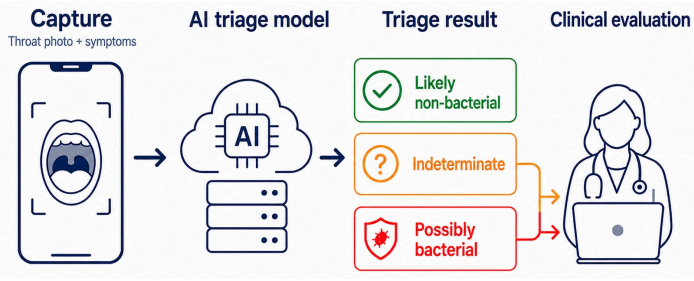
CASE STUDY · AI PRODUCT DEVELOPMENT

Is this strep throat? An AI triage aid for your phone

I developed an AI smartphone tool that helps screen suspected bacterial pharyngitis from a throat photo — built for patients in rural and low-resource areas, where the nearest clinic can be hours away.



ROLE
Product Developer + Researcher

TEAM
Advisor: Prof. Thanassis Rikakis
3 practicing physicians

TIMELINE
Jan – April 2026 · 16 weeks

OUTCOME
Gave a guest lecture on the project to an AI course at USC

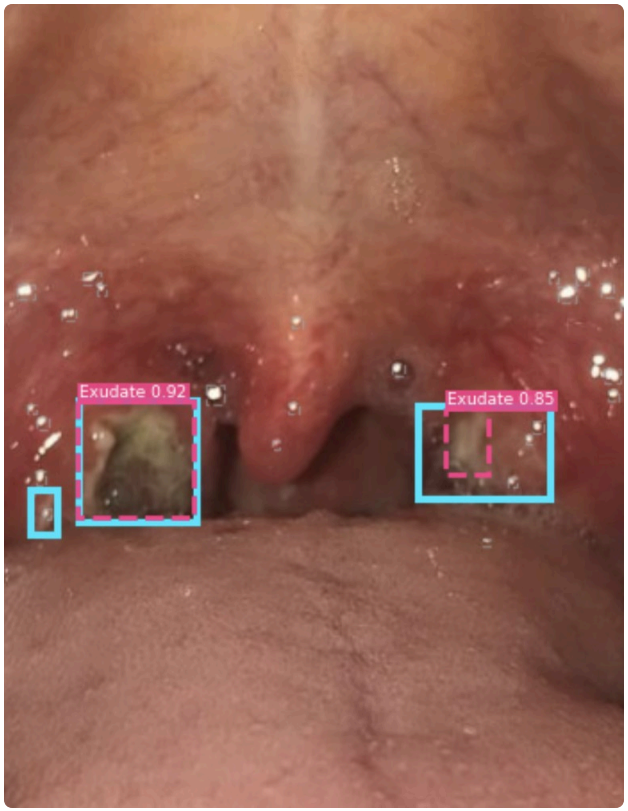
USC University of Southern California

Eddie Simmens · esimmens@gmail.com · Directed research, 2026

Project Snapshot

For a semester-long directed research project, I partnered with a group of physicians to learn how they assess strep throat, and then used those insights to develop a computer vision product that helps patients decide whether they may need an in-person clinical evaluation for suspected strep throat (bacterial pharyngitis). A user takes a picture of their sore throat with a smartphone, and the model returns one of three results: "low likelihood bacterial," "indeterminate," or "possibly bacterial." Under the hood, it is a multi-stage deep learning pipeline built from several independently trained Convolutional Neural Network (CNN) detectors that feed into a Multilayer Perceptron (MLP) fusion classifier. I built it to work on pictures taken by patients, which are often blurry, dimly lit, or partially obstructed. At the end of the project, I was invited to guest lecture this project for a deep learning AI course at the University of Southern California, teaching this work as a case study on developing AI products with noisy real world data.

To guide the design, I ran an image-quality survey with three clinicians and followed up with physician interviews. I also hand-drew 3778 bounding box annotations to train an exudate detector, which reduced false negatives by 44%. The system is optimized to rule out specific presentations of bacterial pharyngitis, and it clears the project's 80% target for negative confidence (average confidence in ruling out the disease), landing at 84.4%. To prioritize patient safety, I calibrated it to only give "rule out" answers when the model is very confident, routing the rest to clinicians for evaluation. Under this calibration, it achieves 98% NPV at 90% sensitivity with 81% specificity (5-fold CV, 387 patients).



The core output. The exudate detector marks the white patches on each tonsil with a confidence score. Exudate on both tonsils is the strongest positive signal the system has, and it overrides a rule-out.

90%
Sensitivity
5-fold CV (n=387)

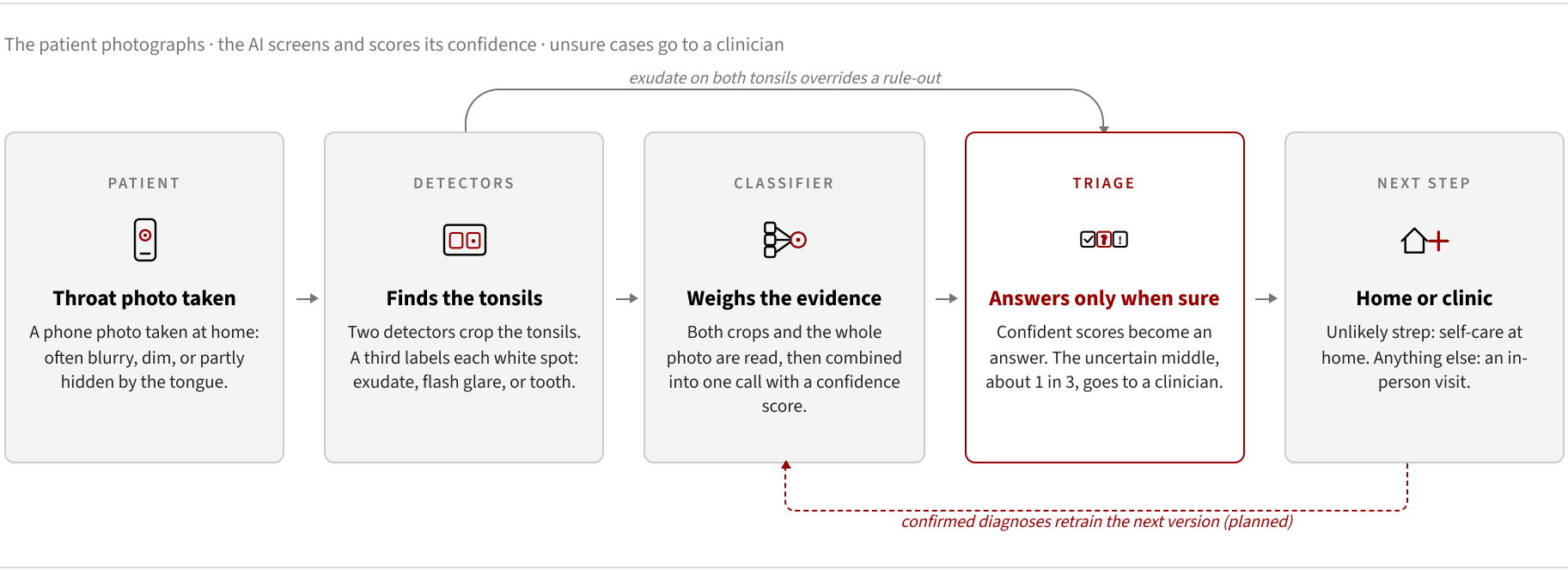
84.4%
Mean negative confidence
(target: 80%)

0.80
AUC, outperforming all existing published models

3500+
Manually annotated bounding boxes for modeling

From a throat photo to a triage answer

The system looks where clinicians look first: the tonsils, for the white or yellow patches called exudate. It scores how sure it is and only answers when that score is high. Everything else goes to a person.



