

AI-Based Multi-Disease Fetal Anomaly Detection Using Prenatal Ultrasound Images

Prof. Parag B. Patil¹, Prof. Sangita Chaudhari², Pankaj Chaudhari³,
Vijay Deshmukh⁴, Sanket Patil⁵, Vishal Borse⁶

Project Guide, Dept. of Information Technology¹

Head of Department, Dept. of Information Technology²

Students, Dept. of Information Technology³⁻⁶

Sandip Institute of Technology and Research Centre (SITRC), Nashik, Maharashtra, India

Parag.patil@gmail.com, sangita.chaudhari@sitrc.org, pankajchaudhari.112023186@sitrc.org
vishalborse.112023024@sitrc.org, vijaydeshmukh.112023498@sitrc.org, sanketpatil.112023040@sitrc.org

Abstract: Prenatal ultrasound is the primary imaging modality for monitoring fetal growth and screening for congenital anomalies, yet its interpretation depends on operator experience, image quality and the availability of specialists. This paper presents the design and interim progress of a deep-learning framework for AI-assisted, multi-disease fetal anomaly detection from prenatal ultrasound images. The framework combines patient-wise data splitting, ultrasound-specific preprocessing, a baseline convolutional neural network (CNN) and transfer learning with ImageNet-pretrained ResNet50, DenseNet and EfficientNet models. It is planned to be extended to multi-disease classification (brain, heart, spine, kidney and other anomalies), Grad-CAM explanations and a Flask-based web interface. A review of six core studies identifies six research gaps: limited multi-disease coverage, small and imbalanced datasets, a weak link between plane recognition and anomaly diagnosis, limited external validation, weak early-stage detection, and limited interpretability. Progress recorded up to October 2026 (approximately 35% of the project) comprises dataset acquisition, cleaning, preprocessing, patient-wise train/validation/test splitting, a baseline CNN in TensorFlow/Keras, and initial fine-tuning experiments with three pretrained networks. Early observations are overfitting, bias toward majority classes and run-to-run variability. Multi-disease classification, full evaluation, Grad-CAM, the web application and testing remain pending, and no performance figures are claimed. The system is a research prototype and not a diagnostic device

Keywords: prenatal ultrasound, fetal anomaly detection, deep learning, convolutional neural network, transfer learning, multi-disease classification, Grad-CAM, medical image analysis

I. INTRODUCTION

Prenatal care depends on timely identification of structural malformations and growth abnormalities. Ultrasound is the primary imaging modality in obstetrics because it is non-ionising, real-time, safe and comparatively inexpensive [1], [3]. Although most examinations show normal development, recognising subtle congenital abnormalities demands careful visual assessment by trained clinicians. Manual interpretation therefore faces four recurring challenges:

1. **Operator dependency and subjectivity.** Visual assessment relies on the experience, fatigue and judgement of the examiner, which can produce inconsistent diagnoses [3].
2. **Shortage of specialists.** In rural or low-resource settings, access to fetal-medicine expertise is limited, which can delay diagnosis.
3. **Image-quality limitations.** Speckle noise, acoustic shadowing, low contrast and poorly defined boundaries make analysis difficult [1].



4. Single-disease scope of existing tools. Most computer-aided systems target one organ or one task, whereas malformations may involve several organ systems [1], [3].

Deep learning, and convolutional neural networks (CNNs) in particular, can learn hierarchical image features directly from data and has produced strong results in fetal ultrasound analysis [1], [3], [21]. However, much of the reported performance comes from small, single-centre or heavily imbalanced datasets, and results are often quoted at image level rather than patient level, which can overstate generalisation [1], [3], [5]. These concerns shape the design choices in this work, notably patient-wise data splitting, class-imbalance handling and explicit, honest reporting of what has and has not been validated.

This paper describes a framework for multi-disease fetal anomaly detection from prenatal ultrasound images and reports the project's interim progress. The project is in progress: dataset preparation, preprocessing, a baseline CNN and initial transfer-learning experiments have been carried out, whereas multi-disease classification, full evaluation, Grad-CAM, the web application and testing are still to be completed (Section XIII). Accordingly, the paper makes no accuracy claims, and quantitative results will be reported once final experiments are complete.

The remainder of the paper is organised as follows. Section II reviews related work, Sections III and IV state the problem, research gap and objectives, Sections V to IX describe the methodology, architecture, workflows, implementation and requirements, Sections X to XII discuss benefits, the evaluation plan and safety considerations, Section XIII reports progress, and Sections XIV and XV give limitations, future scope and conclusions.

II. LITERATURE REVIEW

Research papers published between 2019 and 2026 on fetal ultrasound analysis were searched in online research databases. Six core studies [1]–[6] were reviewed in detail, recording the objective, method, dataset, key findings and limitations of each (Table 1). Further papers [7]–[14] on EfficientNet, ResNet, U-Net and organ-specific fetal studies were reviewed during model selection.

II.1 Surveys and Systematic Reviews

Shahrivar et al. [1] reviewed machine-learning and deep-learning methods for fetal anomaly detection in ultrasound across the heart, head and neck, lung and chromosomal anomalies, summarising about 40 studies. They note that many studies use small, single-centre datasets, that fetal tissue appearance changes with gestational age, and that real-time detection is largely unexplored. Ramirez Zegarra and Ghi [3] review deep learning in fetal imaging, covering classification, localisation, detection and segmentation of standard planes, the central nervous system, heart and placenta. They highlight the need for large labelled datasets, the black-box nature of models and the shortage of prospective validation. D'Alberty et al. [5] pooled studies on AI detection of congenital heart defects in 2D ultrasound (16–40 weeks) and report substantial heterogeneity between studies, with little evidence on integration into clinical workflow.

II.2 AI-Integrated Ultrasound in Clinical Settings

Nath and Guha [2] compare conventional ultrasound with AI-integrated ultrasound using an ensemble of a modified U-Net and ResNet-50 with real-time feedback to the sonographer in a multi-centre prospective setting. The study reports higher detection and fewer false positives with AI assistance, but includes only about 90–100 anomaly cases, uses the same images for both readings and was conducted in tertiary centres with high-end equipment. Validation in low-resource settings is therefore still needed.

II.3 Architecture-Specific and Organ-Specific Studies

Islam et al. [4] propose Fetal-Net, which combines multi-scale CNN blocks (3×3, 5×5 and 7×7 kernels) with six transformer layers and attention maps, evaluated on the BCNatal dataset of 12,400 images from 1,792 patients across nine plane classes. The authors note brain-heavy class imbalance, absence of first-trimester and rare-condition data, and unclear anomaly labels. Qi et al. [6] present a multi-centre deep-learning model for four fetal central-nervous-system



anomalies with heat-maps showing the regions behind each prediction; it is a preprint covering only one organ system. Organ-specific studies on renal anomalies [13] and fetal brain abnormalities [14] were also consulted; their quantitative findings are not tabulated here.

