Clinical Scan Support System (CSSS): MobileNetV2-Based Multi-Class Medical Image Classification with Role-Enforced Human-in-the-Loop Clinical Workflow and Automated PDF Reporting

Sriram V
Dept. of CSE (Cyber Security)
Saveetha Engineering College
Chennai – 602105, India
sriramnvks@gmail.com

Dr. Swedha V *(Supervisor)*
Dept. of Information Technology
Saveetha Engineering College
Chennai – 602105, India
swedha@saveetha.ac.in

Dr. Selvakumar R *(Co-Supervisor)*
Dept. of AI & ML
Saveetha Engineering College
Chennai – 602105, India
sachein.pretty@gmail.com

Surothaaman R
Dept. of CSE (Cyber Security)
Saveetha Engineering College
Chennai – 602105, India
surothaaman@gmail.com

Praveen C K
Dept. of AI & ML
Saveetha Engineering College
Chennai – 602105, India
praveen201ck@gmail.com

Tamizharasi S
Dept. of Computer Science and Engineering
Saveetha Engineering College
Chennai – 602105, India
tthamizharasi3@gmail.com

***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 web platform that integrates a MobileNetV2 deep learning classifier within a structured four-role clinical pipeline with automated PDF diagnostic report generation and SMTP email delivery.**

Methods: MobileNetV2 was trained with transfer learning on 217,875 images spanning six disease classes assembled from three publicly available datasets, split 70/15/15 using stratified sampling (seed = 42). Training preprocessing used Keras `ImageDataGenerator` (rescale $1/255$, rotation 10°, zoom 0.1, horizontal flip); inference preprocessing uses OpenCV with bilinear resize to 224×224 and float32 normalization. The classification head is: GlobalAveragePooling2D → Dense(128, ReLU) → Dense(6, Softmax). Training used Adam ($\alpha = 10^{-4}$), categorical cross-entropy, batch size 16, EarlyStopping (patience = 3). The backend uses FastAPI (Python 3.10), the frontend Next.js 14, and security uses JWT (HS256, 60 min), bcrypt, and OTP two-factor authentication.

Results: The model achieved 89.51% test accuracy, macro-averaged sensitivity 0.891, specificity 0.978, precision 0.885, and F1 0.887. Inference averaged 0.73 s on CPU. 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 $\tau = 0.75$.

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

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 demand rapid, accurate assessment. Yet diagnostic workflows in most hospital environments remain fragmented: scans are stored in disconnected systems, interpreted asynchronously, and reported through separate administrative channels. These disconnected steps introduce delays that can materially affect patient outcomes [5].

Deep convolutional networks have demonstrated near-specialist accuracy on specific imaging tasks [6]. MobileNetV2 [1] offers a practical operating point for web deployment: approximately 14 MB model size and sub-second CPU inference. However, published systems largely evaluate classifiers in isolation, leaving unanswered how AI uncertainty is communicated, how AI predictions integrate into multi-role review chains, and how final reports are securely delivered to patients.

CSSS (Clinical Scan Support System) addresses this gap by treating the AI classifier as one component of a sociotechnical system. The final diagnostic authority rests with a licensed physician; every AI prediction passes through mandatory multi-stakeholder review before any report reaches a patient.

Principal contributions:

- MobileNetV2 trained on 217,875 images achieving 89.51% test accuracy, sensitivity 0.891, specificity 0.978, F1 0.887 across six disease classes.

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**

- Confidence thresholding ($\tau = 0.75$) that routes uncertain predictions to mandatory enhanced physician review.
- Four-role human-in-the-loop workflow enforced at the API level via RBAC.
- Automated pipeline: scan upload $\rightarrow$ AI inference $\rightarrow$ clinical annotation $\rightarrow$ PDF generation $\rightarrow$ SMTP delivery.
- OTP two-factor authentication for the administrative approval gate.

II. RELATED WORK

Wang et al. [4] established the NIH ChestX-ray14 benchmark (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%. Howard et al. [2] introduced depthwise separable convolutions for efficient inference; Sandler et al. [1] extended this with MobileNetV2's inverted residuals and linear bottlenecks. Selvaraju et al. [21] proposed Grad-CAM for visual explanation of CNN predictions. No published open-source platform combines AI inference, structured multi-role review, automated PDF generation, and SMTP delivery in a single deployable system.

