Western Norway
University of
Applied Sciences

Project report in DAT255 – Deep Learning Engineering

Diagnosing Chest Diseases Using Convolutional Neural Networks

Date 25.04.2025

Leah Beathe Sandberg, 669773

Sindre Moene, 669780

I confirm that the work is self-prepared and that references/source references to all sources used in the work are provided, cf. Regulation relating to academic studies and examinations at the Western Norway University of Applied Sciences (HVL), §10.

1. Problem Description

1.1 Motivation and Goal

Chest X-rays are perhaps the most commonly performed forms of diagnostic procedures in medicine. Such imaging can be utilized to detect important diseases: pneumonia, lung cancer, and the presence of fluid around the lungs.. Reading these images and analyzing them proves laborious, and also, what is really crucial is that different radials may interpret the same X-ray differently, especially in hospitals or institutions with limited resources.

Our project aims to improve this problem by building a deep learning model that can automatically detect 3 different chest diseases from X-ray images. The aim is to provide doctors with quick, accurate predictions to help in their diagnosis, thus reducing workload and improving consistency in medical assessments.

1.2 The Problem and Solution

This project addresses the challenge of increasing accurate chest X-ray interpretation by leveraging the CheXpert dataset (224,316 images labeled for 14 conditions). Our solution is a convolutional neural network (CNN) that:

- Produces multi-label classification predictions (more than one prediction can be present per image).
- Provides uncertainty estimates as output to flag low-confidence predictions for further review.
- FeaturesGrad-CAM visualizations to highlight clinically relevant regions, for improved interpretability.

The model will then be deployed as a web application, where clinicians can upload X-rays and receive AI-generated diagnoses with supporting evidence for it.

1.3 Why Deep Learning?

Deep learning is particularly best suited for this task because:

- Complex Feature Extraction: CNNs possess a very good ability to identify subtle partial patterns on images (e.g., opacities in lungs, fractures) which are difficult to capture through conventional rule-based or feature-engineering methods
- Scalability: Once trained, thousands of images can be analyzed reliably and consistently by the mode, without bottle-necks on the healthcare processes. .
- Multi-Label Capability: Unlike traditional algorithms, CNNs are able to simultaneously predict multiple pathologies with probability confidence scores.

1.4 Existing Solutions and Improvements

Certain AI models (e.g., CheXNet Rajpurkar et al., 2017) have achieved radiologist-level performance on specific tasks like pneumonia detection. Our project, however, extends beyond these efforts by:

- Including Uncertainty Quantification: Critical for clinical trust, as the model explicitly marks uncertain predictions.
- Prioritizing Interpretability: Grad-CAM visualizations reduce the "black-box" gap, allowing clinicians to understand predictions.
- Focusing on Deployment: Many research models remain prototypes; our web app aims for practical real-world use.