The resulting tool aims to:

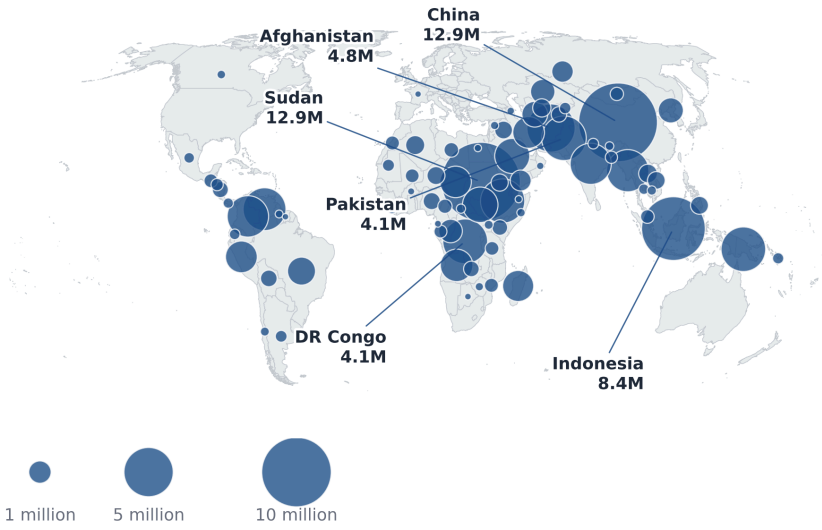
- Work on real patient photos: blurry, dim, and badly framed, the way people actually photograph their own throats
- Look where clinicians look first: the tonsils, for the white or yellow patches called exudate
- Tell true exudate from flash glare and teeth, so the model stops learning shortcuts
- Rule strep out rather than in, the direction the data supports, with a confidence score on every answer
- Hand uncertain cases to a clinician instead of guessing: a triage aid, not a diagnosis
- Improve with use: confirmed diagnoses retrain the next version (planned)

The problem

Bacterial pharyngitis (strep throat) is a common reason people seek care. It's one of the sore throat conditions where antibiotics can help — but left untreated, it can lead to serious complications like rheumatic fever, throat abscesses, or kidney problems.

In rural and low-resource communities, the nearest clinic can sometimes be hours away in driving time, and people have to decide whether the trip is worth it for a sore throat. A triage tool can help by giving a confident recommendation when the signs are clear, and by clearly saying "see a clinician" when they're not. It supports the decision without replacing the clinical visit.

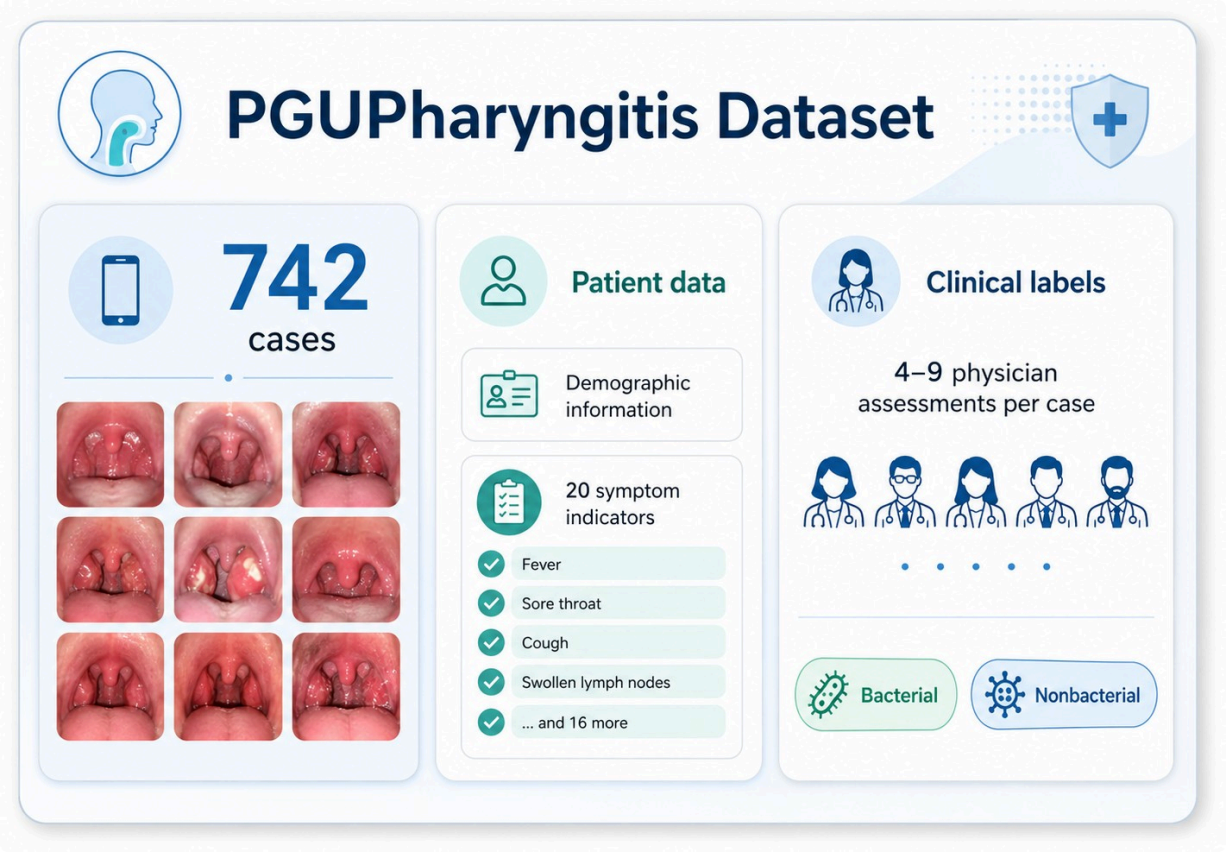
This is a triage aid, not a diagnostic tool.



I created this visual to show the target users.

Choosing a realistic dataset

This project uses a recently published dataset of 742 smartphone throat images, labeled bacterial or nonbacterial, that was intentionally captured with the poor lighting, motion blur, and framing you'd expect from real users. Many images are hard to see, which makes modeling very challenging.

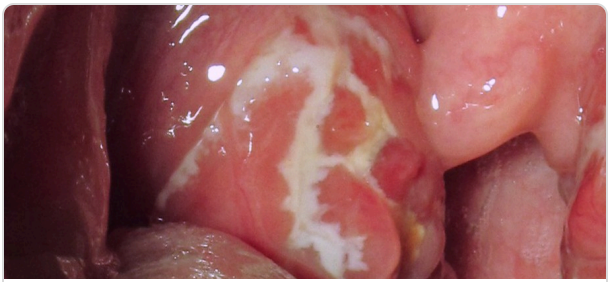


The labels are also unreliable:

- Diagnoses came from physician opinion only (no lab tests), and each image was reviewed by 4 to 9 doctors.
- Doctors disagreed on most images, and some split evenly.
- Each image has a distribution of diagnoses, not a single definitive label.

Potential model features: what clinicians use when diagnosing bacterial pharyngitis

Clinicians rely on a combination of visual findings. Below are some of the cues they use, and the same cues I evaluated as candidate features for the model.



Tonsillar exudate

White or yellow patches on the tonsils. A strong visual indicator of bacterial pharyngitis.



Palatal petechiae

Red dots on the soft palate from broken capillaries. Highly specific when present, but rare in this dataset.

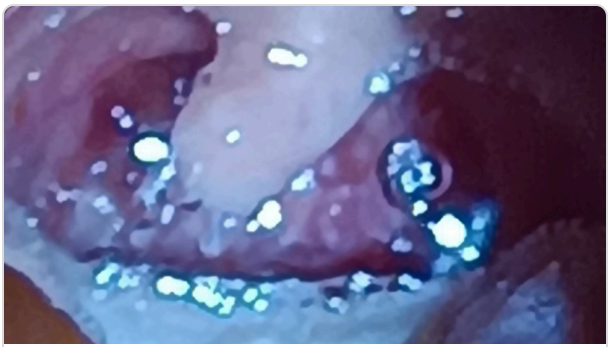


Enlarged tonsils & erythema

Swollen tonsils may suggest infection, especially if asymmetric, but size alone does not confirm a bacterial cause. Erythema is supporting context, not a definitive sign.

