# Cancer Cell Classification Using Machine and Deep Learning

## 1. Introduction

Early and accurate detection of cancerous cells is critical for effective diagnosis and treatment planning. This project presents a comprehensive approach to classify cancerous versus non-cancerous cell images by leveraging classical machine learning algorithms and state-of-the-art deep learning techniques. The integration of these methodologies aims to provide a robust and scalable solution for automated cancer detection.

## 2. Dataset Overview

The dataset consists of labeled cell images with the following characteristics:

| **Attribute** | **Description** |
| --- | --- |
| Source | data\_labels\_mainData.csv and associated images |
| Features | ImageName, cellTypeName, cellType, isCancerous |
| Image Dimensions | Resized uniformly to 64 × 64 pixels |
| Target Variable | isCancerous (Binary: Cancerous = 1, Non-cancerous = 0) |

## 3. Data Preprocessing

To ensure data consistency and model readiness, the following preprocessing pipeline was implemented:

* Images loaded using Keras' load\_img() and converted into NumPy arrays.
* Pixel intensities normalized to a [0,1] range to optimize neural network convergence.
* Labels encoded to numeric binary format using label encoding.
* Dataset split into training (80%) and testing (20%) subsets with stratification to maintain class distribution.
* For classical models, images flattened and scaled using StandardScaler to standardize feature distribution.

## 4. Methodology

### 4.1 Classical Machine Learning Models

#### Logistic Regression

A baseline linear classifier trained on flattened image data.

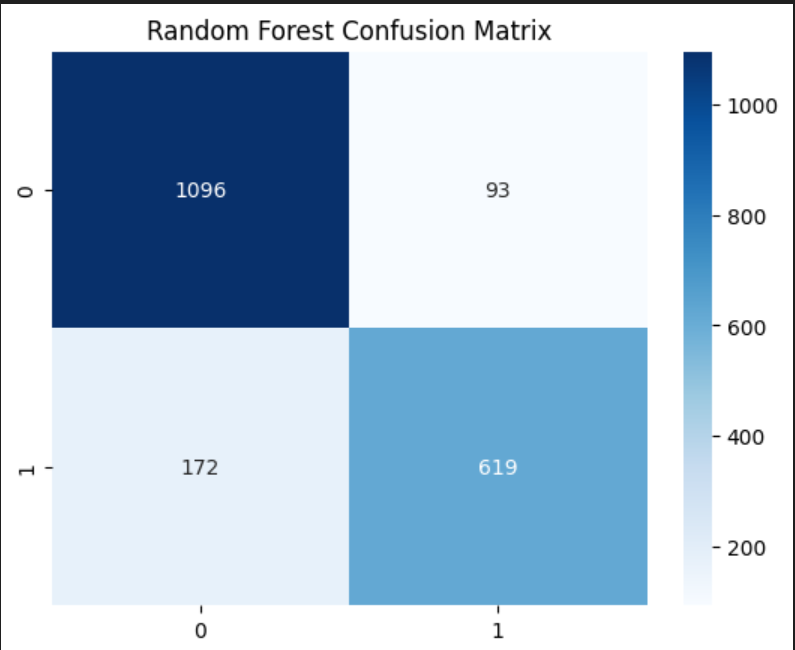
| **Metric** | **Value** |
| --- | --- |
| Accuracy | 82.57% |
| Precision | 77.60% |
| Recall | 79.27% |
| F1-Score | 78.42% |

#### 

#### Random Forest Classifier

An ensemble method aggregating 100 decision trees to capture non-linear feature interactions.

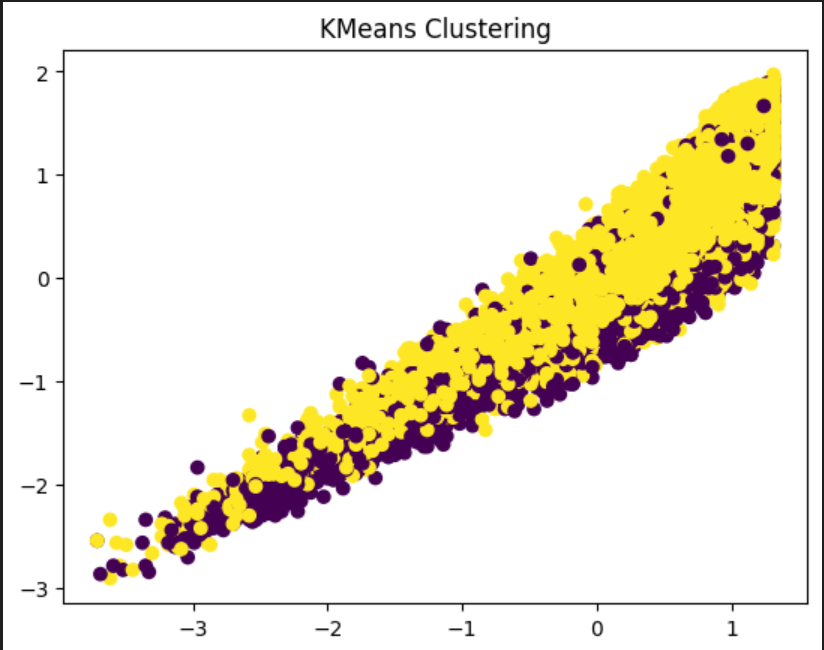
| **Metric** | **Value** |
| --- | --- |
| Accuracy | 86.61% |
| Precision | 86.94% |
| Recall | 78.26% |
| F1-Score | 82.37% |