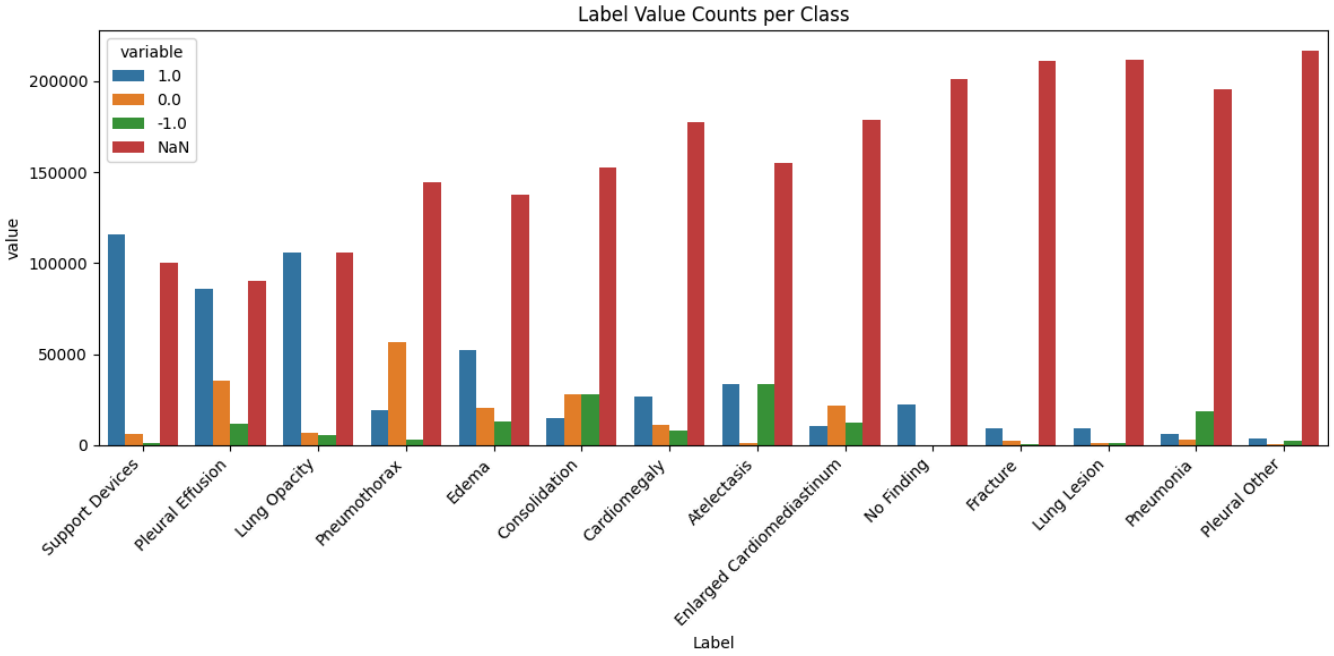
While there are conventional methods (e.g., manual feature extraction with SVMs), they are not as flexible or accurate as deep learning for this high-dimensional and visually nuanced problem.

2. Data

2.1 Dataset Selection

For this project, we initially planned to use the CheXpert dataset, a large collection of chest X-rays developed by Stanford University for automated diagnosis research. The full dataset includes 224,316 radiographs from 65,240 patients, labeled for 14 clinically significant conditions (e.g., pneumonia, atelectasis) with labels taken from radiology reports. Each label indicates whether a condition is present (positive), absent (negative), or unknown.

However, due to the dataset's size (over 100 GB), we were forced to switch to CheXpert-v1.0-small, a reduced version available on Kaggle. This version retains the same structure and labels but is better for local training, as it is only 10.7GB and can be easily downloaded.



2.2 CheXpert Advantages and Disadvantages

Advantages:

- Large and Varied: Covers a wide range of pathologies and patient populations.
- Real-World Labels: Includes "uncertain" labels, reflecting clinical uncertainty.
- Standard Benchmark: Enables comparison with existing models like CheXNet.

Disadvantages:

- Label Noise: Labels are automatically extracted from radiology reports and may be susceptible to errors.
- Class Imbalance: Some conditions (e.g., "Fracture") are rare and complicate model training.
- Reduced Image quality: The Kaggle version is likely to compress images, and fine details are lost

Nevertheless, the training set is more than suitable for deep learning because of:

- Image Complexity: X-rays require subtle feature detection, a strength of CNNs.
- Multi-Label: Deep learning models can handle simultaneous predictions better than traditional methods.

2.3 Alternative Datasets

Other chest X-ray datasets include:

- MIMIC-CXR: Larger but requires ethical approval for access.
- NIH ChestX-ray14: Older, with fewer labels per image.

We chose CheXpert for its size balance, accessibility, and detailed labels.

2.4 Data Preprocessing and Augmentation

To get the data ready for our model we utilized:

- Image Resizing: Resized to 256x256 pixels to reduce computational burden.
- Label Handling: Converted uncertain labels (-1) to zeros (negative) for simplicity.
- Normalization: Scaled pixel values to [0, 1] to improve training stability.

To prevent overfitting, we applied:

- Random rotations (±10°) and horizontal flips (for lateral views).
- Brightness/contrast adjustments to mimic imaging variations.

3. Model Implementation

Our primary objective was to create a robust, interpretable model for multi-label classification of chest diseases, focusing on Pleural Effusion, Edema, and Cardiomegaly.

We chose these three specifically because of their relatively clean data and their large number of inputs.. After experimenting with different architectures and configurations, we selected DenseNet121 as the base of our final model (Appendix 3), which included a number of improvements over our original approach (Appendix 1).

