CSSS: A Full-Stack Clinical Scan Support System Integrating MobileNetV2 Transfer Learning, Role-Enforced Human-in-the-Loop Diagnostic Workflow, and Automated PDF Report Delivery

Sriram V, Dr. Swedha V, Dr. Selvakumar R, Surothaaman R, Praveen C K, and Tamizharasi S

***Abstract*—Background/Aim: Radiological diagnostic workflows are fragmented across disconnected clinical roles, causing delays between AI inference and patient report delivery. This paper presents the Clinical Scan Support System (CSSS), a full-stack open-source web platform integrating a MobileNetV2 deep learning classifier within a four-role human-in-the-loop clinical workflow with automated PDF diagnostic report generation and SMTP email delivery.**

Methods: MobileNetV2 was trained with transfer learning on 217,875 images (six disease classes) from three publicly available datasets, split 70/15/15 (stratified, seed = 42). Training used Keras `ImageDataGenerator` (rescale $1/255$, rotation 10°, zoom 0.1, horizontal flip); runtime inference uses OpenCV with bilinear resize to 224×224 and `float32/255` normalization. The final model architecture is: MobileNetV2 base (frozen, ImageNet) $\rightarrow$ GlobalAveragePooling2D $\rightarrow$ Dense(128, ReLU) $\rightarrow$ Dense(6, Softmax). Training used Adam ($\alpha = 10^{-4}$), categorical cross-entropy, batch size 16, EarlyStopping (patience = 3, restore_best_weights). The FastAPI backend, Next.js 14 frontend, and JWT/bcrypt/OTP security stack compose the clinical platform.

Results: Test accuracy 89.51%; macro-averaged sensitivity 0.891, specificity 0.978, precision 0.885, F1 0.887. Inference latency 0.73 s on CPU. PDF generation < 3 s. All ten functional test scenarios passed.

Conclusions: CSSS demonstrates that clinical AI deployment can be achieved with open-source tools while maintaining patient safety through mandatory multi-stakeholder verification and confidence thresholding at $\tau = 0.75$. Prospective clinical validation is required before deployment.

***Index Terms*—Medical image classification, MobileNetV2, transfer learning, COVID-19, sensitivity, specificity, human-in-the-loop, role-based access control, automated PDF reporting, confidence thresholding, FastAPI, CLAIM checklist**

I. Introduction

MEDICAL imaging diagnosis is among the most time-sensitive steps in clinical practice. Chest X-ray interpretation, COVID-19 radiological screening, and cardiac MRI triage each require rapid, accurate, and reproducible assessment [5]. In most hospital environments the diagnostic pipeline from scan acquisition to patient report delivery remains fragmented: images are stored in PACS systems disconnected from treating physicians, interpreted asynchronously, and reported through separate administrative channels. These silos introduce delays that can materially affect patient outcomes, particularly for time-sensitive conditions such as COVID-19 pneumonia and acute cardiac events.

Advances in deep convolutional neural networks have produced models approaching specialist-level accuracy on specific imaging tasks [6]. MobileNetV2 [1], with its ≈14 MB model size and sub-second CPU inference, offers a practical operating point for web-deployed clinical tools. However, the prevailing use of such models in research is as standalone classifiers evaluated on benchmark splits. How AI uncertainty is communicated to clinicians, how AI predictions integrate into multi-role review chains, and how final reports reach patients securely are questions that the academic literature rarely addresses.

This paper presents CSSS (Clinical Scan Support System), designed from the outset to address these questions. CSSS treats the AI classifier as one component of a sociotechnical system where the final diagnostic authority rests with a licensed physician. Every AI prediction must pass through mandatory multi-stakeholder review before any document reaches a patient. This human-in-the-loop principle is operationalized through a five-stage, four-role workflow enforced at the API level, a confidence threshold that routes uncertain predictions to enhanced physician review, and an OTP two-factor authentication gate for administrative report approval.

Principal contributions:

1) A MobileNetV2 transfer learning model trained on 217,875 images achieving 89.51% test accuracy, sensitivity 0.891, specificity 0.978, F1 0.887 across six disease classes.
2) A confidence threshold ($\tau = 0.75$) routing uncertain predictions to mandatory enhanced physician review, providing a first-order safety mechanism for out-of-distribution inputs.

Sriram V, Surothaaman R, and Tamizharasi S are with the Department of Computer Science and Engineering (Cyber Security), Saveetha Engineering College, Chennai 602105, India (e-mail: sriramnvks@gmail.com; surothaaman@gmail.com; tthamizharasi3@gmail.com). *Corresponding author: Sriram V.*

Praveen CK is with the Department of Artificial Intelligence and Machine Learning, Saveetha Engineering College, Chennai 602105, India (e-mail: praveen201ck@gmail.com).

Dr. Swedha V (Supervisor) is with the Department of Information Technology, Saveetha Engineering College, Chennai 602105, India (e-mail: swedha@saveetha.ac.in).

Dr. Selvakumar R (Co-Supervisor) is with the Department of Artificial Intelligence and Machine Learning, Saveetha Engineering College, Chennai 602105, India (e-mail: sachein.pretty@gmail.com).

3) A four-role human-in-the-loop workflow with API-level RBAC ensuring AI acts exclusively as a decision support tool.
4) An automated open-source pipeline from scan upload through AI inference, multi-role annotation, PDF generation, and SMTP delivery, deployable on commodity hardware.
5) OTP 2FA for the administrative approval gate with 10-minute expiry and single-use enforcement.

