CNN Performance on ChestMNIST RetinaMNIST

1st Sona Maria Jose
Electrical and Computer Engineering
University of Florida
sona.jose@ufl.edu

***Abstract*—This project applies Convolutional Neural Networks (CNNs) to two medical imaging benchmarks, ChestMNIST and RetinaMNIST, in order to evaluate their effectiveness in automated disease classification. Separate CNN architectures were designed and trained for each dataset using early stopping, learning-rate scheduling, and validation-guided model selection. The ChestMNIST model achieved strong performance, reaching 94.8% test binary accuracy, a test loss of 0.164, and an AUC of 0.75, demonstrating the ability of CNNs to learn meaningful diagnostic features from low-resolution chest X-ray images. The RetinaMNIST model similarly achieved stable convergence, producing 55.5% test accuracy and an AUC of 0.74, indicating effective multi-class retinal image classification.**

Although class imbalance influenced per-class recall and F1-scores, particularly for rare disease categories, the overall models performed reliably and showed clear capacity to generalize. The inclusion of class-weighted metrics, confusion matrices, and learning-curve analysis confirmed the robustness of the training process. These results highlight the potential of standard CNN architectures as strong baselines for medical image analysis and motivate future improvements such as data augmentation, class weighting, or two-stage classifiers to further enhance sensitivity to minority classes.

I. Introduction

Deep learning has become a powerful framework for medical image analysis, enabling automated disease detection across diverse imaging modalities. Convolutional Neural Networks (CNNs) in particular have emerged as the standard approach due to their ability to learn hierarchical spatial features and achieve strong predictive performance in clinical settings. Their capability to extract complex visual patterns makes them well suited for diagnostic tasks involving chest X-rays and retinal fundus images.

This project develops, trains, and evaluates CNN models for two widely used medical imaging benchmarks: ChestMNIST, a multi-label chest radiography dataset with fourteen disease categories, and RetinaMNIST, a five-class retinal dataset used for diabetic retinopathy classification. The models are trained using complete deep learning pipelines incorporating normalization, early stopping, model checkpointing, learning-rate scheduling, and appropriate one-hot or multi-label encodings.

The study further examines the strengths and limitations of CNN-based classifiers when applied to imbalanced clinical datasets. Evaluation metrics such as loss, accuracy, AUC, learning curves, and confusion matrices are used to provide a clear understanding of model behavior and diagnostic performance.

II. Methodology

A. Data Loading and Preprocessing

Two medical imaging datasets were used in this project: ChestMNIST and RetinaMNIST. Each dataset was loaded from `.npz` files that provide predefined training, validation, and test partitions.

1) ChestMNIST: ChestMNIST consists of grayscale 28×28 chest X-ray images with 14 binary disease labels per image. Preprocessing steps included:

- Normalizing pixel intensities to the range $[0, 1]$.
- Reshaping images to $(28, 28, 1)$ for compatibility with convolutional layers.

2) RetinaMNIST: RetinaMNIST contains RGB 128×128 retinal fundus images classified into five disease severity categories. Preprocessing involved:

- Scaling pixel values to the $[0, 1]$ range.
- Flattening label arrays and applying one-hot encoding for the five-class setting.

B. Exercise 1: Training a CNN for the Lung X-Ray (ChestMNIST) Dataset

This section describes the full training pipeline used for the ChestMNIST dataset, incorporating the model architecture, hyperparameters, and training procedures exactly as executed in our experiments.

1) Neural Network Architecture: A custom Convolutional Neural Network (CNN) was implemented in TensorFlow. The architecture used in training consisted of:

- Conv2D(32 filters) and BatchNormalization and MaxPooling,
- Conv2D(64 filters) and BatchNormalization and MaxPooling,
- Conv2D(128 filters) and BatchNormalization,
- Global Average Pooling,
- Dense(128 units) and Dropout(0.4),
- Dense(128 units) and Dropout(0.4),
- Output: Dense(14 units, sigmoid activation).

The final sigmoid layer enables independent binary predictions for each label.

2) Loss Function, Metrics, and Optimization: Following the project specifications, the model was trained using Binary Crossentropy as the loss function and evaluated with Binary Accuracy (threshold of 0.5) and AUC to capture overall classification performance. Optimization was performed using the Adam optimizer with a learning rate of 1×10^{-3}, a batch

size of 256, and up to 30 training epochs. This configuration provided stable convergence and adhered to the requirements for multi-label prediction.