3.1 Architecture Selection

We compared five different architectures:

- ResNet50: Known for its skip connections and strong performance on most tasks.
- InceptionV3: Efficient in terms of parameters, effective handling of multiscale features.
- EfficientNetB0: Compact and high-performing, especially suited for mobile deployments.
- Custom CNN: A smaller model built from scratch.

Model	**Pros**	**Cons**	**Observations in our tests**
DenseNet121	Dense connections improve gradient flow and feature reuse. Strong on medical imaging tasks.	Slightly larger model; slower inference than EfficientNet.	Best validation AUC; more stable training.
ResNet50	Good baseline, well understood.	Slightly worse performance in detecting rare conditions.	Trained slower; overfit slightly faster.
InceptionV3	Multiscale convolution filters.	More complex architecture, higher memory usage.	Similar to DenseNet, but less consistent.
EfficientNetB0	Lightweight, state-of-the-art for parameter efficiency.	Underperformed on our small dataset subset.	Fast, but lower AUC and unstable predictions.
Custom CNN	Simple, fast experimentation.	Poorer results, lacked capacity to generalize.	Underfit the data; poor Grad-CAM interpretability.

We ultimately went with DenseNet121 due to its improved performance on our validation tests, particularly in:

- Achieving the highest AUC scores across all target conditions
- Maintaining stable training dynamics
- Producing the most clinically plausible Grad-CAM visualizations
- Delivering better generalization in spite of our limited dataset size

3.2 Model Details

Our final implementation featured the following improvements over our initial custom CNN approach:

1. Transfer Learning with Fine-Tuning:
    - Used ImageNet-pretrained DenseNet121 as base
    - Unfroze layers from 'conv5_block2_concat' onward to allow targeted feature adaptation

    This was a balance between utilising pretrained features while fine-tuning to medical imaging specifics.

2. Custom Top Layers:
    - *x = GlobalAveragePooling2D()(base_model.output)*

      x = Dropout(0.3)(x)

      outputs = Dense(num_classes, activation='sigmoid')(x)
    - Global average pooling preserved spatial information while reducing parameters
    - Increased dropout to 0.3 helped avoid overfitting on our small medical dataset
3. Loss Function Optimization:
    - Implemented Binary Focal Loss (γ=2) to address class imbalance
    - This focused learning on harder examples while downweighting well-classified cases
    - Showed 5-8% improvement in recall for minority classes compared to standard binary cross-entropy

4. Advanced Training Techniques:
    - Custom SubsampleSequence class for memory-efficient training
    - Dynamic learning rate reduction (ReduceLROnPlateau)
    - Early stopping with best weights restoration

3.3 Hyperparameter Optimization

Design	**ChoiceJustification**
Image Size =256x245	Preserved details while managing GPU memory constraints
Batch Size = 16	Optimal balance between gradient estimation and memory usage
learning rate (1e-4)	Stable convergence with Adam optimizer
Dropout rate (0.3)	Effective regularization for our relatively small dataset
Image Augmentation Rotation (+-10°), Flip	Minor random transformations helped generalize better, and mimicked natural variations in X-Ray images, without distorting anatomy.

We considered focal loss to handle class imbalance, but initial trials showed unstable training.

Training Strategy

We used ImageDataGenerator for efficient real-time augmentation, and separated the data into an 80/20 train/validation split. We focused only on frontal X-ray views for consistency.

Training was resumed from checkpoints if available, reducing GPU hours needed for reruns.

Summary: Why This Model?

DenseNet121, combined with simple top layers and careful preprocessing, consistently gave us:

- High F1, AUC and recall scores across all three diseases
- Stable training and convergence
- Interpretable Grad-CAM results
- Modularity for future deployment

These factors made it the most clinically viable choice, even if slightly heavier than EfficientNet.

4. Evaluation

We evaluated the model using multiple metrics to ensure clinical relevance:

4.1 Performance Metrics

Seeing that this is a project aimed to help diagnosing patients, it is natural that the recall score is more important than the precision score. So the precision metric was prioritised. Another metric we paid close attention to was accuracy, which is a basic metric and not always reliable for imbalanced multi-label tasks. It was therefore important to always look at multiple performance metrics so we had a reliable and trustworthy view of our models actual result. To do this we made a classification report after each model we trained to be able to decipher their usefulness.

