (Random Forest) — which lets us compare across several levels of model complexity.

Table 1. Hyperparameter Settings for All ML Classifiers

Classifier	Parameter	Setting / Value
Naive Bayes	var_smoothing	1×10^{-9} (GaussianNB default)
SVM	kernel, C, γ, class_weight	RBF, 1.0, 'scale', 'balanced.'
Random Forest	n_estimators, max_depth, min_samples_split, class_weight	100, None, 2, 'balanced.'
Logistic Regression	C, solver, max_iter, penalty, class_weight	1.0, lbfgs, 1000, L2, 'balanced.'

Table 2. Hyperparameter Settings for Feature Extraction Models (HOG, SIFT, ORB)

Descriptor	Parameter	Setting / Value
HOG	orientations, pixels_per_cell, cells_per_block, block_norm	9, (8, 8), (2, 2), L2-Hys
SIFT	nfeatures, nOctaveLayers, contrastThreshold, edgeThreshold, sigma, BoVW k	0 (all), 3, 0.04, 10, 1.6, k = 500
ORB	nfeatures, scaleFactor, nlevels, edgeThreshold, patchSize, WTA_K	500, 1.2, 8, 31, 31, 2

Validation Strategy:

Every classifier was trained on the predefined training partition (5,216 images) and subsequently tested on the fixed held-out test set (624 images), in line with standard benchmarking practices for this dataset. In addition, we carried out 5-fold cross-validation on the training data to gauge stability and reduce the risk of overfitting. The resulting train-to-test

ratio is approximately 89:11. To determine whether the accuracy differences between any two configurations were statistically meaningful, we applied McNemar's test ($\alpha = 0.05$) to their paired prediction outputs on the test set.

Model Deployment:

The best-performing pipelines — Logistic Regression + ORB for speed, and Random Forest + HOG for accuracy — were deployed on the NVIDIA Jetson Orin Nano Super (shown in Figure 2) using unaccelerated ARM CPU cores in a purely sequential, single-threaded execution path. Pre-trained scikit-learn model weights were loaded into pre-allocated memory buffers during initialization. Each input image was resized to 128 × 128 pixels and normalized with OpenCV. End-to-end inference latency for Logistic Regression + HOG came to roughly 0.325 ms per image, demonstrating the feasibility of real-time operation on resource-constrained hardware.

Figure 2. NVIDIA Jetson Orin Nano Super used for model deployment

Evaluation Metrics:

Classification performance was measured with four standard metrics. Using aaa for true positives, bbb for false positives, ccc for false negatives, and ddd for true negatives:

Accuracy (ACC) = (a + d) / (a + b + c + d) — simply the fraction of all predictions the model gets right.

Precision (Pre) = a / (a + b) — of the cases the model labels as positive, how many actually are. High precision means fewer false alarms and fewer patients sent for unnecessary treatment.

Recall (Rec) = a / (a + c) — of the truly positive cases, how many the model catches. High recall means fewer missed diagnoses.

F1 Score (F1) = 2a / (2a + b + c) — the harmonic mean of precision and recall, giving a single number that captures the balance between precision and recall.

For the edge deployment side, we recorded inference time (ms per sample), frames per second (FPS), and stored model size (KB). Pairwise accuracy differences were tested with McNemar's test ($\alpha = 0.05$). All metrics were computed on the held-out test set.

Experimental Results and Discussion:

In this section, we evaluate all twelve classifier–feature configurations against the four research objectives (O1–O4). The evaluation unfolded in two phases: cloud-based training and testing on Google Colab environment, followed by edge deployment on the NVIDIA Jetson Orin Nano Super Developer Kit.

Classification Performance (Objective O1):

Accuracy (Figure 3): Random Forest paired with HOG achieved the highest performance at 93.52% accuracy and a 95.60% F1 score. Logistic Regression with ORB was

notably fast, while both Logistic Regression and SVM with SIFT features reached 91.81%. Naive Bayes + SIFT trailed the pack at 84.90%, not entirely surprising, given that Naive Bayes assumes feature independence, a condition that high-dimensional, correlated descriptors clearly violate. McNemar's test showed a statistically significant gap between RF+HOG and NB+SIFT ($p < 0.01$), but no significant difference between LogReg+SIFT and SVM+SIFT ($p = 0.83$).

