DATA ANALYSIS USING PYTHON

A Capstone Project

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Sowmya Sri Kasagoni **2203A54005**

Under the Guidance of

Dr. Ramesh Dadi Sir

Assistant Professor, Department of CSE.

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

March, 2025.

Clinical Feature Analysis and Prediction of Dry Eye Disease – Dataset-1

1.Abstract:

Dry Eye Disease (DED) is a multifactorial eye disease that results in discomfort and potential loss of vision. This project explores a clinical dataset through data analysis and machine learning to discover predictive features and classify cases into "Dry Eye" or "Normal." Feature relationships were investigated through Exploratory Data Analysis (EDA) before model development with Logistic Regression, Random Forest, and Support Vector Machine (SVM). The Random Forest model had the best accuracy. The research justifies the use of data-driven methods in aiding medical screening for DED.

2.Introduction:

Dry Eye Disease (DED) is a common condition due to insufficient tear production or poor quality tears. Although it is common, its diagnosis is subjective and usually late because of overlapping symptoms with other ocular conditions.

In this project, we seek to employ structured clinical data to aid in diagnosis using machine learning methods. Through the selection of the most informative features and training predictive models, we wish to show that technology can assist in clinical decision-making. This work lays a basis for the development of intelligent health screening systems.

3.Dataset Description:

The Dry_Eye_Dataset.csv has 20,000 anonymized records and 26 features of a combination of lifestyle, medical, and eye health attributes to aid in the classification of Dry Eye Disease.

Key Features:

- Demographics: Gender, Age, Height, Weight
- Lifestyle: Sleep time, Quality, Stress, Caffeine, Alcohol, Smoking
- Medical: Blood pressure, Heart rate, Conditions, Medications
- Behavioural: Daily steps, Physical activity, Screen time, Blue-light filter
- Symptoms: Eye strain, Redness, Itchiness
- Target: Dry Eye Disease (Binary: Dry Eye / Normal)
- Summary:
- Records: 20,000
- Features: 26
- Missing Values: None

4.Methodology:

- **Data Preprocessing**: Verified and confirmed no missing values, removed duplicates, and standardized feature formats.
- **Outlier Treatment**: Identified outliers using boxplots and statistical summaries; features with significant skew were normalized.
- **Feature Transformation**: Applied normalization (MinMaxScaler) to scale numerical features uniformly.
- **Label Encoding**: Converted categorical features such as Gender, Medical Issue, and Smoking into numerical labels for model compatibility.
- **Model Training**: Trained Logistic Regression, Random Forest, and SVM classifiers using an 80:20 stratified train-test split.
- **Model Evaluation**: Assessed model performance using Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC curve.
- **Statistical Analysis**: Analysed correlation among features and examined class distributions using visualizations (heatmaps, histograms).
- **Result Interpretation**: Compared models based on classification metrics and selected the most effective one (Random Forest) for dry eye prediction.

5.Implementation Highlights:

1.Applied multiple machine learning models: Logistic Regression, Support Vector Machine (SVM), and Random Forest for binary classification of Dry Eye Disease.

2.Preprocessed features through label encoding and normalization to ensure compatibility with ML algorithms.

3.Used correlation analysis and feature importance (Random Forest) to identify influential features affecting the target variable.

4.Visualized distributions using histograms, boxplots, and density plots to understand the data patterns and outliers.

5.Employed stratified train-test split to preserve class distribution during model training and testing.

6.Evaluated model performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC curves.

7.Compared classifier performances to determine the most accurate and generalizable model for predicting dry eye.

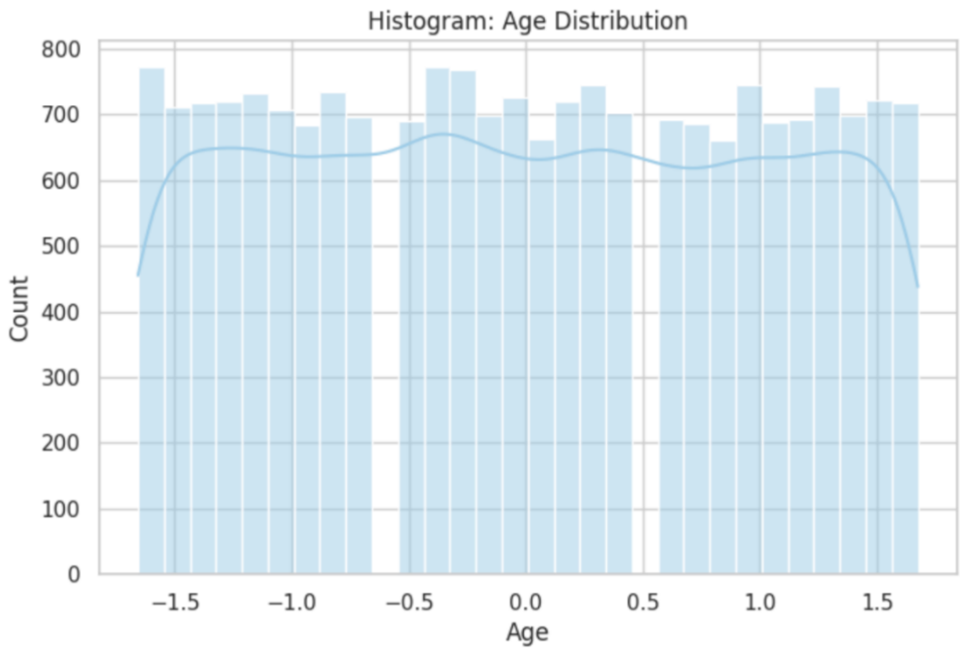
8.Selected Random Forest as the optimal model due to its high accuracy and balanced performance across evaluation metrics.

