Disease Detection from Chest X-ray Images Using Deep Learning

Adithya Asokan and Subhan Ahmad

Faculty Mentor: Professor Ole Forsberg

Funding: Richter Memorial Fund

Can AI detect disease from a simple X-ray?"

Scalable diagnosis in low-resource settings

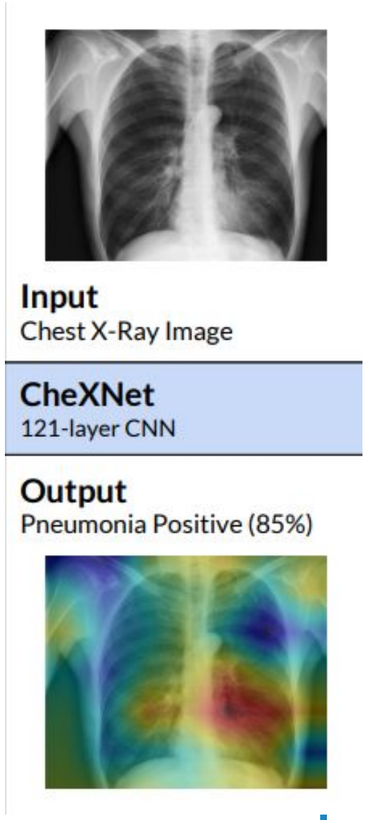
Literature Review

CheXNet (Rajpurkar et al., 2017)

- A 121-layer DenseNet trained on the ChestX-ray14 dataset (over 100,000 images, 14 diseases).
- Surpassed average radiologist performance in detecting pneumonia, achieving an F1-score of 0.435 compared to the average radiologist F1 of 0.387.
- Used Class Activation Maps (CAMs) to provide visual explanations for predictions.

What We Add

- Used two datasets (NIH ChestX-ray14 + Kaggle's pneumonia dataset) for cross-domain evaluation.
- Integrated Grad-CAM extensively — not just to explain predictions, but also to analyze misclassified cases.



Research Question

How accurately can deep learning classify X-rays into NORMAL or PNEUMONIA?

Datasets

- Kaggle Dataset (Guangzhou Medical Center): Balanced pneumonia/normal images

```
# === 1. Load Full Dataset ===
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  './data/chest_xray/chest_xray/train',
  validation_split=0.3,
  subset='training',
  seed=123,
  image_size=(224, 224),
  batch_size=32
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  './data/chest_xray/chest_xray/train',
  validation_split=0.3,
  subset='validation',
  seed=123,
  image_size=(224, 224),
  batch_size=32
)
```

Datasets

- NIH ChestX-ray14: 2881 images, US-based

```
import pandas as pd
import os

# Set path to the downloaded dataset
dataset_path = '/home/jovyan/.cache/kagglehub/datasets/nih-chest-xrays/data/versions/3'

# Load metadata
csv_path = os.path.join(dataset_path, 'Data_Entry_2017.csv')
df = pd.read_csv(csv_path)

# View basic info
print("Number of images:", len(df))
print("Unique diseases:", df['Finding Labels'].str.split('|').explode().nunique())
print("Sample labels:", df['Finding Labels'].unique()[:5])
df.head()
```

	Image Index	Finding Labels	Follow-up #	Patient ID	Patient Age	Patient Gender	View Position	OriginalImage[Width	Height]
0	00000001_000.png	Cardiomegaly	0	1	58	M	PA	2682	2749
1	00000001_001.png	Cardiomegaly|Emphysema	1	1	58	M	PA	2894	2729
2	00000001_002.png	Cardiomegaly|Effusion	2	1	58	M	PA	2500	2048
3	00000002_000.png	No Finding	0	2	81	M	PA	2500	2048
4	00000003_000.png	Hernia	0	3	81	F	PA	2582	2991

Unique labels in the NIH Chest X-ray dataset:

- Atelectasis
- Cardiomegaly
- Consolidation
- Edema
- Effusion
- Emphysema
- Fibrosis
- Hernia
- Infiltration
- Mass
- No Finding
- Nodule
- Pleural_Thickening
- Pneumonia
- Pneumothorax