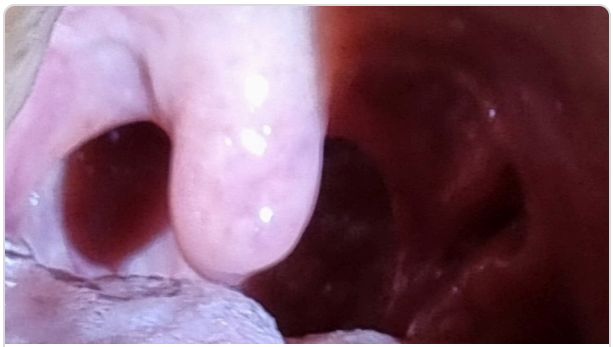
Distractor features

Before training anything, I looked at every image in the 742-patient PGUPharyngitis dataset to examine distractor features. These can pull CNNs away from clinically relevant regions, so many of the architectural decisions I made respond to these patterns.



Specular reflections

Phone flash bouncing off saliva and mucosal surfaces; a large source of model confusion.



Shadows

The mouth's natural geometry produces unavoidable dark regions that can obscure key structures.



Occlusion: tongue, lips, teeth, objects

No model can recover information that isn't in the frame.

Insights from clinicians that shaped the architecture

Because I am not a clinician, one of the biggest challenges was figuring out what the model should look for, and how physicians actually read a throat image. To answer that, I ran a study with three practicing physicians: an image-usability survey, where each rated a sample of dataset images on a 1–5 scale across anatomical framing, lighting, sharpness, color accuracy, obstructions, and overall diagnostic usability — and a follow-up interview focused on the visual features clinicians prioritize when evaluating sore throats.

Tonsillar exudate is the most predictive visual finding

When asked what they check first, clinicians said they scan the tonsils for exudate. I dropped my uvula and posterior pharyngeal wall detectors and refocused solely on the tonsils.

Absence of exudate is a partial rule-out signal

The absence of visible exudate provides a meaningful negative signal — not sufficient on its own, but informative enough to anchor a rule-out approach when combined with other information.

Clinicians tolerate visual imperfection, but not missing anatomy

Blurry or poorly lit images were sometimes still usable if the tonsillar regions were framed. Clean images with cropped or occluded tonsils were not. This shaped the training set and guided image capture requirements.

Confirmation of triage output

In the survey, clinicians also validated the 3 proposed model outputs:

OUTPUT 1

Likely non-bacterial

Conservative self-care delivered to the patient. Option to share results with their PCP.

OUTPUT 2

Indeterminate

Anatomy or model confidence insufficient. The system declines to make a judgment; case routed to a clinician for review.

OUTPUT 3

Possibly bacterial

Visual signs consistent with bacterial infection. Patient prompted to seek in-person evaluation with a clinician.

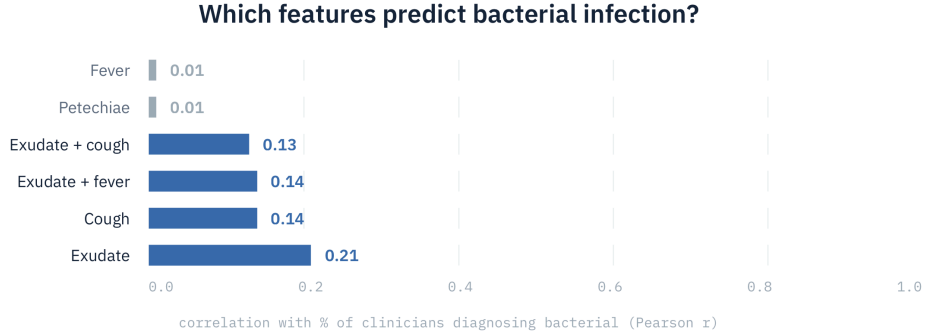
The goal shifted: ruling bacterial pharyngitis out instead of ruling it in

After auditing the labels, I manually annotated rough exudate and petechiae across the dataset to see how individual visual features correlated with clinician diagnosis and how common they were. Every feature was weakly correlated, with the strongest (exudate) at $r \approx 0.21$.

However, there was a silver lining: among images with nothing resembling exudate, almost none were bacterial. The dataset supported negative confidence rather than positive confidence, potentially helping rule out exudative bacterial pharyngitis.

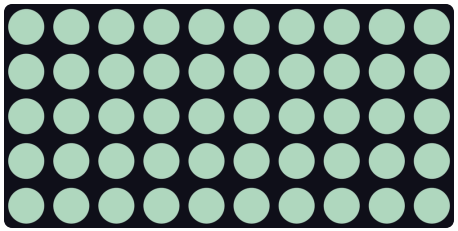
The objective became reframed: instead of predicting "this is strep" (which the data does not support), I made the system identify "this is unlikely to be strep."

One consequence: I excluded the dataset's 20 structured symptom indicators (fever, cough, etc.) from model training because their correlations were too weak in this sample to justify the added complexity. This is dataset-specific. In other contexts (e.g. Centor scoring), symptoms carry more signal, so future versions trained on populations with stronger symptom-image correlation should revisit inclusion.

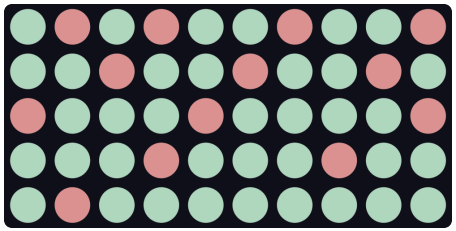


Pearson correlation of individual visual and symptom features in the PGUPharyngitis dataset with the percent of clinicians diagnosing bacterial. Exudate should be a strong signal — but its only weakly correlated in this data.

In this particular dataset, absence of exudate strongly suggests "not bacterial" (presence of exudate is much less conclusive).



Patients with NO visible exudate
~99.5% turn out to be non-bacterial

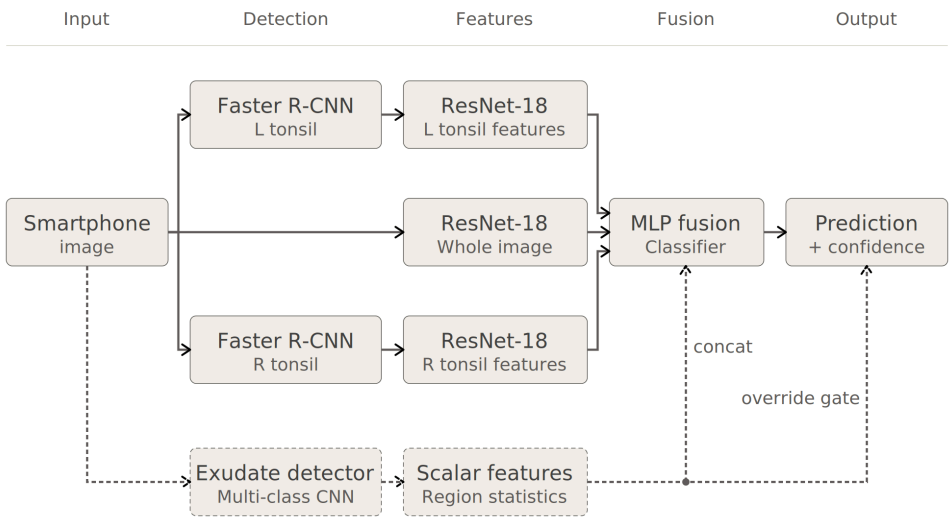


Patients WITH visible exudate
A meaningful share are diagnosed bacterial

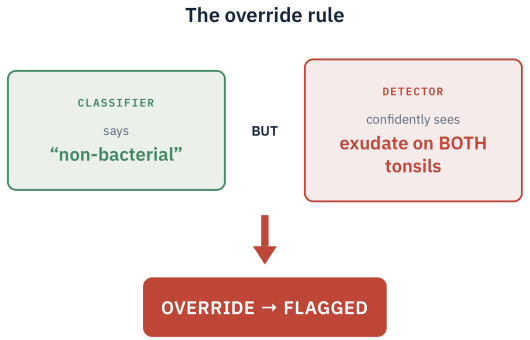
The Technical Pipeline

The system is built from multiple independently trained CNN detectors and a separately trained classifier.