II.4 Foundation Architectures

The deep CNN of Krizhevsky et al. [9] established large-scale image classification with CNNs. ResNet [8] introduced residual skip connections that ease the training of deep networks; DenseNet [16] connects each layer to all subsequent layers within a block; and EfficientNet [7] scales depth, width and resolution jointly. These networks are commonly pretrained on ImageNet [23] and fine-tuned for medical tasks, which is the basis of the transfer-learning candidates used in this project. U-Net [10] is a segmentation architecture and is cited as background only, because segmentation is not part of the present scope. Vision Transformers [11] and contrastive image-text models [12] were reviewed but are not adopted in the current experiments.

Table 1 Comparison of reviewed studies (values as reported by the original authors)

Ref.	Method / task	Data	Reported findings	Limitations / relevance
[1]	Review of ML/DL for fetal anomaly detection (CNN, U-Net, GAN, RNN/LSTM)	Literature review; about 40 studies	Heart ensemble AUC 0.99 on 107,823 images; GAN-based model 85% accuracy; brain U-Net/ResNet about 91% Dice	Small single-centre data; variable image quality; real-time detection unexplored
[2]	Conventional vs AI-integrated ultrasound (modified U-Net + ResNet-50 ensemble)	1,250 women, 3 tertiary centres; 28,500 training images	Detection 72.4% to 91.7%; false positives 4.2% to 1.6%	About 90–100 anomaly cases; same images for both reads; no low-resource validation
[3]	Review of deep learning in fetal imaging	Literature review; about 130 references	Best of 19 plane-classification models matched a trained sonographer and ran 25× faster	Needs large labelled sets; black-box; mostly retrospective; calls for prospective validation
[4]	Fetal-Net: multi-scale CNN + 6 transformer layers; plane classification and anomaly flag	BCNatal: 12,400 images, 1,792 patients, 9 plane classes	Anomaly detection accuracy 97.5%, precision 96.5%, recall about 97.8%, F1 96.6%	Brain-heavy imbalance; unclear anomaly labels; no first-trimester or rare data
[5]	Systematic review and meta-analysis of AI for fetal heart defects (2D ultrasound)	30,121 fetuses across studies	Pooled sensitivity 0.89, specificity 0.91	High heterogeneity (I^2 about 78%); excludes 3D/4D and < 16 weeks
[6]	Deep learning for four fetal CNS anomalies with heatmaps	Multi-centre dataset (size not verified)	Patient-level accuracy 94.5%, AUROC 99.3%	Preprint; CNS only; no other organ systems
[7], [8], [16]	EfficientNet, ResNet, DenseNet backbones	ImageNet pretraining	General-purpose architectures; basis of the transfer-learning candidates	Not ultrasound-specific; need fine-tuning



Ref.	Method / task	Data	Reported findings	Limitations / relevance
[10], [11]	U-Net; Transformer	Vision Biomedical segmentation; ImageNet-scale	Background architectures	Not used in the current scope

II.5 Synthesis

The reported results in Table 1 come from different datasets, tasks and metrics, so they are not directly comparable with each other or with this project. Very high accuracy values must be interpreted with attention to dataset size and whether splits were made by patient or by image, since image-level splits can leak near-identical frames from the same patient into the test set. This observation directly motivates the patient-wise splitting adopted here.

III. PROBLEM STATEMENT AND RESEARCH GAP

Prenatal ultrasound is safe, non-invasive and widely available, but accurate interpretation remains difficult. Diagnostic quality depends on examiner expertise, and manual examination is time-consuming and subjective, which may delay or miss the detection of fetal anomalies. Most computer-aided systems detect a single anomaly or concentrate on one organ and cannot identify several abnormalities within one framework. Achieving high accuracy while remaining robust across ultrasound datasets is still an open challenge. There is therefore a need for an AI-based system that analyses prenatal ultrasound images, classifies multiple fetal conditions, supports clinicians in decision-making and, through transparent evaluation, avoids over-stating its reliability.

The review of [1]–[6] identifies six gaps:

- Limited multi-disease and multi-organ coverage.** Most models address one task and one organ. Heart studies dominate, and CNS models cover few anomaly types (four in [6]). Few systems address heart, brain, kidney and skeleton together [1], [3].
- Small, imbalanced, single-centre datasets.** Many studies use a few hundred to a few thousand images from one hospital, raising the risk of overfitting [1]. Even larger studies contain few abnormal cases (about 100 in [2]) or are dominated by one class (brain views in [4]). Publicly available, well-labelled anomaly datasets are scarce.
- Weak link between plane recognition and anomaly diagnosis.** Plane classification and anomaly detection are usually treated separately; several models output only normal or abnormal without the malformation type, and abnormal labelling is often unclear [1], [4].
- Limited external validation and inconsistent evaluation.** Data are mostly retrospective and rarely tested on other hospitals or machines [3]; studies report different metrics on different datasets, producing heterogeneity [5].
- Weak performance on early-stage and subtle anomalies.** Detection is lowest in the first trimester and for subtle anomalies [2], yet many models are trained only on second- and third-trimester images [4], [5]; anatomy also changes with gestational age.
- Limited interpretability.** Deep models act as black boxes, reducing clinician trust [3]. Attention maps and heat-maps appear in recent work [4], [6] but are rarely checked against expert annotations.

IV. RESEARCH OBJECTIVES AND CONTRIBUTIONS

The primary objective is to design and develop an AI-based system for automated detection of multiple fetal anomalies from prenatal ultrasound images, to assist healthcare professionals in identifying congenital abnormalities at an early stage. The specific objectives are:

- To collect and preprocess a fetal ultrasound dataset (resizing, histogram equalisation, normalisation, augmentation) and to handle class imbalance with augmentation and weighted loss.



2. To develop a CNN-based classification model using transfer learning (for example ResNet50, DenseNet or EfficientNet pretrained on ImageNet) to identify fetal anatomical planes and classify normal versus abnormal cases, and to extend it to multi-disease classification.
3. To compare the proposed model with baseline models (a simple CNN and at least two pretrained networks) in order to select the best-performing architecture.
4. To evaluate the model with standard metrics (accuracy, precision, recall, F1-score, specificity, AUC and confusion matrix) on a held-out test set split by patient.
5. To improve interpretability with Grad-CAM heat-maps that show which image regions drive each prediction.