Datasets

- NIH ChestX-ray14: 2881 images, US-based

```
# Load NIH labels and map to binary
def binary_label(finding):
  if 'Pneumonia' in finding:
    return 'PNEUMONIA'
  elif finding.strip() == 'No Finding':
    return 'NORMAL'
  return None

label['binary_label'] = label['label'].apply(binary_label)
balanced_df = label.dropna(subset=['binary_label'])
```

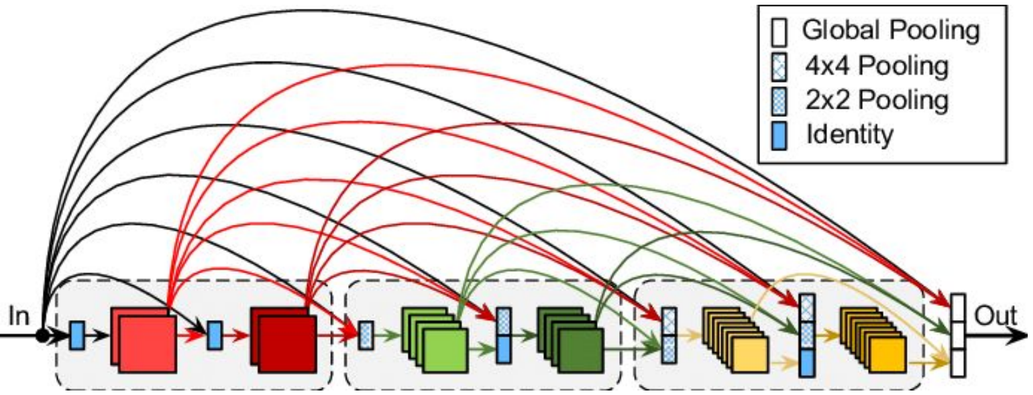
	image_path	image_name	binary_label
0	/home/jovyan/.cache/kagglehub/datasets/nih-che...	00011062_001.png	NORMAL
1	/home/jovyan/.cache/kagglehub/datasets/nih-che...	00000468_029.png	PNEUMONIA
2	/home/jovyan/.cache/kagglehub/datasets/nih-che...	00017710_009.png	PNEUMONIA
3	/home/jovyan/.cache/kagglehub/datasets/nih-che...	00003028_050.png	PNEUMONIA
4	/home/jovyan/.cache/kagglehub/datasets/nih-che...	00011721_000.png	NORMAL

Preprocessing

```
def load_image(path, label):
    image = tf.io.read_file(path)
    image = tf.image.decode_png(image, channels=3)
    image = tf.image.resize(image, [224, 224])
    image = image / 255.0
    return image, label

train_ds = tf.data.Dataset.from_tensor_slices((paths, labels))
train_ds = train_ds.map(load_image).batch(32).prefetch(tf.data.AUTOTUNE)
```

- 224x224 resizing, normalization
- Augmentation: flip, rotate, zoom
- Balanced classes (e.g., 1450 normal samples)