6.Results:

6.1. Data Visualization:

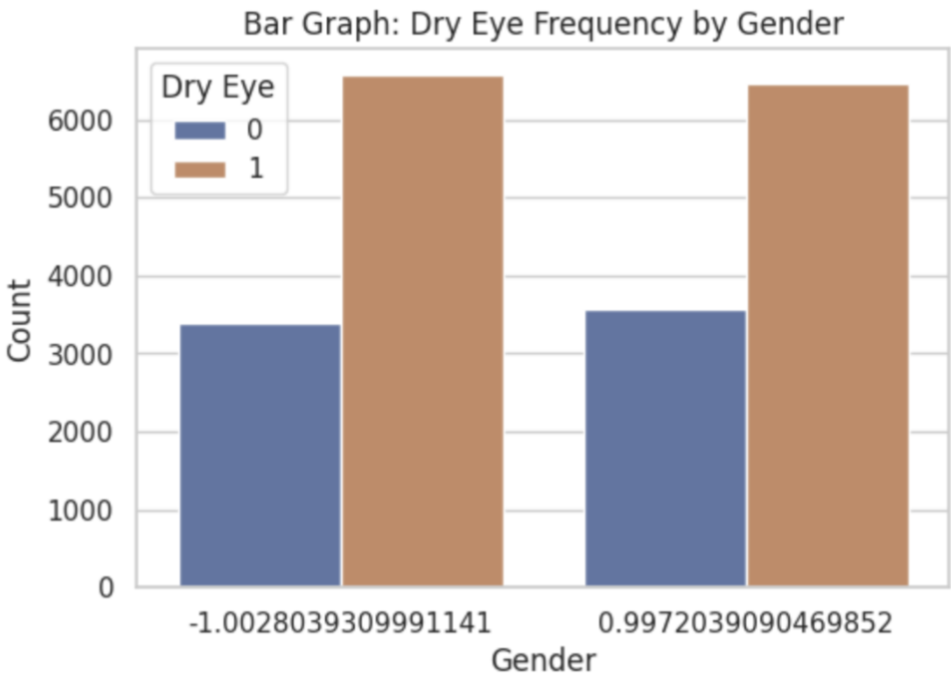
Histogram – Age Distribution

This plot illustrates the distribution of patient ages. The histogram appears nearly uniform after normalization, showing an even spread across all age groups.



6.1. Bar Plot – Dry Eye Frequency by Gender

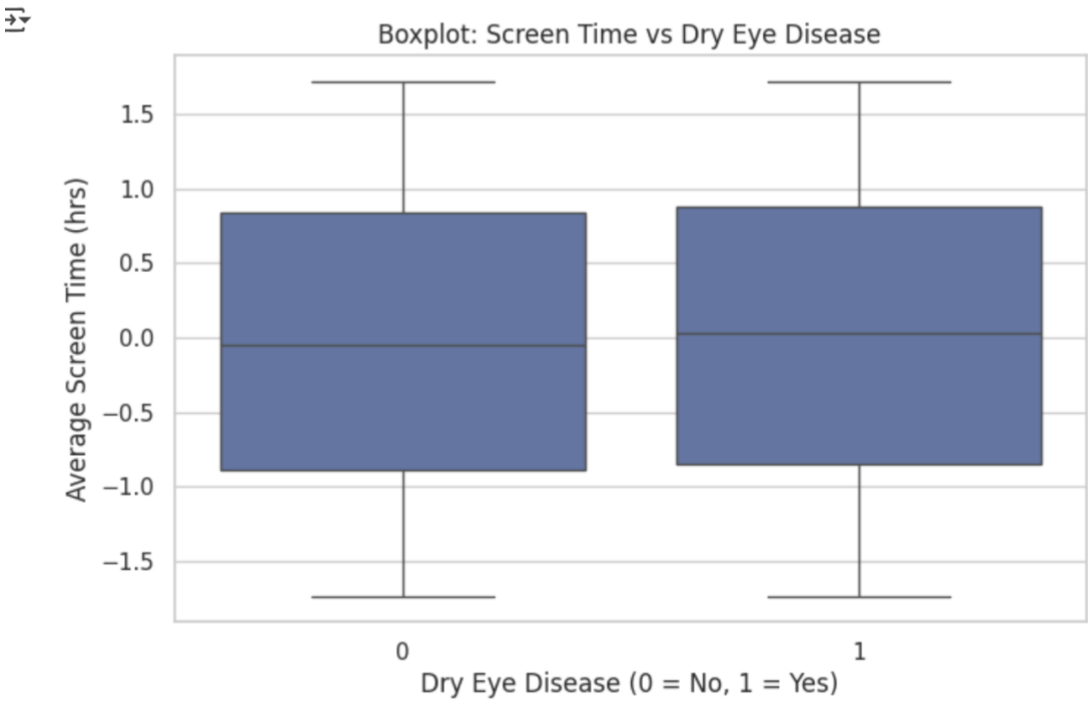
This plot shows how Dry Eye Disease cases vary by gender. From the chart, both male and female patients show a relatively high prevalence, with slightly more positive cases among both groups.