- Two Faster R-CNN tonsil detectors locate and crop the left and right tonsillar regions; a multi-class exudate detector runs in parallel on the whole image.
- The classifier consumes three feature streams — the two crops plus the original whole image — each processed by its own ResNet-18 backbone.
- Features are concatenated and passed to an MLP whose softmax output produces both the predicted class and a confidence score, which drives the indeterminate band.
- The exudate detector contributes scalar features into the same MLP fusion and also applies a deterministic override when it detects high-confidence exudate on both tonsils.



ResNet-18 is used due to the small dataset and to avoid overfitting.

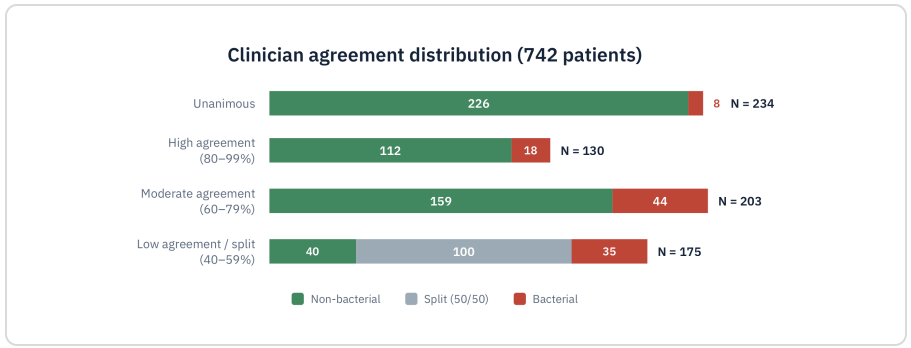


The override rule. The exudate detector helps catch false negatives with a rule-based override feature after the MLP has made its classification.

Key design rationales

I only included images where most doctors agreed on the diagnosis.

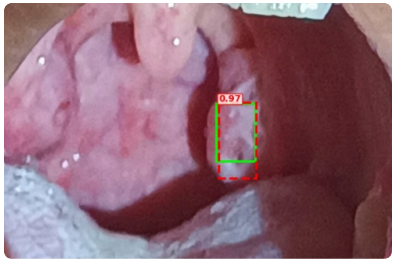
My rule was: if at least 75% of the doctors who reviewed an image agreed on a diagnosis, it was included and labeled with their majority call. Out of 742 images, only 234 had full agreement from every doctor — requiring unanimity would have thrown out too many images, especially bacterial cases. But allowing too much disagreement would have left me with untrustworthy labels.



ROIs were cropped to focus the model on relevant features

Clinically relevant tissue is concentrated in the tonsillar regions. Teeth, tongue, and lips are not diagnostic and can add noise the classifier may latch onto.

By cropping to the relevant areas, we teach the CNN where to focus. This also improves interpretability: clinicians can see what drove the decision, overall impression, exudate, and which tonsil.

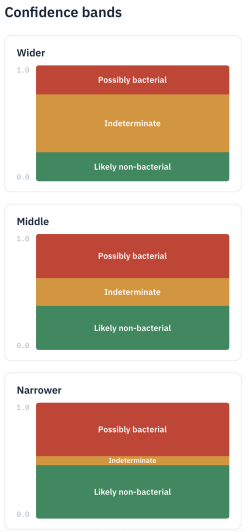


Green = my annotation. Red = the right tonsil detector's prediction, found with high confidence.

Confidence scores were used with triage predictions

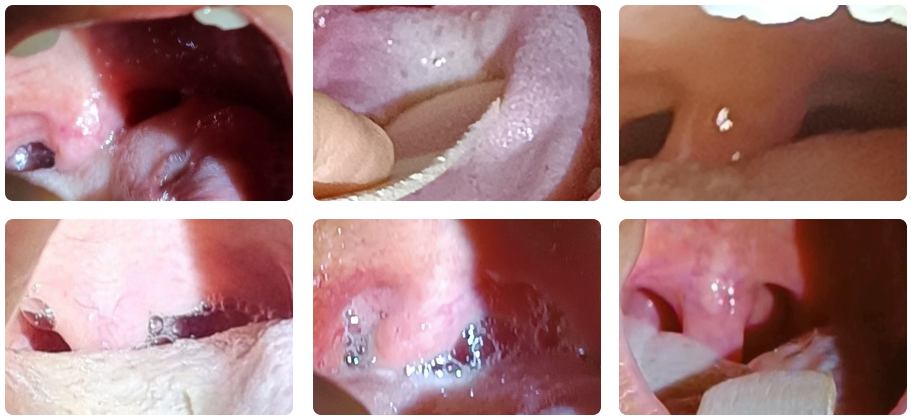
The MLP outputs a confidence score for each prediction, not just a bacterial or non-bacterial label. I then grouped the results into the three triage bands based on that score.

This approach keeps the system flexible because the model probabilities stay the same while the triage thresholds can be tuned.



I also removed 100 clinically unusable images from the training data

I looked through the images the clinicians had rated as unusable and then removed similar-looking ones from the dataset. They wouldn't have given the model much to learn from:

