**Mahesh Reddy D**

**Technical Report Chest X-Ray Clustering (Pneumonia vs Normal)**  
**Dataset:** Chest X-Ray Images (Pneumonia), Kaggle  
 **1. Executive summary**

This project builds an unsupervised clustering pipeline for chest X-ray images to explore natural groupings in the data and evaluate whether visual features separate NORMAL and PNEUMONIA cases. The pipeline extracts intensity, shape, texture and invariant moment features, normalizes them, reduces dimensionality with PCA, and applies three clustering algorithms (K-Means, Agglomerative, DBSCAN). Evaluation uses t-SNE/PCA visualizations and intrinsic clustering metrics (Silhouette Score and Davies-Bouldin Index). All code and outputs are saved under clustering\_outputs/.

**2. Data & preprocessing**

**Source:** Kaggle *Chest X-Ray Images (Pneumonia)*.  
**Files used:** All images under the chest\_xray folder (train, val, test combined).  
**Number of images processed:** **11,712**. *(Insert your exact number if different.)*

**Preprocessing steps**

* Read image (OpenCV)
* Convert to grayscale (if not already)
* Resize to **256×256** pixels (uniform dimensions for feature extraction)
* Normalize pixel scale when required (converted to uint8 0–255)

**3. Feature extraction descriptors, parameters, and output dimensions**

For each preprocessed image we computed four feature sub-vectors and concatenated them to form a single high-dimensional vector:

1. **Intensity Histogram (vhist)**
   * Method: global intensity histogram over the entire grayscale image
   * Parameters: **256 bins** (range 0–255)
   * Dimension: **256**
2. **Histogram of Oriented Gradients (vHOG)**
   * Method: HOG (skimage) on the full resized image
   * Parameters:
     + orientations = **9**
     + pixels\_per\_cell = **(8, 8)**
     + cells\_per\_block = **(2, 2)**
     + block\_norm = L2-Hys
   * Dimension: **~34,596** (depends on image size and HOG configuration).
     + In this run the *full feature vector* dimension was **34,918**, and removing the other sub-vectors (256 + 59 + 7 = 322) leaves **34596** HOG features.
3. **Local Binary Patterns (vLBP)**
   * Method: uniform LBP (skimage local\_binary\_pattern), histogram of the LBP map
   * Parameters: **P = 8** (neighbors), **R = 1** (radius), **bins = 59** (uniform patterns)
   * Dimension: **59**
4. **Hu Moments (vHu)**
   * Method: 7 Hu invariant moments (log-transformed) from image moments via OpenCV
   * Parameters: standard 7 Hu moments; log-scaling to reduce dynamic range
   * Dimension: **7**

**Concatenated feature vector**  
v = [vhist (256) , vHOG (~34596) , vLBP (59) , vHu (7)] → **Total dimension ≈ 34,918**.

Note: HOG dominates the dimensionality. This is expected with the chosen HOG granularity (8×8 cells on 256×256 images). You can reduce dimensionality/time by increasing cell size (e.g., 16×16) or lowering image resolution (e.g., 128×128).

**4. Feature normalization & dimensionality reduction**

* **Normalization:** StandardScaler applied to the full feature matrix (zero mean, unit variance) so that each feature dimension contributes comparably to Euclidean distances used by clustering algorithms.
* **Dimensionality reduction for clustering:** PCA to **100 components** (PCA\_N\_COMPONENTS\_FOR\_CLUSTERING = 100) was applied before clustering. This reduces noise, speeds clustering, and helps density methods like DBSCAN work better in lower-dimensional space.
* **Dimensionality reduction for visualization:** PCA to **50 components** followed by **t-SNE (2D)** for plotting (perplexity = 30).

**5. Similarity / distance function justification**

**Chosen metric:** **Euclidean distance** (L2) between normalized feature vectors.

**Justification**

* Euclidean distance is compatible with K-Means (which optimizes within-cluster squared Euclidean distance) and with Ward linkage used for Agglomerative clustering.
* After StandardScaler, Euclidean distance treats each standardized feature equally, making comparisons across heterogeneous features (histogram bins, HOG coefficients, LBP bins, Hu moments) meaningful.
* For DBSCAN we also use Euclidean distance in PCA space (DBSCAN typically performs poorly in very high dimensions; reducing via PCA helps).
* Alternative metrics (cosine, Mahalanobis) could be explored, but Euclidean is a robust default when features are standardized.