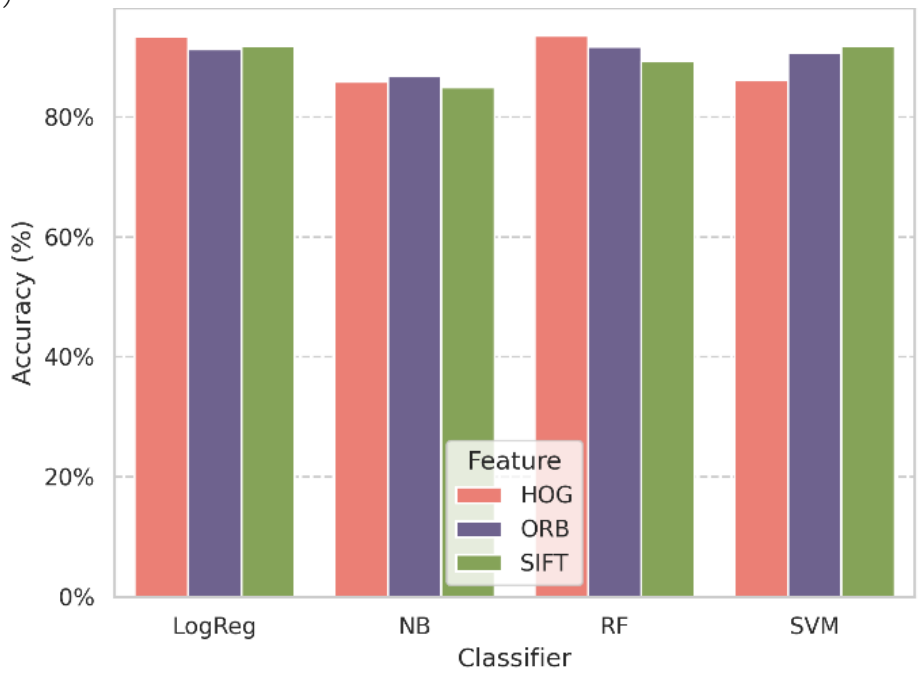

Figure 3. Accuracy of classifiers using HOG, ORB, and SIFT.

Precision (Figure 4): Logistic Regression + HOG led with 98.89% precision, with SVM + HOG very close behind at 98.87%. From a clinical standpoint, high precision matters because each false positive sends a healthy patient down an unnecessary treatment path. Random Forest + SIFT showed the lowest precision at 90.64%. at 90.64%.

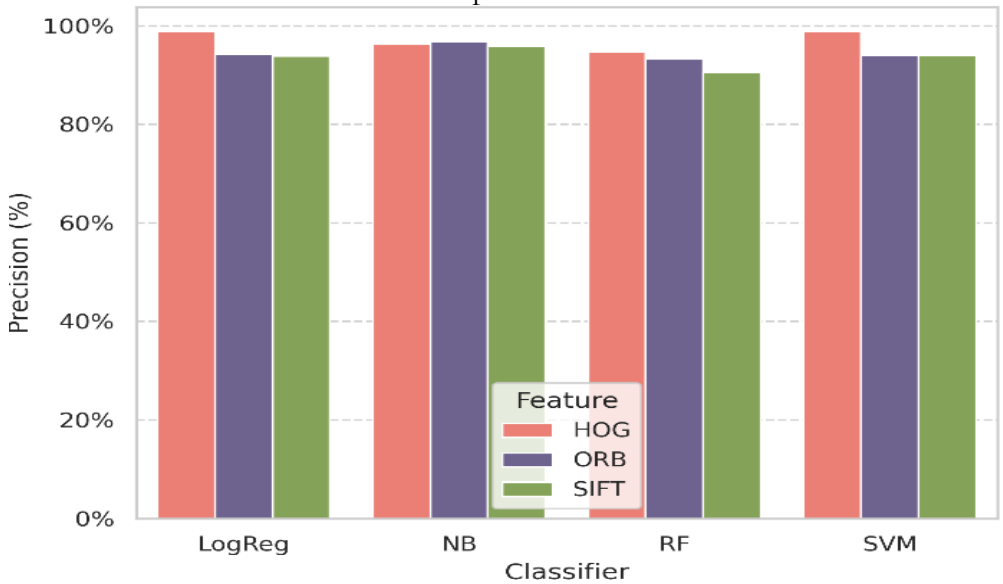

Figure 4. Precision of classifiers using HOG, ORB, and SIFT.