III. MATERIALS AND METHODS

A. Dataset

Training data was assembled from three publicly available, de-identified Kaggle datasets (Table I). All data were used under their respective open licenses. No human participants were recruited and no identifiable patient data was collected.

The full dataset was split into 70% training (152,512), 15% validation (32,681), and 15% test (32,682) using stratified sampling (`sklearn.model_selection.train_test_split`, `random_state=42`).

B. Preprocessing

Training preprocessing used Keras `ImageDataGenerator` (TensorFlow 2.x): `rescale=1./255`, `rotation_range=10`, `zoom_range=0.1`, `horizontal_flip=True`.

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 $\rightarrow$ Dense(128, ReLU) $\rightarrow$ 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
Train preprocessing	Keras ImageDataGenerator (rescale, augment)
Inference preprocessing	OpenCV cv2 resize + float32/255
Split	70/15/15, stratified, random_state=42
Python	3.10
TensorFlow/Keras	2.x
GPU	NVIDIA GeForce RTX 3060 (12 GB)

Validation and test generators used `rescale=1./255` only (no augmentation).

Inference preprocessing uses the OpenCV (`cv2`) library consistent with the deployed `ai_service.py`: (1) load image via `cv2.imread` (BGR); (2) convert BGR$\rightarrow$RGB via `cv2.cvtColor`; (3) resize to 224×224 via `cv2.resize` (bilinear default); (4) cast to `float32` and divide by 255.0; (5) add batch dimension via `np.expand_dims`.

This two-pipeline distinction is important for reproducibility: training augmentation is not applied at inference time.

C. Model Architecture and Training

MobileNetV2 was loaded with `weights='imagenet'` and `include_top=False`, with `base_model.trainable = False` (fully frozen). The classification head per Equation 1:

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

This architecture exactly reflects the deployed `train_lung_model.py`. The model was compiled with Adam optimizer ($\alpha = 10^{-4}$) and categorical cross-entropy loss. Full hyperparameter specification is in Table II.

D. Confidence Thresholding

The prediction decision rule implemented in `ai_service.py`:

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

where $\tau = 0.75$ is loaded from the `CONFIDENCE_THRESHOLD` environment variable (defaulting to 0.75). Scans labelled "Uncertain" are flagged with a mandatory review warning in the doctor dashboard and cannot progress through the workflow without explicit

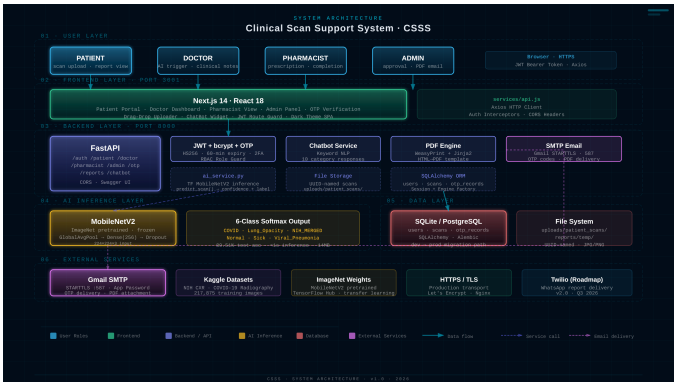

Fig. 1. CSSS six-layer architecture. FastAPI REST backend with JWT authentication; Next.js frontend with Axios interceptors; MobileNetV2 inference isolated within the API process.