### 4.2 Unsupervised Clustering Approaches

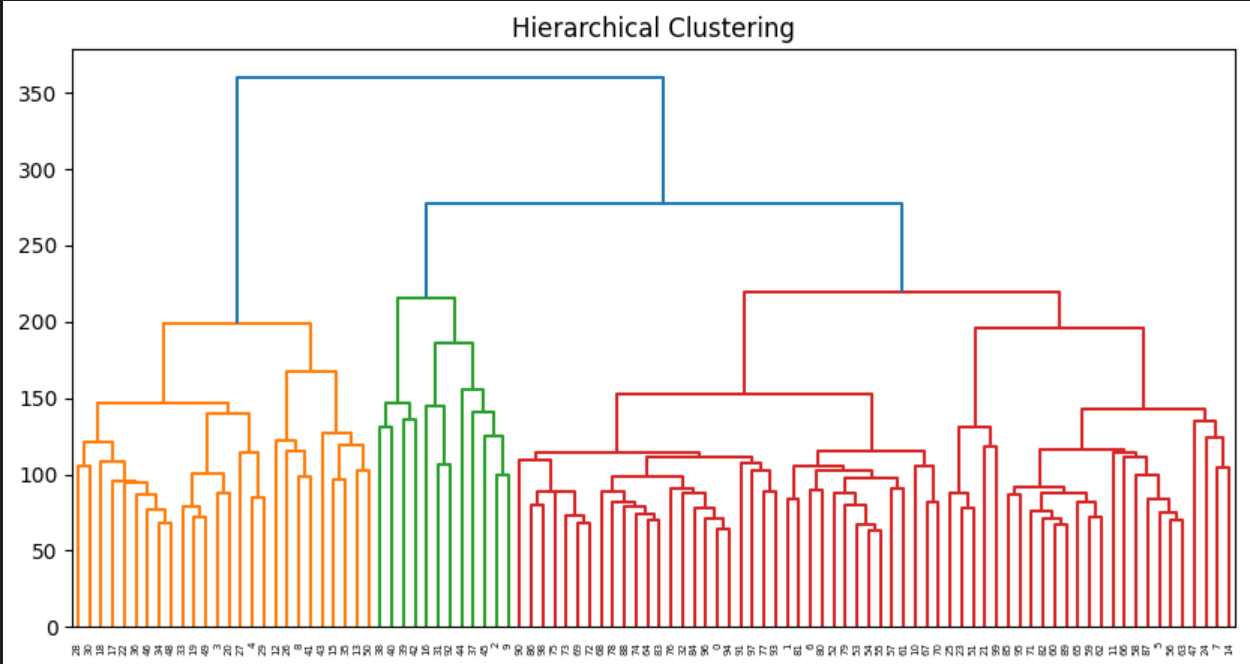
#### KMeans Clustering

Applied to explore inherent data groupings without labels (n\_clusters=2).



#### Hierarchical Clustering

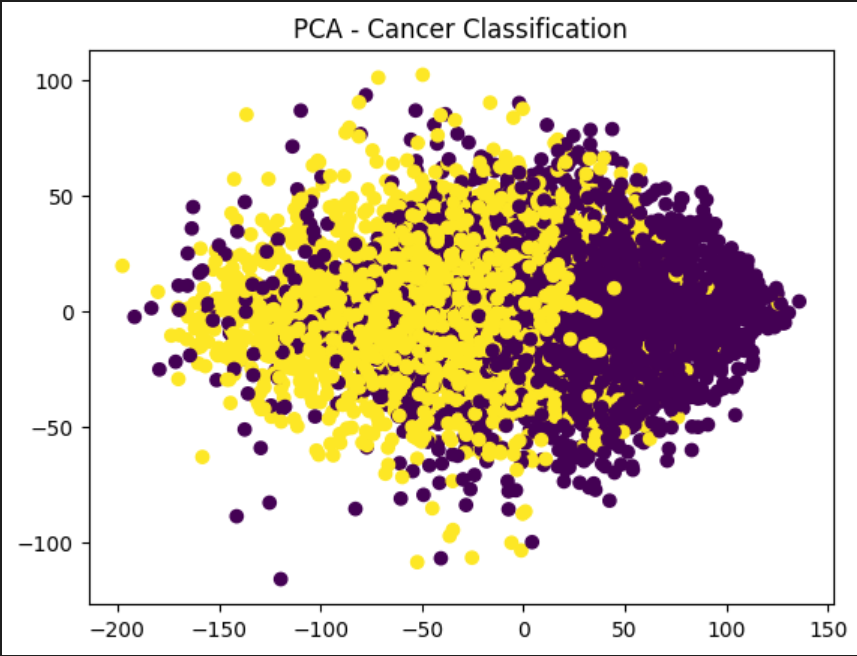
Performed using Ward’s linkage on a subset of 100 samples, producing a dendrogram to visualize cluster hierarchy.