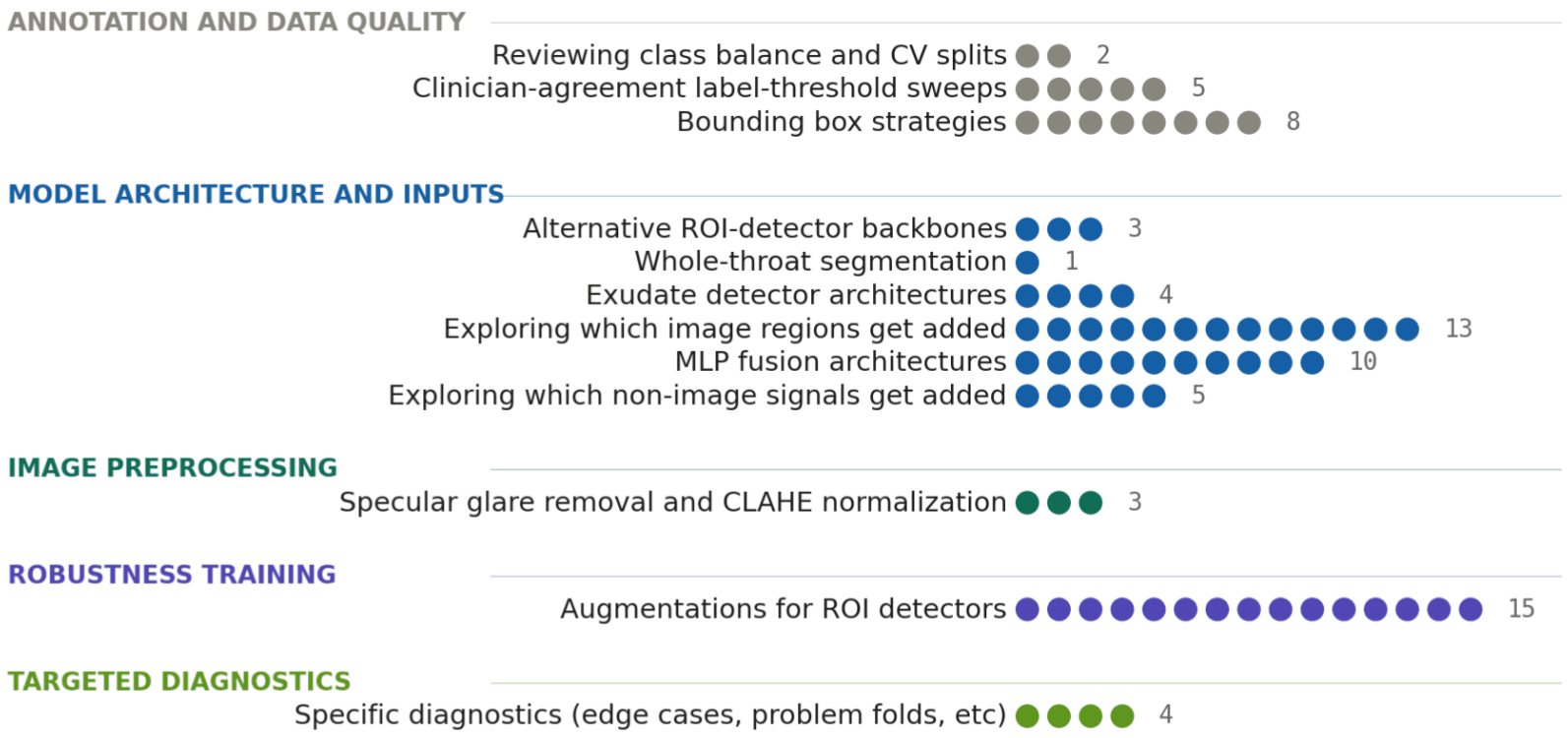


Iterating: identifying which interventions worked

Most research directions I pursued did not improve performance; the final system is built on the few that did. The sections that follow cover some of the many iterations I did -- both successful and unsuccessful. I ended up logging 500+ runs in Weights and Biases for analysis and storage.

Directions tested, and how much

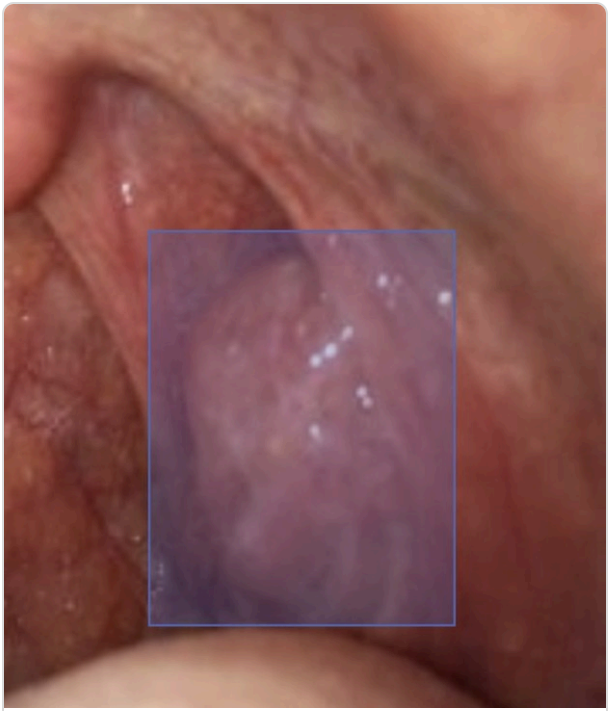
Here is a summary of the directions I tested and how much I tested each, one dot equals one experiment



Each "experiment" included many, many runs -- a lot of time was spent iterating.

Bounding-box strategy: multiple iterations

I started tonsil detection with loose boxes around each anatomical structure, but performance was poor. A MedSAM-based technique recommended by my advisor helped improve accuracy. After follow-up clinician interviews clarified that exudate on the tonsillar region is the priority, and I saw exudate sometimes appeared on the tonsil pillars, I expanded the boxes to a broader region. Three iterations:



V1 — Loose boxes

Loose crops around each structure. Detectors underperformed; the boxes captured too much non-discriminative tissue.



V2 — Tight patches

Conservative crops avoiding adjacent tissue. Improved precision but excluded areas where exudate occasionally appeared.



V3 — Broad tonsillar region

Includes pillars and some soft palate to capture exudate wherever it appears. This version was used by the final tonsil detectors. Boxes all end at the anterior pillar arch to give the CNN consistent edge cues.

Refinement of the annotation strategy alone produced significant improvements in model performance.

Iterative troubleshooting with attention maps: the classifier focused on the wrong feature

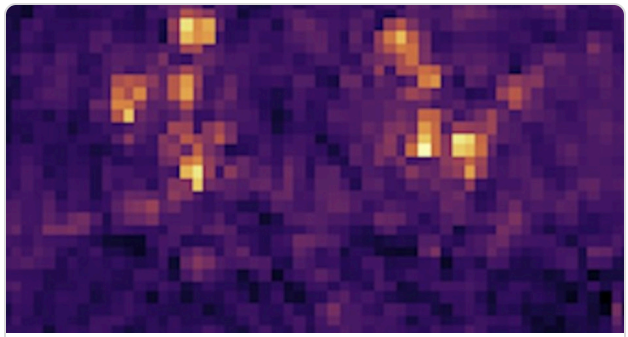
The classifier consistently missed positive (bacterial) cases. To diagnose why, I extracted early layer feature maps from representative bacterial cases and compared them to the input.

The classifier was relying on **shortcut learning** — picking up on glare and reflections instead of the true visual signs of strep throat. I tried inpainting the reflections, but that just introduced texture artifacts the model fixated on instead. Adding synthetic reflections as augmentation helped only slightly. Another strategy was needed.



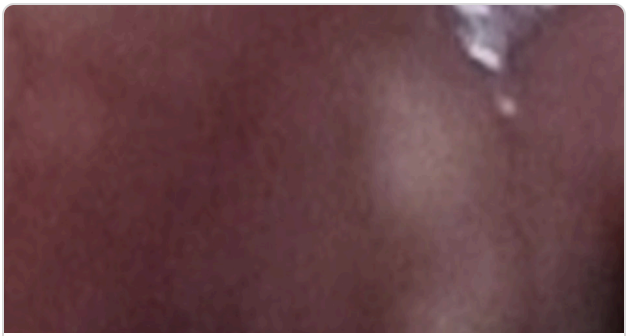
Whole image · original

A bacterial case. The white spots are phone-flash specular reflections off saliva — not exudate. The actual exudate sits in the dimmer mid-tonsil region.



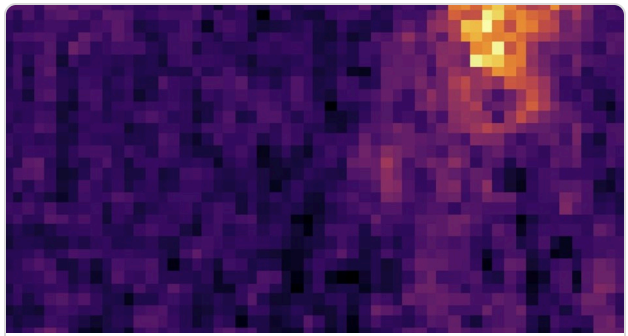
Whole image · feature map

The model attends to the bright glare patches, not the actual exudate.



Left tonsil close up · original

Close-up of the same case. The exudate is faded — low contrast, no clear edges. CNN early layers respond primarily to edges.



Left tonsil close up · feature map

Even in close-up, the model attends to reflections and mostly ignores the subtle gray exudate patch below.

The exudate detector

The solution was a separate detector that was specifically trained to tell exudate apart from its look-alikes. I trained a multiclass object detector to label each white spot in a throat photo as exudate, flash reflection, or teeth, using the same higher-quality external dataset (PMC7440230) as I had primarily used for the tonsil detectors.

