

LOCALLY ROOTED,
GLOBALLY RESPECTED

Traditional Machine Learning for TB Detection on Chest X-rays

By Group 5:

Anders Emmanuel Tan

Evan Razzan Adytaputra

Indratanaya Budiman

Daffa Maulana Siddiq

A Digital Image Processing Approach
based on the AESFERM Pipeline

ugm.ac.id

INTRODUCTION

Problem, Deep Learning, and
Alternative Approach



Why TB Detection Matters



UNIVERSITAS
GADJAH MADA

- TB is still a leading infectious killer worldwide.
- Chest X-ray is cheap and widely used for screening.
- Reading X-rays is slow, subjective, and depends on expert radiologists.

Existing Methods

Deep Learning is Powerful but Heavy

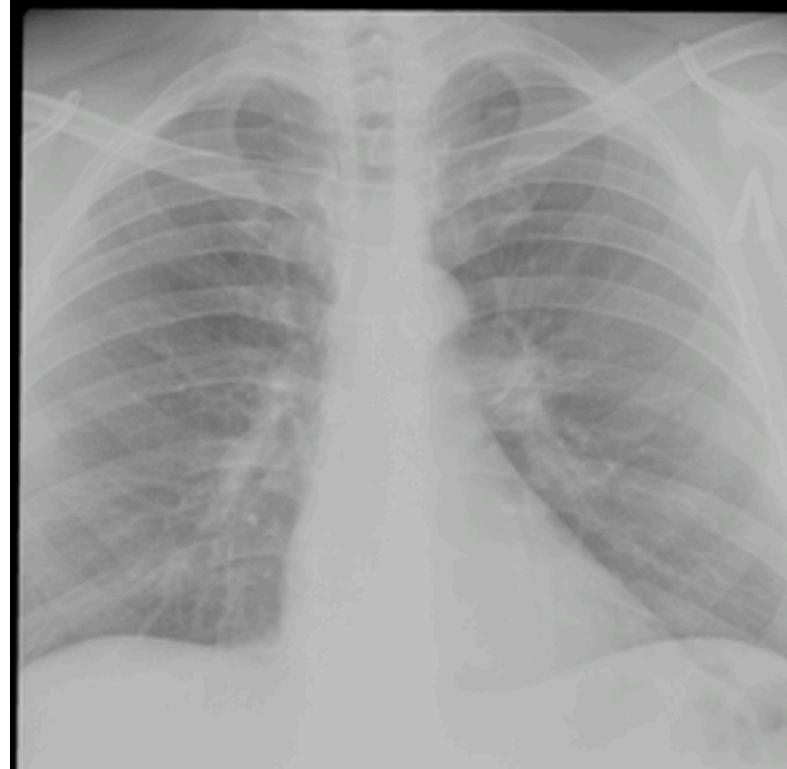
- CNN-based models for TB detection reach high accuracy in papers.
- But they need large labeled datasets, GPUs, and long training time.
- Hard to deploy in low-resource hospitals.

Alternative Method

Lightweight Traditional ML

- Use classical Digital Image Processing + handcrafted features.
- Train traditional ML (no deep learning).
- Goal: A lighter system that still achieves high accuracy on TB vs Normal.

Dataset & Sample CXRs

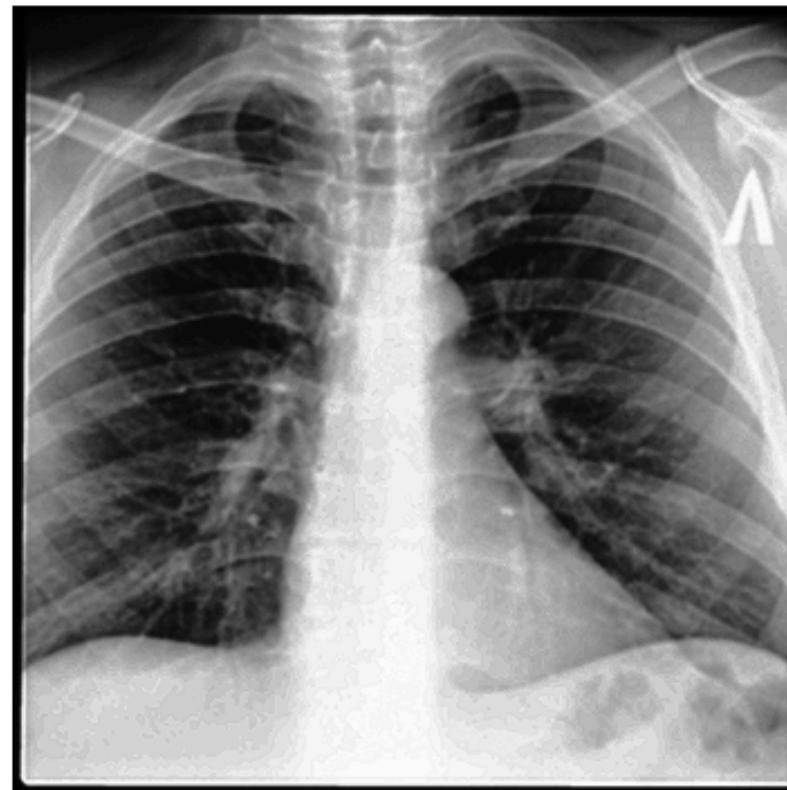


TBC



Normal

- Kaggle TB Chest X-ray dataset.
- 4,200 images: 3,500 Normal, 700 TB.
- All resized to 512×512 and converted to grayscale.