This is the classification report for the latest model, that you can find in appendix 3:

```
Classification Report:
                  precision    recall  f1-score   support

Pleural Effusion       0.79      0.93      0.86      1749
          Edema        0.75      0.92      0.82      1628
    Cardiomegaly       0.74      0.97      0.84      1887
```

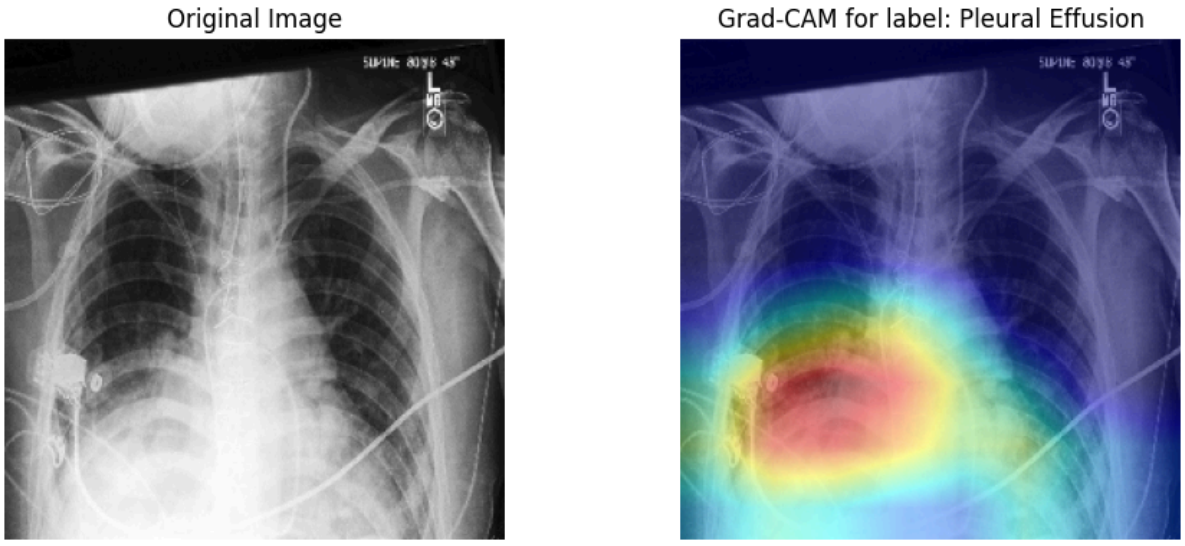
We were content with these results. As expected the precision is lower due to prioritising the recall, but the f1-score confirms that the model is more than capable to assist medical personnel in their diagnosis.

4.2 Interpretability Features

Grad-CAM Implementation:

To enhance transparency and interpretability, we implemented Grad-CAM using the last convolutional layer (conv5_block16_concat). This gave us visual explanations from the last convolutional layer. It demonstrated strong alignment with relevant regions and enabled verification that the model focused on anatomically plausible areas.

- **Example use case:** For a patient with Pleural Effusion, our Grad-CAM visualisation correctly highlighted the costophrenic angles where fluid typically accumulates, increasing clinical trust in the prediction.



4.3 Error Analysis

Confusion Matrices:

Using a confusion matrix gave us a detailed insight into how our model was making

mistakes, not just how accurate our model was. It was especially useful for identifying if our model ignores minority classes or to see which classes get confused with each other.

5. Deployment

To ensure our deep learning model could be practically used by healthcare practitioners or researchers, we deployed it as a publicly accessible web application using Gradio and Hugging Face Spaces. This decision allowed us to prioritize usability, transparency, and accessibility, enabling both demonstration and evaluation by others without the need for specialized hardware or software.