Recall (Figure 5): Random Forest + HOG posted the highest recall at 96.49%, meaning it missed very few actual pneumonia cases — a critical property when delayed

treatment can have serious consequences. In contrast, SVM + HOG recorded the lowest recall at 81.87%.

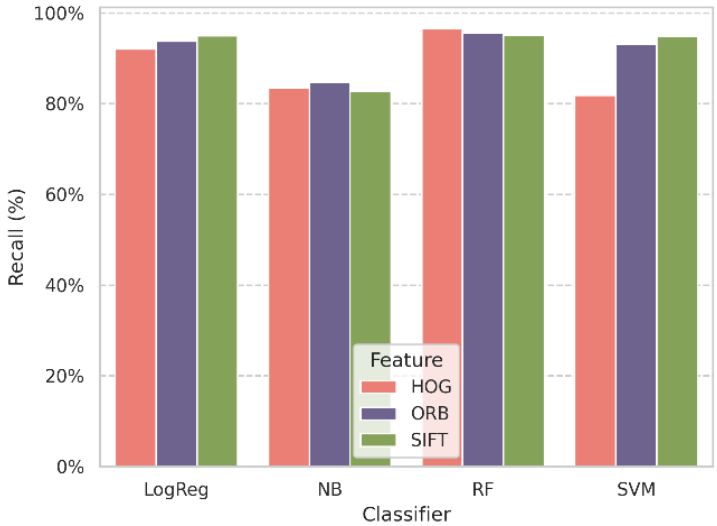

Figure 5. Recall of classifiers using HOG, ORB, and SIFT.

F1 Score (Figure 6): Random Forest + HOG achieved the highest F1 of 95.60%, balancing precision and recall well. Naive Bayes + SIFT had the lowest F1 at 88.89%.

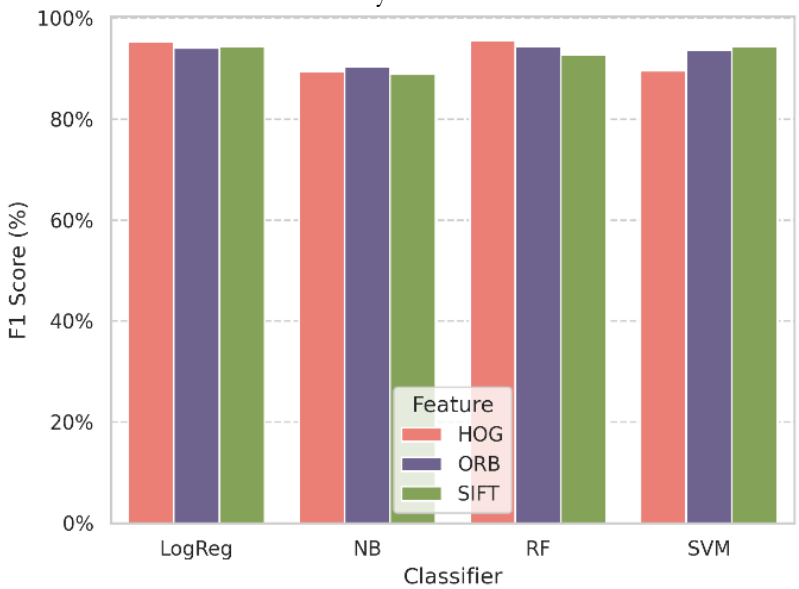

Figure 6. F1 Score of classifiers using HOG, ORB, and SIFT**.**

How do these results compare with deep learning models? Our RF+HOG pipeline (93.52%) sits comfortably within the 90–96% accuracy range reported by recent lightweight CNN-based approaches tested on the same Kaggle dataset [27], but does so without a GPU, with sub-millisecond inference, and with model files under 6 MB. For context, [12] achieved 89% using a classical SVM pipeline, and [13] reached 91% with classical features — both below what we report here. The takeaway is that a well-tuned classical pipeline can still achieve competitive performance when deployment constraints make deep learning impractical.