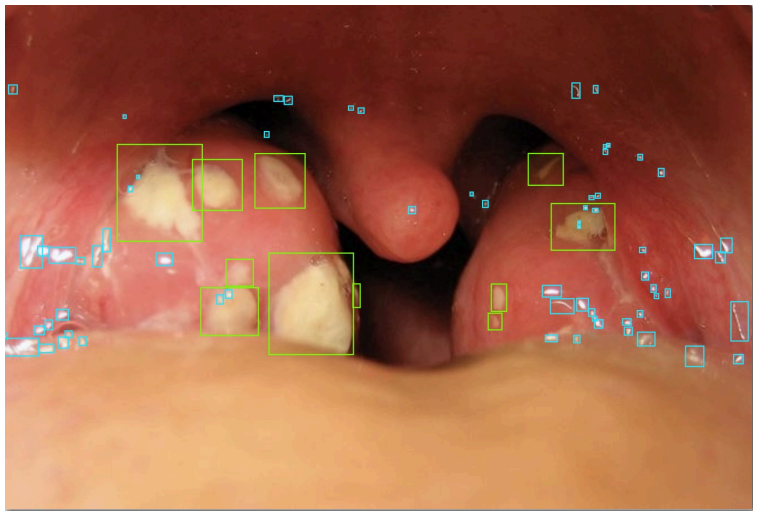
My first version had only tried to detect exudate, but it kept confusing exudate with teeth or reflections. Adding the look-alikes as explicit classes solved that.

Annotating each individual reflection was time consuming: I drew 3,778 bounding boxes by hand, but it paid off in accuracy.

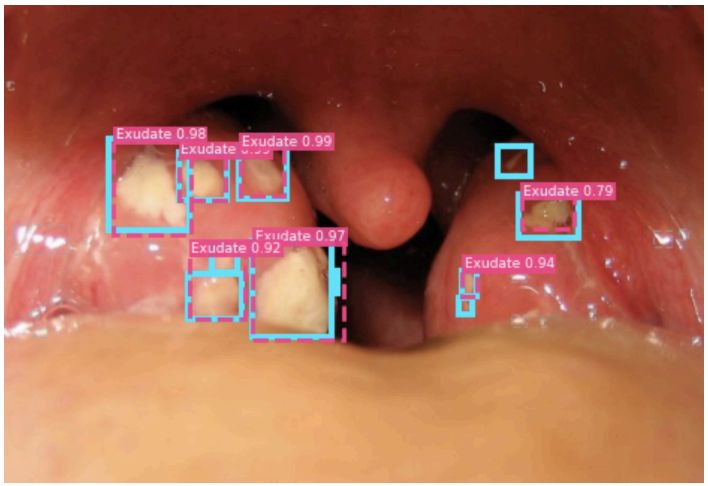
This detector supported the final model by generating numeric features (size, count, color, and location of exudate, including which tonsil) for the MLP, and by enabling the deterministic override rule that flipped a non bacterial prediction when it detected high confidence bilateral exudate.



A bacterial case with numerous reflections that distract a CNN from the gray exudate patches on the tonsils.



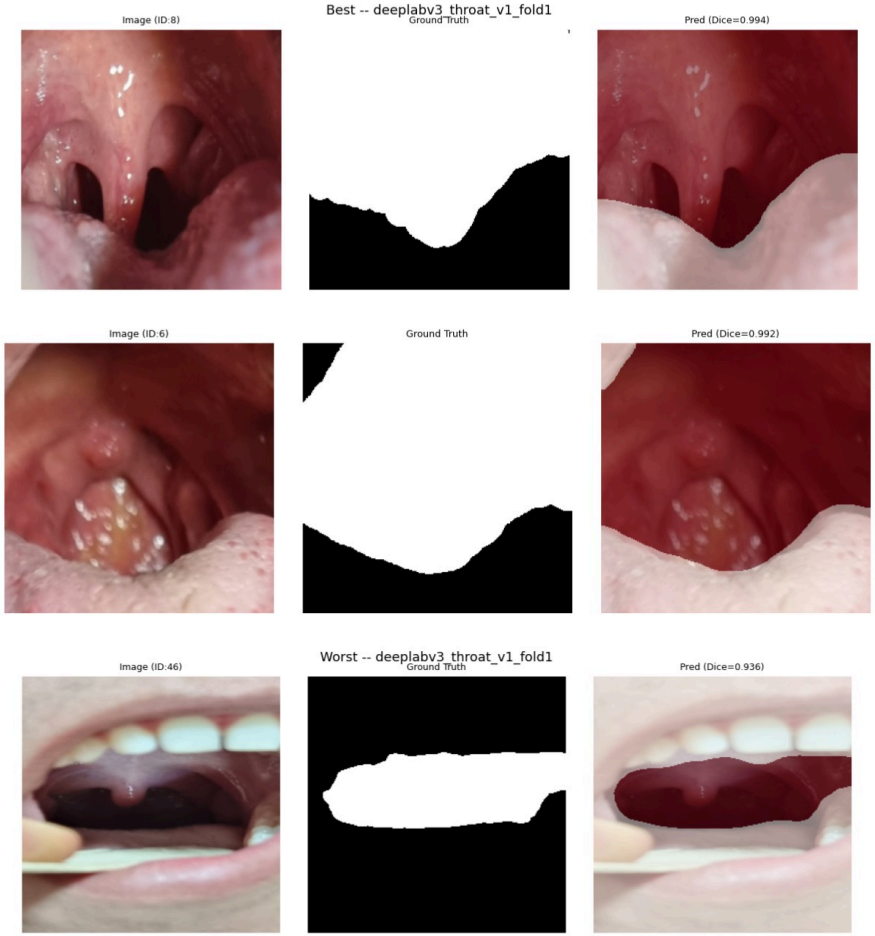
Annotations. Green boxes: exudate patches. Cyan boxes: specular reflections. The hand-labeled distinction is what trained the detector to distinguish between them.



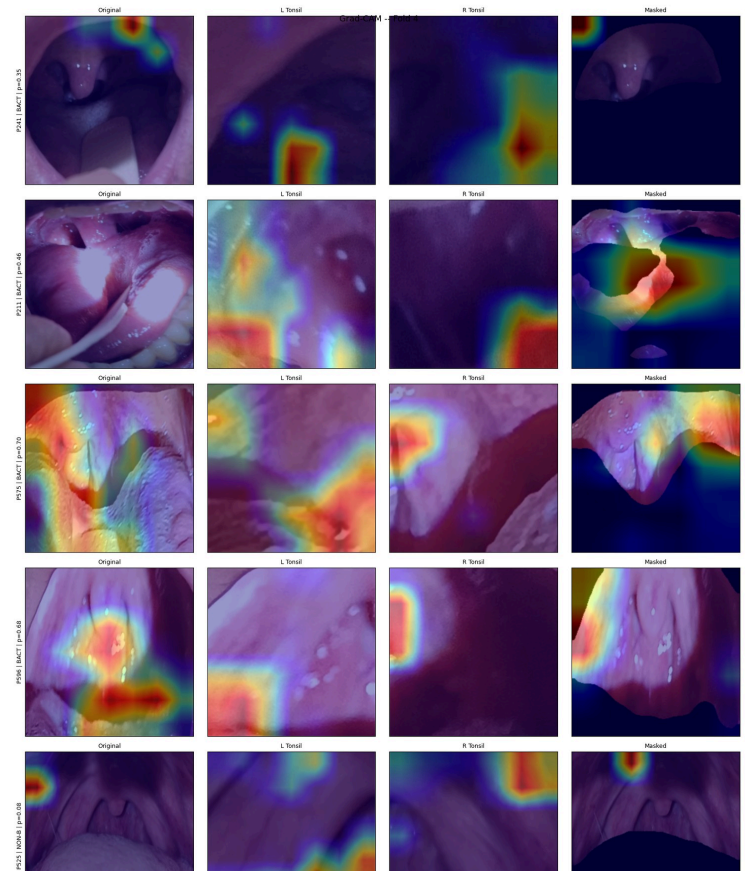
Predictions. Predictions on that bacterial image in the validation fold. The detector correctly localizes most exudate tissue while ignoring nearby reflections. Blue = my annotations, pink = model prediction.

I tried segmenting out the tongue, teeth and lips — but it did not help

I trained a DeepLabV3 segmentation model to isolate the throat interior from teeth, lips, and surrounding tissue, intending to feed the mask into the MLP as a fourth feature stream. The segmenter produced clean crops, but adding it didn't improve classifier performance — evidence that the existing three streams may have already been capturing the relevant signal, and that further gains would need to come from elsewhere.