**6. Clustering methods, parameter selection & results**

**General workflow**

1. Reduce features with PCA to 100 components for clustering.
2. Use elbow and silhouette analysis to choose K for K-Means.
3. Fit Agglomerative clustering (Ward linkage) with the chosen K.
4. Fit DBSCAN on the same PCA representation (experiment with eps and MinPts guided by a k-distance plot).

**K-Means**

* **K candidates tested:** K = 2..6 (range)
* **Selection method:** silhouette score used to pick the best K (higher = better internal separation). The elbow plot (inertia vs K) is saved for reference.
* **Chosen K (best by silhouette):** **best\_k = (insert best\_k here)** — *replace with the value reported by your run.*
* **Commands/outputs:** k\_elbow.png, k\_silhouette.png.

**Agglomerative (Hierarchical)**

* **Linkage:** **Ward** (minimizes variance within clusters and uses Euclidean distance)
* **Number of clusters:** set to the same K chosen for K-Means for direct comparison.
* **Dendrogram:** Because computing a full dendrogram on all ~11k samples is infeasible to read, a **sampled dendrogram** (e.g., 500 samples) is generated and saved as dendrogram\_sample.png.

**DBSCAN**

* **Space used:** PCA reduced space (100 components).
* **Parameters tested / selected:**
  + eps: **start with 2.5** (example); refine using k-distance plot (k\_distance\_plot.png).
  + MinPts (min\_samples): **5** (baseline).
* **Guidance:** Inspect the k-distance plot and identify the “knee” value as a candidate eps. DBSCAN often marks some points as noise (cluster = -1).
* **Files:** k\_distance\_plot.png shows the sorted k-th neighbor distances to help pick eps.

**7. Visualizations**

All 2D visualizations are saved in clustering\_outputs/. Key files:

* kmeans\_tsne.png — t-SNE (2D) colored by K-Means cluster labels
* agg\_tsne.png — t-SNE (2D) colored by Agglomerative cluster labels
* dbscan\_tsne.png — t-SNE (2D) colored by DBSCAN cluster labels
* true\_labels\_tsne.png — t-SNE (2D) colored by ground truth labels (NORMAL vs PNEUMONIA)
* dendrogram\_sample.png — truncated/sample dendrogram for hierarchical clustering
* k\_distance\_plot.png — k-distance plot used to choose DBSCAN eps
* cluster\_examples\_kmeans/ — example image thumbnails per K-Means cluster

Include these images in the report (figures) with short captions describing the color mapping.

**8. Quantitative evaluation metrics table**

Compute intrinsic metrics for each clustering method using the **PCA** space used for clustering. The script already calculates **Silhouette Score** (higher is better) and **Davies-Bouldin Index** (lower is better). Below is the table format replace the placeholders with your actual values from clustering\_outputs/clustering\_evaluation.csv.

| **Method** | **# Clusters** | **Silhouette Score** | **Davies-Bouldin Index** |
| --- | --- | --- | --- |
| K-Means | best\_k | sil\_kmeans | db\_kmeans |
| Agglomerative | best\_k | sil\_agg | db\_agg |
| DBSCAN | n\_clusters\_dbscan | sil\_dbscan | db\_dbscan |

**Notes on interpreting metrics**

* **Silhouette Score:** ranges from -1 to 1. Values near +1 indicate well-separated clusters; near 0 indicate overlapping clusters; negative values indicate misassignment.
* **Davies-Bouldin Index:** lower values are better; a small value indicates a good clustering structure.
* Because DBSCAN may return a variable number of clusters and some noise points, its silhouette may be computed only when at least 2 clusters are present.

**9. Sample results (how to fill this section)**

Insert the actual numbers from your run. For example:

* best\_k = 2 (if silhouette favored 2)
* K-Means silhouette = 0.12, DB index = 3.4
* Agglomerative silhouette = 0.10, DB index = 3.8
* DBSCAN found 3 clusters + noise; silhouette = 0.05, DB index = 4.2

*(Do not use these example numbers in the final report; replace with values from clustering\_outputs/clustering\_evaluation.csv.)*

**10. Discussion & interpretation**

**Quality of clusters**

* If **K = 2** performs best (by silhouette) and purity/confusion shows that one cluster is dominated by PNEUMONIA and the other by NORMAL, that suggests the chosen features capture useful shape/texture differences between infected and healthy lungs.
* If clusters do not align well with ground truth (low purity), possible causes:
  + Feature extraction parameters (HOG granularity, LBP radius) may not capture discriminative patterns for pneumonia.
  + Image variability (view angle, exposure, artifacts) may dominate features.
  + Pneumonia appearance is heterogeneous — unsupervised clusters might split dataset by other imaging factors (contrast, patient age, view).
* DBSCAN is sensitive to eps in PCA space; if it produces many noise points or trivial clusters, try adjusting eps or reducing PCA dims (e.g., 20–50) before running DBSCAN.