Edge Deployment Performance (Objectives O2, O3):

Table I shows inference time, FPS, and model size as measured on the Jetson Orin Nano. Logistic Regression + ORB was the fastest configuration, finishing each inference in 0.187 ms and reaching 5,342.79 FPS — thereby addressing O2. Random Forest consistently took the longest (8.0–8.2 ms) and had the largest models (up to 5,695.81 KB with SIFT features). SVM fell in between, with FPS ranging from 2,419.50 (ORB) to 3,047.58 (HOG).

The results indicate that: pairing a simpler classifier with a compact descriptor like ORB gives the best real-time throughput, which addresses O3.

Table 2. Analysis of Inference Time, FPS, and Model Size on NVIDIA Jetson Orin Nano

Classifier	Feature	Inf. Time (ms)	FPS	Size (KB)
LogReg	ORB	0.187	5342.79	1.62
NB	ORB	0.218	4489.43	3.88
RF	ORB	8.034	125.37	5233.62
SVM	ORB	0.442	2419.50	798.39
LogReg	SIFT	0.398	2512.60	1.62
NB	SIFT	0.374	2690.77	3.88
RF	SIFT	8.245	122.65	5695.81
SVM	SIFT	0.345	2811.47	1.62
LogReg	HOG	0.325	3078.36	64.12
NB	HOG	0.630	1587.67	253.88
RF	HOG	8.179	122.26	2676.10
SVM	HOG	0.328	3047.58	64.00

Research Objectives Achievement Summary:

Looking across the full set of experiments, each of the four objectives has been addressed. For O1, we systematically compared all twelve configurations, and RF+HOG came out ahead on accuracy (93.52%), F1 (95.60%), and recall (96.49%); McNemar's test confirmed a significant difference between the top and bottom performers ($p < 0.01$). For O2, every pipeline was deployed and profiled on the Jetson Orin Nano Super, with LogReg+ORB posting the lowest latency (0.187 ms) and highest throughput (5,342.79 FPS) — all without GPU acceleration. O3 was answered by mapping out which configuration best balances detection quality against speed, giving practitioners a clear basis for choosing a setup that fits their clinical constraints. O4 was met through McNemar's test, which verified that the gap between the strongest and weakest pipelines is not due to chance ($p < 0.01$). In short, the proposed framework does what it set out to do and produces repeatable results for pneumonia screening on low-cost edge hardware.

Implications:

Clinical Implications:

A recall of 96.49% means RF+HOG misses very few pneumonia cases — something that matters enormously in paediatric wards and emergency departments where a delayed diagnosis can be fatal. On the precision side, LogReg+HOG at 98.89% keeps false positives low, which in turn avoids unnecessary courses of antibiotics and supports better antimicrobial stewardship. The fact that inference takes less than a millisecond opens the door to bedside screening in clinics that have no radiologist on staff. And because these models rely on interpretable features (HOG, SIFT, ORB) rather than opaque neural network weights, it is easier for clinicians and regulators to understand — and trust — the output.

Technical Implications:

The deployment numbers make a straightforward point: classical ML pipelines run in real time on standard ARM processors with no GPU at all. Model sizes range from 1.62 KB (LogReg+ORB/SIFT) to 5,695.81 KB (RF+SIFT), all well within the memory capacity of typical embedded boards. Running inference entirely on-device also removes the need for internet connectivity and the privacy concerns that come with sending patient X-rays to external servers. Because the pipeline is modular — swap in a different feature extractor or classifier and the rest stays the same — it can be adapted to other medical imaging tasks without starting from scratch.

Societal Implications:

The countries hit hardest by pneumonia are the same ones with the fewest radiologists. A screening tool that runs on a sub-$200 board (the Jetson Orin Nano Super) and needs no internet connection could meaningfully improve early detection in rural and remote clinics. Scaled up, such a tool could help bring down pediatric pneumonia deaths by getting children onto antibiotics sooner — an outcome aligned with the WHO's priorities and the United Nations Sustainable Development Goal 3 (Good Health and Well-Being).