DeepLabV3 throat-interior masks: image, ground truth, and prediction, from best to worst.



The Grad-CAM analysis was informative: the model sometimes used the segmentation mask inconsistently or not at all. Grad-CAM is a regular tool in my failure-analysis workflow.

Cross-Validated Results

5-fold stratified cross-validation on 387 patients (44 bacterial, 343 non-bacterial; 11.4% bacterial prevalence).

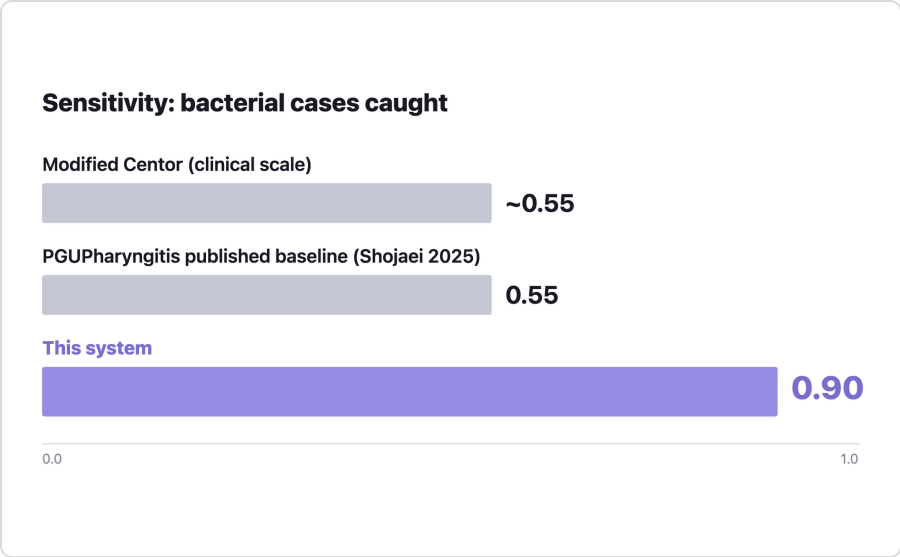
0.80	0.90	0.81	0.98
AUC	Sensitivity	Specificity	NPV

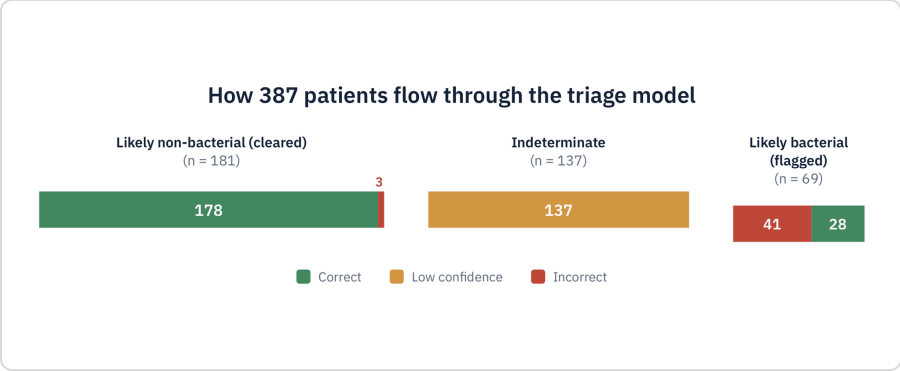
At my chosen threshold, the model correctly identifies most patients a physician panel would agree are unlikely to have a bacterial infection.

It reaches a 98% negative predictive value (NPV), meaning when the system says “non bacterial,” it is correct 98% of the time. However, positive predictive value is only 37% (0.37). Similarly, negative confidence is 84%, but positive confidence is 46%. This is expected given the class imbalance -- there are only 44 bacterial cases out of 387 training images. The model has enough data to learn what bacterial infection does not look like, but not enough to learn what it does look like.

Part of how the system catches 90% of bacterial cases is by "knowing when it doesn't know." About 35% of patients automatically get routed to a clinician because the model isn't confident enough to answer (i.e. indeterminate) — and the 90% reflects performance on the cases it does answer.

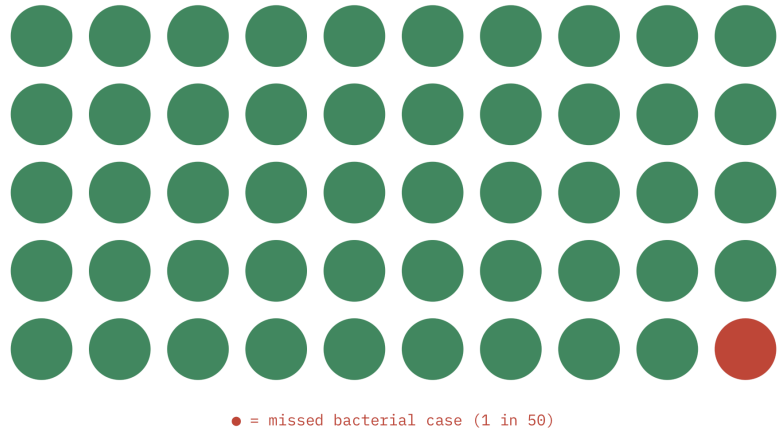
Even if you force the model to answer every case, no exceptions, it still catches 80% of bacterial infections — well above the 55% caught by the PGUPharyngitis paper's best published baseline.





Performance against targets

My advisor set two targets: ≥80% mean confidence on negative predictions and ≥50% on positive predictions. The system met the first (84.4%) but missed the second (45.8%). This matches the dataset-level asymmetry: exudate is strongly negative-predictive but weakly positive-predictive, leading to high NPV but low PPV.



Does it generalize?

I wanted to see if the model would work in the real world, which meant I needed to test it on a completely different set of images the model had never seen before. So I ran the trained model against two other public smartphone throat datasets: Roboflow's "Pharyngitis Detecting" (358 images) and Kaggle's "Pharyngitis Images" (335 images). The model held up: AUC of 0.87 and 0.89, respectively.

0.87

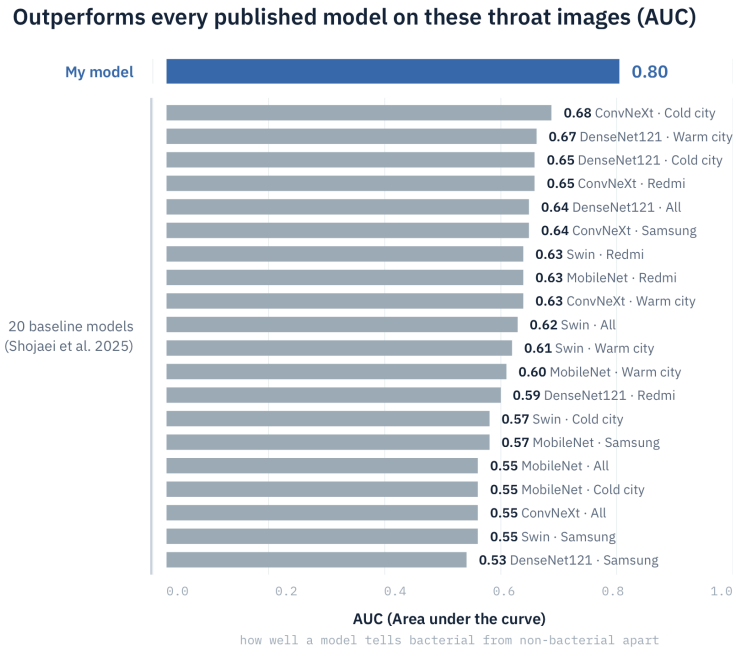
AUC on Roboflow "Pharyngitis Detecting" (358 images)