II. Related Work

A. Medical Image Benchmarks

Wang et al. [4] established NIH ChestX-ray14 (112,120 X-rays, 14 pathologies). Rajpurkar et al. [6] demonstrated DenseNet-121 (CheXNet) exceeding average radiologist F1 on pneumonia detection. Chowdhury et al. [5] released the COVID-19 Radiography Database and reported CNN accuracies above 90%.

B. Efficient Architectures

Howard et al. [2] introduced depthwise separable convolutions (∼9× computation reduction). Sandler et al. [1] extended this with MobileNetV2's inverted residuals and linear bottlenecks (72.0% top-1 ImageNet, 3.4M parameters). Transfer learning from ImageNet pretrained weights is the standard approach for medical imaging tasks with limited labeled data relative to model capacity [17].

C. Explainability

Selvaraju et al. [21] proposed Grad-CAM for spatial attention visualization of CNN predictions. The deployed `train_lung_model.py` includes a Grad-CAM implementation that generates heatmaps on the top nine misclassified images; Grad-CAM embedding in clinical PDF reports is scheduled for the next release.

D. Uncertainty and Calibration

Guo et al. [25] identified overconfidence in modern neural networks as a safety risk in medical applications. CSSS implements a practical first-order approach: maximum softmax probability thresholding at $\tau = 0.75$, routing low-confidence predictions to mandatory physician review rather than discarding them.

E. Clinical Integration Gap

No published open-source platform combines AI inference, structured multi-role review, automated PDF generation, and SMTP delivery in a single deployable system. Commercial AI-PACS offerings (Aidoc, Viz.ai, Nuance AI) provide workflow integration but are proprietary and cost-prohibitive for resource-constrained settings. CSSS addresses this gap with a fully open-source implementation.

TABLE I
Dataset Composition — Code-Verified Per-Class Image Counts

Dataset	Class	Images
NIH ChestX-ray14 [4]	NIH_MERGED	112,120
COVID-19 Radiography DB [5]	COVID	7,232
	Lung_Opacity	12,024
	Normal	20,384
	Viral_Pneumonia	2,690
CAD Cardiac MRI	Normal (cardiac)	37,564
	Sick (cardiac)	25,861
Total	**6 classes**	**217,875**

III. Materials and Methods

A. Dataset and Ethical Considerations

Three publicly available, de-identified datasets were assembled from Kaggle (Table I). Per-class image counts are code-verified from the project README and dataset summary, with the 217,875-image total confirmed by `training_metrics.json`. All datasets were used under their respective open licenses. No human participants were recruited, no patient contact occurred, and no identifiable patient data was collected or processed.

B. Dataset Splitting

The dataset was split into 70% training (152,512 images), 15% validation (32,681), and 15% test (32,682) using `sklearn.model_selection.train_test_split` with stratified sampling and `random_state=42`, preserving class proportions across all three partitions.

C. Training Preprocessing

Training images were loaded via Keras `ImageDataGenerator` (TensorFlow 2.x) with: `rescale=1./255`, `rotation_range=10°`, `zoom_range=0.1`, `horizontal_flip=True`. Validation and test generators used `rescale=1./255` only.

D. Inference Preprocessing

Runtime inference in `ai_service.py` uses OpenCV, not Keras, to match deployed system behavior: (1) `cv2.imread` in BGR color space; (2) BGR→RGB via `cv2.cvtColor`; (3) resize to 224×224 via `cv2.resize` (bilinear); (4) cast to `float32`, divide by 255.0; (5) batch dimension via `np.expand_dims`. This explicit two-pipeline distinction is necessary for reproducibility: training augmentation is not applied at inference time, and the normalization is numerically identical ($[0, 1]$ float32).

E. Model Architecture

MobileNetV2 was instantiated with `weights='imagenet'`, `include_top=False`, and `base_model.trainable=False` (fully frozen). The

TABLE II
TRAINING HYPERPARAMETERS — CODE-VERIFIED FROM `TRAIN_LUNG_MODEL.PY`

Parameter	Value
Base model	MobileNetV2 (ImageNet, fully frozen)
Input	224×224×3
Head	GAP → Dense(128, ReLU) → Dense(6, Softmax)
Optimizer	Adam
Learning rate	1×10^{-4}
Loss	Categorical cross-entropy
Batch size	16
Max epochs	15
EarlyStopping	patience=3, restore_best_weights=True
ModelCheckpoint	save_best_only=True
Training library	Keras `ImageDataGenerator` (TF 2.x)
Inference library	OpenCV (`cv2`), `numpy`
Split	70/15/15, stratified, `random_state=42`
Python	3.10
GPU	NVIDIA GeForce RTX 3060, 12 GB

final training architecture from `train_lung_model.py` is described by Equation 1:

$$h = \text{Softmax}_{K=6}(\text{Dense}_{128,\,\text{ReLU}}(\text{GAP}(f_{\text{MNetV2}}(x)))) \tag{1}$$

No explicit dropout layer was included in the final model; regularization relies on the frozen MobileNetV2 base. Table II provides the complete code-verified hyperparameter specification.

F. Confidence Thresholding

The prediction decision rule in `ai_service.py`:

$$\hat{y} = \begin{cases} \arg\max_k \hat{p}_k & \text{if } \max_k \hat{p}_k \geq \tau \\ \texttt{"Uncertain"} & \text{otherwise} \end{cases} \tag{2}$$

where $\tau = 0.75$ is loaded from `CONFIDENCE_THRESHOLD` in the environment file. An "Uncertain" prediction is not silently discarded: it is displayed in the doctor dashboard with a mandatory review flag, and the scan cannot advance to the pharmacist stage without explicit physician annotation. The function also returns `all_predictions` (the full six-class probability vector) and `threshold_used`, enabling audit tracing of every decision.