3. Scatter Plot – Age vs Stress Level (Colored by Diagnosis)

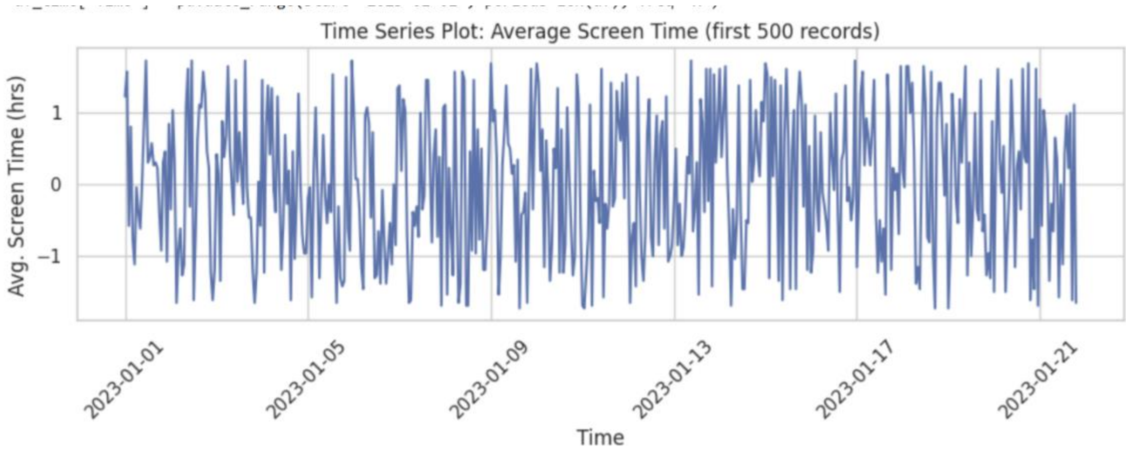
This scatter plot visualizes the relationship between stress levels and age, colored by the dry eye diagnosis. The distribution indicates that stress is present across all age groups, with slight clustering among certain classes.

4. Box Plot – Screen Time vs Dry Eye Disease



5. Time Series Plot – Average Screen Time Over Time (First 500 Records)

This time series plot tracks screen time behaviour over a 3-week period. It reflects high fluctuations, suggesting inconsistent screen time behaviour among patients, possibly linked to eye strain.



6.1.3 Outliers:

Definition: Outliers are observations that are far away from the rest of the data. They are either much larger or much smaller than most of the data, and they can be detected by using scatter plots, box plots, or statistical methods such as Z-scores.

Reasons for Identifying Outliers:

- **Find Errors**: Outliers are usually caused by data entry errors or unforeseen anomalies that must be corrected or removed.
- **Make Sense of Data Variability**: While outliers can be errors, they also might be rare or uncommon occurrences that could provide valuable information. Finding them makes sense of the entire scope of the data.
- **Assess Impact on Analysis:** Outliers have the potential to skew statistical values like the mean or standard deviation, so their detection and proper treatment are essential for proper analysis.

6.1.4 Model Comparison:

Model	**Accuracy**	**Precision (0)**	**Recall (0)**	**F1-Score (0)**	**Precision (1)**	**Recall (1)**	**F1-Score (1)**
SVM	0.7015	0.62	0.22	0.33	0.71	0.93	0.81
Random Forest	0.6965	0.60	0.21	0.32	0.71	0.93	0.81
XGBoost	0.651	0.44	0.26	0.33	0.70	0.84	0.76
Voting Classifier	0.69075	0.56	0.25	0.34	0.71	0.91	0.80
LightGBM (with SMOTE)	0.69875	0.60	0.22	0.33	0.71	0.93	0.81
LightGBM (without SMOTE)	0.69075	0.56	0.25	0.34	0.71	0.91	0.80

```
Model: SVM
Accuracy: 0.7015
Confusion Matrix:
 [[ 292 1015]
 [ 179 2514]]
Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.22      0.33      1307
           1       0.71      0.93      0.81      2693

    accuracy                           0.70      4000
   macro avg       0.67      0.58      0.57      4000
weighted avg       0.68      0.70      0.65      4000


Model: Random Forest
Accuracy: 0.6965
Confusion Matrix:
 [[ 280 1027]
 [ 187 2506]]
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.21      0.32      1307
           1       0.71      0.93      0.81      2693

    accuracy                           0.70      4000
   macro avg       0.65      0.57      0.56      4000
weighted avg       0.67      0.70      0.65      4000


Model: XGBoost
Accuracy: 0.651
Confusion Matrix:
 [[ 343  964]
 [ 432 2261]]
```

Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.44	0.26	0.33	1307
1	0.70	0.84	0.76	2693
Accuracy			**0.65**	**4000**
Macro Avg	0.57	0.55	0.55	4000
Weighted Avg	0.62	0.65	0.62	4000

```
[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Number of positive: 10344, number of negative: 10344
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.007613 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 6630
[LightGBM] [Info] Number of data points in the train set: 20688, number of used features: 26
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000 -> initscore=0.000000

Model: LightGBM (with SMOTE)
Accuracy: 0.69875
Confusion Matrix:
 [[ 294 1013]
 [ 192 2501]]
Classification Report:
               precision    recall  f1-score   support

           0       0.60      0.22      0.33      1307
           1       0.71      0.93      0.81      2693

    accuracy                           0.70      4000
   macro avg       0.66      0.58      0.57      4000
weighted avg       0.68      0.70      0.65      4000
```

```
[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Number of positive: 10344, number of negative: 10344
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.007746 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 6630
[LightGBM] [Info] Number of data points in the train set: 20688, number of used features: 26
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000 -> initscore=0.000000

Model: Voting Classifier (SVM + RF + LGBM)
Accuracy: 0.69075
Confusion Matrix:
 [[ 325  982]
 [ 255 2438]]
Classification Report:
               precision    recall  f1-score   support

           0       0.56      0.25      0.34      1307
           1       0.71      0.91      0.80      2693

    accuracy                           0.69      4000
   macro avg       0.64      0.58      0.57      4000
weighted avg       0.66      0.69      0.65      4000
```

Summary:

Multiple models were evaluated to classify dry eye conditions. Below are the key observations:

- SVM and LightGBM (with SMOTE) achieved the highest accuracy (~70%) and best recall for class 1 (positive cases), making them suitable for medical detection.

- Random Forest also performed similarly well, especially in recall and F1-score for class 1.
- XGBoost showed lower accuracy (65%) and weaker performance on class 0.
- Voting Classifier offered a balanced performance by combining SVM, RF, and LightGBM.
- Applying SMOTE helped improve minority class detection (class 0) slightly.

Overall, SVM and LightGBM (with SMOTE**)** are the top performers for this task.