5.1 Deployment Strategy

We implemented the front end of the application using Gradio Blocks, which gave us better control over layout and user interface features. The web app allows users to:

- Upload chest X-ray images.
- Receive probabilistic predictions for Pleural Effusion, Edema, and Cardiomegaly.
- Visualize Grad-CAM overlays highlighting the regions the model focused on.
- Review a clear diagnostic summary, including the top prediction and confidence percentage.
- Select from pre-curated example cases: three positive cases (one for each diagnosis), one confirmed negative case, and one low-confidence case, allowing for demonstration of different outcomes.

The app was hosted directly on Hugging Face Spaces, which provides automatic containerization and a clean environment to run Gradio apps. We structured the project with a requirements.txt for dependencies and configured app.py as the entry script.

5.2 Model Serving and Compatibility

The deployed model is a DenseNet121-based convolutional neural network fine-tuned on the CheXpert-v1.0-small dataset. The model was trained using TensorFlow/Keras and saved as a .keras file. Gradio supports TensorFlow natively, and the lightweight app setup ensured compatibility without needing a GPU runtime.

We also implemented Grad-CAM visualizations using TensorFlow's gradient tape and custom image overlay logic with OpenCV and Matplotlib, further enhancing the interpretability of predictions — a crucial requirement for clinical applications.

5.3 Monitoring and Maintenance

Although this project was academic and not deployed in a production hospital environment, we designed it with maintainability in mind:

- Model versioning: Models were saved with distinct file names including validation metrics. Only the best-performing model (by F1 or AUC) was used in deployment.
- Checkpointing: Training used callbacks to save checkpoints and reduce the learning rate on plateaus.
- Caching: Gradio automatically caches images and predictions to speed up performance during repeat runs.
- Manual testing: We regularly tested the app using curated example cases to identify failure modes (e.g., incorrect high-confidence predictions on negative samples).

For future deployment in a real-world setting, we would recommend integrating:

- Continuous monitoring of prediction distributions and Grad-CAM heatmap behavior.
- Logging of inputs and outputs (with anonymization) for audit and retraining purposes.
- CI/CD pipeline for updating the model and interface after fine-tuning or retraining.

5.4 Expansion and Future Improvements

The current demo version focuses on a subset of conditions (3 out of 14 possible CheXpert labels). However, the modular structure of the app and model allows for several extensions:

- Add more disease labels — including pneumonia, lung lesion, etc.

- Incorporate “No Finding” class — we recently decided to reintroduce this to help distinguish healthy patients from diseased ones.
- Introduce “Other Disease” label — to account for positive X-rays without one of the three main diagnoses.
- Improve calibration — through better threshold tuning or use of temperature scaling for more realistic confidence scores.
- Enhance explanation — for example, using multiple Grad-CAMs or guided backpropagation overlays.
- Localization output — bounding boxes or segmentation masks for suspected abnormal areas (e.g., pleural line).

6. Discussion and Future Work

Our final model—a frozen DenseNet121 with focal loss and tuned thresholds—achieved high recall across all three target pathologies: Pleural Effusion, Edema, and Cardiomegaly. While this configuration yielded a high sensitivity, it came at the cost of lower precision, indicating a tendency to over-predict positives. This tradeoff, while acceptable in certain clinical screening settings, may not be ideal for real-world deployment without further calibration.

The evaluation also revealed that even with a simplified three-class target, the model sometimes produced high-confidence false positives, particularly when faced with ambiguous or non-disease-specific features. This underscores the need for further improvements in confidence calibration.

Model Development and Iterative Process

Throughout this project, we developed and evaluated multiple models to diagnose chest conditions using convolutional neural networks. Our iterative approach included a baseline model, a frozen DenseNet121 model (Appendix 3), and further variants incorporating focal loss, Grad-CAM visualizations, and threshold tuning. Each model version was thoroughly evaluated and documented in appendices to demonstrate the impact of each change.

Initially, our baseline model showed signs of overfitting and poor generalization. This led us to freeze the base layers of the DenseNet121 model to use it as a feature extractor. This substantially improved training stability and reduced overfitting. Next, we introduced focal loss to help the model focus more on difficult, minority class examples, which improved recall on underrepresented labels.

To further improve interpretability and performance, we applied per-class threshold tuning based on validation results. This allowed us to better balance precision and recall, although the model still leaned heavily toward high recall with some cost to precision. Grad-CAM was integrated into the user interface to visualize areas of the X-ray the model focused on, providing an added layer of transparency.