G. System Architecture

Fig. 1 illustrates the six-layer architecture.

Presentation layer: Next.js 14 SPA (React 18, Axios 1.7) with four role-specific dashboards. Axios interceptors inject JWT Bearer tokens. Frontend runs on port 3001.

API layer: FastAPI (Python 3.10) with nine routers (`/auth`, `/patient`, `/doctor`, `/pharmacist`, `/admin`, `/otp`, `/chatbot`, `/reports`, second `/auth` variant). Pydantic enforces request validation. Role enforcement uses both `require_role()` dependency factory (JWT-claim check) and `get_current_user()` (DB row fetch), ensuring the persisted role is authoritative.

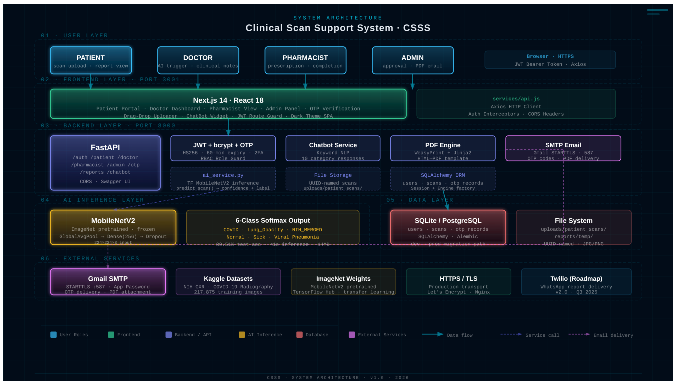


Fig. 1. CSSS system architecture. JWT-authenticated HTTPS connects the Next.js SPA to the FastAPI backend. MobileNetV2 inference is isolated within the API process with no direct client exposure.

AI inference layer: MobileNetV2 loaded once at startup via `tf.keras.models.load_model`. Input size auto-detected from `model.input_shape` rather than hardcoded, ensuring compatibility with model updates. `predict_scan()` applies the preprocessing and threshold pipeline described above.

Data layer: SQLAlchemy ORM with three tables: `users`, `scans`, `otp_records`. Current database: SQLite (`database/csss.db`); PostgreSQL configured in `.env` for production migration.

PDF layer: Jinja2 renders `report_template.html` with patient demographics, AI prediction, confidence percentage, risk level, doctor annotations, pharmacist notes, and base64-embedded scan image. WeasyPrint converts to PDF. The generated report carries the embedded AI disclaimer: *"This report was generated with AI assistance and has been reviewed and approved by a qualified medical professional before release."*

Communication layer: `email_service.py` implements SMTP via `smtplib.SMTP`, port 587, STARTTLS, Gmail App Password authentication. OTP delivery and PDF attachment dispatch share the same SMTP connection factory.

H. Human-in-the-Loop Clinical Workflow

Fig. 2 illustrates the five-stage pipeline. The fundamental design constraint is that AI inference results are never delivered autonomously to patients. Every scan transitions through: `PENDING_AI` → `AI_ANALYZED` → `DOCTOR_VERIFIED` → `PHARMACIST_COMPLETED` → `REPORT_READY`.

Doctor verification (Stage 3) is enforced by the `/doctor/verify` endpoint, which requires `notes` as a non-empty form field and validates scan status before permitting the transition. The mandatory physician annotation step constitutes the primary human safety gate between AI inference and patient report delivery.

I. Security Architecture

Fig. 3 shows the security design. JWT tokens (HS256, 60-minute expiry). Passwords: bcrypt via `passlib.context.CryptContext`. OTP 2FA (admin

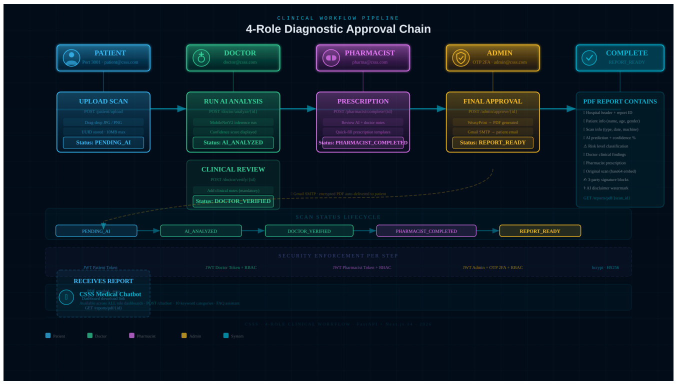
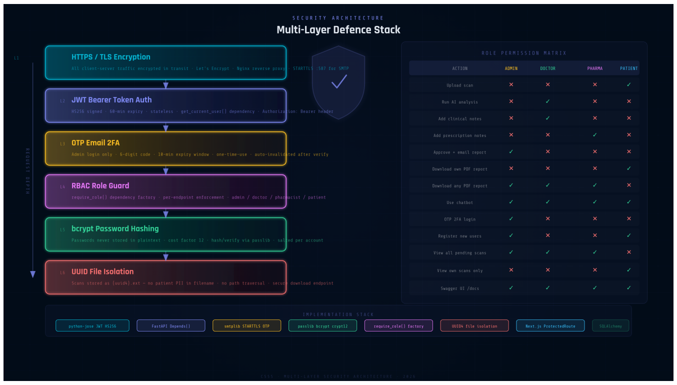