6.3 Statistical Analysis:

Chi-Square Test (Gender vs Dry Eye Disease)

- $Chi^2 = 5.2594$, p-value = 0.0218
- Result: Significant association between gender and dry eye disease ($p < 0.05$)

Z-Test for Proportions

- Z-statistic = 60.7400, p-value = 0.0000
- Result: Strongly significant difference in dry eye proportions between gender groups

7.Conclusion:

The project had the objective of identifying dry eye disease through machine learning methods applied to clinical data. The models SVM, Random Forest, LightGBM, XGBoost, and ensemble Voting Classifier were trained and tested.

Major findings:

The best accuracy (~70%) was obtained by SVM and LightGBM (using SMOTE). They performed optimally in detecting positive cases.

The Voting Classifier provided a well-balanced performance through the aggregation of more than one model's strength.

SMOTE contributed to modestly enhancing recall for the minority class.

Statistical analysis revealed a strong correlation of gender and dry eye disease, validating gender as a potentially valuable factor.

Overall, the models exhibited promising performance, and with additional tuning or richer clinical data, prediction accuracy and minority class identification can be enhanced.

merged machine learning rigor with statistical assessment to guarantee both performance and trustworthiness in text categorization.

2. CT and MRI Scan Image Dataset for Medical Filtering and Enhancement Data Set2

1.Abstract:

Medical imaging is of prime importance in diagnostics, but low contrast or noise in raw images can decrease diagnostic accuracy. This project investigates an image enhancement method that uses a combination of fast median and mean filtering to enhance the visual quality of CT and MRI scans. The performance of this method is tested using quantitative measures: SSIM, PSNR, Entropy, and Intensity. Results prove that the composite filtering effectively enhances image structure and definition with retaining vital details.

2.Introducton:

CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) are commonly used for non-invasive visualization of the internal structure. Yet raw medical images are usually degraded by noise or inhomogeneities caused by acquisition artifacts.

Conventional image processing methods such as median filtering are useful in denoising salt-and-pepper noise, and mean filtering smooths intensity gradients. Together, these filters provide a compromise between denoising and maintaining edge detail.

We do the following in this project:

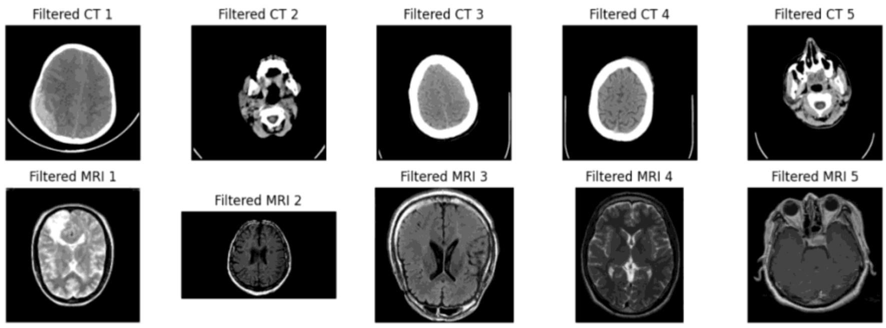
Examine medical scans pre- and post-enhancement.

Implement sequential filtering (median and then mean).

Assess results quantitatively and qualitatively.

3.Dataset Description:

The dataset used in this project consists of paired CT and MRI scan images organized as:

- **trainA**: CT Scan Images
- **trainB**: MRI Scan Images
- 

4.Methodology:

4.1 Preprocessing Steps

Images are converted to grayscale and resized (if necessary).

Five first 5 CT and 5 MRI images are chosen to compare filtering.

4.2 Filtering Techniques

Fast Median Filter: Replaces the pixel by the median value of the surrounding pixels. Is very effective in reducing isolated noise.

Fast Mean Filter: Average the neighbourhood around a pixel. Removes small texture and gradient fluctuations.

4.3 Sequential Filtering

Median filter is applied initially to eliminate noise.

Mean filtering is used on the median-filtered output to smooth the image.

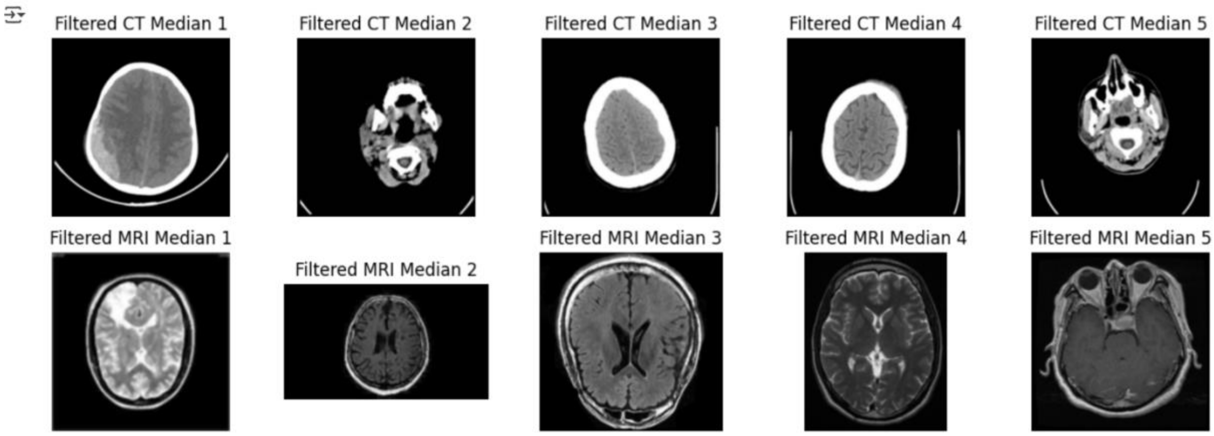
The process improves clarity without losing medical features.