The contributions of this paper, stated with their current status, are: (i) a layered framework that separates offline model development from online inference; (ii) a gap analysis of six core studies leading to the research gaps above; (iii) a leakage-aware evaluation protocol using patient-wise splitting, training-only augmentation and a locked test set; (iv) a documented, status-labelled account of the progress achieved so far, including the problems encountered; and (v) a plan for Grad-CAM explanation and a lightweight web interface, which are not yet implemented.

The project is aligned with three United Nations Sustainable Development Goals. SDG 3 (Good Health and Well-being): supporting earlier, more consistent prenatal screening. SDG 9 (Industry, Innovation and Infrastructure): applying AI and deep learning to a healthcare solution. SDG 10 (Reduced Inequalities): aiming at accessible screening support for underserved and rural settings. These alignments describe intent; they are not outcomes that have been demonstrated.

V. PROPOSED FRAMEWORK AND METHODOLOGY

The framework has two phases: an offline development phase (data audit, splitting, preprocessing, augmentation, model training, comparison and evaluation) and an online inference phase (image upload, validation, preprocessing, prediction, confidence handling, optional explanation and result display). Section VI presents the architecture and Section VII the workflows.

V.1 Dataset and Labels

Publicly available annotated fetal ultrasound datasets were identified and downloaded (Week 7). Exploration of the data showed class imbalance: some anatomical planes and anomaly classes contain few images (Week 8). Duplicate and corrupted images were removed, and train, validation and test sets were created patient-wise so that all images of one patient fall in a single subset (Week 8). Current experiments address fetal plane classification and normal-versus-abnormal classification (Week 9).

V.2 Preprocessing and Augmentation

The logbook confirms implementation of resizing, histogram equalisation, noise removal, intensity normalisation and augmentation (Week 8). The parameter values listed below follow the earlier project draft and should be verified against the code. Images are resized to $224 \times 224 \times 3$ to match the network input; speckle noise is attenuated with median or Gaussian filtering; local contrast is improved with contrast-limited adaptive histogram equalisation (CLAHE) [20]; and intensities are scaled to $[0, 1]$ by min-max normalisation:

$$I_{norm}(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \quad (1)$$

where $I(x, y)$ is the pixel intensity and I_{min} , I_{max} are the minimum and maximum intensities. To avoid information leakage, such statistics should be computed from the training set only. Online augmentation (rotation of $\pm 20^\circ$, horizontal flip, zoom of $\pm 15\%$ and translation) is applied to training images only [22]. Horizontal flipping should be reviewed for organ classes in which left-right orientation is diagnostically meaningful, such as cardiac views.



V.3 Convolutional Feature Extraction

The baseline CNN uses standard building blocks. A 2D convolution slides a learnable $k \times k$ kernel K over the input I :

$$F(i, j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I(i+m, j+n) K(m, n) + b \quad (2)$$

followed by the rectified linear unit (ReLU) non-linearity and batch normalisation [15]:

$$f(z) = \max(0, z) \quad (3)$$

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \quad y = \gamma \hat{x} + \beta \quad (4)$$

where μ_B and σ_B^2 are the mini-batch mean and variance, ε is a small constant, and γ, β are learnable parameters. Max-pooling over a local region Ω reduces spatial size while retaining dominant features:

$$P(i, j) = \max_{(m, n) \in \Omega} F(i+m, j+n) \quad (5)$$

The flattened features pass through a dense layer with dropout (rate 0.5) [18] to reduce co-adaptation. The layer configuration is shown in Fig. 4 (Section VIII.4).

V.4 Transfer Learning

Pretrained ResNet50 [8], DenseNet [16] and EfficientNet [7] networks, initialised with ImageNet weights [23], are fine-tuned on the ultrasound data. ResNet50 fine-tuning started in Week 11, and DenseNet and EfficientNet runs started in Week 12. Because the models overfit quickly on the small training set, stronger augmentation and dropout were added in Week 13. The specific DenseNet and EfficientNet variants and the fine-tuning schedule (frozen layers, learning rates) are not recorded in the logbook and are to be stated by the authors.

V.5 Output Layer, Loss and Class Imbalance

For C categories the output layer applies the softmax function:

$$p_c = \frac{e^{z_c}}{\sum_{k=1}^C e^{z_k}} \quad (6)$$

where z_c is the logit of class c and p_c is the predicted probability, which serves as the confidence score. Because the dataset is imbalanced and the baseline favoured majority classes (Week 9), a class-weighted categorical cross-entropy is used (class weighting was applied in Week 10):

$$\mathcal{L}_{WCE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C w_c y_{i,c} \log(p_{i,c}) \quad (7)$$

where N is the number of samples, $y_{i,c}$ is the one-hot label and $p_{i,c}$ the predicted probability. A common inverse-frequency choice for the weights is

$$w_c = \frac{N}{C \cdot N_c} \quad (8)$$



with N_c the number of training samples in class c . The weighting scheme actually used must be confirmed by the authors. The Adam optimiser [17] is specified in the project draft and should likewise be confirmed.

V.6 Multi-Disease Stage (Planned)

The planned multi-disease stage targets brain, heart, spine, kidney and other anomalies, with a list of available datasets per class prepared in Week 15. Few labelled abnormal images exist for some organ categories, which may force rare classes to be merged or dropped. Two design decisions remain open and are flagged for confirmation: (a) whether the task is single-label multi-class (softmax, one label per image) or multi-label (sigmoid outputs), and (b) whether a single shared head or organ-specific heads are used. The softmax formulation in (6) assumes the single-label case. Until these are settled, multi-disease classification is described as proposed and not as implemented.

VI. SYSTEM BLOCK DIAGRAM AND ARCHITECTURE

Fig. 1 shows the architecture supported by the project documents. Training and inference are drawn as separate pipelines: the offline pipeline produces a saved model and its exact preprocessing configuration, and the online pipeline loads them to serve predictions through a web interface. Colour and line style indicate status according to the progress logbook: solid green for completed, amber for preliminary or in progress, and dashed grey for planned.