**Impact of feature choices**

* HOG carries the largest representational weight — lowering its resolution (larger cell sizes) reduces dimensionality and may reduce noise.
* LBP adds micro-texture cues useful for local texture differences (consolidation, opacities).
* Hu moments add global invariant descriptors but are only 7 dims; their influence is minor after scaling.

**11. Practical recommendations & next steps**

1. **Parameter tuning**
   * Run K-Means across a wider K range (e.g., 2–10) and inspect silhouette/elbow.
   * Evaluate DBSCAN across a grid of eps (guided by the k-distance plot) and min\_samples (e.g., 5–20).
2. **Feature engineering**
   * Try lower HOG resolution (pixels\_per\_cell = (16,16)) and smaller image size (128×128) to speed experiments and reduce HOG dimension.
   * Consider adding global contrast features (CLAHE histogram), Gabor filters, or CNN embeddings (pretrained ResNet/VGG feature vectors) for more semantic features.
3. **Use supervised checks**
   * Compute confusion matrix / per-cluster precision/recall vs true labels (script saves a contingency table).
   * Consider training a small supervised classifier on the extracted features to measure separability (for diagnostic purposes only — the assignment requires unsupervised work).
4. **Reporting**
   * Include sample X-rays from each cluster (script saves examples).
   * Include dendrogram (sampled) to show hierarchical relationships.

**12. Conclusion (guidance text choose the paragraph that matches your results)**

**If K-Means performed best:**

K-Means with **K = *X*** gave the most coherent clustering by silhouette and by visual separation on the t-SNE plot. The clusters align reasonably with the ground truth labels: cluster 0 corresponds predominantly to NORMAL and cluster 1 corresponds predominantly to PNEUMONIA. This indicates that the combined intensity/HOG/LBP/Hu feature set captures discriminative information relevant to the pneumonia vs normal separation in this dataset.

**If Agglomerative performed best:**

Agglomerative clustering (Ward linkage) produced cluster assignments that were slightly more consistent with the silhouette metric than K-Means. The dendrogram (sampled) shows a hierarchical separation that may be useful for multi-level analysis (coarse grouping followed by sub-clusters). This suggests there are hierarchical structures in the visual data that Ward linkage can exploit.

**If DBSCAN performed best:**

DBSCAN identified dense groupings and flagged some outlier images as noise; when tuned correctly, it provided compact clusters that matched some pneumonia patterns. Because DBSCAN does not require a pre-specified number of clusters, it can be useful to identify anomalous cases. However, DBSCAN’s sensitivity to eps in PCA space means careful parameter selection is essential.

**If none aligned strongly:**

All three unsupervised methods produce clusters that capture some visual structure but do not perfectly separate clinical labels. This indicates that while low-level features (HOG/LBP/histogram) capture imaging differences, pneumonia vs normal often requires more semantic / context information — which pretrained CNN embeddings or supervised learning could better capture.

**13. Files & reproducibility**

The following outputs are saved in clustering\_outputs/ (include in submission):

* xray\_features\_scaled.csv — normalized feature matrix + paths + true labels
* xray\_features\_with\_clusters.csv — cluster labels (KMeans/Agg/DBSCAN) + PCA components
* clustering\_evaluation.csv — Silhouette & Davies-Bouldin metrics for each method
* k\_elbow.png, k\_silhouette.png, k\_distance\_plot.png, dendrogram\_sample.png
* kmeans\_tsne.png, agg\_tsne.png, dbscan\_tsne.png, true\_labels\_tsne.png
* contingency\_true\_vs\_kmeans.csv & contingency\_true\_vs\_kmeans.png
* cluster\_examples\_kmeans/ — example thumbnails for each KMeans cluster
* report\_summary.txt — small text summary generated by the script

**Reproducibility notes**

* Set random seeds (RANDOM\_STATE) to make PCA / KMeans / t-SNE runs repeatable.
* Feature extraction can be cached; the script stores extracted features for quick re-runs.

**How to regenerate the main evaluation table from CSV using pandas**

import pandas as pd

eval\_df = pd.read\_csv('clustering\_outputs/clustering\_evaluation.csv')

print(eval\_df)

**How to view contingency table**

contig = pd.read\_csv('clustering\_outputs/contingency\_true\_vs\_kmeans.csv', index\_col=0)

print(contig)