SIIM CAIMI26 Conference on Artificial Intelligence in Medical Imaging
OCT 26-27 | University of Pennsylvania, Philadelphia, PA

Afri CXR- A Chest Xray Triage for African Hospitals

Aniekanabasi Umoh, MBBS; University of Calabar Teaching Hospital

Briefly describe what attendees will see during your demo.
A live walkthrough of CXR Triage, an educational chest X-ray triage prototype, running end to end in a browser at cxrtriage.vercel.app. Attendees will see me upload a chest X-ray, watch the app call a deployed medical foundation model, and return a triage read (likely normal vs likely pneumonia) with a confidence score and a cautious, clinically framed interpretation. I will then walk through the model journey behind it: a baseline CNN, a transfer-learning model, and the MedSigLIP-based pipeline that now powers the live demo, including the confusion matrices and the false-positive reduction that made it the chosen model. I will be explicit throughout about what the tool is not: not a diagnostic device, not a radiologist replacement. The through-line is the design principle: AI that assists, people who decide.

Problem & Motivation: What clinical or operational problem does your MVP address?
In many low-resource settings, including Nigerian hospitals where I trained and now practice, the limiting factor in a busy department is rarely medical knowledge. It is time, attention, and the availability of imaging expertise. A clinician covering a large patient load cannot give every chest X-ray the same scrutiny in the same minute, and radiologist coverage is often thin or delayed. Pneumonia is a high-volume, time-sensitive condition where earlier prioritisation of the sickest patients changes outcomes. The specific pain point: there is no fast, accessible way for a frontline clinician to get a second-opinion flag on a chest X-ray to help decide what to look at first. Existing radiology AI is largely built for, and priced for, well-resourced systems with PACS integration and vendor contracts. I wanted to build something that demonstrates the opposite end of that spectrum: a lightweight, deployable triage aid that keeps the human clinician firmly in the decision seat. The motivation is not to automate diagnosis. It is to support prioritisation when attention is the scarcest resource.

What You Built: Describe your MVP, tool, or workflow in plain language
CXR Triage is a web app. You upload a chest X-ray image (JPG or PNG), and it returns three things: a triage label (Likely Normal Pattern, Likely Pneumonia Pattern, or Low Confidence Result), a confidence score, and a short, cautious interpretation note. There is no login, no setup, and no patient data stored. Under the hood, the image is converted into a numerical representation by a medical foundation model (Google's MedSigLIP), and a lightweight classifier reads that representation to produce the triage flag. The output is deliberately framed as a triage insight, not a diagnosis, and a non-diagnostic disclaimer is shown before and after every result. It is an honest MVP with rough edges. It runs on free-tier infrastructure, so the first request after idle is slow while the model wakes up. It also includes built-in sample X-rays so anyone can try it without needing their own image. The whole thing, from model training to a live public inference API, is deployed and usable today at cxrtriage.vercel.app.

Tech Stack & Development Approach: What tools, frameworks, or methods did you use to build it?

Honestly, this was built as a learning project that I deliberately pushed all the way to deployment, by a medical doctor learning ML. Models (Python): I trained and compared three approaches in Jupyter/Colab. A baseline CNN from scratch (Keras), a MobileNetV2 transfer-learning model (Keras), and the final pipeline: MedSigLIP (google/medsiglip-448) image embeddings fed into a scikit-learn logistic regression. Grad-CAM (via the transfer model) was used for explainability and error analysis. Backend: FastAPI serving the MedSigLIP + logistic regression pipeline, with PyTorch and Hugging Face transformers, containerised with Docker and deployed on a Hugging Face Space (CPU tier). Frontend: Next.js (App Router, TypeScript) with Tailwind, deployed on Vercel, calling the live backend API directly. Approach: a mix of structured learning (working through an ML course), heavy use of AI coding assistants for the engineering, and a lot of debugging real deployment problems: dependency pinning, gated-model authentication, embedding parity between training and serving, and CORS. I treated "it works in a notebook" and "it works as a deployed product" as two genuinely different milestones.

Validation & Evidence: Have you done any informal testing or validation? If so, what did you find?

This is informal benchmark validation, not clinical validation, and I want to be clear about that distinction. On the public Kaggle chest X-ray pneumonia test set (624 images), the three models compared as follows: Baseline CNN: 62.66% accuracy. Confusion matrix [[6, 228], [5, 385]]. It behaved like an aggressive screener, catching pneumonia but massively over-calling it on normal scans (228 false positives). MobileNetV2 transfer: 76.12% accuracy. Confusion matrix [[86, 148], [1, 389]]. MedSigLIP + logistic regression: 91.03% test accuracy (93.75% validation). Confusion matrix [[180, 54], [2, 388]]. The most clinically relevant finding was not the headline accuracy but the error profile. The foundation-model pipeline kept false negatives very low (2) while cutting false positives from 228 to 54. For triage, that balance matters more than accuracy alone: you want to avoid missing sick patients without crying wolf on every normal film. Surprising result: a tiny logistic regression on frozen medical-foundation-model embeddings beat a fully trained CNN by nearly 30 points. The representation did the heavy lifting, not a bigger task-specific network. Grad-CAM on the transfer model showed clinically plausible lower-lung attention on some true positives, but broad, non-specific attention on failures, a useful honesty check that a model can be right for the wrong reasons.