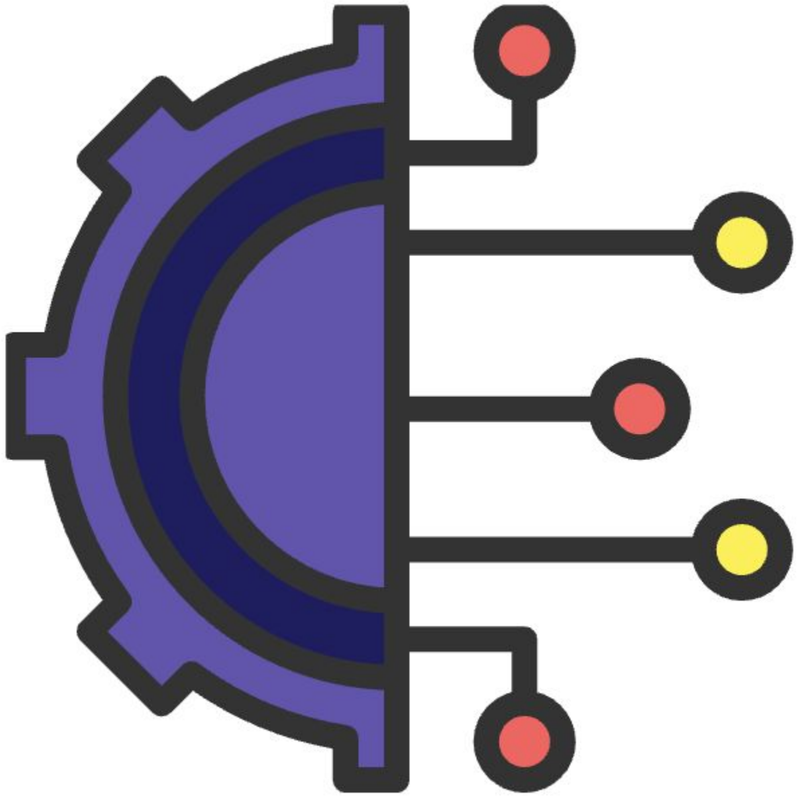
Model Architecture




```
base_model = DenseNet121(include_top=False, weights='imagenet', input_shape=(224, 224, 3))
base_model.trainable = False

model = tf.keras.Sequential([
  base_model,
  tf.keras.layers.GlobalAveragePooling2D(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.5),
  tf.keras.layers.Dense(1, activation='sigmoid')
])
```

- Block diagram: DenseNet121 → GAP → Dense → Dropout → Sigmoid
- Transfer learning (ImageNet weights)

Training Details

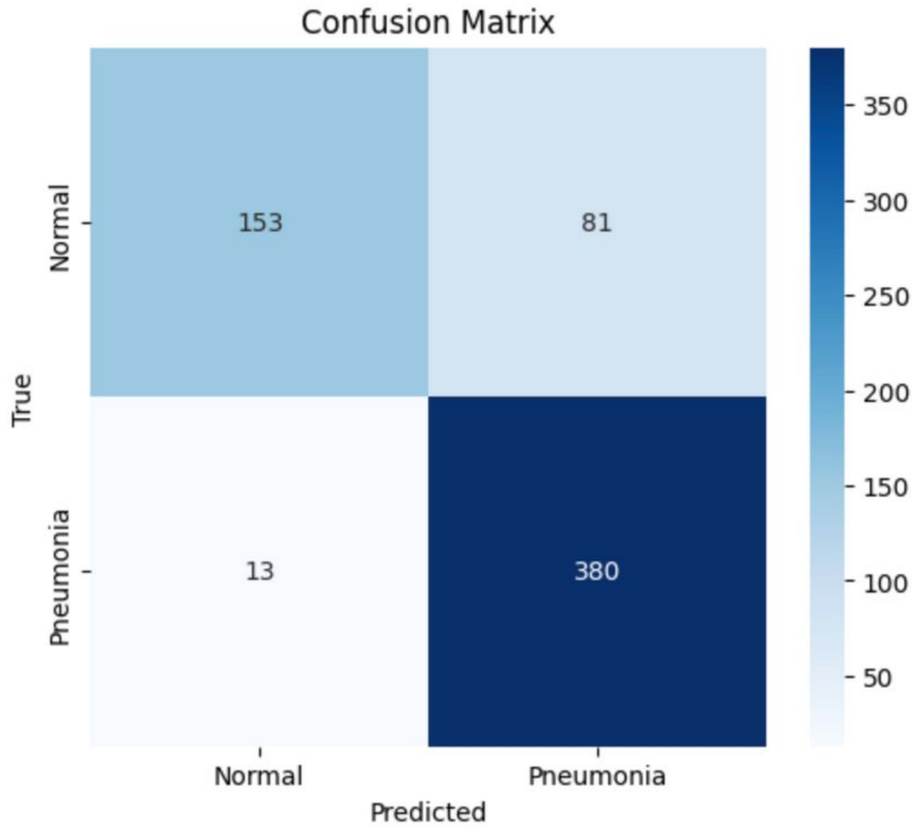


- Binary cross-entropy loss
- Adam optimizer (lr=1e-5)
- Early stopping, ReduceLROnPlateau
- Class weights