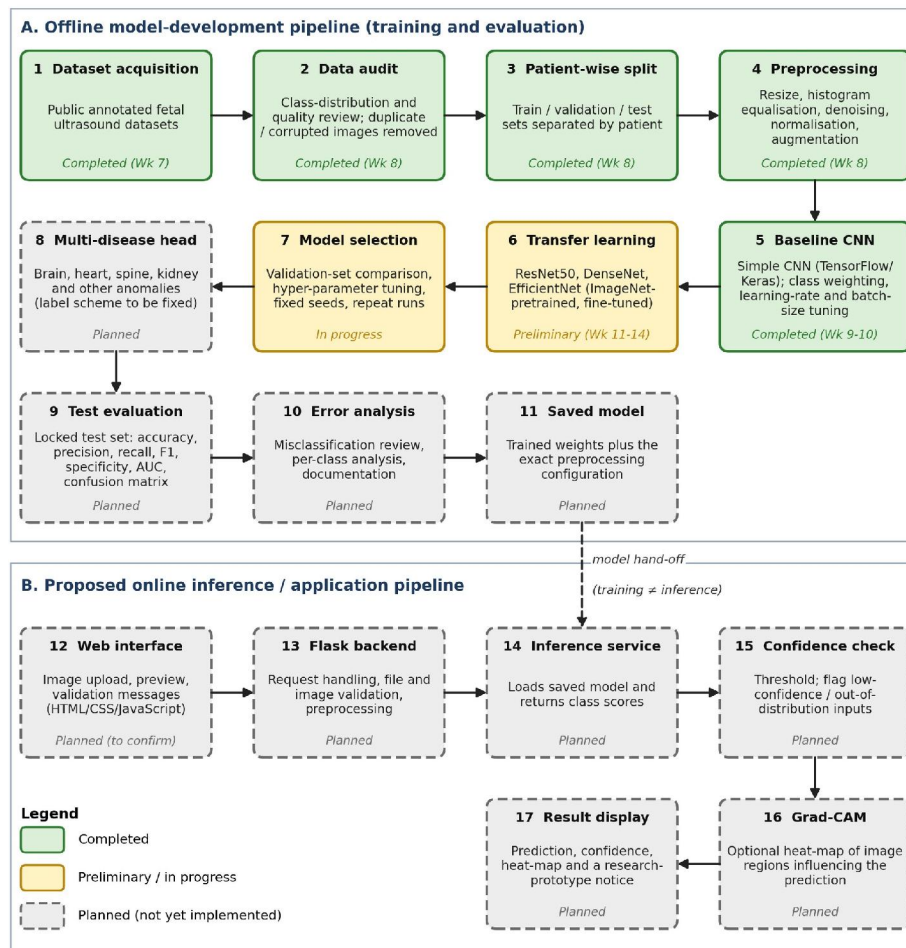


Fig. 1. Proposed architecture of the AI-based multi-disease fetal anomaly detection system, with implementation status.



Table 2 lists the input, operation and output of each block.

Table 2 Block-wise input, operation and output

Block	Input	Operation	Output
1–2 Dataset acquisition and audit	Public annotated fetal ultrasound datasets	Download; review class distribution and image quality; remove duplicate and corrupted images	Cleaned images, labels, class statistics
3 Patient-wise split	Cleaned images with patient identity	Assign all images of a patient to one subset (train, validation or test)	Train / validation / test lists
4 Preprocessing and augmentation	Split images	Resize, histogram equalisation, noise removal, normalisation; augmentation on training images	Model-ready tensors
5–6 Baseline CNN and transfer learning	Training and validation tensors	Train baseline CNN; fine-tune ResNet50, DenseNet, EfficientNet; class weighting	Trained candidate models; training logs
7–8 Model selection and multi-disease head	Validation results	Compare candidates under identical conditions; tune; design multi-disease output	Selected architecture (pending)
9–10 Test evaluation and error analysis	Locked test set; selected model	Compute metrics and confusion matrix; review errors	Performance report (pending)
11 Saved model	Final trained weights	Export with preprocessing configuration and version record	Deployable model artefact
12–13 Web interface and Flask backend	User-supplied image	Upload, preview, validate file and image, preprocess	Validated tensor or error message
14–15 Inference and confidence handling	Validated tensor; saved model	Compute class probabilities; apply threshold; flag low-confidence or out-of-distribution input	Scores, confidence, flag
16–17 Grad-CAM and result display	Scores; convolutional activations	Generate optional heat-map; show result with research-prototype notice	Result page

VII. SYSTEM FLOW DIAGRAM AND WORKFLOW

VII.1 Model-Training Workflow

The training workflow (Fig. 2) starts with the dataset audit and a patient-wise split, followed by training-only augmentation and baseline training. Candidate models are compared and tuned on validation data only, and the loop repeats until validation criteria are met. Only then is the locked test set evaluated, once, followed by error analysis and documentation. Steps 1 to 4 are completed; comparison and tuning are in progress; test evaluation and error analysis are planned.



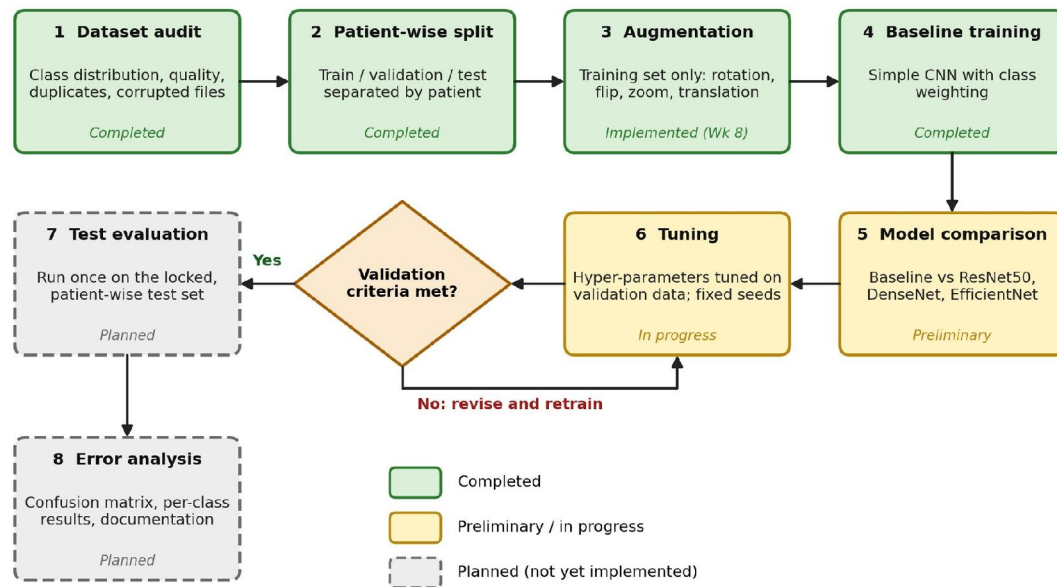


Fig. 2. Model-training and evaluation workflow with status of each step.