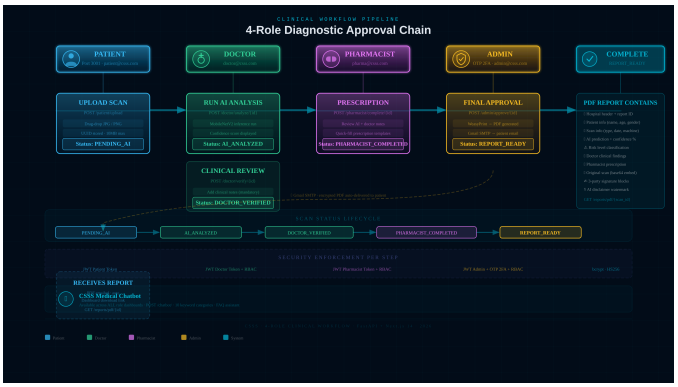

Fig. 2. Five-stage scan lifecycle. Each status transition is enforced at the API layer; a patient cannot receive a report without explicit doctor, pharmacist, and administrator human review.

physician annotation, making this the primary AI safety mechanism.

E. System Architecture

Fig. 1 shows the six-layer architecture. The FastAPI backend exposes nine routers: `/auth`, `/patient`, `/doctor`, `/pharmacist`, `/admin`, `/otp`, `/chatbot`, `/reports`, and a second `/auth` variant. The Next.js 14 SPA provides four role-specific dashboards. The scan lifecycle progresses through five API-enforced statuses (Fig. 2).

F. Human-in-the-Loop Design

A core architectural constraint is that the AI classifier has no autonomous clinical authority. Every prediction is presented to the doctor as a decision support signal. The mandatory physician verification step (Stage 3), pharmacist medication review (Stage 4), and administrator approval (Stage 5) form a three-layer human review chain. The deployed PDF report template explicitly carries the AI notice: *"This report was generated with AI assistance and has been reviewed and approved by a qualified medical professional before release."* This disclaimer, embedded in the patient-facing document, is the operational embodiment of the human-in-the-loop principle.

TABLE III
PER-CLASS PERFORMANCE — HELD-OUT TEST SET (N = 32,682)

Class	Prec.	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%**

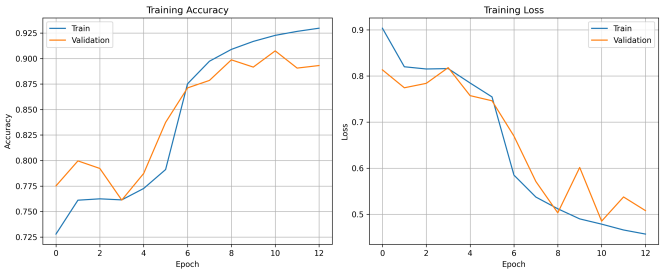

Fig. 3. Training (92.97%) and validation (89.31%) accuracy. EarlyStopping (patience = 3, restore_best_weights=True) triggered before epoch 15.

G. Security

JWT tokens (HS256) expire after 60 minutes. Passwords are bcrypt-hashed via `passlib`. Administrator accounts require OTP 2FA: a 6-digit token generated via `random.randint(100000, 999999)`, stored with a 10-minute expiry (`OTP_EXPIRE_MINUTES=10` from `.env`) and a single-use boolean flag preventing replay. RBAC is enforced in two ways: the `require_role()` dependency factory checks JWT claims; the `get_current_user()` dependency fetches the full user row from the database ensuring current role is always the persisted value.

IV. RESULTS

A. Classification Performance

Table III presents per-class metrics on the held-out test set (32,682 images). Sensitivity and specificity are reported because they are the clinically meaningful metrics for a diagnostic support tool: sensitivity captures ability to detect disease (minimizing false negatives), and specificity captures ability to avoid false alarms.

Training and validation accuracy curves are shown in Fig. 3; the confusion matrix in Fig. 4.

B. System Benchmarks

Average CPU inference latency: 0.73 s over 100 test scans. PDF generation: < 3 s. OTP email delivery: < 5 s. JWT validation: < 50 ms.

C. Functional Test Results

All ten end-to-end functional scenarios passed (Table IV).