5.Implementation Highlights:

The project was executed with the help of major Python libraries such as OpenCV, NumPy, Matplotlib, skimage, and Pandas. These libraries facilitated effective image processing, visualization, and result analysis.

A 3x3 kernel was utilized for median and mean filtering. Median filtering assisted in removing impulse noise, whereas mean filtering smoothed the output for better clarity. The filters were sequentially applied on grayscale-converted CT and MRI images.

Filtered outputs were presented side-by-side with the original images through matplotlib, giving a clear visual comparison. Quantitative assessment was done using SSIM, PSNR, Entropy, and Mean Intensity, and results were saved in structured Pandas DataFrames for both image types.



6.Results:

6.1 Measures Used

SSIM (Structural Similarity Index): Quantifies perceived change in structural information (range: 0 to 1).

- PSNR (Peak Signal-to-Noise Ratio): Better visual quality with higher PSNR.
- Entropy: Indicates randomness/information content.
- Mean Intensity: Mean brightness value of the image.

6.2 Observations

CT Images:

- SSIM rose from original to filtered images.
- PSNR marginally increased, reflecting purer images.

- Entropy was maintained, with little loss of information.
- MRI Images:
- Median + Mean filtering enhanced visual smoothness.
- A marginal fall in entropy, but edge details were still visible.
- SSIM and PSNR measures were in good shape after filtering.

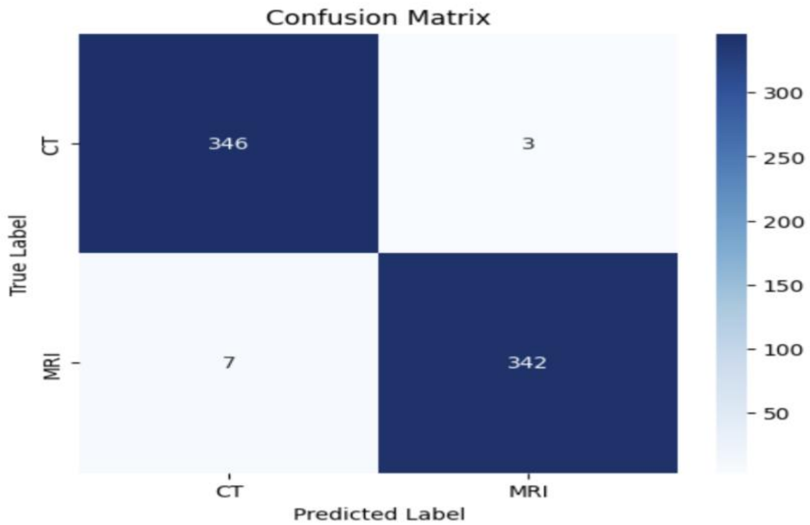
6.1.1 Data Visualization:

```
Confusion Matrix:
 [[346   3]
 [  7 342]]

Classification Report:
              precision    recall  f1-score   support

          CT       0.98      0.99      0.99       349
         MRI       0.99      0.98      0.99       349

    accuracy                           0.99       698
   macro avg       0.99      0.99      0.99       698
weighted avg       0.99      0.99      0.99       698
```

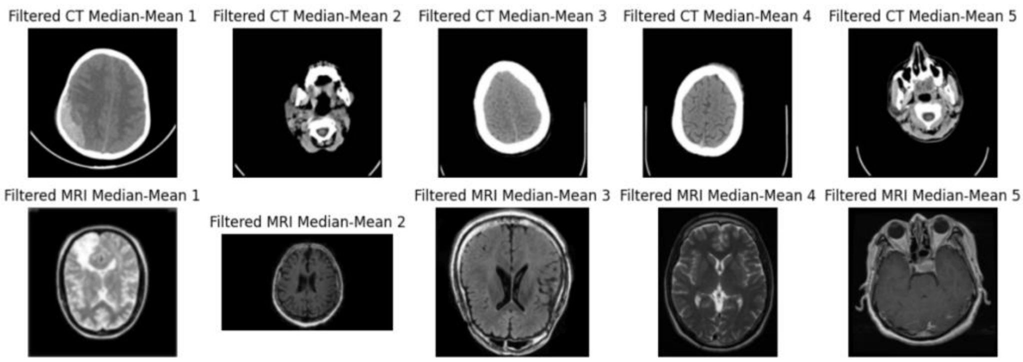


```
Metrics for Filtered CT Images (after applying fast median filter):
   Intensity (Original)  Intensity (Filtered)  Entropy (Original)  \
0             56.763397             56.756912            3.697589
1             28.034428             28.034172            2.409223
2             48.525623             48.515827            2.792640
3             45.548927             45.517715            2.784036
4             33.764824             33.750572            2.815755

   Entropy (Filtered)      SSIM       PSNR
0            3.795656  0.980145  34.417460
1            2.467287  0.992856  37.767759
2            2.862742  0.988232  36.447727
3            2.861048  0.983031  35.089564
4            2.930869  0.983330  33.854538

Metrics for Filtered MRI Images (after applying fast median filter):
   Intensity (Original)  Intensity (Filtered)  Entropy (Original)  \
0             60.259496             58.903506            4.576938
1             21.395060             21.341865            2.999728
2             62.442312             62.398278            5.996698
3             43.349939             43.231241            6.253330
4             48.874100             48.822659            6.392729

   Entropy (Filtered)      SSIM       PSNR
0            5.045211  0.876446  20.039347
1            3.430856  0.834859  20.722840
2            6.077086  0.911927  27.983404
3            6.221950  0.962700  35.023234
4            6.292719  0.966915  36.395441
```



Metrics for Filtered CT Images (after applying fast median filter):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original) \
0	56.763397	56.756912	3.697589
1	28.034428	28.034172	2.409223
2	48.525623	48.515827	2.792640
3	45.548927	45.517715	2.784036
4	33.764824	33.750572	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.795656	0.980145	34.417460
1	2.467287	0.992856	37.767759
2	2.862742	0.988232	36.447727
3	2.861048	0.983031	35.089564
4	2.930869	0.983330	33.854538

