forbeginners.html

workshop.css

Building a ‘CNN’ to Detect Pneumonia in Chest X-rays {

[A Deep Learning Approach to Medical Image Classification and Localization]

}

Programming Language

forbeginners.html   workshop.css

Table Of ‘Contents’ {

}

Programming Language

01 {

[Introduction]

< We’re building an AI model to detect pneumonia in chest X-rays. It finds lung opacities, which are signs of infection, and highlights them automatically. We’re using a Convolutional Neural Network (CNN) model to help make diagnosis faster and more accurate.>

}

forbeginners.html workshop.css

Concepts < /1 > { Medical Imaging

< We're working with chest X-rays, a type of medical imaging used to view the lungs. These grayscale images show differences in tissue density, helping us identify abnormalities like pneumonia. >

}

Concepts < /2 > { CNN

< A CNN is a deep learning model designed to analyze images. It breaks the image into patterns and learns to detect features like lung opacities over time. >

}

Programming Language

02 {

[Problem]

< Pneumonia is a serious lung infection that can be hard to catch on X-rays, especially in **understaffed clinics** or by **overworked doctors**. Subtle lung opacities, poor image quality, and overlap with other conditions all make diagnosis challenging, leading to missed or delayed treatment.>

}

forbeginners.html workshop.css

Concepts < /1 > { Diagnosis Challenges

< Pneumonia often shows up as faint or irregular patterns on X-rays. These can vary a lot between patients, making it hard to catch consistently without specialized training or support tools. >

}

Concepts < /2 > { Undermanned Clinics

< Many healthcare systems, especially in rural or underfunded areas, lack enough trained radiologists. This leads to delays in diagnosis and increases the burden on already overworked staff. >

}

Programming Language

forbeginners.html workshop.css

03 {

[Solution]

< We built a Convolutional Neural Network (CNN) that analyzes chest X-rays and predicts whether pneumonia is present. If it is, the model also highlights the location of the infection using bounding boxes. This helps support faster and more consistent diagnoses, especially in clinics with limited staff. >

}

forbeginners.html   workshop.css

Concepts < /1 > { Bounding Boxes

}

< These are rectangles drawn around infected areas on the X-ray. Our model predicts them to show where pneumonia may be present. >

Concepts < /2 > { Two-Part Prediction

}

< The model both detects if pneumonia exists and locates it – combining classification and localization in one step. >

What Our Code Does {

< Our code loads chest X-ray images and labeled bounding boxes, processes them, and trains a convolutional neural network (CNN). The CNN learns to both classify pneumonia and predict where it appears in the lungs. The workflow moves from loading data, to training, to testing, and ends with visualizing predictions. >

}

Step-by Step Using the Code {

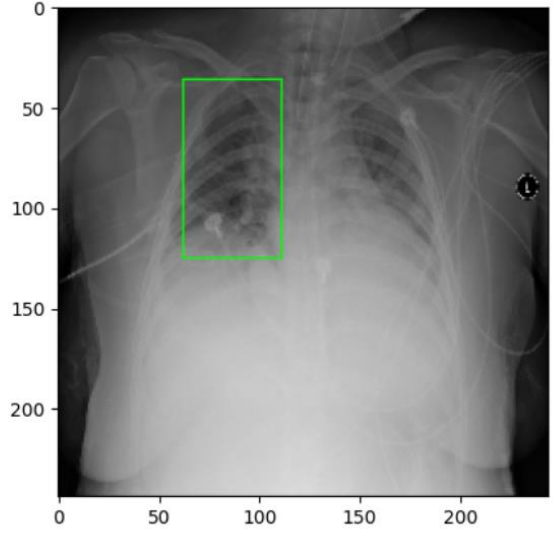
1. Import Libraries:
Import TensorFlow, pydicom, OpenCV, and other packages.
2. Load data:
Read the label CSV and link patient IDs to DICOM images.
3. Preprocess images:
Resize to 244×244, normalize pixel values, and adjust boxes.
4. Split dataset:
Divide the data into training, validation, and test sets.
5. Build model:
Create a CNN with outputs for classification and bounding boxes.
6. Train model:
Train for 100 epochs while tracking accuracy and loss.
7. Evaluate model:
Test the model and calculate accuracy and IoU scores.
8. Visualize predictions:
Plot sample images with predicted and true bounding boxes.

}

Model Architecture; {

< Our CNN takes chest X-ray images and passes them through layers that detect key visual features. The model has two outputs: one for predicting pneumonia presence and another for predicting a bounding box around affected areas. >

}



< Sample prediction: pneumonia detected and localized. >

Dataset Breakdown; {

< The full RSNA dataset contains over 30,000 labeled X-ray images. For this project, we used a subset of **6,500 images** for training, validation, and testing, selected to balance performance with training time and available resources.

Dataset Split:

Training: 5,200 | **Validation:** 700 | **Testing:** 601

>

}

Model Parameters; {

< These are the settings we used to train our model:

- **Batch size:** 32 images processed at a time during training.
- **Input size:** 244 × 244 pixels to standardize all X-ray images.
- **Epochs:** 100 full passes through the training set to help the model learn patterns.
- **Loss functions:**
  - **Classification loss** to decide if pneumonia is there.
  - **Bounding box loss** to find and outline the infected area.
- **Optimizer:** Adam, to adjust the model quickly and efficiently during training.