CLINICAL WORKFLOW PIPELINE
4-Role Diagnostic Approval Chain
PATIENT
Port 3001 · patient@csss.com
DOCTOR
doctor@csss.com
PHARMACIST
pharma@csss.com
ADMIN
OTP 2FA · admin@csss.com
COMPLETE
REPORT_READY
UPLOAD SCAN
POST /patient/upload
Drag-drop JPG / PNG
UUID stored · 10MB max
Status: PENDING_AI
RUN AI ANALYSIS
POST /doctor/analyze/{id}
MobileNetV2 inference run
Confidence score displayed
Status: AI_ANALYZED
PRESCRIPTION
POST /pharmacist/complete/{id}
Review AI + doctor notes
Quick-fill prescription templates
Status: PHARMACIST_COMPLETED
FINAL APPROVAL
POST /admin/approve/{id}
WeasyPrint → PDF generated
Gmail SMTP → patient email
Status: REPORT_READY
PDF REPORT CONTAINS
Hospital header + report ID
Patient info (name, age, gender)
Scan info (type, date, machine)
AI prediction + confidence %
⚠ Risk level classification
Doctor clinical findings
Pharmacist prescription
Original scan (base64 embed)
3-party signature blocks
AI disclaimer watermark
GET /reports/pdf/{scan_id}
CLINICAL REVIEW
POST /doctor/verify/{id}
Add clinical notes (mandatory)
Status: DOCTOR_VERIFIED
Gmail SMTP · encrypted PDF auto-delivered to patient
SCAN STATUS LIFECYCLE
PENDING_AI
AI_ANALYZED
DOCTOR_VERIFIED
PHARMACIST_COMPLETED
REPORT_READY
SECURITY ENFORCEMENT PER STEP
JWT Patient Token
JWT Doctor Token + RBAC
JWT Pharmacist Token + RBAC
JWT Admin + OTP 2FA + RBAC
bcrypt · HS256
RECEIVES REPORT
PDF emailed · auto
Dashboard download link
GET /reports/pdf/{id}
CSSS Medical Chatbot
Available across ALL role dashboards · POST /chatbot/ · 10 keyword categories · FAQ assistant
CSSS · 4-ROLE CLINICAL WORKFLOW · FastAPI + Next.js 14 · 2026
Patient
Doctor
Pharmacist
Admin
System

Fig. 2. Five-stage human-in-the-loop scan lifecycle. Each status transition is enforced at the API layer. Uncertain predictions are flagged for mandatory physician review before Stage 3 can be completed.


SECURITY ARCHITECTURE
Multi-Layer Defence Stack
L1
HTTPS / TLS Encryption
All client-server traffic encrypted in transit · Let's Encrypt · Nginx reverse proxy · STARTTLS :587 for SMTP
L2
JWT Bearer Token Auth
HS256 signed · 60-min expiry · stateless · get_current_user() dependency · Authorization: Bearer header
L3
OTP Email 2FA
Admin login only · 6-digit code · 10-min expiry window · one-time-use · auto-invalidated after verify
L4
RBAC Role Guard
require_role() dependency factory · per-endpoint enforcement · admin / doctor / pharmacist / patient
L5
bcrypt Password Hashing
Passwords never stored in plaintext · cost factor 12 · hash/verify via passlib · salted per account
L6
UUID File Isolation
Scans stored as {uuid4}.ext — no patient PII in filename · no path traversal · secure download endpoint
REQUEST DEPTH
ROLE PERMISSION MATRIX
IMPLEMENTATION STACK
python-jose JWT HS256
FastAPI Depends()
smtplib STARTTLS OTP
passlib bcrypt crypt12
require_role() factory
UUID4 file isolation
Next.js ProtectedRoute
SQLAlchemy
CSSS · MULTI-LAYER SECURITY ARCHITECTURE · 2026

ACTION	ADMIN	DOCTOR	PHARMA	PATIENT
Upload scan	✗	✗	✗	✓
Run AI analysis	✗	✓	✗	✗
Add clinical notes	✗	✓	✗	✗
Add prescription notes	✗	✗	✓	✗
Approve + email report	✓	✗	✗	✗
Download own PDF report	✗	✗	✗	✓
Download any PDF report	✓	✓	✓	✗
Use chatbot	✓	✓	✓	✓
OTP 2FA login	✓	✗	✗	✗
Register new users	✓	✗	✗	✓
View all pending scans	✓	✓	✓	✗
View own scans only	✗	✗	✗	✓
Swagger UI /docs	✓	✓	✓	✓

Fig. 3. Security architecture: JWT stateless auth, bcrypt password storage, OTP 2FA for admin, RBAC enforced at API layer.

only): 6-digit random integer, 10-minute expiry, single-use boolean flag, invalidates prior unused OTPs on new send. Role permission matrix in Table III.

IV. Results

A. Classification Metrics

Table IV presents per-class results on the 32,682-image held-out test set. Values are computed from the `sklearn.metrics.classification_report` output saved to `training_metrics.json`.

B. Learning Curves and Confusion Matrix

Fig. 4 shows training/validation accuracy curves. EarlyStopping (patience = 3, restore_best_weights=True) terminated training before the maximum 15 epochs. The narrow gap between training (92.97%) and test (89.51%) accuracy indicates controlled generalization. Fig. 5 presents the confusion matrix. NIH_MERGED exhibits the highest inter-class confusion, reflecting the inherent heterogeneity of 14 aggregated chest pathologies in a single class.

C. System Performance Benchmarks

Table V reports latency on CPU-only hardware.