AESFERM

PIPELINE



AESFERM Overview



UNIVERSITAS
GADJAH MADA

Acquisition → Enhancement →
Segmentation & Morphology →
Feature Extraction →
Feature Representation →
Machine Learning

- We designed a step-by-step pipeline.
- Each step transforms the image into something more “ML-friendly”.

Acquisition

- Load PNG CXRs from Kaggle.
- Resize to 512×512 .
- Convert to grayscale,
standardize format.

Enhancement

- Remove noise, emphasize edges,
improve contrast.
- We use: Gaussian smoothing →
Laplacian → Histogram
Equalization.

Segmentation

- Apply Otsu thresholding to
separate lungs from background.
- Invert so lung becomes foreground
(white), background black.

- Morphology: Opening & Closing
- Connected Components: Isolate the
Lungs
- Final ROI: Lungs Only



AESFERM

Feature Extraction

- LBP: small-scale local texture patterns.
- GLCM: texture co-occurrence (smooth vs mottled).
- HOG: edge and shape distribution.

Representation

- Combine feature vectors:
 - GLCM only, HOG only, LBP only, and all combinations.

Matching

- Train a traditional classifier [Logistic Regression] to classify TB vs Normal.

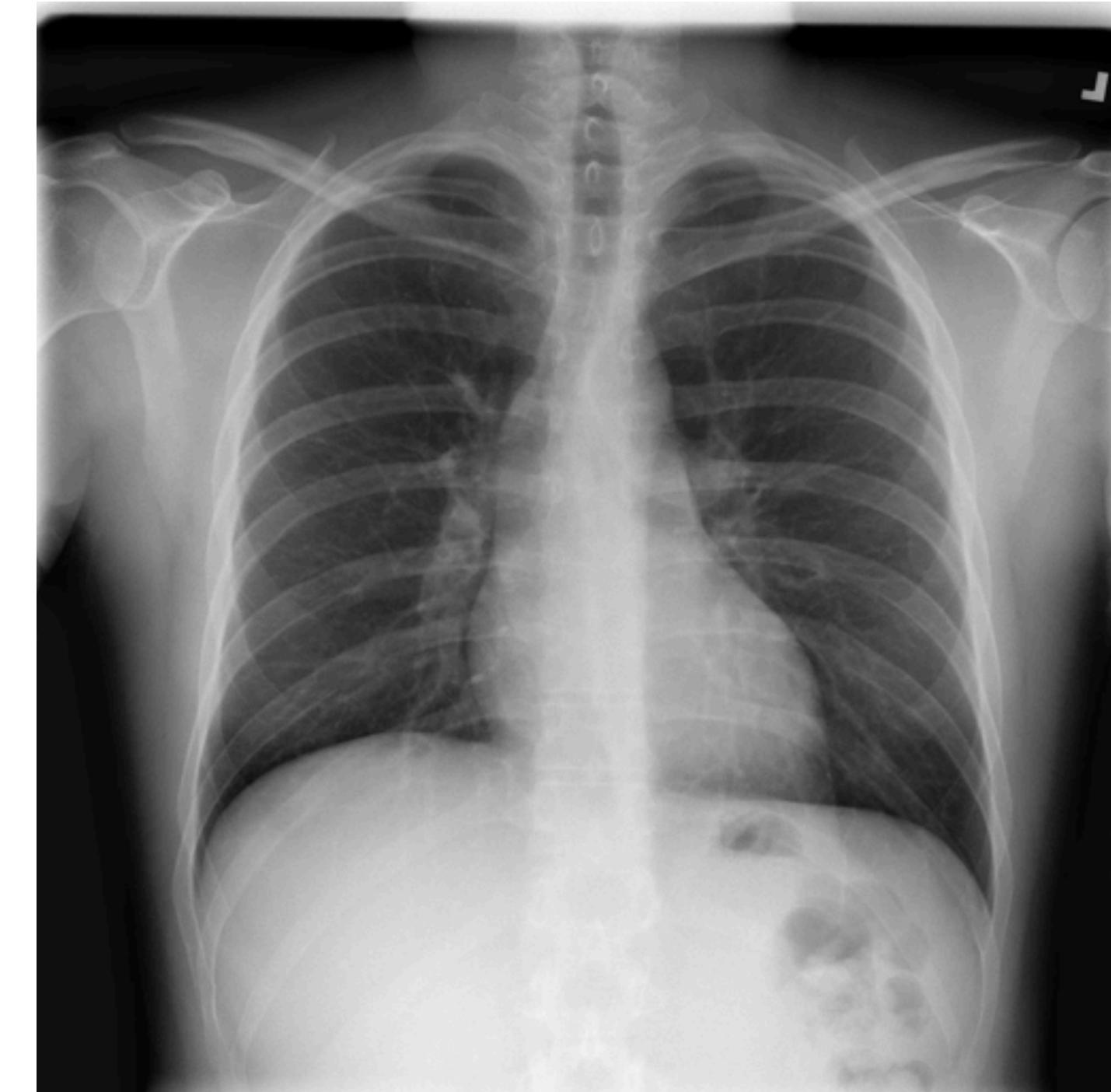


AESFERM

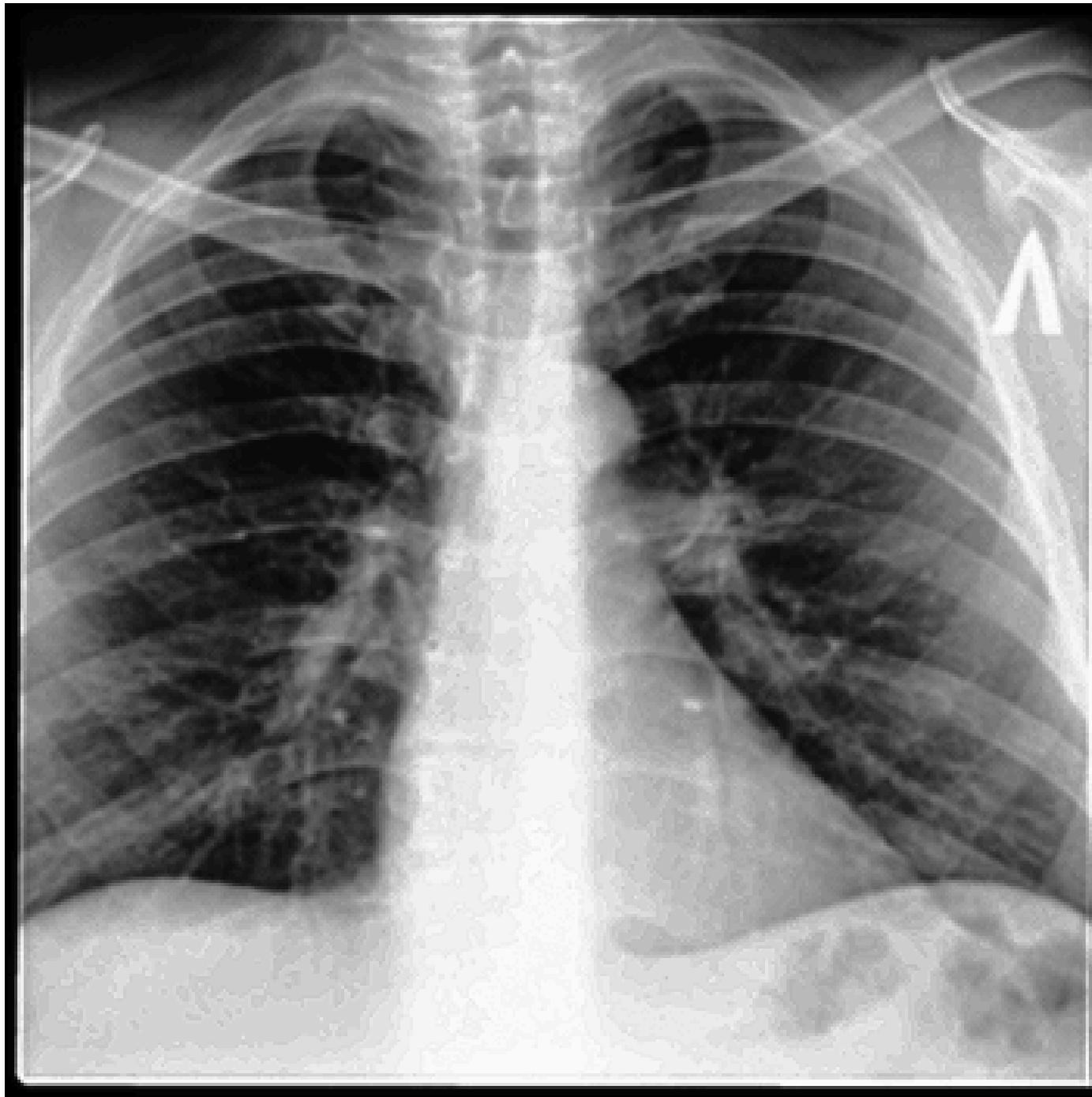


UNIVERSITAS
GADJAH MADA

PROCESS VISUALISATION AESFERM PIPELINE



TBC



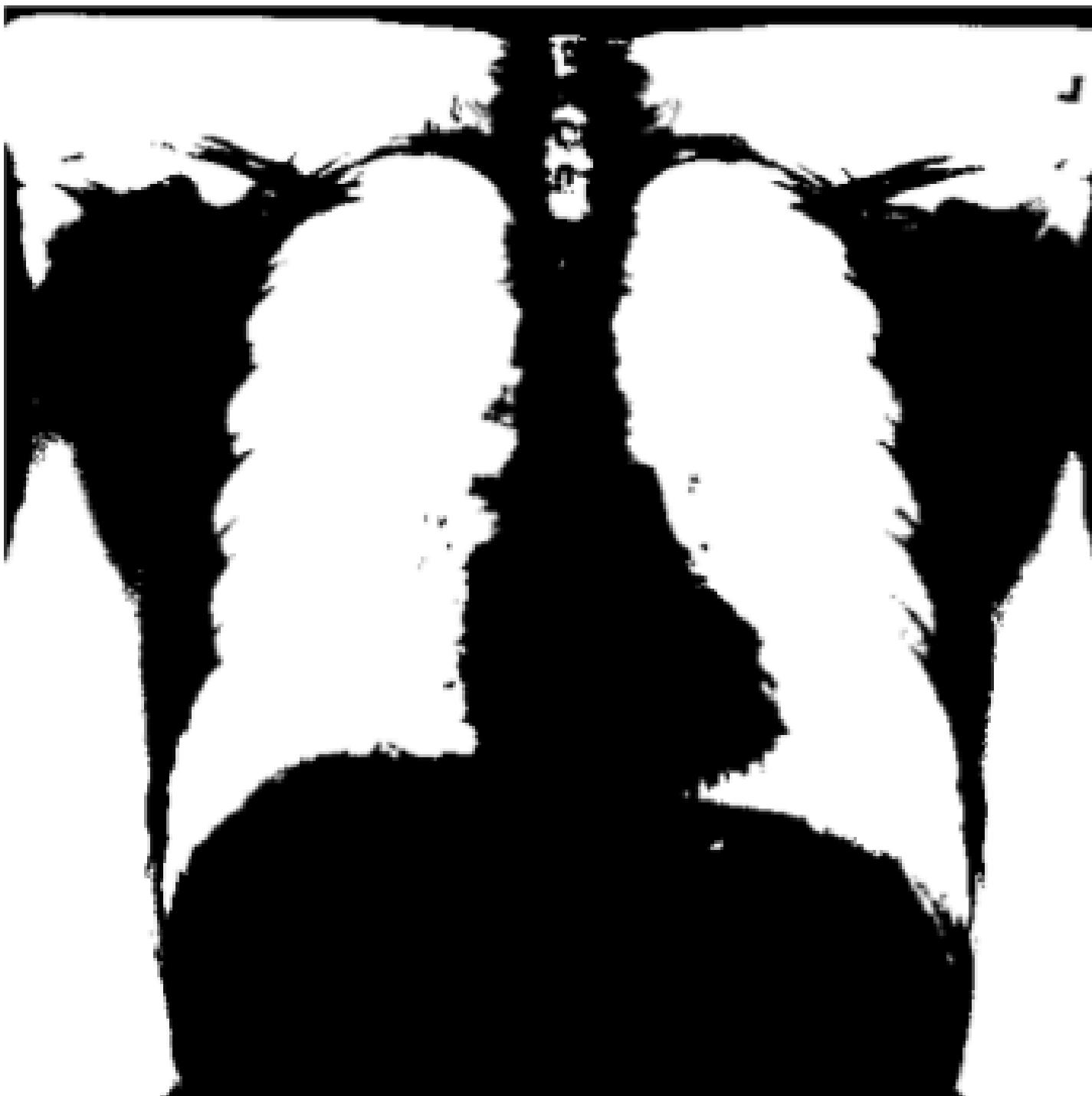
Normal

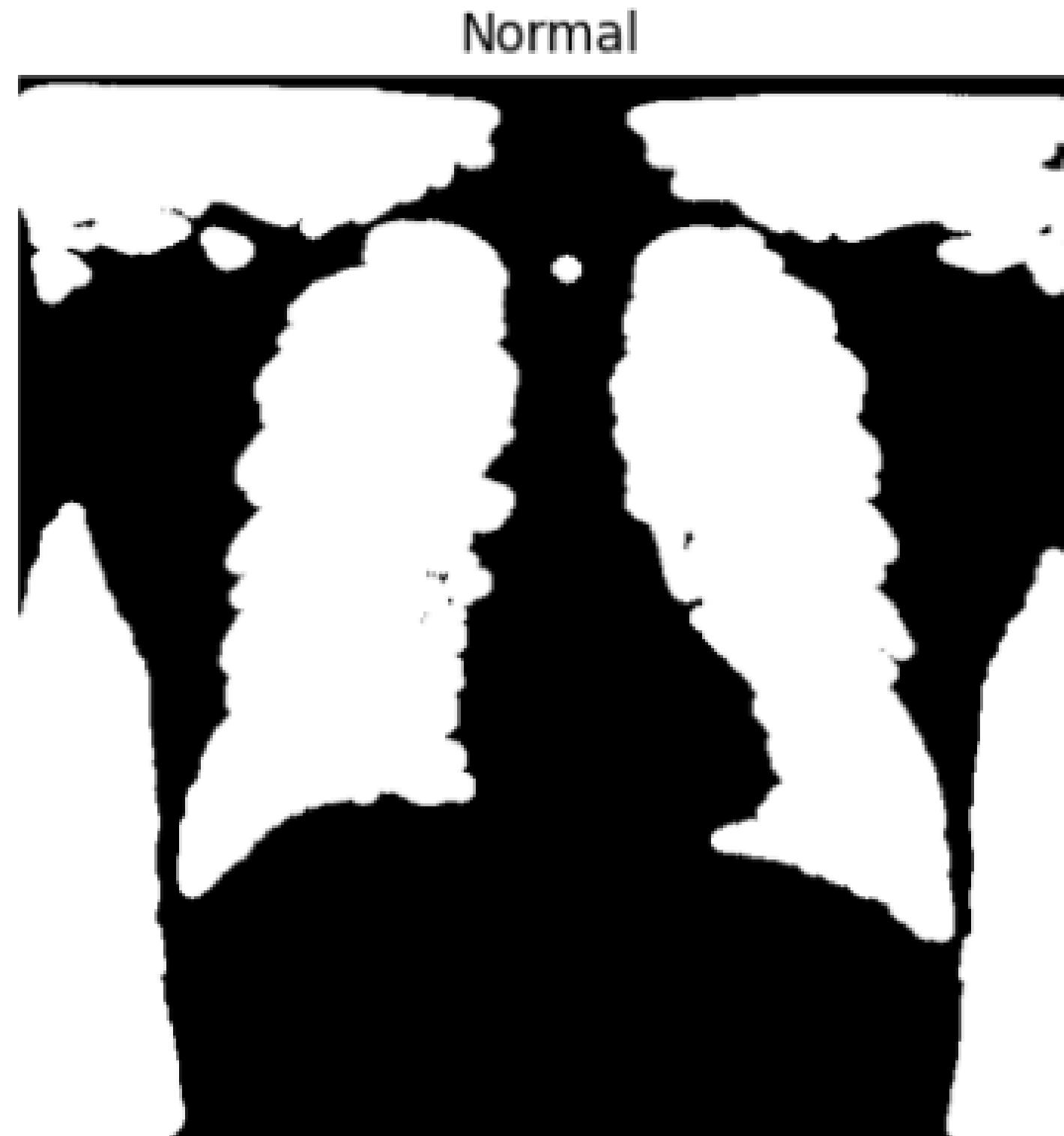
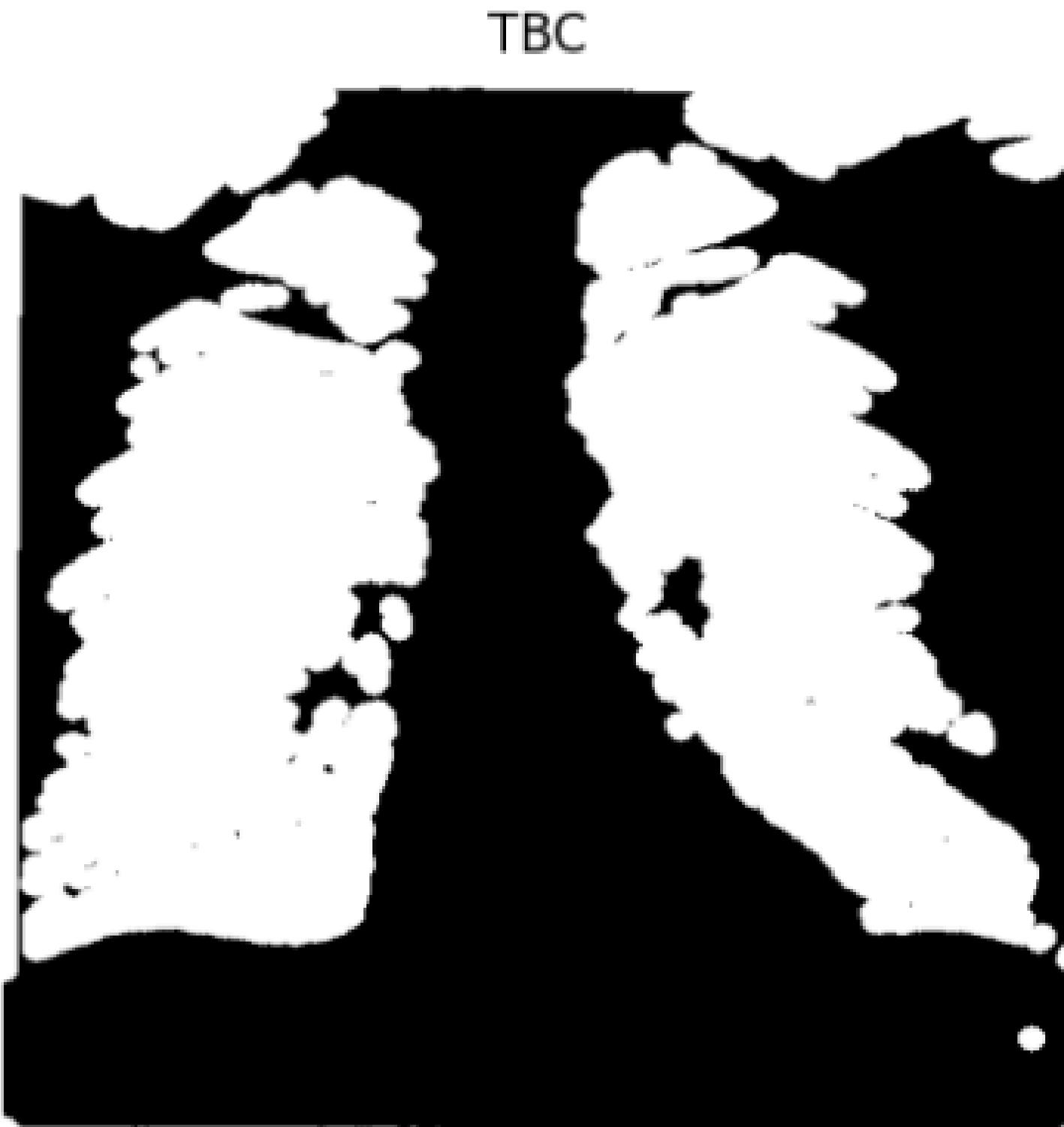