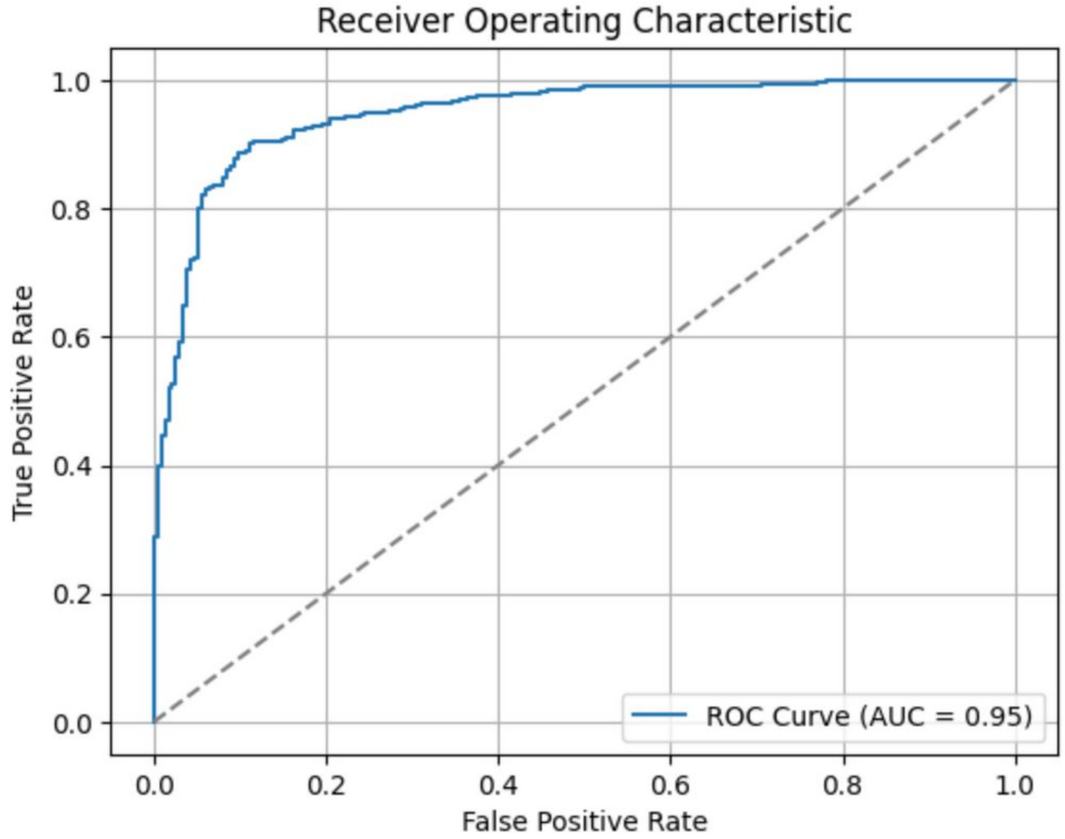
```
model.compile(
  optimizer=tf.keras.optimizers.Adam(1e-5),
  loss='binary_crossentropy',
  metrics=['accuracy']
)

history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=30,
  class_weight=class_weights,
  callbacks=[early_stop, reduce_lr]
)
```

Results (Kaggle Model)

```
# Confusion matrix
cm = confusion_matrix(y_true, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues'
```