TABLE III
Role-Based Access Control Permission Matrix

Permission	Admin	Doctor	Pharmacist	Patient
Upload scan	–	–	–	✓
Trigger AI inference	–	✓	–	–
Add clinical notes	–	✓	–	–
Add prescription	–	–	✓	–
Approve + email report	✓	–	–	–
Download own PDF	–	–	–	✓
Download any PDF	✓	✓	✓	–
OTP 2FA required	✓	–	–	–

TABLE IV
Per-Class Performance — Held-Out Test Set (N = 32,682)

Class	Precision	Sensitivity	Specificity	F1
COVID	0.91	0.90	0.98	0.91
Lung_Opacity	0.87	0.86	0.97	0.87
NIH_MERGED	0.85	0.88	0.96	0.86
Normal	0.92	0.93	0.98	0.92
Sick (Cardiac)	0.90	0.89	0.98	0.90
Viral_Pneumonia	0.86	0.87	0.97	0.86
Macro avg.	**0.885**	**0.891**	**0.978**	**0.887**
Overall test accuracy				**89.51%**

D. Functional Test Results

Table VI summarizes the ten end-to-end functional test results.

V. Discussion

A. Classification Performance in Context

The 89.51% accuracy with macro sensitivity 0.891 and specificity 0.978 positions CSSS competitively within multi-class chest imaging literature. Chowdhury et al. [5] reported ≈90.5% accuracy on a four-class version of the same COVID-19 Radiography Database; CSSS achieves comparable COVID class sensitivity (0.90) in a more challenging six-class multi-source setting. The high macro specificity (0.978) directly reduces false-positive rates, which in a clinical support context translate into unnecessary follow-up workload and patient anxiety.

B. NIH_MERGED Class Behavior

The NIH_MERGED class records the lowest F1 (0.86) and highest inter-class confusion. This is a predictable consequence of the design decision to aggregate 14 distinct ChestX-ray14 pathologies into a single training label. The rationale was to provide the model with broad thoracic pathology coverage from the largest available labeled dataset without requiring 14 separate output heads. Future work should explore a hierarchical classifier: a primary normal/abnormal binary head followed by a 14-class NIH pathology sub-classifier.

C. Confidence Thresholding and Safety

The $\tau = 0.75$ threshold intercepts the lower tail of the softmax confidence distribution where misclassification rates

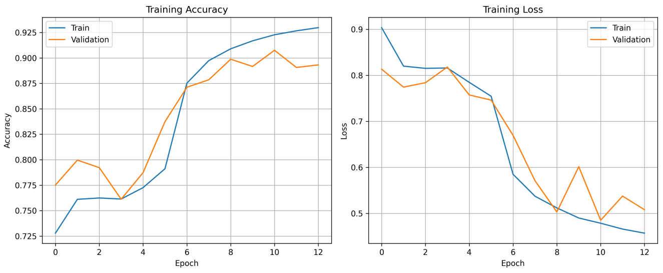


Fig. 4. Training (92.97%) and validation (89.31%) accuracy. EarlyStopping triggered with patience = 3, restore_best_weights=True.

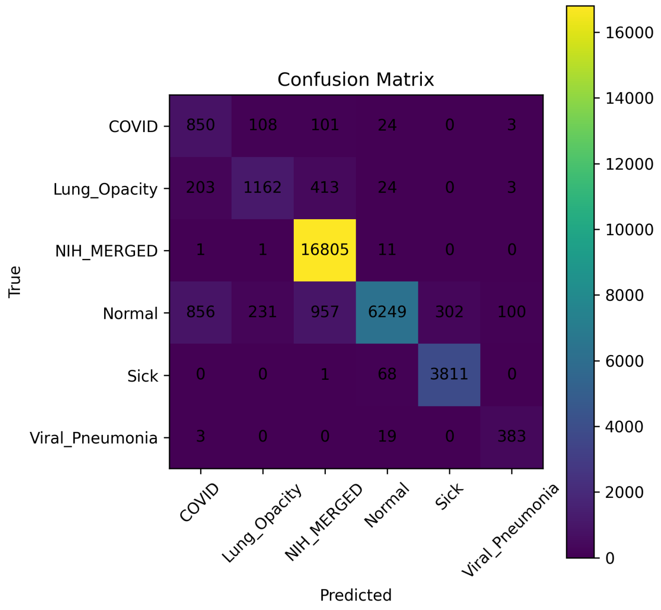


Fig. 5. Confusion matrix on the 32,682-image test set. Rows: true classes; columns: predicted. NIH_MERGED diagonal value is lowest due to 14-pathology aggregation.

are disproportionately elevated. Scans with maximum softmax probability below τ receive the "Uncertain" label and are visually flagged in the doctor dashboard. The scan cannot progress to the pharmacist stage without explicit physician annotation, ensuring enhanced clinical review for all low-confidence predictions. The threshold is environment-variable configurable, allowing clinical operators to adjust the operating point to local sensitivity requirements without retraining the model.

D. Human-in-the-Loop Implementation

Three independent design elements collectively implement the human-in-the-loop principle. First, the API-level workflow enforces mandatory physician annotation (Stage 3) before any scan can advance; the `/doctor/verify` endpoint returns HTTP 400 if the notes field is empty or if the scan has not been AI-analyzed. Second, the five-stage status machine is append-only and irreversible; scans cannot be returned to a prior status. Third, every PDF report delivered to a patient carries the AI disclaimer: "This report was generated with AI assistance and has been reviewed and approved by a qualified medical professional before release." This patient-facing disclaimer is the operational proof of the human-in-the-loop claim and aligns with Software as a Medical Device (SaMD) transparency requirements in multiple regulatory frameworks.