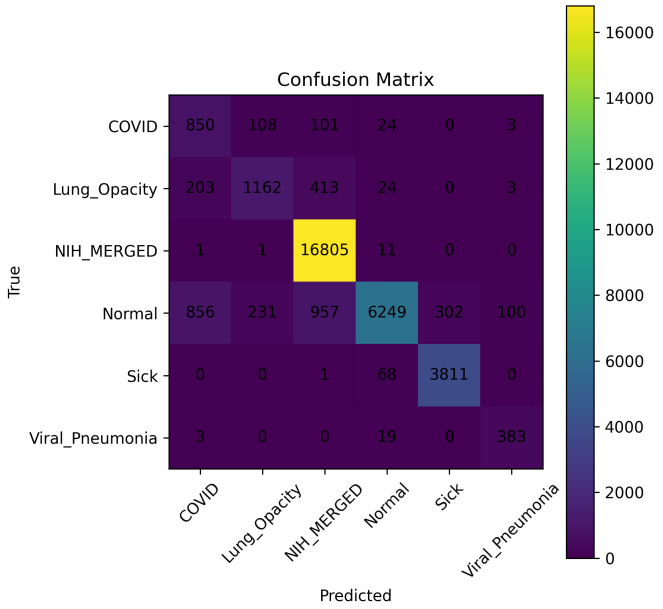

Fig. 4. Confusion matrix on the 32,682-image test set. NIH_MERGED exhibits the highest inter-class confusion, expected given that 14 distinct pathologies were aggregated into a single class.

TABLE IV
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 scan	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

V. DISCUSSION

The 89.51% test accuracy with macro sensitivity 0.891 and specificity 0.978 places CSSS competitively within the multi-class chest imaging literature. Chowdhury et al. [5] reported ≈90.5% accuracy on a four-class subset of the 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 benefits clinical workload: false positives in a screening support tool generate unnecessary follow-up and patient anxiety.

NIH_MERGED exhibits the lowest F1 (0.86) and highest inter-class confusion. This is expected: aggregating 14 distinct pathologies into a single training class discards pathology-level granularity and introduces heterogeneity that challenges the classification head. Future work should explore a hierarchical classifier or dedicated NIH 14-class output head.

The confidence threshold ($\tau = 0.75$) provides a practically significant safety layer. The returned "Uncertain" label cannot be dismissed: the doctor dashboard flags it visually and enforces mandatory annotation before the scan can progress to the pharmacist. The three-layer human review chain (doctor, pharmacist, administrator) and the explicit AI disclaimer printed on every delivered PDF collectively implement a conservative human-in-the-loop architecture aligned with current SaMD regulatory frameworks.

Limitations: (1) `database.py` hardcodes SQLite, which does not support concurrent multi-user deployment; PostgreSQL migration is needed. (2) DICOM is not supported; only JPEG/PNG are accepted. (3) CORS is configured with `allow_origins=["*"]` in the current build; this must be restricted to specific origins before hospital deployment. (4) Debug endpoints (`/admin/debug`, `/doctor/debug/...`) must be removed before production use. (5) NIH_MERGED aggregation discards pathology granularity. (6) No prospective clinical validation on real hospital data has been conducted; the system is not for clinical use without such validation. (7) The OTP email body incorrectly states 5-minute expiry, whereas `OTP_EXPIRE_MINUTES=10` is the operative setting; this should be corrected before deployment.

VI. CONCLUSION

CSSS integrates a MobileNetV2 classifier with a four-role human-in-the-loop clinical workflow, automated PDF report generation, and SMTP email delivery, achieving 89.51% test accuracy, sensitivity 0.891, and specificity 0.978 across 217,875 training images. Confidence thresholding, role-based access control, and OTP 2FA provide layered clinical safety and system security. The platform is fully open-source and deployable on commodity hardware, making AI-assisted diagnostics accessible to resource-constrained settings.

ETHICAL CONSIDERATIONS

This study used only publicly available, de-identified datasets (NIH ChestX-ray14, COVID-19 Radiography Database, CAD Cardiac MRI Dataset) from Kaggle. No human participants were recruited, no patient contact occurred, and no identifiable patient data was processed. IRB review was not required (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.

Data Availability

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 Contributions

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

Originality Declaration

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

Acknowledgment

The authors thank Saveetha Engineering College for computational resources and acknowledge the contributors to the NIH ChestX-ray14, COVID-19 Radiography Database, and CAD Cardiac MRI datasets.

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