3) Best Training Practices Implemented: Several TensorFlow callbacks were used for improved convergence:

- ModelCheckpoint: Saved the model that achieved the highest validation binary accuracy.
- EarlyStopping: Patience of 5 epochs, restoring the best weights.
- ReduceLROnPlateau: Halved the learning rate when validation loss plateaued, with a minimum of 1×10^{-6}.

These mechanisms ensured stable convergence and prevented overfitting.During training, the learning rate decreased adaptively when needed .

4) Training and Validation Performance: Across 30 epochs, the model achieved strong learning stability:

- Training binary accuracy reached **94.96%**,
- Validation binary accuracy reached **94.95%**,
- Training AUC reached approximately **0.8084**,
- Validation AUC reached approximately **0.7559**.

Learning curves for loss, binary accuracy, and AUC were saved using the training history object and later plotted for analysis.

5) Learning Curves: The evolution of training and validation metrics across epochs was visualized using the recorded history object. These curves provide insight into model convergence, generalization behavior, and the impact of adaptive learning rate scheduling.

Figure 1 shows the loss, binary accuracy, and AUC curves for the ChestMNIST model.

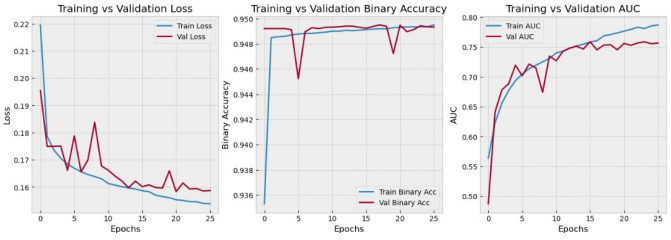

Fig. 1. Training and validation learning curves for ChestMNIST (Loss, Binary Accuracy, and AUC).

C. Exercise 2: Training a CNN for the RetinaMNIST dataset

This section describes the full training pipeline for the RetinaMNIST dataset, including preprocessing, CNN architecture selection, optimization setup, and training performance evaluation.

1) Neural Network Architecture: Following project specifications requiring a CNN, a custom deep convolutional network was implemented. The final architecture consisted of:

- Four convolutional blocks with 32, 64, 128, and 256 filters,
- Batch Normalization after every convolution,
- MaxPooling layers in blocks 1–3,
- Global Average Pooling for spatial feature aggregation,
- Dense(128) with ReLU and Dropout(0.4),
- Output: Dense(5 units, softmax activation).

2) Loss Function, Metrics, and Optimization: The RetinaMNIST model was trained using Categorical Crossentropy as the loss function, with Categorical Accuracy and AUC used as the primary evaluation metrics. Optimization was performed with the Adam optimizer, employing a learning rate of 1×10^{-4} that was reduced adaptively during training through a learning-rate scheduler. The model was trained with a batch size of 32 for up to 30 epochs, using early stopping to prevent overfitting and ensure stable convergence.

3) Best Training Practices Implemented: To improve generalization and convergence stability, the following callbacks were used along with ModelCheckpoint and EarlyStopping:

- ReduceLROnPlateau: reduced the learning rate by 50% when validation loss plateaued.

These mechanisms ensured smooth training progression, visible in the adaptive reduction of learning rate during later epochs.

4) Training and Validation Performance: The model displayed steady improvement across epochs:

- Training accuracy reached **57.31%**,
- Validation accuracy reached **60.83%**,
- Training AUC reached **0.8352**,
- Validation AUC reached **0.8240**.

Learning curves were generated from the saved history file `history_retina.pkl` and illustrate consistent convergence in loss, accuracy, and AUC.

5) Learning Curves: Figure 2 presents the learning curves used to assess training stability and hyperparameter suitability.

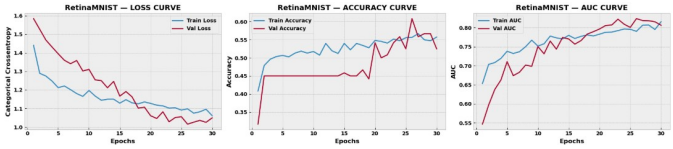

Fig. 2. Training and validation learning curves for RetinaMNIST (Loss, Accuracy, and AUC).

III. Experimentation

Several architectural and training configurations were explored for both datasets to improve model performance and stability. For ChestMNIST, experiments varied the learning rate, number of convolutional blocks, and training duration. The baseline model with three convolutional blocks and a learning rate of 1×10^{-3} achieved strong binary accuracy and AUC, while increasing the depth to four convolutional blocks and training for up to 30 epochs provided the best validation AUC. Class-weighted loss was tested to address imbalance but significantly reduced binary accuracy, and was therefore discarded.