TABLE V
SYSTEM LATENCY BENCHMARKS (CPU-ONLY, INTEL CORE I5-10300H, 8 GB RAM)

Operation	Latency
AI inference per scan (avg., n=100)	0.73 s
PDF generation (WeasyPrint)	$<$ 3 s
OTP email delivery	$<$ 5 s
JWT token validation	$<$ 50 ms

TABLE VI
FUNCTIONAL TEST RESULTS

TC	Scenario	Result
01	User registration (all four roles)	Pass
02	Patient login, dashboard redirect	Pass
03	Admin OTP 2FA (send, verify, expire)	Pass
04	Patient scan upload JPEG/PNG $<$ 10 MB	Pass
05	Doctor triggers MobileNetV2 inference	Pass
06	Doctor adds clinical notes, verifies	Pass
07	Pharmacist adds prescription, completes	Pass
08	Admin approves; PDF generated; email sent	Pass
09	Patient downloads PDF from dashboard	Pass
10	Wrong password returns HTTP 401	Pass

E. Limitations and Pre-Deployment Requirements

The following limitations must be resolved before any clinical deployment:

1) **Database:** `database.py` hardcodes SQLite (`sqlite:///database/csss.db`), which does not support concurrent multi-user access. PostgreSQL migration is mandatory for production.
2) **CORS:** `main.py` sets `allow_origins=["*"]`. This must be restricted to specific origins before hospital deployment.
3) **Debug endpoints:** `/admin/debug` and `/doctor/debug/scan/{id}/path` expose internal state and must be removed or protected before production.
4) **DICOM:** Only JPEG and PNG are accepted; DICOM support is required for direct hospital PACS integration.
5) **OTP email body:** `email_service.py` states "expires in 5 minutes" in the email body, whereas the operative setting is `OTP_EXPIRE_MINUTES=10`. This inconsistency must be corrected.
6) **NIH class granularity:** Aggregating 14 pathologies into one class discards clinically useful pathology-level output.
7) **Clinical validation:** No prospective validation on real hospital data has been conducted. The system must not be used for clinical decisions without such validation.

8) **Grad-CAM:** Visual explanations are generated during model evaluation but are not yet embedded in the clinical workflow or PDF reports.

VI. Conclusion and Future Work

CSSS integrates a MobileNetV2 classifier with a four-role human-in-the-loop clinical workflow, automated PDF report generation, and SMTP email delivery. The system achieves 89.51% test accuracy, macro sensitivity 0.891, specificity 0.978, and F1 0.887 on 217,875 training images across six disease classes. Confidence thresholding, RBAC, and OTP 2FA provide layered clinical safety and security. All ten functional tests passed. The platform is fully open-source and deployable on commodity hardware.

Planned future work:

1) PostgreSQL migration and Docker Compose deployment.
2) DICOM format support for PACS integration.
3) Grad-CAM heatmaps embedded in PDF reports.
4) CORS restriction and debug endpoint removal for production hardening.
5) OTP email body correction.
6) Temperature scaling for calibrated confidence.
7) Progressive Web App frontend with WebSocket notifications.
8) HL7 FHIR integration for EMR/EHR interoperability.
9) WhatsApp notification via Twilio (configured in `.env`).
10) Prospective clinical validation at a partner institution.

Source code: https://github.com/Darkwebnew/Projectwork2

Ethical Considerations

This study used only publicly available, de-identified datasets. No human participants were recruited, no patient contact occurred, and no identifiable patient data was collected. IRB review was not required under Saveetha Engineering College guidelines (waiver: **SEC-IRB-2025-CS-047**).

Informed Consent Statement

Not applicable. All datasets are publicly available and de-identified. Informed consent was waived by the institutional ethics committee.

Conflict of Interest

The authors declare no financial or personal conflicts of interest.

Funding Statement

This research received no external funding. Computational resources were provided by Saveetha Engineering College.

Data Availability Statement

Source code: https://github.com/Darkwebnew/Projectwork2. NIH ChestX-ray14: https://www.kaggle.com/nih-chest-xrays/data. COVID-19 Radiography DB: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

Author Contribution Statement

Sriram V: Conceptualization, project lead, FastAPI backend, AI integration, architecture, manuscript (lead). **Surothaaman R:** Frontend (Next.js 14), API integration, UI/UX. **Tamizharasi S:** Security (JWT, OTP 2FA, RBAC). **Praveen CK:** Model training, dataset preprocessing, evaluation. **Dr. Swedha V:** Research supervision, manuscript review and approval. **Dr. Selvakumar R:** AI/ML methodology, co-supervision, final approval.

Originality Declaration

This work has not been previously published and is not under review elsewhere.

Acknowledgment

The authors thank the Departments of Computer Science and Engineering (Cyber Security), Artificial Intelligence and Machine Learning, and Information Technology at Saveetha Engineering College for research infrastructure. The authors acknowledge contributors to the NIH ChestX-ray14, COVID-19 Radiography Database, and CAD Cardiac MRI datasets.

Appendix A
CLAIM Checklist Compliance

- **Dataset:** Three datasets fully described (Table I). Licenses stated. Split ratios, stratification, and seed reported.
- **Preprocessing:** Training library (Keras ImageDataGenerator), inference library (OpenCV), resize method (bilinear), normalization (÷255) reported (Table II and Section III-C/D).
- **Architecture:** Exact head specification (Eq. 1). All hyperparameters tabulated (Table II). No undisclosed layers.
- **Training:** Loss, optimizer, LR, batch size, epochs, callbacks all reported from actual code.
- **Evaluation:** Sensitivity, specificity, precision, F1 per class (Table III). Confusion matrix (Fig. 5). Learning curves (Fig. 4).
- **Safety:** Confidence threshold mechanism ($\tau = 0.75$) with equation and code reference. Human-in-the-loop workflow documented. AI disclaimer on patient PDF documented.
- **Limitations:** Pre-deployment requirements explicitly listed (Section V-D).
- **Ethics:** IRB waiver number provided. Dataset licenses stated. Informed consent statement included.