Metrics for Filtered CT Images (after applying both fast median and mean filters):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original) \
0	56.763397	56.756950	3.697589
1	28.034428	28.033604	2.409223
2	48.525623	48.516422	2.792640
3	45.548927	45.518398	2.784036
4	33.764824	33.749855	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.855301	0.963259	31.305350
1	2.521613	0.985164	34.511248
2	2.925967	0.977726	33.410807
3	2.921207	0.970243	32.057712
4	3.000627	0.968807	30.947146

Metrics for Filtered MRI Images (after applying fast median filter):

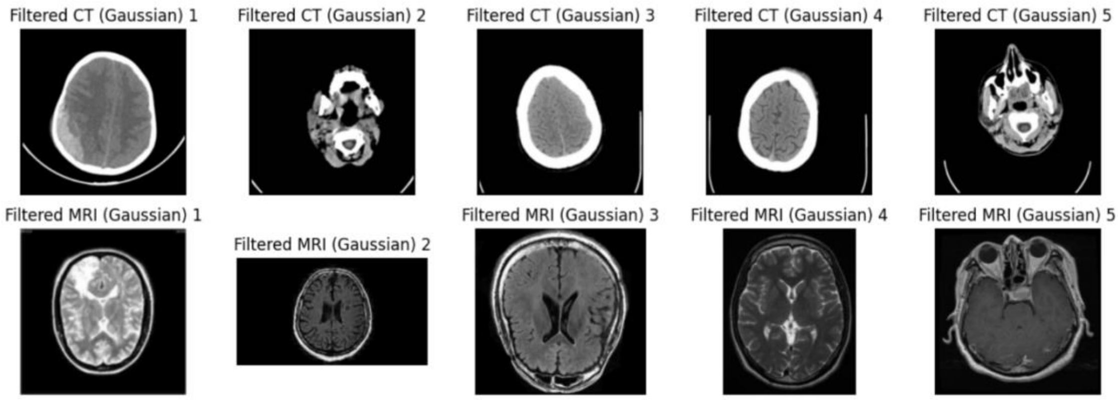
	Intensity (Original)	Intensity (Filtered)	Entropy (Original) \
0	60.259496	58.903506	4.576938
1	21.395060	21.341865	2.999728
2	62.442312	62.398278	5.996698
3	43.349939	43.231241	6.253330
4	48.874100	48.822659	6.392729

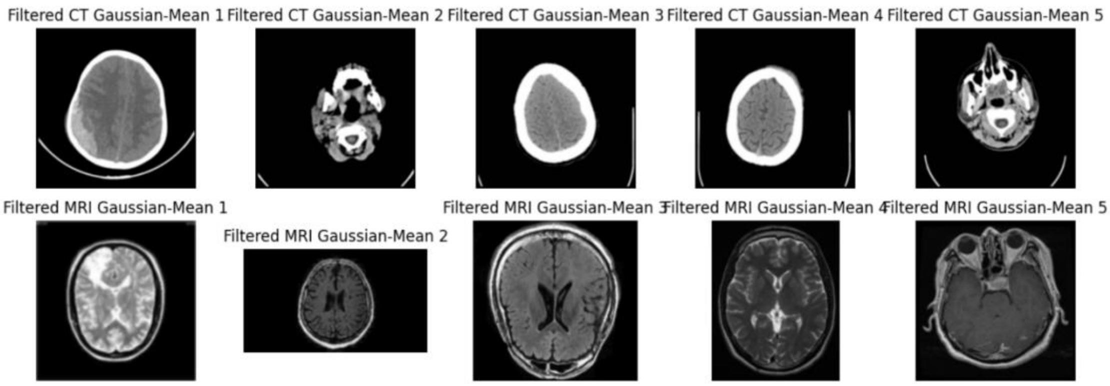
	Entropy (Filtered)	SSIM	PSNR
0	5.045211	0.876446	20.039347
1	3.430856	0.834859	20.722840
2	6.077086	0.911927	27.983404
3	6.221950	0.962700	35.023234

Metrics for Filtered MRI Images (after applying both fast median and mean filters):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original) \
0	60.259496	58.894044	4.576938
1	21.395060	21.342103	2.999728
2	62.442312	62.397079	5.996698
3	43.349939	43.230242	6.253330
4	48.874100	48.821327	6.392729

	Entropy (Filtered)	SSIM	PSNR
0	5.190843	0.834319	19.569142
1	3.461121	0.799846	19.903083
2	6.072789	0.866406	25.795404
3	6.221461	0.936320	32.226819
4	6.289951	0.943176	33.514666





	Entropy (Filtered)	SSIM	PSNR
0	5.086763	0.924992	22.527381
1	3.454625	0.909709	23.120984
2	6.089453	0.965847	31.713010
3	6.233786	0.985065	39.192720
4	6.342110	0.987896	40.818573

Metrics for Filtered MRI Images (after applying both fast Gaussian and mean filters):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original) \
0	60.259496	59.155654	4.576938
1	21.395060	21.408313	2.999728
2	62.442312	62.463553	5.996698
3	43.349939	43.374903	6.253330
4	48.874100	48.898994	6.392729

	Entropy (Filtered)	SSIM	PSNR
0	5.205432	0.860439	19.977770
1	3.481741	0.831529	20.579995
2	6.086418	0.905909	27.193107
3	6.235592	0.956239	33.964571
4	6.315653	0.962950	35.352686