Recommendations:

For Clinical Practitioners:

Where diagnostic accuracy and recall matter most — emergency triage, pediatric wards — RF+HOG is the stronger choice. Where speed on a low-power device is the priority, LogReg+ORB gives the best accuracy-to-latency ratio. Either way, we recommend keeping a clinician in the loop; the system works best as a second opinion or pre-screening filter, not as a standalone diagnostic.

For Policymakers:

For high-burden regions, investing in edge-based pneumonia screening makes financial sense: the entire pipeline runs on a standard laptop or single-board computer with no cloud subscription. Policymakers in South Asia and Sub-Saharan Africa should consider incorporating automated chest X-ray pre-screening into national pneumonia programs, while updating regulatory frameworks to cover AI-assisted diagnostics.

For Future Researchers:

Several directions follow naturally from this work. First, fusing handcrafted features with CNN-derived representations could push accuracy higher. Second, model compression techniques — quantization, pruning — could shrink the Random Forest models further. Third, the framework should be tested on adult pneumonia datasets and broader benchmarks like NIH ChestX-ray14. Fourth, explainability methods such as Grad-CAM adaptations for classical features deserve investigation. Finally, profiling the pipeline on other edge platforms — Jetson Xavier NX, Raspberry Pi 5, Coral Edge TPU — would show how broadly the approach generalizes.

Conclusion:

This study set out to fill a gap: lightweight, interpretable pneumonia detection designed for edge hardware. Judged against the four objectives, the quantitative picture is as follows: (O1) RF+HOG delivered the strongest classification numbers — 93.52% accuracy, 95.60% F1, 96.49% recall — and was statistically superior to the weakest pairing (NB+SIFT at 84.90%; $p < 0.01$, McNemar's test). (O2) LogReg+ORB was the most efficient on the edge, finishing each inference in 0.187 ms and reaching 5,342.79 FPS on the Jetson Orin Nano Super without GPU help. (O3) A clear accuracy-versus-speed trade-off emerged: RF+HOG maximizes detection quality but costs 8.179ms per inference, while LogReg+ORB cuts latency at the expense of 4.62 percentage points of accuracy (88.90%). The optimal configuration therefore depends on the specific clinical requirements. (O4) McNemar's test confirmed that the best pipeline significantly outperforms the worst ($p < 0.01$), giving confidence that the comparisons hold up.

In summary, the results indicate that handcrafted descriptors paired with classical classifiers can reach clinically relevant accuracy on hardware that costs under $200 and draws minimal power — a useful result for any setting where pneumonia screening is needed but computing infrastructure is limited. The framework is ready for deployment. Future work will explore deep feature fusion, model pruning, and benchmarking across additional edge platforms to close the remaining gap with deep learning methods.

References:

[1] "Pneumonia - Our World in Data." Accessed: Mar. 25, 2026. [Online]. Available:

https://ourworldindata.org/pneumonia

[2] Daniele Piovani, Gisella Figlioli, "The global burden of enteric fever, 2017–2021: a systematic analysis from the global burden of disease study 2021," *EClinicalMedicine*, vol. 77, p. 102883, 2024, [Online]. Available: https://www.thelancet.com/journals/eclinm/article/PIIS2589-5370(24)00462-0/fulltext

[3] "Pneumonia: Causes, Symptoms, Diagnosis & Treatment." Accessed: Mar. 25, 2026. [Online]. Available: https://my.clevelandclinic.org/health/diseases/4471-pneumonia

[4] A. D. Nguyen, D. R. Stamm, and H. A. Stankewicz, "Atypical Bacterial Pneumonia," *StatPearls*, Apr. 2025, Accessed: Mar. 25, 2026. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK532239/

[5] Pedro Cruz, Ana M Meireles, "COVID-19-Associated Cognitive Biases on Pneumonia Differential Diagnosis," *Cureus*, vol. 16, no. 2, 2024, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/38558668/

[6] "An Avoidable Cognitive Error in Chest Radiography - PMC." Accessed: May 05, 2026. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10756156/

[7] "Comparison of Chest Radiograph Impressions for Diagnosing Pneumonia: Accounting for Categories of Language Certainty - PubMed." Accessed: May 05, 2026. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35792164/

[8] Cihan Aydin, Hafize Kızılkaya, "AI Models for Accurate Bacterial Pneumonia Diagnosis in Chest X-ray Images," *Bozok Tıp Derg.*, vol. 15, no. 2, pp. 169–177, 2025, doi: 10.16919/bozoktip.1593097.