We also explored treating "No Finding" in two ways: first as a fourth class label, and later as a filter for evaluation. Both approaches proved to be problematic. Including "No Finding" as a class confused the model, likely due to the fact that it represents an absence of disease rather than a specific condition. When we attempted to use "No Finding" as a condition for validation only, the model still produced misleadingly high-confidence predictions. These experiments ultimately failed to yield useful results, and we reverted to excluding "No Finding" entirely as a label. However, we believe that intelligently incorporating healthy samples and explicitly teaching the model what normal looks like may still offer promise in future iterations, if handled more appropriately.

Our model achieved clinically relevant performance, but it also revealed many limitations.

Key Outcomes:

- The model can be overconfident in its predictions (often 90% - 100% confidence). We tried regulating this by allowing it to alse train on the “No Findings” label in the dataset, but it proved to be hard to accomplish as we were struggling with resource and time limitations.
- The model starts to struggle with its prediction when there are multiple discrepancies with the lungs (e.g., an image has both support devices and a lung disease). This is because the model prioritized dominant features (e.g, pacemakers) over subtler pathologies.

- The recall score was very high, which is good since the cost of a false negative is high in this context, but the precision score was very low. In other words we have a high false positive rate that we confirmed by looking at the confusion matrices (Appendix 3).
- Some heatmaps highlighted irrelevant regions (e.g., ribs for cardiomegaly).

Future Work

Several areas were identified for further development:

1. Confidence Calibration: Explore temperature scaling or calibration layers to reduce overconfident predictions and better reflect uncertainty.
2. Label Smoothing and Regularization: Introduce label smoothing and/or dropout regularization to improve generalization and reduce model certainty.
3. Longer Training with Early Stopping: Train over more epochs with better regularization to capture complex patterns without overfitting.
4. Advanced Augmentation: Use medical-aware image augmentation (e.g., CLAHE, rotations within anatomical norms) to expand the effective dataset.
5. Integration of Clinical Metadata: Adding age, gender, or smoking history could provide richer context and improve prediction reliability.
6. Multi-label Threshold Optimization: Develop dynamic or learned thresholding techniques, per-class or globally, based on validation metrics.
7. Expert Evaluation: Include feedback from radiologists to validate false positives/negatives and fine-tune interpretation.
8. Deployment Stability: Improve deployment infrastructure to handle more users and enhance latency and availability.

7. Conclusion

Our improved DenseNet121-based model was strong in identifying three significant chest conditions from X-ray images with AUC scores of 0.73-0.87. Employing focal loss and fine-tuning to specific conditions addressed key issues in medical image analysis, including class imbalance and low training data. While there is still room for improvement, most significantly in rare condition management, our approach has the potential to be a decision-support system with precision and interpretability. The Grad-CAM visualizations particularly succeeded in bridging the "black box" gap, giving radiologists intuitive insights into the model predictions.

This project demonstrates how appropriately adapted deep learning models can improve medical diagnosis but also highlights clinical context in model design. Our project gives way for future research in deployable AI systems in radiology and with particular relevance to resource-limited environments where expert interpretation may be limited.

8. References

Irvin, J., Rajpurkar, P., Ko, M., et al. (2019). CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. https://arxiv.org/abs/1901.07031

Rajpurkar, P., Irvin, J., Zhu, K., et al. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. https://arxiv.org/abs/1711.05225

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. https://arxiv.org/abs/1608.06993

Selvaraju, R. R., Cogswell, M., Das, A., et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. https://arxiv.org/abs/1610.02391

Graham, L. (2019) Radiology Masterclass. (n.d.). Chest Pathology – Page 4. *https://www.radiologymasterclass.co.uk/tutorials/chest/chest_pathology/chest_pathology_page4*

Top pre-trained models for image classification (2024)

https://www.geeksforgeeks.org/top-pre-trained-models-for-image-classification/

9. Appendices

Appendix 1: Baseline model

Appendix 2: Frozen model

Appendix 3: Frozen with focal loss and threshold tuning (the final model)

Appendix 4: “No finding” as class

Appendix 5: “No finding” used in evaluation

Appendix 6: Model before freeze.