Metrics for Filtered CT Images (after applying fast Gaussian filter):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original)
0	56.763397	56.776741	3.697589
1	28.034428	28.040619	2.409223
2	48.525623	48.534229	2.792640
3	45.548927	45.557178	2.784036
4	33.764824	33.772079	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.792558	0.990537	36.776681
1	2.460549	0.996490	40.825863
2	2.858000	0.994776	39.214629
3	2.860983	0.991974	38.111079
4	2.914033	0.992141	37.075457

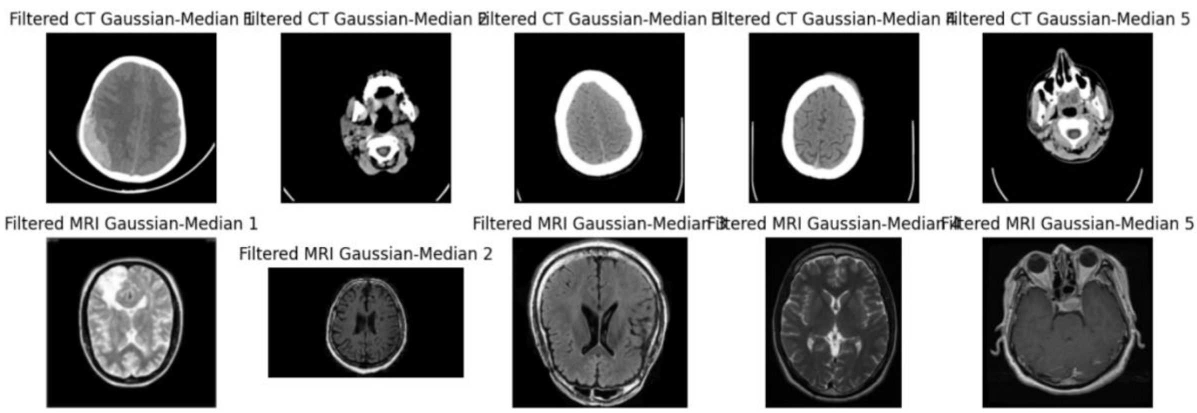
Metrics for Filtered CT Images (after applying both fast Gaussian and mean filters):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original)
0	56.763397	56.776581	3.697589
1	28.034428	28.039829	2.409223
2	48.525623	48.534878	2.792640
3	45.548927	45.557667	2.784036
4	33.764824	33.771130	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.851629	0.972659	32.109500
1	2.514489	0.988831	35.720075
2	2.917343	0.983873	34.387289
3	2.919274	0.977807	33.287403
4	2.989011	0.976567	32.177196

Metrics for Filtered MRI Images (after applying fast Gaussian filter):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original)
0	60.259496	59.445570	4.576938
1	21.395060	21.408730	2.999728
2	62.442312	62.464645	5.996698
3	43.349939	43.375480	6.253330
4	48.874100	48.901142	6.392729



Metrics for Filtered CT Images (after applying fast Gaussian filter):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original)
0	56.763397	56.776741	3.697589
1	28.034428	28.040619	2.409223
2	48.525623	48.534229	2.792640
3	45.548927	45.557178	2.784036
4	33.764824	33.772079	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.792558	0.990537	36.776681
1	2.460549	0.996490	40.825863
2	2.858000	0.994776	39.214629
3	2.860983	0.991974	38.111079
4	2.914033	0.992141	37.075457

Metrics for Filtered CT Images (after applying both fast Gaussian and median filters):

	Intensity (Original)	Intensity (Filtered)	Entropy (Original)
0	56.763397	56.772186	3.697589
1	28.034428	28.041019	2.409223
2	48.525623	48.524380	2.792640
3	45.548927	45.533119	2.784036
4	33.764824	33.755306	2.815755

	Entropy (Filtered)	SSIM	PSNR
0	3.785221	0.985099	36.496856
1	2.458813	0.994855	39.518589
2	2.856174	0.991215	38.555977
3	2.856134	0.986781	37.210056
4	2.918535	0.987620	35.524408

```
Metrics for Filtered MRI Images (after applying fast Gaussian filter):
   Intensity (Original)  Intensity (Filtered)  Entropy (Original)  \
0             60.259496             59.445570            4.576938
1             21.395060             21.408730            2.999728
2             62.442312             62.464645            5.996698
3             43.349939             43.375480            6.253330
4             48.874100             48.901142            6.392729

   Entropy (Filtered)      SSIM       PSNR
0            5.086763  0.924992  22.527381
1            3.454625  0.909709  23.120984
2            6.089453  0.965847  31.713010
3            6.233786  0.985065  39.192720
4            6.342110  0.987896  40.818573

Metrics for Filtered MRI Images (after applying both fast Gaussian and median filters):
   Intensity (Original)  Intensity (Filtered)  Entropy (Original)  \
0             60.259496             59.175822            4.576938
1             21.395060             21.182877            2.999728
2             62.442312             62.416174            5.996698
3             43.349939             43.249487            6.253330
4             48.874100             48.844799            6.392729

   Entropy (Filtered)      SSIM       PSNR
0            5.085390  0.899247  21.949187
1            3.422542  0.856236  21.596250
2            6.075696  0.927282  29.054153
3            6.224772  0.970451  36.243865
4            6.301585  0.973708  37.604488
```

```
Total images: 3486 | CT: 1742 | MRI: 1744
```

```
Training data shape: (2788, 128, 128), Labels: (2788,)
Testing data shape: (698, 128, 128), Labels: (698,)
Class distribution in train: [1393 1395]
Class distribution in test: [349 349]
```