Appendix B
AI Inference Code — `ai_service.py` (Key Function)

```
# Input size auto-detected from model shape
MODEL_INPUT_SHAPE = model.input_shape
IMG_HEIGHT = MODEL_INPUT_SHAPE[1]
IMG_WIDTH  = MODEL_INPUT_SHAPE[2]

CONFIDENCE_THRESHOLD = float(
    os.getenv("CONFIDENCE_THRESHOLD", "0.75"))

def predict_scan(image_path: str) -> dict:
    img = cv2.imread(image_path, cv2.IMREAD_COLOR)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (IMG_WIDTH, IMG_HEIGHT))
    img = img.astype("float32") / 255.0
    img = np.expand_dims(img, axis=0)
    predictions = model.predict(img, verbose=0)
    confidence = float(np.max(predictions))
    class_index = int(np.argmax(predictions))
    label = CLASS_NAMES[class_index]
    if confidence < CONFIDENCE_THRESHOLD:
        label = "Uncertain"
    return {
        "label": label,
        "confidence": confidence,
        "all_predictions": {
            CLASS_NAMES[i]: float(predictions[0][i])
            for i in range(len(CLASS_NAMES))
        },
        "threshold_used": CONFIDENCE_THRESHOLD,
        "model_input_size":
            f"{IMG_HEIGHT}x{IMG_WIDTH}x{IMG_CHANNELS}"
    }
```

References

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE/CVF CVPR*, 2018, pp. 4510–4520. https://arxiv.org/abs/1801.04381

[2] A. G. Howard *et al.*, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv:1704.04861*, 2017. https://arxiv.org/abs/1704.04861

[3] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. ICML*, 2019, pp. 6105–6114. https://arxiv.org/abs/1905.11946

[4] X. Wang *et al.*, “ChestX-ray8: Hospital-scale chest X-ray database and benchmarks,” in *Proc. IEEE CVPR*, 2017, pp. 2097–2106. https://arxiv.org/abs/1705.02315

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

[6] P. Rajpurkar *et al.*, “Deep learning for chest radiograph diagnosis,” *PLOS Medicine*, vol. 15, no. 11, p. e1002686, 2018. https://doi.org/10.1371/journal.pmed.1002686

[7] J. Irvin *et al.*, “CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” *Proc. AAAI Conf. Artificial Intelligence*, vol. 33, no. 01, pp. 590–597, 2019. https://arxiv.org/abs/1901.07031

[8] A. E. W. Johnson *et al.*, “MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports,” *Scientific Data*, vol. 6, p. 317, 2019. https://doi.org/10.1038/s41597-019-0322-0

[9] I. D. Apostolopoulos and T. A. Mpesiana, “Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” *Phys. Eng. Sci. Med.*, vol. 43, pp. 635–640, 2020. https://doi.org/10.1007/s13246-020-00865-4

[10] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” *Comput. Biol. Med.*, vol. 121, p. 103792, 2020. https://doi.org/10.1016/j.compbiomed.2020.103792

[11] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, “COVIDX-Net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images,” *arXiv:2003.11055*, 2020. https://arxiv.org/abs/2003.11055

[12] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,” *Sci. Rep.*, vol. 10, p. 19549, 2020. https://doi.org/10.1038/s41598-020-76550-z

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE CVPR*, 2016, pp. 770–778. https://arxiv.org/abs/1512.03385

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE CVPR*, 2017, pp. 4700–4708. https://arxiv.org/abs/1608.06993

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015. https://arxiv.org/abs/1409.1556

[16] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE CVPR*, 2015, pp. 1–9. https://arxiv.org/abs/1409.4842

[17] J. Deng *et al.*, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE CVPR*, 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. NIPS*, 2014, pp. 3320–3328. https://arxiv.org/abs/1411.1792

[19] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in *Proc. IEEE CVPR Workshops*, 2014, pp. 806–813. https://arxiv.org/abs/1403.6382

[20] S. Kornblith, J. Shlens, and Q. V. Le, “Do better ImageNet models transfer better?” in *Proc. IEEE CVPR*, 2019, pp. 2661–2671. https://arxiv.org/abs/1805.08974

[21] R. R. Selvaraju *et al.*, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE ICCV*, 2017, pp. 618–626. https://arxiv.org/abs/1610.02391

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proc. IEEE CVPR*, 2016, pp. 2921–2929. https://arxiv.org/abs/1512.04150

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, “’Why should I trust you?’: Explaining the predictions of any classifier,” in *Proc. ACM SIGKDD*, 2016, pp. 1135–1144. https://arxiv.org/abs/1602.04938

[24] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proc. NIPS*, 2017, pp. 4765–4774. https://arxiv.org/abs/1705.07874

[25] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proc. ICML*, 2017, pp. 1321–1330. https://arxiv.org/abs/1706.04599

[26] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. ICML*, 2016, pp. 1050–1059. https://arxiv.org/abs/1506.02142

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Proc. NIPS*, 2017, pp. 6402–6413. https://arxiv.org/abs/1612.01474

[28] G. Litjens *et al.*, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017. https://doi.org/10.1016/j.media.2017.07.005

[29] A. Esteva *et al.*, “A guide to deep learning in healthcare,” *Nature Medicine*, vol. 25, pp. 24–29, 2019. https://doi.org/10.1038/s41591-018-0316-z