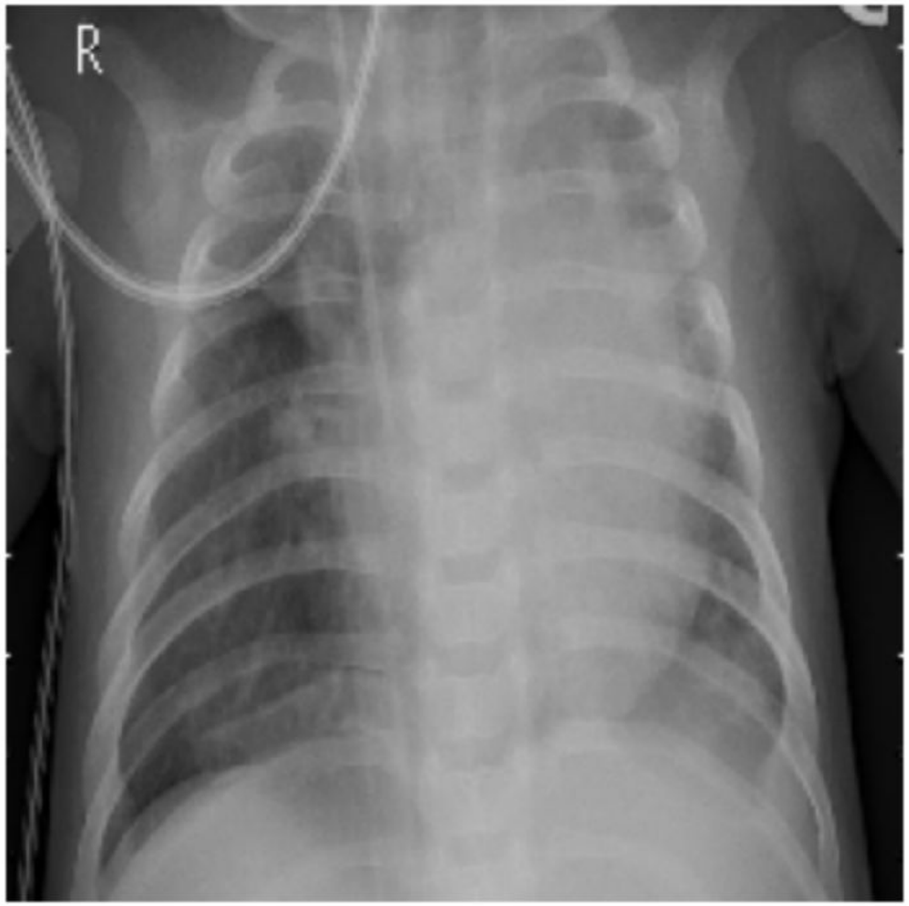



```
# ROC Curve
fpr, tpr, _ = roc_curve(y_true, y_probs)
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
```



Results (Kaggle Model)