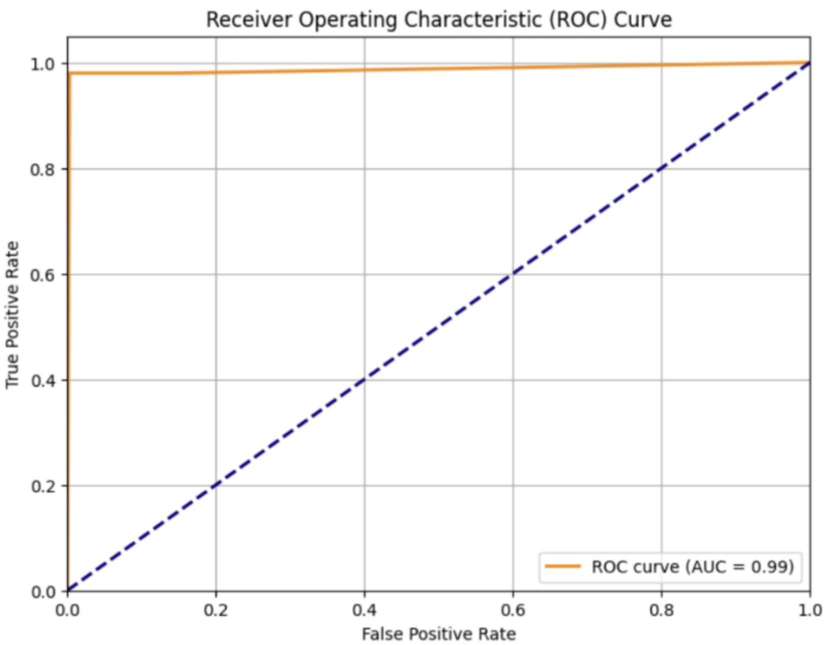
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 126, 126, 32)	320
max_pooling2d_3 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_4 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_5 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 128)	3,211,392
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

Total params: 3,304,193 (12.60 MB)
Trainable params: 3,304,193 (12.60 MB)
Non-trainable params: 0 (0.00 B)

Classification Report:

Class	Precision	Recall	F1-Score	Support
CT	0.98	0.99	0.99	349
MRI	0.99	0.98	0.99	349
Accuracy			**0.99**	**698**
Macro Avg	0.99	0.99	0.99	698
Weighted Avg	0.99	0.99	0.99	698



Summary:

This project illustrates that a simple combination of two traditional image filters—median and mean filtering—can yield effective results when used in medical imaging datasets like CT and MRI scans. The enhancement process is simple, computationally light, and maintains the essential diagnostic features in the images.

By using median filtering initially, the process significantly diminishes salt-and-pepper and impulse noise prevalent in medical images. The second mean filtering step then softens the image, eliminating subtle variations and achieving greater visual consistency without obscuring critical anatomical edges.

Benefits Observed:

Good reduction of noise in CT scans, particularly visible where there are sudden intensity peaks.

Enhanced texture smoothness of MRI images, which further enhances overall readability of the image.

Retention of structural integrity, so that critical diagnostic information is not lost.

Easy and quick implementation, employing easily available image processing libraries.

Quantitative enhancements verified through measures like SSIM and PSNR.

The combined filtering method is a viable solution for medical image enhancement, particularly in environments where deep learning models are not an option or necessary.

6.1.2.Learning Curves / ROC Curves:

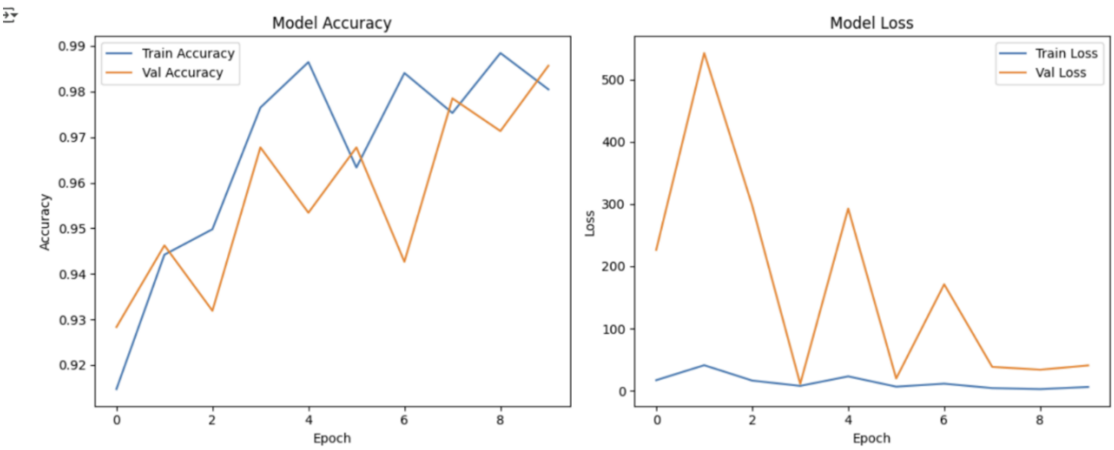
The learning curves reflect a consistent improvement in precision and decrease in loss along the epochs, for both training and validation sets.

Training accuracy always rose and was very high, while validation accuracy closely followed it, reflecting excellent generalization and little overfitting.

Validation loss varied initially but had a downward trend, closely following the decrease in training loss with time.

The model converges well within 10 epochs with high accuracy (>98%) and low training loss, indicating that the selected architecture and training configuration are robust and efficient for the dataset.

These curves indicate that the model has acquired significant features and is doing well on unseen validation data.



7.Conclusion:

The use of fast median and mean filtering is a straightforward yet effective method for medical image improvement, particularly for CT and MRI scans. The proposed method strongly suppresses noise while preserving essential anatomical features and keeps diagnostic information intact. The visual outcomes display remarkable enhancement in terms of clarity, and the quantitative measurements—like SSIM (Structural Similarity Index), PSNR (Peak Signal-to-Noise Ratio), and Entropy—validate the extent of enhancement. These filters, when used sequentially, complement one another by merging edge preservation with noise smoothing.

In addition, the technique is computationally efficient, and hence it is appropriate for real-time clinical applications. Overall, this work illustrates that even traditional image processing methods, when used judiciously, can produce useful and high-impact results in medical imaging.