>

}

Confusion Matrix; {

< We coded a confusion matrix:

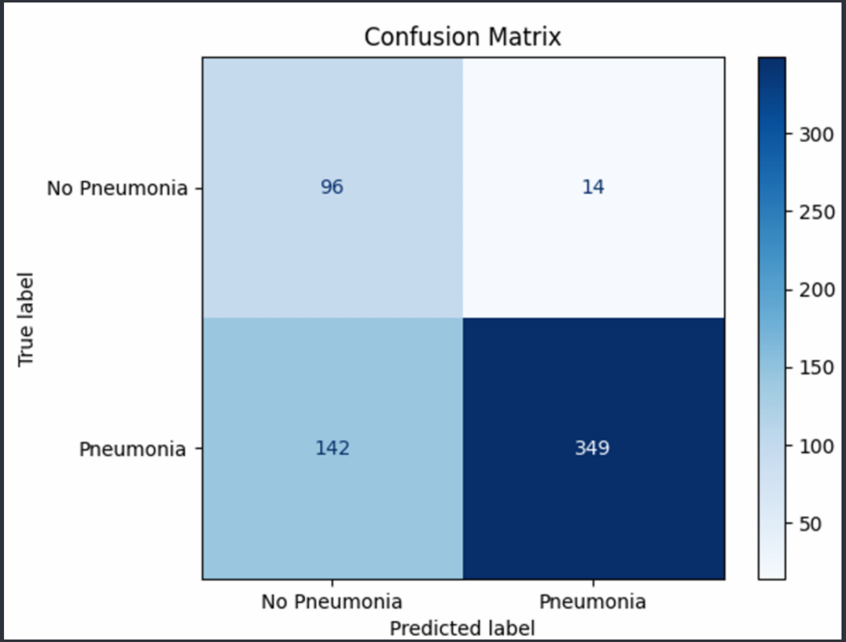
- TP: 349
- FP: 14
- FN: 142
- TN: 96

From the matrix, we calculated:

- Recall: 0.7108
- Precision: 0.9614
- F1 Score: 0.8173
- Accuracy: 0.7404

>

}



Performance Metrics; {

< We evaluated the model using a confusion matrix for categorization and Intersection over Union (IoU) on the test set.

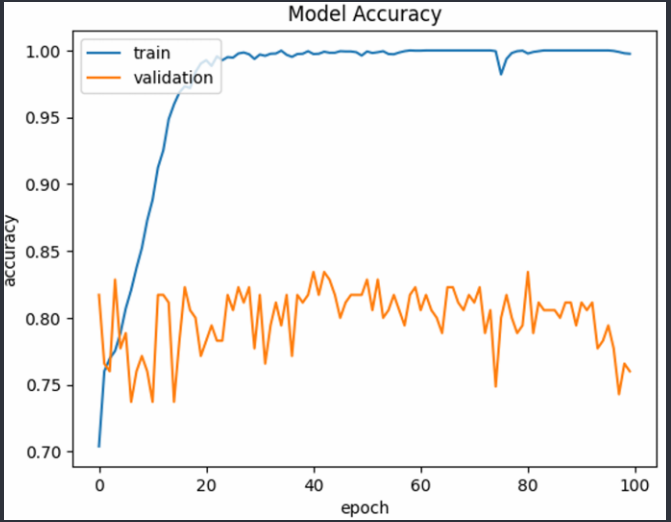
- **Test data size:** 200 images
- **Accuracy:** 75.5%
- **Mean IoU:** 0.0198

IoU values near 0 indicate poor alignment between predicted and actual boxes.

The model classified pneumonia correctly in many cases but struggled to localize the exact infected area, as shown by the low IoU scores. >

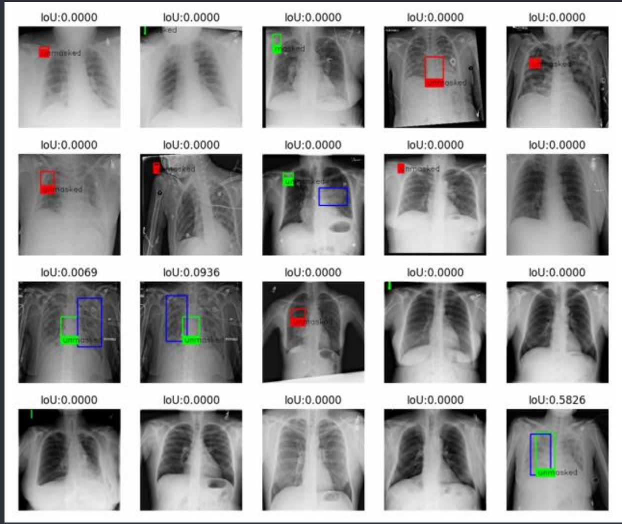
}

Visual Results; {



< Training accuracy increased over time on the training data only and does not reflect real-world performance. >

}



< Predicted boxes (green/blue) often missed the actual infection areas (red), leading to low IoU scores.>

forbeginners.html

workshop.css

Conclusion; {

< Our CNN model was able to classify pneumonia in chest X-rays with reasonable accuracy, reaching 75.5% on the test set. However, accurately localizing pneumonia with bounding boxes proved much more difficult, as shown by low IoU scores. While the model successfully learned key patterns during training, real-world performance still leaves room for improvement. Future work could focus on better bounding box prediction and using more advanced architectures or larger datasets. >

}