Prediction: PNEUMONIA



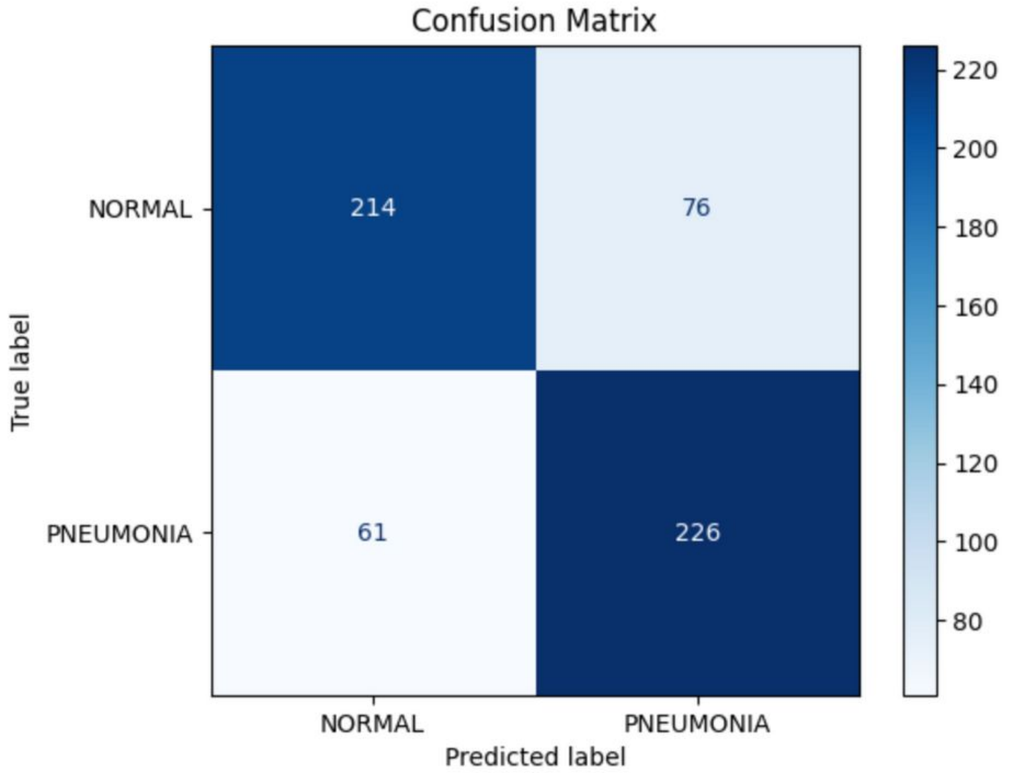
Grad-CAM

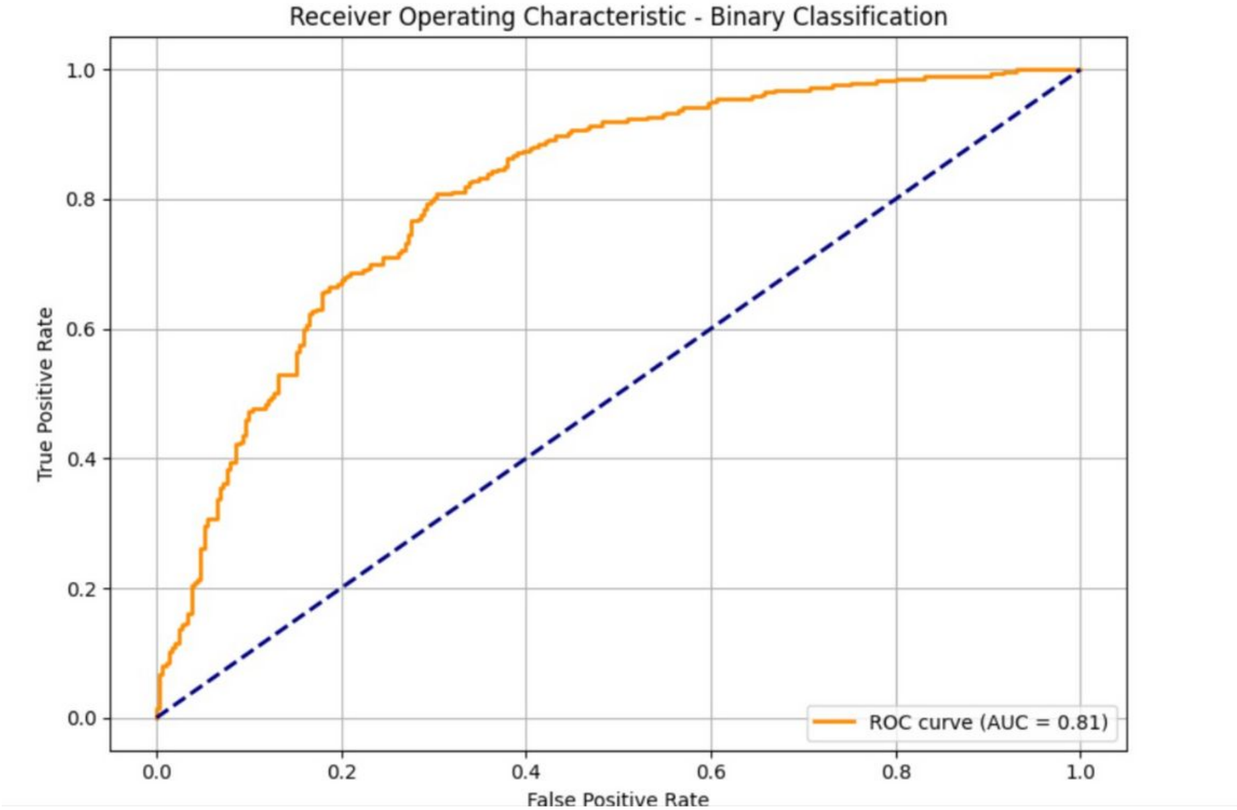
Results (Kaggle Model)