VII.2 Inference Workflow

Fig. 3 shows the end-to-end handling of one request. All components of this flow are planned; none has been implemented yet according to the logbook.

1. The user uploads an image (or an image from the dataset is supplied for testing).
2. The file and image are validated for type, size, readability and dimensions. If invalid, an error message is displayed and the request stops.
3. A valid image is preprocessed and normalised with the same pipeline used in training.
4. The saved model is loaded and produces class probabilities.
5. If the confidence is below a threshold, the image is flagged as low-confidence or possibly out-of-distribution and expert review is recommended. The threshold must be chosen on validation data.
6. An optional Grad-CAM heat-map is generated and the result is displayed together with a limitations notice.



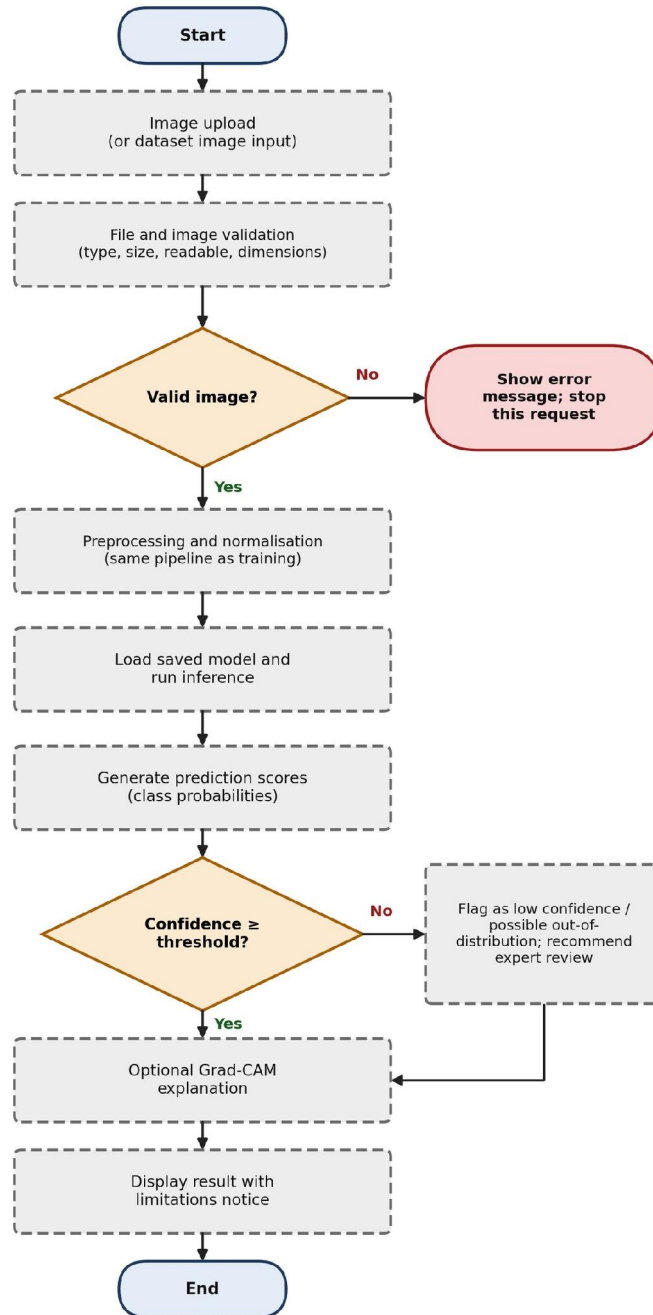


Fig. 3. End-to-end inference flow with validation and confidence decision nodes. Dashed boxes are planned components.



VIII. PROPOSED IMPLEMENTATION FRAMEWORK

VIII.1 Frontend

The interface is planned to provide image upload, an image preview, validation messages, display of the predicted class with its confidence, and display of a Grad-CAM overlay. HTML, CSS and JavaScript are proposed. React.js appears in a team member's skills list in the synopsis, but no document confirms its use, so it is not described as part of the implementation.

VIII.2 Backend

A Python/Flask service is planned (Weeks 6 and 16). It would receive the upload, check extension, content type, size and decodability, apply the shared preprocessing function, run inference with a model loaded once at start-up, and return class probabilities, a confidence indicator and an optional heat-map. Error messages should be generic and must not expose file paths or internal details.

VIII.3 Image-Processing Module

OpenCV and NumPy are the intended libraries for resizing, histogram equalisation (or CLAHE), median or Gaussian filtering and normalisation [20], [25]. Training and inference must call one shared preprocessing implementation to avoid train-serve mismatch. Which individual filters were used in the completed preprocessing pipeline is to be confirmed by the authors.

VIII.4 Deep-Learning Module

The baseline CNN was implemented in TensorFlow/Keras (Week 9) [24]. Its layer configuration in the project draft is shown in Fig. 4: three convolutional blocks (32, 64 and 128 filters of size 3×3 , each with ReLU, batch normalisation and 2×2 max-pooling), a flattened 100,352-dimensional vector with dropout 0.5, a dense layer of 256 units and a softmax output. These values must be checked against the trained baseline.

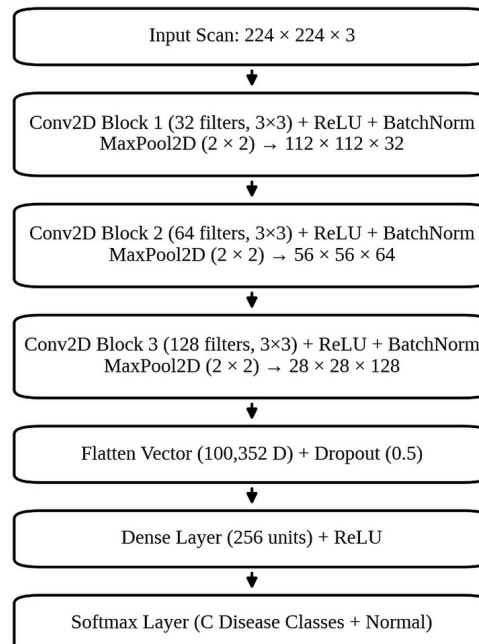


Fig. 4. Baseline CNN architecture as specified in the project draft (to be verified against the trained model).



Models actually trained so far are the baseline CNN, ResNet50, DenseNet and EfficientNet. Vision Transformers and U-Net were reviewed but not trained. TensorFlow/Keras is the framework supported by the logbook and synopsis. PyTorch appears only in a skills list and is not claimed as used.