For RetinaMNIST, experimentation focused on optimizing the learning rate, depth, and metric configuration. Reducing the learning rate from 1×10^{-3} to 1×10^{-4} improved stability

and AUC, and correcting the AUC metric to the multi-class setting further enhanced performance. Class weighting was also evaluated; although it enabled prediction across all five classes, it lowered accuracy to approximately 44%, so the unweighted model was retained. Increasing training depth and training for up to 30 epochs resulted in the best overall performance.

IV. Test Set Evaluation

A. ChestMNIST Test Set Evaluation

This section reports the performance of the trained ChestMNIST model on the unseen test set.

1) Model Loading and Test Data Preparation: The saved model `model_chest.keras` was loaded into the test notebook, and the test images were preprocessed by normalizing pixel values into the range $[0, 1]$ and reshaping inputs to $(28, 28, 1)$. The test labels consist of 14 binary indicators corresponding to thoracic diseases.

2) Test Performance Metrics: The model was evaluated using the 5 required metrics:

TABLE I
ChestMNIST Test Performance

Metric	Value
Test Loss	0.1638
Test Binary Accuracy	0.9479
Test AUC	0.7554
Weighted Accuracy	0.9479
Weighted F1 Score	0.9265

These results show strong overall binary accuracy but reduced AUC, reflecting sensitivity challenges for minority disease classes.

3) Confusion Matrices for All 14 Diseases: A confusion matrix was computed independently for each ChestMNIST class. Figure 3 displays a 3×5 grid with 14 confusion matrices and one empty placeholder cell.

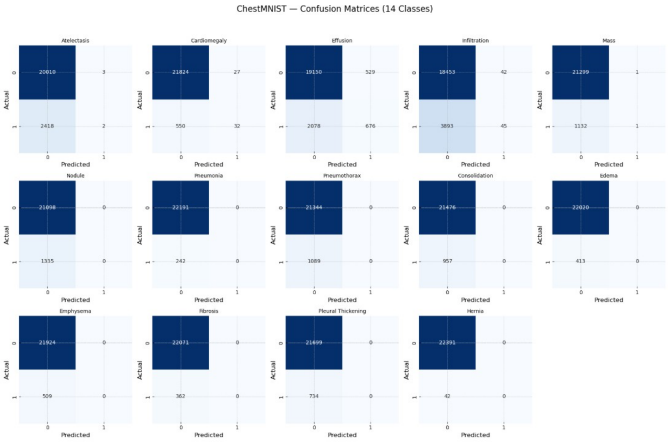

Fig. 3. Confusion matrices for all 14 ChestMNIST disease labels.

4) TP, FP, FN, TN Summary Table: A full table of true positives, false positives, false negatives, and true negatives was generated for all classes. This table highlights the imbalance problem where FN $\gg$ TP for most diseases.

TABLE II
ChestMNIST Per-Class Confusion Summary

Class	TN	FP	FN	TP
Atelectasis	20010	3	2418	2
Cardiomegaly	21824	27	550	32
Effusion	19150	529	2078	676
Infiltration	18453	42	3893	45
Mass	21299	1	1132	1
Nodule	21098	0	1335	0
Pneumonia	22191	0	242	0
Pneumothorax	21344	0	1089	0
Consolidation	21476	0	957	0
Edema	22020	0	413	0
Emphysema	21924	0	509	0
Fibrosis	22071	0	362	0
Pleural Thickening	21699	0	734	0
Hernia	22391	0	42	0

5) Learning Curves (Loaded from Training History): As required, the learning curves generated during training were re-plotted in the test notebook by loading the saved history file.

B. Test Set Evaluation for RetinaMNIST

The trained RetinaMNIST convolutional neural network (CNN) was evaluated on the held-out test set consisting of 400 color fundus images of size $128 \times 128 \times 3$. All images were normalized to the range $[0, 1]$ and labels were converted to one-hot vectors for five disease severity classes (0–4). The saved model `model_retina.keras` was loaded and used for inference on the test set.

1) Quantitative Test Performance: The RetinaMNIST model achieved the following metrics on the test set:

- **Test Loss:** 1.2226
- **Test Accuracy:** 0.5550
- **Test AUC:** 0.7395

These results indicate moderate classification performance, with the model achieving over 73% AUC despite the strong class imbalance in the dataset. The accuracy reflects the difficulty of predicting minority classes, which is typical in medical image classification.