Classification Report

Class	Precision	Recall	F1-Score	Support
Normal	0.92	0.65	0.77	234
Pneumonia	0.82	0.97	0.89	393
Accuracy			**0.85**	627
Macro Avg	0.87	0.81	0.83	627
Weighted Avg	0.86	0.85	0.84	627

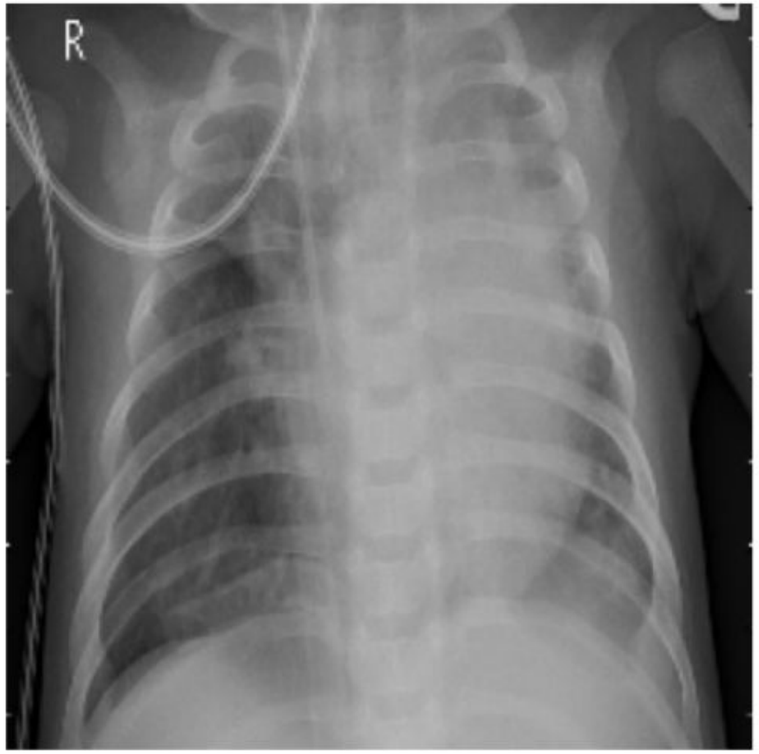
Results (NIH Model)



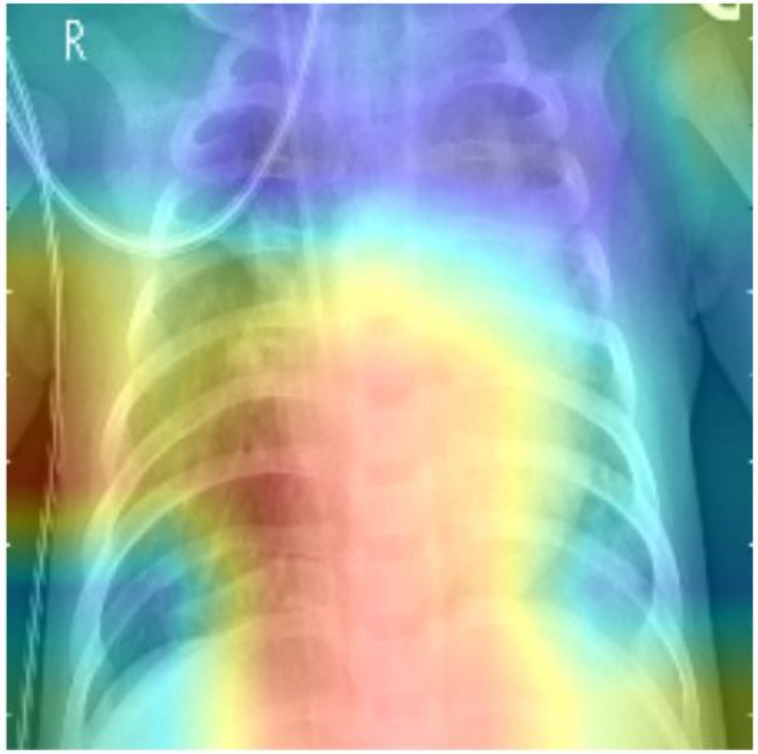


Results (NIH Model)