TBC



Normal





TBC



Normal



TBC



Normal





```

#Extract LBP Feature
def extract_lbp(image, P=8, R=1):
    lbp = feature.local_binary_pattern(image, P, R, method="uniform")
    # Histogram (59 bins for uniform LBP)
    n_bins = P + 2
    hist, _ = np.histogram(lbp.ravel(), bins=n_bins, range=(0, n_bins))
    hist = hist.astype("float")
    hist /= hist.sum() # normalize
    return hist

#Extract GLCM Feature
def extract_glcm(image):
    image = img_as_ubyte(image) #ensure image is in uint8 format
    distances = [1, 2, 3] #takes texture from 3x3 neighborhood
    angles = [0, np.pi/4, np.pi/2, 3*np.pi/4]
    glcm = feature.graycomatrix(image,
                                 distances=distances,
                                 angles=angles,
                                 levels=256,
                                 symmetric=True,
                                 normed=True)
    features = []
    #Calculate these Haralick features (might add/remove some later)
    props = ['ASM', 'variance', 'contrast', 'dissimilarity', 'homogeneity', 'energy']
    for prop in props: #Extract Values for each feature
        values = feature.grayscaleprops(glcm, prop)
        features.extend(values)

    return np.array(features) #Return as an numpy array

#Extract HOG Features
def extract_hog(image):
    #Calculated Histogram of Oriented Gradients
    hog = skimage.feature.hog(image, orientations=9,
                             pixels_per_cell=(8, 8),
                             cells_per_block=(2,2),
                             block_norm='L2-Hys',
                             transform_sqrt = True,
                             feature_vector=True)
    return hog

#Use all prev functions to get the final vector
def extract_features(image):
    lbp_features = extract_lbp(image)
    glcm_features = extract_glcm(image)
    hog_features = extract_hog(image)

    final_vector = np.concatenate((lbp_features, glcm_features, hog_features)) #Combine the vectors
    return final_vector

def extract_glcml_hog(image):
    #Use all prev functions to get the final vector
    lbp_features = extract_lbp(image)
    glcm_features = extract_glcm(image)
    hog_features = extract_hog(image)

    final_vector = np.concatenate((lbp_features, glcm_features, hog_features)) #Combine the vectors
    return final_vector

```

TBC Features: (142990,)
Normal Features: (142990,)