VIII.5 Explainability

Grad-CAM [19] is planned as an interpretability aid; so far only the method has been reviewed (Week 16). It produces a heat-map from gradients flowing into the last convolutional layer. A heat-map shows where the model attended, not whether the prediction is medically correct, and it can highlight irrelevant regions for a wrong or right prediction alike. It should be checked against expert annotation before being relied upon, and it must never be presented as a diagnosis.

VIII.6 Training and Reproducibility

Run-to-run differences were observed in Week 13, so reproducibility measures are required: fixed random seeds for Python, NumPy and TensorFlow; repeated runs reported as mean and standard deviation; saving of the best checkpoint by validation metric; a recorded environment (Python, TensorFlow/CUDA and library versions in a requirements file); versioned code in Git/GitHub; and the existing experiment log of learning rate, batch size and results (Week 12), extended to record seed, split identifier and checkpoint. Patient-wise splitting, class weighting and training-only augmentation are the leakage and imbalance controls.

IX. FRONTEND, BACKEND, SOFTWARE AND HARDWARE REQUIREMENTS

Table 3 lists technologies with their purpose and the evidence supporting their status.

Table 3 Tools, technologies and status

Component	Technology	Purpose	Status / evidence
Language and environment	Python 3.x; Jupyter / VS Code	Development of pipeline and models	Implemented: GPU-enabled Python environment set up (Wk 7); editor not recorded
Deep-learning framework	TensorFlow / Keras [24]	Baseline CNN and transfer learning	Used: baseline built in Wk 9; pretrained models in Wk 11–12
Pretrained models	ResNet50, DenseNet, EfficientNet (ImageNet)	Transfer-learning candidates	Used in preliminary experiments (Wk 11–14); variants not recorded
Image processing	OpenCV [25], NumPy	Resize, histogram equalisation, denoising, normalisation	Used: preprocessing implemented (Wk 8); library calls to be confirmed
Evaluation	scikit-learn [26]	Metrics, confusion matrix, ROC	Proposed: full evaluation pending (Wk 17)
Visualisation	Matplotlib, Seaborn	Curves, confusion matrix, charts	Proposed
Data handling	Pandas, NumPy	Label tables, split lists	Listed in synopsis; use not recorded in logbook
Explainability	Grad-CAM [19]	Visual explanation	Planned: method reviewed only (Wk 16)
Backend	Flask (Python)	Upload, validation, inference API	Planned (Wk 6, 16)



Component	Technology	Purpose	Status / evidence
Frontend	HTML, CSS, JavaScript	Upload, preview, result display	Proposed; React.js not confirmed
User login	E-mail one-time-password login page	Restrict access to the web page	Awaiting confirmation: reported by the team, not in logbook
Database	MySQL / SQLite	Optional storage of reports or users	Optional; not required by the core scope
Version control	Git, GitHub	Code and experiment tracking	Implemented (Wk 7)
Experiment log	Tabular log of learning rate, batch size, results	Reproducibility	Implemented (Wk 12)
Hardware	NVIDIA CUDA GPU; recommended 16 GB RAM, 512 GB SSD	Model training	GPU used but model not recorded; availability limited

The synopsis recommends an Intel Core i5 (10th generation or above) or AMD Ryzen 5 processor, at least 8 GB RAM (16 GB recommended), a 256 GB SSD (512 GB recommended) and an NVIDIA CUDA-capable GPU, running Windows 10/11 or Ubuntu. These are recommended resources; the actual GPU model and memory used in the experiments are not recorded and should be added. No cloud service has been adopted and none is assumed.

X. ADVANTAGES AND EXPECTED BENEFITS

The framework is expected to offer the following potential benefits, each subject to the stated conditions:

Assistance in reviewing images. Automated analysis may reduce routine review effort, provided that performance is validated on independent data.

Consistent computational workflow. The same preprocessing and model are applied to every image of the defined dataset, reducing variability relative to manual reading; this does not guarantee correctness.

Support for screening and prioritisation. Confidence scores could help prioritise images for expert review, provided that thresholds are calibrated and validated.

Systematic model comparison. A baseline CNN is compared with three pretrained networks under identical splits, supporting an evidence-based model choice.

Multi-disease analysis. A single model may cover several organ systems, depending on the final label scheme and on how much labelled data exists for each class.

Visual explanation. Grad-CAM may help users inspect model attention, with the limits noted in Section VIII.5.

Accessibility. A lightweight web interface may make the tool easy to use in settings without specialist software, subject to security safeguards in Section XII.

Reproducible and transparent evaluation. Patient-wise splits, fixed seeds and a locked test set support credible reporting.

These benefits are potential rather than demonstrated. The system does not replace radiologists or obstetricians, cannot guarantee early diagnosis and has not been clinically validated. Its limitations include dependence on the dataset, class imbalance, limited GPU resources and the absence of external validation (Section XIV).

XI. EVALUATION PLAN AND PERFORMANCE METRICS

XI.1 Dataset and Split Protocol

Dataset source, labels, class distribution and split proportions are to be reported once confirmed (Section V.1). Training, validation and test sets are separated at patient level to prevent leakage between images of the same patient. If



patient identifiers are unavailable for any dataset, this must be stated as a limitation and a leakage audit performed, for example by checking near-duplicate images across subsets. Augmentation is applied to the training set only. Validation data are used for model selection and hyper-parameter tuning, and the test set remains locked until the final evaluation.

XI.2 Metrics

For each class the confusion-matrix counts TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives respectively, computed in a one-versus-rest manner for multi-class problems. The metrics are:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \quad (11)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (12)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

Accuracy can be misleading under class imbalance, so precision, recall (sensitivity), specificity and F1-score are reported for every class together with the confusion matrix. For multi-class results an averaging strategy must be stated explicitly. Macro averaging weights all classes equally and exposes poor performance on rare classes, whereas weighted averaging weights classes by support:

$$F_1^{\text{macro}} = \frac{1}{C} \sum_{c=1}^C F_{1,c}, \quad F_1^{\text{weighted}} = \sum_{c=1}^C \frac{N_c}{N} F_{1,c} \quad (14)$$

ROC-AUC is computed per class (one-versus-rest) and macro-averaged for the multi-class setting. Inference time per image, model size and memory use are measured on a stated hardware configuration. Calibration (for example reliability diagrams and expected calibration error) is recommended where feasible, because the softmax probability is used as a confidence score. Segmentation metrics such as Dice or IoU are not used, since segmentation is not part of this project.