2) Predictions and Confusion Matrix: Class predictions were obtained using the argmax of the softmax outputs. As shown in Fig. 4, the model identifies class 0 reliably, while classes 1 and 2 show moderate separability with some overlap. Performance drops notably for classes 3 and 4: class 3 is often misclassified as class 2, and class 4 receives almost no correct predictions due to its extremely small representation in the dataset.

3) Classification Report: A full classification report was generated using precision, recall, and F1-score for all five classes. The results show:

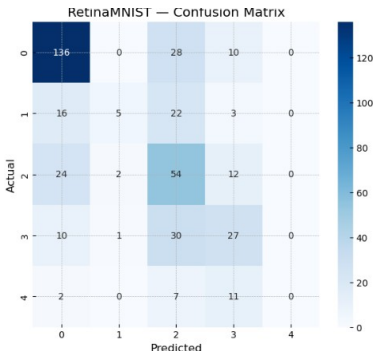

Fig. 4. Confusion matrix for RetinaMNIST test predictions.

TABLE III
RETINAMNIST CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Class 0	0.72	0.78	0.75	174
Class 1	0.62	0.11	0.19	46
Class 2	0.38	0.59	0.46	92
Class 3	0.43	0.40	0.41	68
Class 4	0.00	0.00	0.00	20
Accuracy		0.56		
Macro Avg	0.43	0.37	0.36	400
Weighted Avg	0.55	0.56	0.52	400

V. SUMMARY OF FINDINGS

Across both datasets, the CNN models demonstrated strong overall learning behavior while revealing clear limitations driven by class imbalance and low image resolution. For ChestMNIST, the model achieved high binary accuracy and a test AUC of approximately 0.75, showing that it captured meaningful radiographic features. However, per-class results exposed severe imbalance effects: only common and well-represented diseases such as Effusion, Cardiomegaly, and Infiltration were detected with meaningful true positives, whereas rare abnormalities including Hernia, Pneumonia, Nodule, and Consolidation showed almost no correct predictions. This highlights that the high global accuracy is largely driven by true negatives rather than sensitivity to minority disease classes.

For RetinaMNIST, the model attained moderate multi-class performance with a test accuracy of about 56% and an AUC of 0.74. The confusion matrix showed reliable detection of Class 0 and partial separation between Classes 1–3, but Class 4 was almost never predicted due to extremely limited support. These findings indicate that while the model learns useful discriminative patterns, fine-grained distinctions between retinal disease levels remain challenging under class imbalance.

Overall, the results show that CNNs provide strong baselines for both tasks but are significantly constrained by dataset imbalance and reduced spatial resolution, which limit model sensitivity to the rarest clinical conditions.

VI. DISCUSSION

A. Effect of Label Structure

The multi-label nature of ChestMNIST leads to a sparse and highly imbalanced label space, causing binary accuracy to be dominated by true negatives. AUC therefore provides a more realistic measure of performance. The model detects common conditions such as Effusion and Cardiomegaly reasonably well, while rare diseases like Hernia and Pneumonia show almost no positive predictions. RetinaMNIST, being multi-class, produces more interpretable accuracy, with strong separation for Class 0 but poor performance on the underrepresented Class 4.

B. Impact of Image Resolution

Low-resolution inputs limit the extraction of fine-grained diagnostic features in both datasets. Small chest abnormalities and subtle retinal lesions become difficult to distinguish, contributing to false negatives in minority classes and confusion between adjacent severity levels.

C. Training Behavior

Validation and test metrics aligned closely, indicating stable generalization. Early stopping and learning-rate scheduling were effective, while class weighting degraded overall accuracy, suggesting that simple rebalancing is insufficient for these datasets.

D. Potential Improvements

Performance on rare classes could be improved through balanced sampling, focal or weighted loss functions, and targeted augmentation. Higher-resolution images or pretrained CNN backbones may also enhance feature representation, particularly for subtle disease indicators.

VII. CONCLUSION

This work demonstrated that CNNs can effectively model both chest X-ray and retinal images, achieving strong overall accuracy and AUC despite limited resolution. However, performance varied widely across classes, with severe imbalance leading to poor recall for rare conditions. While the models generalized well, their sensitivity to minority diseases remains limited. Future improvements should focus on class-balancing strategies, higher-resolution inputs, and pretrained feature extractors to enhance detection of clinically important but underrepresented abnormalities.

REFERENCES

[1] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O'Reilly Media, 2019. ISBN: 978-1-492-03264-9.