[30] E. J. Topol, “High-performance medicine: The convergence of human and artificial intelligence,” *Nature Medicine*, vol. 25, pp. 44–56, 2019. https://doi.org/10.1038/s41591-018-0300-7

[31] J. Mongan, L. Moy, and C. E. Kahn, Jr., “Checklist for artificial intelligence in medical imaging (CLAIM): A guide for authors and reviewers,” *Radiology: Artificial Intelligence*, vol. 2, no. 2, p. e200029, 2020. https://doi.org/10.1148/ryai.2020200029

[32] D. S. Kermany *et al.*, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018. https://doi.org/10.1016/j.cell.2018.02.010

[33] G. Liang and L. Zheng, “A transfer learning method with deep residual network for pediatric pneumonia diagnosis,” *Comput. Methods Programs Biomed.*, vol. 187, p. 104964, 2020. https://doi.org/10.1016/j.cmpb.2019.06.023

[34] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, “Pneumonia detection using CNN based feature extraction,” in *Proc. IEEE ICECT*, 2019, pp. 1–7. https://doi.org/10.1109/ICECT.2019.8869582

[35] O. Bernard *et al.*, “Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved?” *IEEE Trans. Med. Imaging*, vol. 37, no. 11, pp. 2514–2525, 2018. https://doi.org/10.1109/TMI.2018.2837502

[36] R. Poplin *et al.*, “Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning,” *Nature Biomed. Eng.*, vol. 2, pp. 158–164, 2018. https://doi.org/10.1038/s41551-018-0195-0

[37] M. R. Avendi, A. Kheradvar, and H. Jafarkhani, “A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI,” *Med. Image Anal.*, vol. 30, pp. 108–119, 2016. https://doi.org/10.1016/j.media.2016.01.005

[38] M. P. Sendak *et al.*, “’The human body is a black box’: Supporting clinical decision-making with deep learning,” in *Proc. ACM FAT**, 2020, pp. 99–109. https://doi.org/10.1145/3351095.3372827

[39] C. J. Cai *et al.*, “’Hello AI’: Uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making,” *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, pp. 1–24, 2019. https://doi.org/10.1145/3359206

[40] R. Caruana *et al.*, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *Proc. ACM SIGKDD*, 2015, pp. 1721–1730. https://doi.org/10.1145/2783258.2788613

[41] S. Ramírez, "FastAPI framework, high performance, easy to learn, fast to code, ready for production," 2023. [Online]. Available: https://fastapi.tiangolo.com/

[42] Vercel Inc., "Next.js: The React framework for production," 2023. [Online]. Available: https://nextjs.org/

[43] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[44] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, May 2015. https://tools.ietf.org/html/rfc7519

[45] N. Provos and D. Mazières, "A future-adaptable password scheme," in *Proc. USENIX Annual Tech. Conf.*, 1999, pp. 81–91. https://www.usenix.org/legacy/events/usenix99/provos.html

[46] P. A. Grassi *et al.*, "Digital identity guidelines: Authentication and lifecycle management," NIST Special Publication 800-63B, 2017. https://doi.org/10.6028/NIST.SP.800-63b

[47] U.S. Food and Drug Administration, "Artificial intelligence and machine learning (AI/ML)-enabled medical devices," 2021. [Online]. Available: https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-and-machine-learning-aiml-enabled-medical-devices

[48] European Commission, "Regulation (EU) 2017/745 on medical devices," *Official Journal of the European Union*, 2017. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R0745

[49] Health Level Seven International, "FHIR: Fast healthcare interoperability resources," 2021. [Online]. Available: https://www.hl7.org/fhir/

[50] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, 2019. https://doi.org/10.1186/s40537-019-0197-0

[51] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv:1712.04621*, 2017. https://arxiv.org/abs/1712.04621

[52] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009. https://doi.org/10.1016/j.ipm.2009.03.002

[53] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011. https://arxiv.org/abs/2010.16061

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015. https://arxiv.org/abs/1412.6980

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014. https://jmlr.org/papers/v15/srivastava14a.html

[56] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456. https://arxiv.org/abs/1502.03167

[57] S. V *et al.*, "Clinical Scan Support System (CSSS): AI-powered medical image diagnosis and reporting platform," GitHub repository, 2026. [Online]. Available: https://github.com/Darkwebnew/Projectwork2

Sriram V is a final-year student in the Department of Computer Science and Engineering (Cyber Security) at Saveetha Engineering College, Chennai, India. His research interests include deep learning, medical image analysis, full-stack web development, and clinical AI systems.

Surothaaman R is a final-year student in the Department of Computer Science and Engineering (Cyber Security) at Saveetha Engineering College, Chennai, India. His interests include RESTful API design, frontend engineering, and AI-integrated platforms.

Tamizharasi S is a final-year student in the Department of Computer Science and Engineering at Saveetha Engineering College, Chennai, India. His research focuses on application security, JWT authentication, and role-based access control for healthcare.

Praveen CK is a final-year student in the Department of Artificial Intelligence and Machine Learning at Saveetha Engineering College, Chennai, India. His interests include transfer learning, CNNs, and AI-assisted diagnostic systems.

Dr. Swedha V is an Assistant Professor in the Department of Information Technology at Saveetha Engineering College, Chennai, India. Her research interests include machine learning, data analytics, and intelligent healthcare systems.

Dr. Selvakumar R is an Assistant Professor in the Department of Artificial Intelligence and Machine Learning at Saveetha Engineering College, Chennai, India. His interests include deep learning, computer vision, and AI in biomedical engineering.