TBC Features: [0.03556442 0.03373337 0.02246475 ... 0. 0. 0.]
Normal Features: [0.01786423 0.01780319 0.01591492 ... 0. 0. 0.]



```
Normal directory exists: True
Tuberculosis directory exists: True
Found 3500 images in Normal.
Processed Normal-1.png (1/3500)
Processed Normal-10.png (2/3500)
Processed Normal-100.png (3/3500)
Processed Normal-1000.png (4/3500)
Processed Normal-1001.png (5/3500)
Processed Normal-1002.png (6/3500)
Processed Normal-1003.png (7/3500)
Processed Normal-1004.png (8/3500)
Processed Normal-1005.png (9/3500)
Processed Normal-1006.png (10/3500)
Processed Normal-1007.png (11/3500)
Processed Normal-1008.png (12/3500)
Processed Normal-1009.png (13/3500)
Processed Normal-101.png (14/3500)
Processed Normal-1010.png (15/3500)
Processed Normal-1011.png (16/3500)
Processed Normal-1012.png (17/3500)
Processed Normal-1013.png (18/3500)
Processed Normal-1014.png (19/3500)
Processed Normal-1015.png (20/3500)
Processed Normal-1016.png (21/3500)
Processed Normal-1017.png (22/3500)
Processed Normal-1018.png (23/3500)
Processed Normal-1019.png (24/3500)
Processed Normal-102.png (25/3500)
Processed Normal-1020.png (26/3500)
Processed Normal-1021.png (27/3500)
Processed Normal-1022.png (28/3500)
Processed Normal-1023.png (29/3500)
Processed Normal-1024.png (30/3500)
Processed Normal-1025.png (31/3500)
Processed Normal-1026.png (32/3500)
Processed Normal-1027.png (33/3500)
Processed Normal-1028.png (34/3500)
Processed Normal-1029.png (35/3500)
Processed Normal-103.png (36/3500)
Processed Normal-1030.png (37/3500)
Processed Normal-1031.png (38/3500)
Processed Normal-1032.png (39/3500)
Processed Normal-1033.png (40/3500)

Processed Normal-1104.png (119/3500)
Processed Normal-1105.png (120/3500)
Processed Normal-1106.png (121/3500)
Processed Normal-1107.png (122/3500)
Processed Normal-1108.png (123/3500)
Processed Normal-1109.png (124/3500)
Processed Normal-111.png (125/3500)
Processed Normal-1110.png (126/3500)
Processed Normal-1111.png (127/3500)
Processed Normal-1112.png (128/3500)
Processed Normal-1113.png (129/3500)
Processed Normal-1114.png (130/3500)
Processed Normal-1115.png (131/3500)
Processed Normal-1116.png (132/3500)
Processed Normal-1117.png (133/3500)
Processed Normal-1118.png (134/3500)
Processed Normal-1119.png (135/3500)
Processed Normal-1120.png (136/3500)
Processed Normal-1121.png (137/3500)
Processed Normal-1122.png (138/3500)
Processed Normal-1123.png (139/3500)
Processed Normal-1124.png (140/3500)
Processed Normal-1125.png (141/3500)
Processed Normal-1126.png (142/3500)
Processed Normal-1127.png (143/3500)
Processed Normal-1128.png (144/3500)
Processed Normal-1129.png (145/3500)
Processed Normal-1130.png (146/3500)
Processed Normal-1131.png (147/3500)
Processed Normal-1132.png (148/3500)
Processed Normal-1133.png (149/3500)
Processed Normal-1134.png (150/3500)
Processed Normal-1135.png (151/3500)
Processed Normal-1136.png (152/3500)
Processed Normal-1137.png (153/3500)
Processed Normal-1138.png (154/3500)
Processed Normal-1139.png (155/3500)
Processed Normal-1140.png (156/3500)

Processed Tuberculosis-598.png (554/700)
Processed Tuberculosis-599.png (555/700)
Processed Tuberculosis-6.png (556/700)
Processed Tuberculosis-60.png (557/700)
Processed Tuberculosis-600.png (558/700)
Processed Tuberculosis-601.png (559/700)
Processed Tuberculosis-602.png (560/700)
Processed Tuberculosis-603.png (561/700)
Processed Tuberculosis-604.png (562/700)
Processed Tuberculosis-605.png (563/700)
Processed Tuberculosis-606.png (564/700)
Processed Tuberculosis-607.png (565/700)
Processed Tuberculosis-608.png (566/700)
Processed Tuberculosis-609.png (567/700)
Processed Tuberculosis-61.png (568/700)
Processed Tuberculosis-610.png (569/700)
Processed Tuberculosis-611.png (570/700)
Processed Tuberculosis-612.png (571/700)
Processed Tuberculosis-613.png (572/700)
Processed Tuberculosis-614.png (573/700)
Processed Tuberculosis-615.png (574/700)
Processed Tuberculosis-616.png (575/700)
Processed Tuberculosis-617.png (576/700)
Processed Tuberculosis-618.png (577/700)
Processed Tuberculosis-619.png (578/700)
Processed Tuberculosis-620.png (579/700)
Processed Tuberculosis-621.png (580/700)
Processed Tuberculosis-622.png (581/700)
Processed Tuberculosis-623.png (582/700)
Processed Tuberculosis-624.png (583/700)
Processed Tuberculosis-625.png (584/700)
Processed Tuberculosis-626.png (585/700)
Processed Tuberculosis-627.png (586/700)
Processed Tuberculosis-628.png (587/700)
Processed Tuberculosis-629.png (588/700)
Processed Tuberculosis-63.png (589/700)
Processed Tuberculosis-630.png (590/700)
Processed Tuberculosis-631.png (591/700)
Processed Tuberculosis-632.png (592/700)
Processed Tuberculosis-633.png (593/700)
Processed Tuberculosis-634.png (594/700)
Processed Tuberculosis-635.png (595/700)
Processed Tuberculosis-636.png (596/700)
Processed Tuberculosis-637.png (597/700)
Processed Tuberculosis-638.png (598/700)
Processed Tuberculosis-639.png (599/700)
Processed Tuberculosis-640.png (600/700)
Processed Tuberculosis-641.png (601/700)
Processed Tuberculosis-642.png (602/700)
Processed Tuberculosis-643.png (603/700)
Processed Tuberculosis-644.png (604/700)
Processed Tuberculosis-645.png (605/700)
Processed Tuberculosis-646.png (606/700)
Processed Tuberculosis-647.png (607/700)
Processed Tuberculosis-648.png (608/700)
Processed Tuberculosis-649.png (609/700)
Processed Tuberculosis-65.png (610/700)
Processed Tuberculosis-650.png (611/700)
Processed Tuberculosis-651.png (612/700)
Processed Tuberculosis-652.png (613/700)
Processed Tuberculosis-653.png (614/700)
Processed Tuberculosis-654.png (615/700)
Processed Tuberculosis-655.png (616/700)
Processed Tuberculosis-656.png (617/700)
Processed Tuberculosis-657.png (618/700)
Processed Tuberculosis-658.png (619/700)
Processed Tuberculosis-659.png (620/700)
Processed Tuberculosis-660.png (621/700)
Processed Tuberculosis-661.png (622/700)
Processed Tuberculosis-662.png (623/700)
Processed Tuberculosis-663.png (624/700)
Processed Tuberculosis-664.png (625/700)
Processed Tuberculosis-665.png (626/700)
Processed Tuberculosis-666.png (627/700)
Processed Tuberculosis-667.png (628/700)
Processed Tuberculosis-668.png (629/700)
Processed Tuberculosis-669.png (630/700)
Processed Tuberculosis-67.png (631/700)
Processed Tuberculosis-670.png (632/700)
Processed Tuberculosis-671.png (633/700)
Processed Tuberculosis-672.png (634/700)
Processed Tuberculosis-673.png (635/700)
Processed Tuberculosis-674.png (636/700)
Processed Tuberculosis-675.png (637/700)
Processed Tuberculosis-676.png (638/700)
Processed Tuberculosis-677.png (639/700)
Processed Tuberculosis-678.png (640/700)
Processed Tuberculosis-679.png (641/700)
```