Original Image
Prediction: PNEUMONIA (score: 0.61)



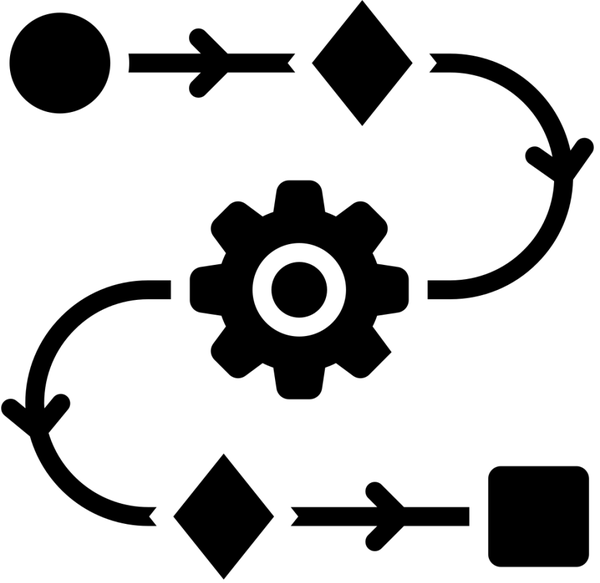
Grad-CAM Overlay



Results (NIH Model)

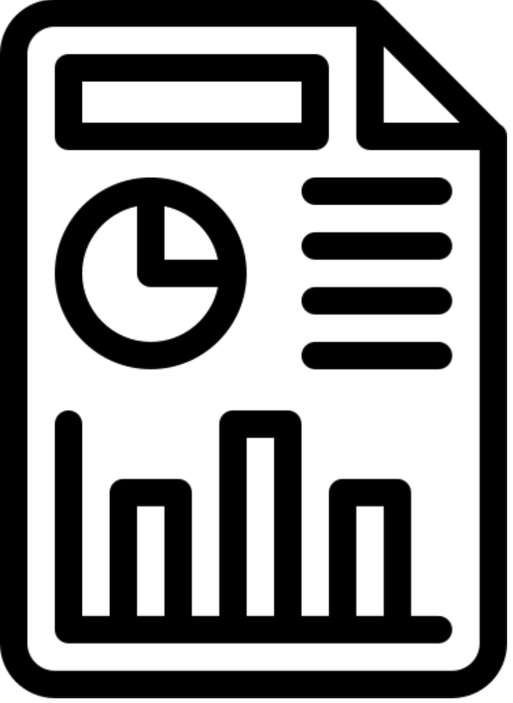
Classification report

Class	Precision	Recall	F1-Score	Support
NORMAL	0.78	0.74	0.76	290
PNEUMONIA	0.75	0.79	0.77	287
Accuracy			**0.76**	577
Macro Avg	0.76	0.76	0.76	577
Weighted Avg	0.76	0.76	0.76	577



Analysis & Future Work

- Class imbalance in NIH dataset (more pneumonia cases)
- Noisy labels from automated report extraction
- Lower recall for "Normal" class due to skewed training
- Tune prediction threshold using ROC/F1 to optimize clinical performance
- Implement multi-label classification (detect multiple conditions per image)
- Integrate Med-GEMMA for vision-language reasoning and report generation
- Expand dataset with CheXpert, MIMIC-CXR, RSNA for better generalization
- Fine-tune model on balanced datasets for improved reliability



Conclusion

- Used NIH and Kaggle (Guangzhou) chest X-ray datasets
- Kaggle dataset (balanced) gave better performance than NIH (imbalanced, noisy)
- Achieved up to 85% accuracy on validation set
- Integrated Grad-CAM for model explainability
- Model focused on clinically relevant lung regions
- Shows potential for AI-assisted diagnosis in real-world use cases

Thank You For Watching and ask any questions