Feedback You're Seeking: What specific feedback, help, or collaboration are you hoping to get from the CAIMI26 community?

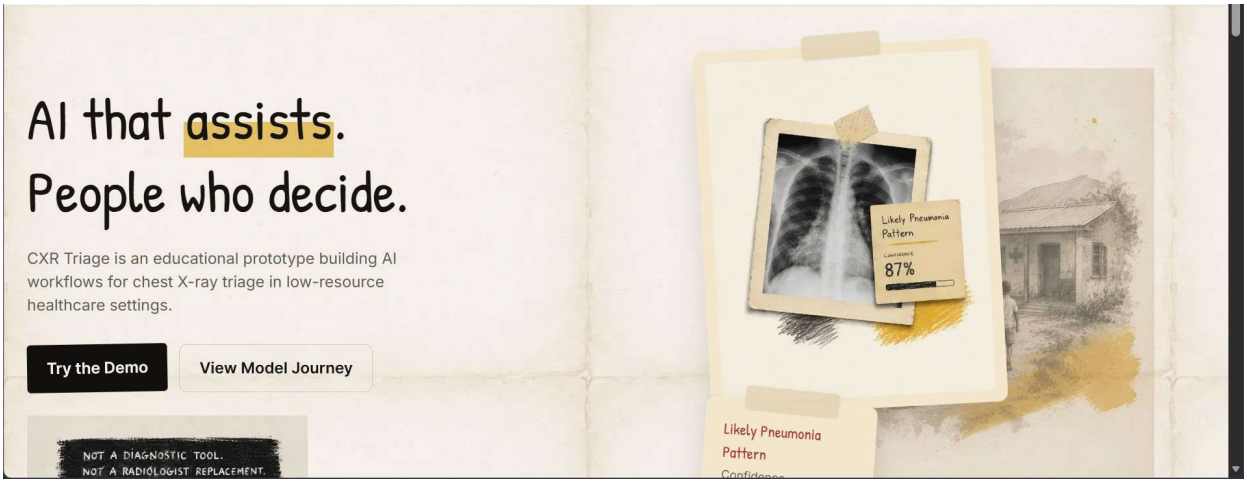
A few specific things: Clinical validity and framing. I am a clinician, but I would value radiologists and clinical-AI practitioners stress-testing the triage framing. Is "Likely Pattern + confidence + human review required" the right level of claim for a frontline triage aid, or does even that overstate it? Real-world evaluation design. The biggest gap is external validation on data that looks like an actual African hospital population. I would love guidance on how to design a credible, low-cost external evaluation, and ideally a clinical champion or site willing to collaborate on it. The deployment-vs-research gap. I am interested in feedback from people who have taken models from benchmark to bedside on what breaks in practice: workflow integration, where a triage flag actually fits (and where it does harm), and failure modes I am not yet seeing. Honestly, community and direction. I am a doctor self-teaching ML and shipping in public. Pointers on where to focus next would be genuinely valuable.

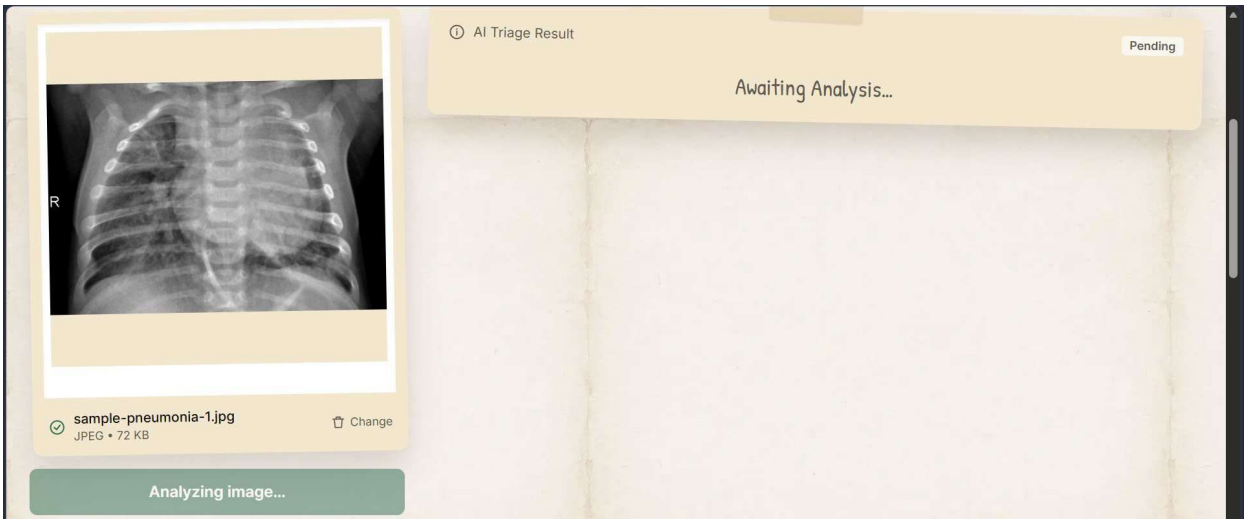
Known Limitations & Honest Failures: What isn't working yet, or what have you already tried that failed?