[9] Amer Kareem, Haiming Liu & Paul Sant, "Review on Pneumonia Image Detection: A Machine Learning Approach," *Human-Centric Intell. Syst.*, vol. 2, pp. 31–43, 2022, [Online]. Available: https://link.springer.com/article/10.1007/s44230-022-00002-2

[10] Apurv Verma, D R Suman Kumar Swarnkar, "Advanced Deep Learning Models for Pneumonia Detection," *Proc. Int. Conf. Smart Heal. Intell. Technol.*, 2025, [Online]. Available: https://www.atlantis-press.com/proceedings/icshit-24/126010389#:~:text=To address these challenges%2C deep,in chest X-ray pictures.

[11] S. Sharma and K. Guleria, "A systematic literature review on deep learning approaches for pneumonia detection using chest X-ray images," *Multimed. Tools Appl. 2023 838*, vol. 83, no. 8, pp. 24101–24151, Aug. 2023, doi: 10.1007/s11042-023-16419-1.

[12] S. Parveen and K. B. Khan, "Detection and classification of pneumonia in chest X-ray images by supervised learning," *Proc. - 2020 23rd IEEE Int. Multi-Topic Conf. INMIC 2020*, Nov. 2020, doi: 10.1109/INMIC50486.2020.9318118.

[13] T. B. Chandra and K. Verma, "Pneumonia Detection on Chest X-Ray Using Machine Learning Paradigm," *Adv. Intell. Syst. Comput.*, vol. 1022 AISC, pp. 21–33, 2020, doi: 10.1007/978-981-32-9088-4_3.

[14] "Machine Learning Model Based on Radiomic Features for Differentiation between COVID-19 and Pneumonia on Chest X-ray - PubMed." Accessed: May 05, 2026. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/36081170/

[15] "In Search of an Efficient and Reliable Deep Learning Model for Identification of COVID-19 Infection from Chest X-ray Images - PubMed." Accessed: May 05, 2026. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/36766679/

[16] S. Hu, "Weakly Supervised Deep Learning for COVID-19 Infection Detection and Classification From CT Images," *IEEE Access*, vol. 8, pp. 118869–118883, 2020, doi: 10.1109/ACCESS.2020.3005510.

[17] A T ; Nagi, M J ; Awan, "Performance Analysis for COVID-19 Diagnosis Using

Custom and State-of-the-Art Deep Learning Models," *Appl. Sci.*, vol. 12, no. 13, p. 6364, 2022, doi: https://doi.org/10.3390/app12136364.

[18] Janmenjoy Nayak, Bighnaraj Naik, "Significance of deep learning for Covid-19: state-of-the-art review," *Res. Biomed. Eng.*, vol. 38, no. 1, pp. 243–266, 2021, doi: 10.1007/s42600-021-00135-6.

[19] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, Gabriel F. Manso, "The Computational Limits of Deep Learning," *arXiv:2007.05558*, 2020, [Online]. Available: https://arxiv.org/abs/2007.05558

[20] T. Dhar, N. Dey, S. Borra, and R. Simon Sherratt, "Challenges of Deep Learning in Medical Image Analysis - Improving Explainability and Trust," *IEEE Trans. Technol. Soc.*, vol. 4, no. 1, pp. 68–75, Mar. 2023, doi: 10.1109/TTS.2023.3234203.

[21] "Chest X-Ray Images (Pneumonia)." Accessed: Mar. 25, 2026. [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

[25] Corinna Cortes & Vladimir Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995, doi: https://doi.org/10.1007/BF00994018.

[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324/METRICS.

[27] M. E. H. Chowdhury, "Can AI help in screening viral and COVID-19 pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.