XI.3 Experimental Comparison

The baseline CNN and the pretrained candidates are compared under identical conditions: the same patient-wise splits, preprocessing, augmentation policy, input size and evaluation code. Each model is trained with several fixed random seeds, and the mean and standard deviation of validation metrics are reported. Because differences between some models were small (Week 14), uncertainty is quantified with patient-level bootstrap confidence intervals before a model is declared best. The final model is chosen on validation data and evaluated once on the test set.

XI.4 Planned Charts and Tables

The following figures will be produced from logged experiments once final runs are complete: class-distribution chart, training and validation loss curves, training and validation accuracy curves, confusion matrix, per-class precision/recall/F1 chart, ROC curves, model-comparison table and inference-time comparison. None is shown in this



manuscript because the numerical values are not recorded in the supplied logbook, and illustrative values would be misleading. Table 4 is the template for per-class reporting.

Table 4 Template for per-class test results (no values yet)

Class	Support	Precision	Recall	Specificity	F1	AUC (one-vs-rest)
Brain abnormality	[]	[]	[]	[]	[]	[]
Heart defect	[]	[]	[]	[]	[]	[]
Spine abnormality	[]	[]	[]	[]	[]	[]
Kidney abnormality	[]	[]	[]	[]	[]	[]
Other anomalies	[]	[]	[]	[]	[]	[]
Normal	[]	[]	[]	[]	[]	[]
Macro / weighted average	—	[]	[]	[]	[]	[]

XII. SECURITY, PRIVACY, ETHICS AND CLINICAL SAFETY

Status statement. The system is a research prototype and not a standalone diagnostic device. No security testing, ethics approval, regulatory authorisation or compliance certification has been performed or is claimed. Of the safeguards below, only duplicate/corrupted-image removal and patient-wise splitting are recorded as implemented (Week 8); all others are recommendations for the proposed system.

XII.1 Data Privacy, De-identification and Licensing

Only the minimum data needed should be collected. Patient names, dates, identifiers and embedded metadata (for example DICOM headers or text burned into the image) should be removed or masked before use. The licence, permitted use and any access agreements of every dataset must be confirmed and cited; dataset access formalities already delayed setup (Week 7).

XII.2 Secure Upload, Storage and Access

The web service should validate file type, size limit and image integrity, reject malformed files, process uploads in memory or in a temporary location and delete them after the response unless retention is justified. Access control should restrict the interface to authorised users. Model files, logs and any stored images must be protected, and application logs and error messages must not reveal patient information, file paths or system internals. A data-retention and deletion policy should be defined before any real clinical image is processed.

XII.3 Bias, Imbalance and Out-of-Distribution Inputs

Class imbalance already affects the baseline (Week 9), so per-class and macro metrics are required. If metadata such as gestational age, machine type or site is available and ethically usable, performance should be compared across these groups. Images from other devices, planes or non-fetal content may fall outside the training distribution; confidence thresholds and rejection of invalid inputs reduce, but do not remove, this risk.

XII.4 Human Oversight and Interpretation Limits

Qualified clinicians must interpret all outputs. Confidence scores from softmax are not guaranteed to be calibrated, and Grad-CAM heat-maps show attention rather than medical correctness (Section VIII.5). The interface should display a clear research-prototype notice. Deployment in care would additionally require prospective, multi-centre validation and the applicable regulatory authorisation.



XII.5 Research Ethics

If clinical images beyond public, de-identified datasets are used, institutional ethics committee approval and, where applicable, patient consent are required. The applicable approvals for this project have not been documented and should be confirmed.

XIII. PRELIMINARY PROGRESS AND DISCUSSION

Table 5 summarises progress from the weekly logbook (1 July to 25 October 2026). The logbook records overall progress at approximately 35% in Week 17, with the schedule rated “on schedule”, apart from weeks 10 to 12, which were “slightly behind” because of training time and GPU availability.

Table 5 Project activities and status according to the progress logbook

Activity	Period	Status	Evidence in logbook
Topic selection, group formation	Wk 1–3	Completed	Topic finalised after discussion with guide (Wk 3)
Literature review of six core papers	Wk 3–5	Completed	Literature comparison table; six research gaps identified
Problem statement, objectives, synopsis	Wk 5–7	Completed	Synopsis prepared (Wk 6); feedback incorporated, title finalised (Wk 7)
Dataset identification and download	Wk 7	Completed	Public annotated datasets downloaded; names not recorded
GPU Python environment; Git/GitHub	Wk 7	Completed	GPU availability limited
Data exploration and cleaning	Wk 8	Completed	Class distribution reviewed; duplicates and corrupted images removed
Preprocessing, augmentation, patient-wise splits	Wk 8	Completed	Resizing, histogram equalisation, noise removal, normalisation, augmentation
Baseline CNN and first training	Wk 9–10	Completed	Baseline results recorded (values not in logbook); class weighting, learning-rate and batch-size tuning
ResNet50 transfer learning	Wk 11	Preliminary	First fine-tuning; early validation comparison with baseline
DenseNet and EfficientNet	Wk 12–13	Preliminary	Initial runs; quick overfitting; stronger augmentation and dropout added
Validation comparison and shortlisting	Wk 14–15	Preliminary	Best candidate shortlisted (not named); differences small
Multi-disease classification planning	Wk 15	Planning only	Class list and dataset inventory prepared
Grad-CAM and Flask review	Wk 16	Review only	Methods studied; not implemented
Multi-disease classification model	—	Planned	Pending per Wk 17
Full evaluation; final model selection	—	Planned	Pending per Wk 17
Grad-CAM; Flask web application; testing	—	Planned	Pending per Wk 17
Research paper and final report	—	Planned	Pending per Wk 17



Activity	Period	Status	Evidence in logbook
Dataset names, split ratios, metric values, shortlisted model, web-page and login status	—	Needs confirmation	Not recorded in the supplied files

XIII.1 Documented Observations

The logbook documents the following observations. The baseline CNN overfitted and favoured majority classes because of class imbalance (Week 9). Rare classes remained harder to classify after class weighting (Week 10). Pretrained models overfitted quickly on the small training set (Week 12), and results differed between runs, indicating a need for fixed seeds and repeated experiments (Week 13). Differences between some models were small, so more experiments are needed to confirm the best choice (Week 14). Few labelled abnormal images were found for some organ categories (Week 15). The logbook records that baseline and early ResNet50 validation results were compared, but does not state numerical values; therefore none are reported here.