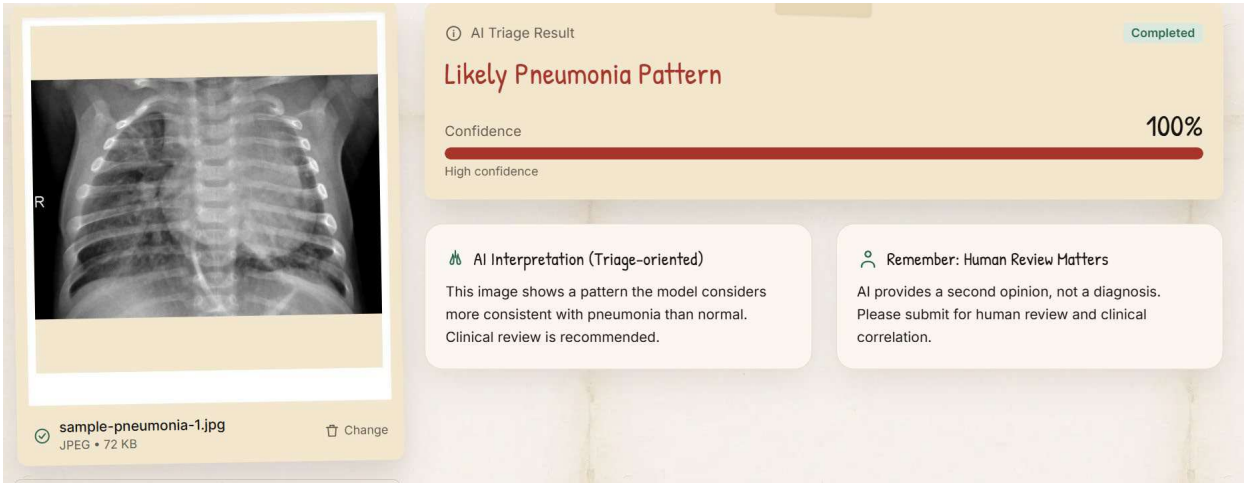
This is the part I care most about being straight on. The data is a public benchmark, not clinical reality. The

Kaggle dataset is imbalanced toward pneumonia, has a tiny validation split (16 images), likely contains label noise, and does not represent the population I want to serve. The 91% number should be read as a benchmark result, not a clinical performance claim. No external validation. None. This has never seen real prospective hospital data. It cannot localise findings. The model outputs a class, not a region. I deliberately removed any interface text implying it can point to consolidation or specific lung zones, because the logistic-regression-on-embeddings pipeline genuinely cannot, and showing that would be dishonest. I dropped Grad-CAM from the public app on purpose, because exposing attention maps would imply a diagnostic precision the model does not have. Infrastructure honesty: it runs on free CPU tiers, so cold starts are slow (20–40s on first request). This is a demo, not a production service. Things that were hard or impossible without more support: getting access to real, labelled, consented chest X-ray data from the target setting is the single biggest blocker, and it is not something I can solve alone. PACS/EHR integration and any path toward regulatory validation are well beyond a solo learning project. Those are exactly the areas where vendor or institutional support would change what is possible.

Potential for Impact: If this MVP were fully developed and deployed, who would benefit and how?
If fully developed and validated, the clearest beneficiaries are frontline clinicians in resource-limited settings: house officers, general practitioners, and emergency staff who interpret chest X-rays without immediate radiologist backup. A trustworthy triage flag could help them prioritise which films and patients to escalate first, shortening time-to-review for the sickest. Patients benefit if that prioritisation reduces missed or delayed pneumonia. Trainees benefit from a tool that shows confidence and explicitly defers to human judgment, reinforcing good clinical habits rather than replacing them. Hospital systems with thin imaging coverage benefit from a lightweight, low-cost layer that does not require heavyweight PACS integration to provide value. The broader point is a model for how this kind of tool should be built and framed: cheap to deploy, honest about uncertainty, and designed so the human always decides. CXR Triage is a small proof of concept for that approach, with pneumonia as the starting condition and a framework meant to extend to others.







GitHub Repo:
https://cxrtriage.vercel.app/
https://huggingface.co/spaces/Aniekanabasiumoh/africxr-triage
https://github.com/AniekanabasiUmoh/cxrtriage