0.89

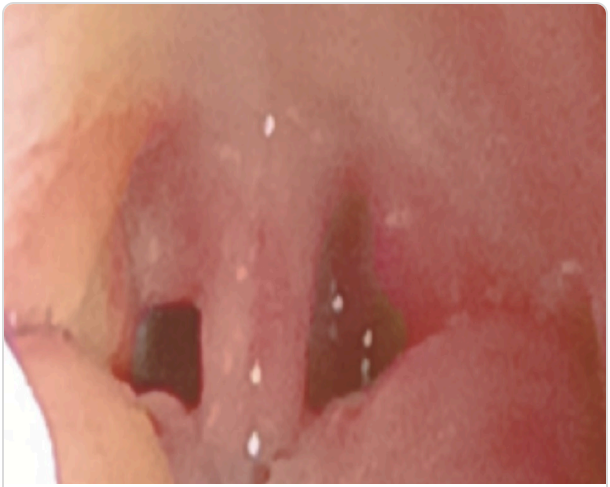
AUC on Kaggle "Pharyngitis Images" (335 images)

My pipeline outperforms all published baseline models on the throat images

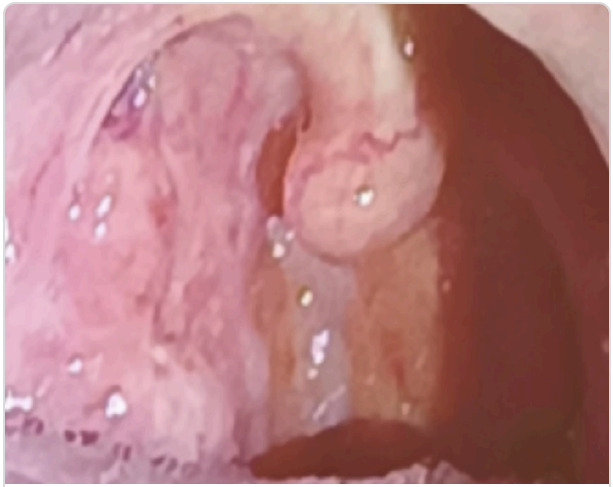
The 20 published baseline models reached 0.53 to 0.68 AUC when distinguishing bacterial from non-bacterial throats. My model achieved 0.80 AUC, showing major improvement.



Some difficult images my model was able to correctly identify as likely having strep throat



True = bacterial · predicted = bacterial · p(bact) = 0.40

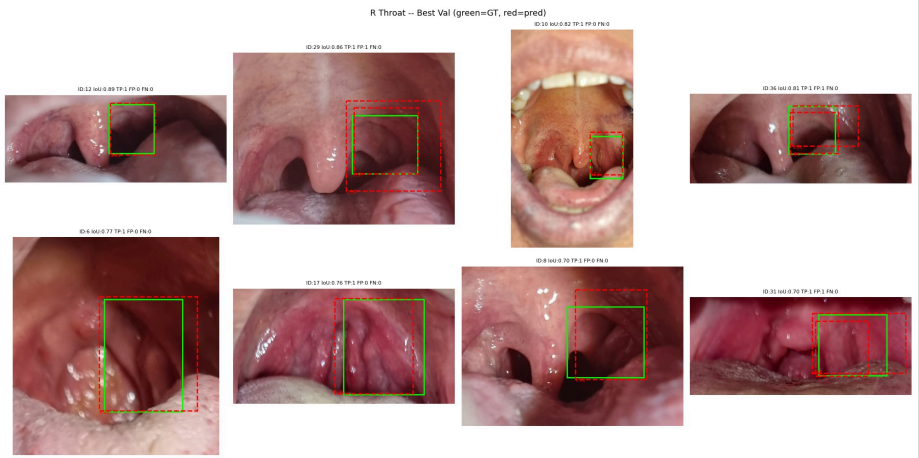


True = bacterial · predicted = bacterial · p(bact) = 0.40

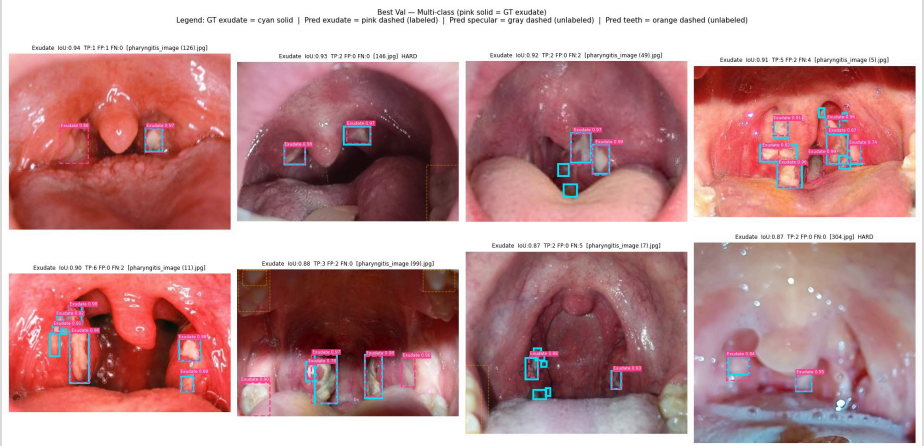


True = bacterial · predicted = bacterial · p(bact) = 0.13

Examples of ROI detections



Left tonsil detector, best cases (green = ground truth, red = prediction)



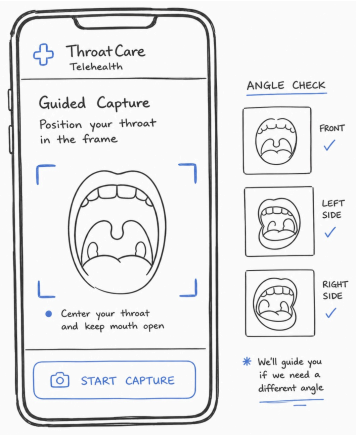
Exudate detector: exudate (cyan), reflections, and teeth

Next Steps

Guided image capture: requirements

The capture step is the second half of the product. At the moment it is just a specification, I haven't built it yet. It would run an on-device, real-time check that tonsils are framed, lighting is sufficient, and glare is below threshold before allowing submission. Key requirements:

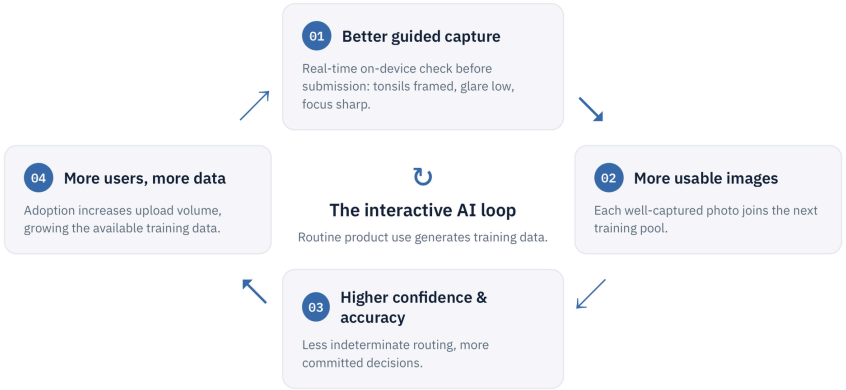
- On-device edge detection for anatomical framing (not cloud)
- Detect teeth dominance, shadow coverage, and whether tonsil edges are in frame
- Trigger capture when alignment is correct
- Support multi-angle capture to reduce glare and improve tonsil visibility



An example of how guided capture might work. The key is getting images from multiple angles.

Deployment-driven model iteration

After launch, the system keeps improving. Every high quality photo a patient submits can become a training example for the next model version. That means everyday product use steadily creates new labeled data, without needing additional expert annotation. Each image can be labeled automatically once the patient's confirmed diagnosis is available.



In a conventional supervised learning setup, additional training data would require additional expert labeling. An interactive AI loop replaces some of that labeling effort with guided capture and confident routing — patients submitting well-captured photos generate training examples as a byproduct of using the product. Labels come from diagnoses.

THANK YOU

Triaging Strep with your Phone

An AI triage aid built with clinicians, on real patient photos, designed to know when it doesn't know.

esimmens@gmail.com

Eddie Simmens · Product Developer + Researcher · Directed research, 2026

USC University of Southern California