Accuracy: 0.969047619047619
Model saved as model.pkl

RESULTS

Findings on Best Combinations
and Future Work



Feature Comparison

Performance of Different Feature Sets

Method	Accuracy	Precision	F1-Score	Sensitivity	Specificity
HOG	93.33%	0.824	0.7863	0.7518	0.9687
LBP	83.69%	0	0	0	1
GLCM	95.23%	0.876	0.8496	0.8248	0.9772
HOG + LBP	93.33%	0.8189	0.7879	0.7591	0.9673
GLCM + LBP	95.59%	0.8906	0.8604	0.8321	0.9801
GLCM + HOG	95.23%	0.876	0.8496	0.8248	0.9772
GLCM + HOG + LBP	96.90%	0.9237	0.903	0.8832	0.9858

Result on real test set

	Prediction	Confidence	Ground truth
0	TBC	0.949361	TBC
1	TBC	0.969892	TBC
2	TBC	0.999999	TBC
3	TBC	0.992888	TBC
4	TBC	0.999688	TBC
5	TBC	0.880674	TBC
6	TBC	1.000000	TBC
7	TBC	1.000000	TBC
8	TBC	0.962212	TBC
9	TBC	1.000000	TBC
correct: 393			
incorrect: 7			
total score: 0.9825			
Confidence average: 0.9799214146508848			

Best Combination :

GLCM + HOG + LBP

- GLCM: captures global texture patterns of lung parenchyma.
- HOG: captures edges, cavities, fibrosis lines typical in TB.
- LBP: adds some fine local variations, but only useful when combined.
- Combined GLCM + LBP + HOG reaches ~97% accuracy with high TB sensitivity.

Limitations & Future Work

- Trained and tested on a single public dataset
→ need external validation.
- Traditional features may be sensitive to changes in scanner, exposure, or demographics.
- Future directions:
 - Cross-dataset testing (different hospitals).
 - Expansion on various CXR-based lung diseases detection

Takeways

- We built a full AESFERM pipeline: from raw CXRs to TB vs Normal.
- With only traditional DIP + ML, we reached ≈97% accuracy.
- This suggests that in low-resource settings, well-designed classic methods can still be very competitive.

Thank You!

This work demanded our full effort,
and we hope it helps push TB
screening a little further, and stops a
few cases before they start.



One step at a time.
One X-Ray at a time.

Credits

group 5

Anders Emmanuel Tan → Image Processing Code, Report, Editing, and Presentation

Evan Razzan Adytaputra → Machine Learning/Model Evaluation, Report and
Presentation

Indratanaya Budiman → Frontend Flask Code

Daffa M. Siddiq → Report and Presentation

made and presented by
Daffa, Anders, and Evan

LOCALLY ROOTED, GLOBALLY RESPECTED