Table 6 Challenges recorded in the logbook and the methodological response

Challenge	Week	Response	Status
Class imbalance; few images in some planes and classes	8, 9, 15	Class weighting and augmentation applied; per-class and macro metrics, oversampling or merging of rare classes proposed	Partly applied; effect not quantified
Overfitting on small training set	9, 12	Stronger augmentation and dropout applied (Wk 13); early stopping and checkpoint selection proposed	Partly applied
Run-to-run variation	13	Fixed seeds and repeated runs with mean and SD reported	Proposed
Small differences between models	14	Identical splits, repeated runs, bootstrap confidence intervals	Proposed
Long training time; limited GPU	7, 10–12, 17	Scheduling, early stopping, checkpoint reuse, smaller or mixed-precision training	Ongoing
Scarce labelled abnormal data per organ	15	Dataset inventory per class; merge rare classes; transfer learning	Planning
Interpreting very high reported accuracy	5	Patient-wise splits; compare only like with like	Applied in design

XIII.2 Results

Quantitative evaluation is pending. Table 7 lists the models and the information that will be inserted from the experiment log; no values are presented because none are recorded in the supplied documents.

Table 7 Model results (documented values only)

Model	Task	Status	Validation metrics	Test metrics
Baseline CNN	Plane / normal-vs-abnormal	Trained	Not recorded	Pending
ResNet50 (fine-tuned)	Plane / normal-vs-abnormal	Preliminary	Not recorded	Pending



Model	Task	Status	Validation metrics	Test metrics
DenseNet (variant to be stated)	Plane / normal-vs-abnormal	Preliminary	Not recorded	Pending
EfficientNet (variant to be stated)	Plane / normal-vs-abnormal	Preliminary	Not recorded	Pending
Multi-disease model	Brain, heart, spine, kidney, other	Not started	—	—

The discussion that can be supported is limited to the qualitative observations above: imbalance, overfitting and variability dominate the early experiments, and the choice of the best architecture is not yet established.

XIV. LIMITATIONS AND FUTURE SCOPE

The principal limitations are: no quantitative results yet; a small, imbalanced training set; unconfirmed dataset provenance and licences; no external or prospective validation; limited GPU availability; scarce labelled abnormal data for several organs; and the absence of first-trimester coverage unless the datasets include it. Model confidence and heat-maps are aids, not clinical evidence.

Future work includes finalising the label scheme, building the multi-disease classifier, completing the controlled model comparison and locked-test evaluation, implementing Grad-CAM and its comparison with expert annotations, developing and security-testing the Flask interface, evaluating on data from other hospitals and machines, and exploring first-trimester and subtle-anomaly detection and calibrated uncertainty estimation.

XV. CONCLUSION

This paper presented a framework for AI-assisted multi-disease fetal anomaly detection from prenatal ultrasound images and an honest account of the project's progress. Dataset preparation, preprocessing, patient-wise splitting, a baseline CNN and initial fine-tuning of ResNet50, DenseNet and EfficientNet have been completed or begun, and early experiments reveal overfitting, class-imbalance effects and run-to-run variability. Multi-disease classification, full evaluation, Grad-CAM, the web application and testing remain to be done. The proposed protocol (patient-wise splits, training-only augmentation, a locked test set, repeated seeded runs and per-class reporting) is intended to make the final results credible. The system is a research prototype; any clinical use would require prospective validation and regulatory authorisation.

REFERENCES

- [1] R. Yousefpour Shahrivar, F. Karami, and E. Karami, "Enhancing fetal anomaly detection in ultrasonography images: A review of machine learning-based approaches," *Biomimetics*, vol. 8, no. 7, Art. no. 519, Nov. 2023, doi: 10.3390/biomimetics8070519.
- [2] T. Nath and M. Guha, "AI-integrated ultrasound technology: A blessing for early pregnancy fetal defect detection," *Int. J. Multidisciplinary Research (IJFMR)*, vol. 7, no. 5, Sep.–Oct. 2025, Art. ID IJFMR250555464.
- [3] R. Ramirez Zegarra and T. Ghi, "Use of artificial intelligence and deep learning in fetal ultrasound imaging," *Ultrasound Obstet. Gynecol.*, vol. 62, no. 2, pp. 185–194, 2023, doi: 10.1002/uog.26130.
- [4] U. Islam, Y. A. Ali, M. Al-Razgan, H. Ullah, M. A. Almaiah, Z. Tariq, and K. M. Wazir, "Fetal-Net: Enhancing maternal-fetal ultrasound interpretation through multi-scale convolutional neural networks and transformers," *Sci. Rep.*, vol. 15, Art. no. 25665, 2025, doi: 10.1038/s41598-025-06526-4.
- [5] E. D'Alberty, O. V. Patey, C. Smith, B. Šalović, N. Hernandez-Cruz, J. A. Noble, and A. T. Papageorgiou, "Artificial intelligence-enabled prenatal ultrasound for the detection of fetal cardiac abnormalities: A systematic review and meta-analysis," *eClinicalMedicine*, vol. 84, Art. no. 103250, Jun. 2025, doi: 10.1016/j.eclinm.2025.103250.



- [6] Y. Qi et al., "Multi-center study on deep learning-assisted detection and classification of fetal central nervous system anomalies using ultrasound imaging," arXiv:2501.02000, 2025 (preprint; not peer-reviewed).
- [7] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning (ICML), vol. 97, 2019, pp. 6105–6114.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
- [11] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in Int. Conf. Learning Representations (ICLR), 2021.
- [12] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. 38th Int. Conf. Machine Learning (ICML), vol. 139, 2021, pp. 8748–8763.
- [13] Y. Li et al., "Deep learning prediction of renal anomalies for prenatal ultrasound diagnosis," Sci. Rep., vol. 14, Art. no. 9013, 2024, doi: 10.1038/s41598-024-59248-4.
- [14] [Author names to be completed] "Development of an integrated deep learning approach for detecting fetal brain abnormalities at routine second trimester US scan: A multicenter study," Radiology: Artificial Intelligence, 2026, doi: 10.1148/ryai.250737.
- [15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. ICML, 2015, pp. 448–456.
- [16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE CVPR, 2017, pp. 4700–4708.
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
- [18] 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, no. 1, pp. 1929–1958, 2014.
- [19] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.
- [20] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," J. VLSI Signal Process. Syst., vol. 38, no. 1, pp. 35–44, 2004.
- [21] G. Litjens et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017.
- [22] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," J. Big Data, vol. 6, Art. no. 60, 2019.
- [23] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in Proc. IEEE CVPR, 2009, pp. 248–255.
- [24] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," arXiv:1603.04467, 2016.
- [25] G. Bradski, "The OpenCV library," Dr. Dobbs's Journal of Software Tools, vol. 25, pp. 120–125, 2000.
- [26] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