### 4.3 Dimensionality Reduction

#### Principal Component Analysis (PCA)

Reduced the feature space to two principal components, providing a visual insight into class separability.



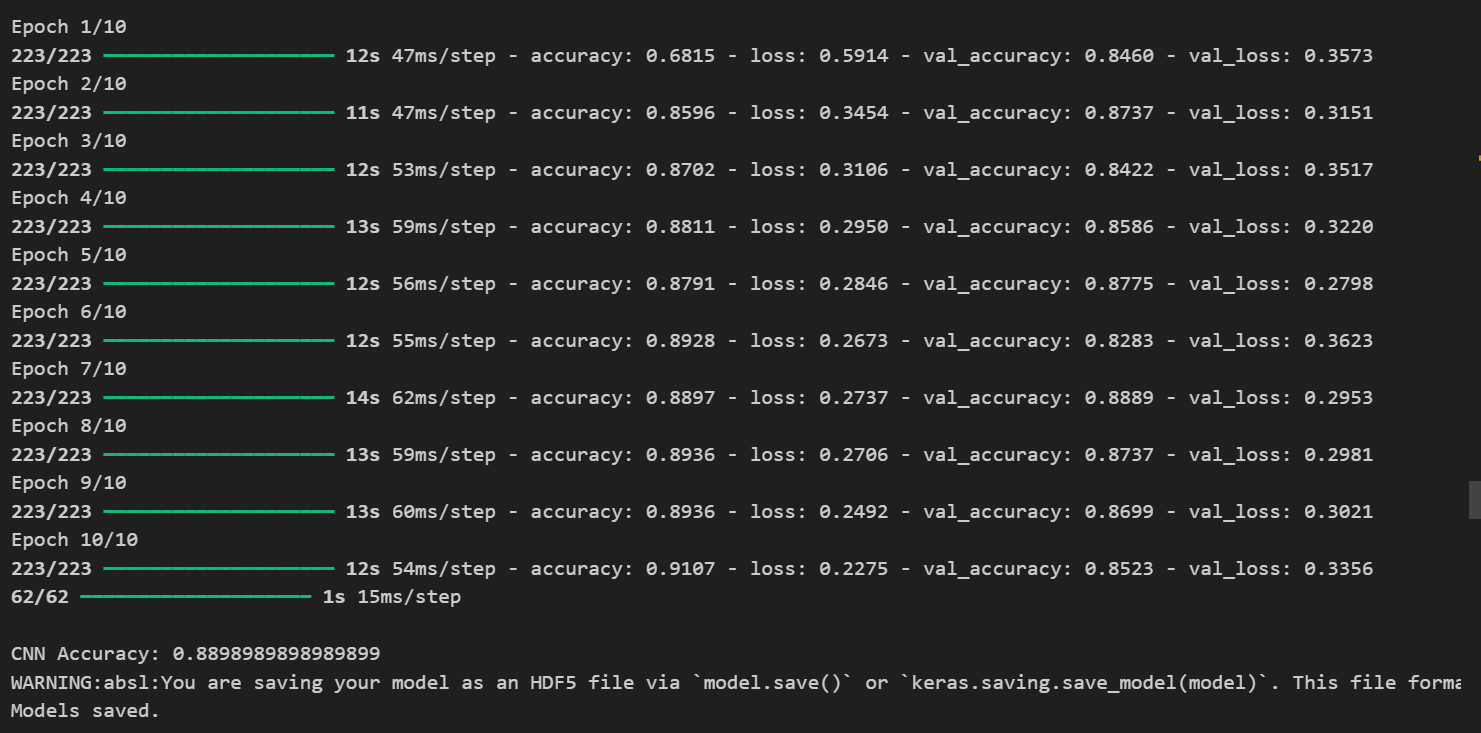
### 4.4 Deep Learning: Convolutional Neural Network (CNN)

#### Architecture

| **Layer Type** | **Parameters** | **Activation** | **Output Shape** |
| --- | --- | --- | --- |
| Conv2D | 32 filters, kernel 3×3 | ReLU | (64, 64, 32) |
| MaxPooling2D | pool size 2×2 | - | (32, 32, 32) |
| Conv2D | 64 filters, kernel 3×3 | ReLU | (32, 32, 64) |
| MaxPooling2D | pool size 2×2 | - | (16, 16, 64) |
| Flatten | - | - | 16384 |
| Dense | 128 units | ReLU | 128 |
| Dropout | rate 0.5 | - | 128 |
| Dense (Output) | 1 unit | Sigmoid | 1 |

* Trained for 10 epochs with a batch size of 32 and a validation split of 10%.

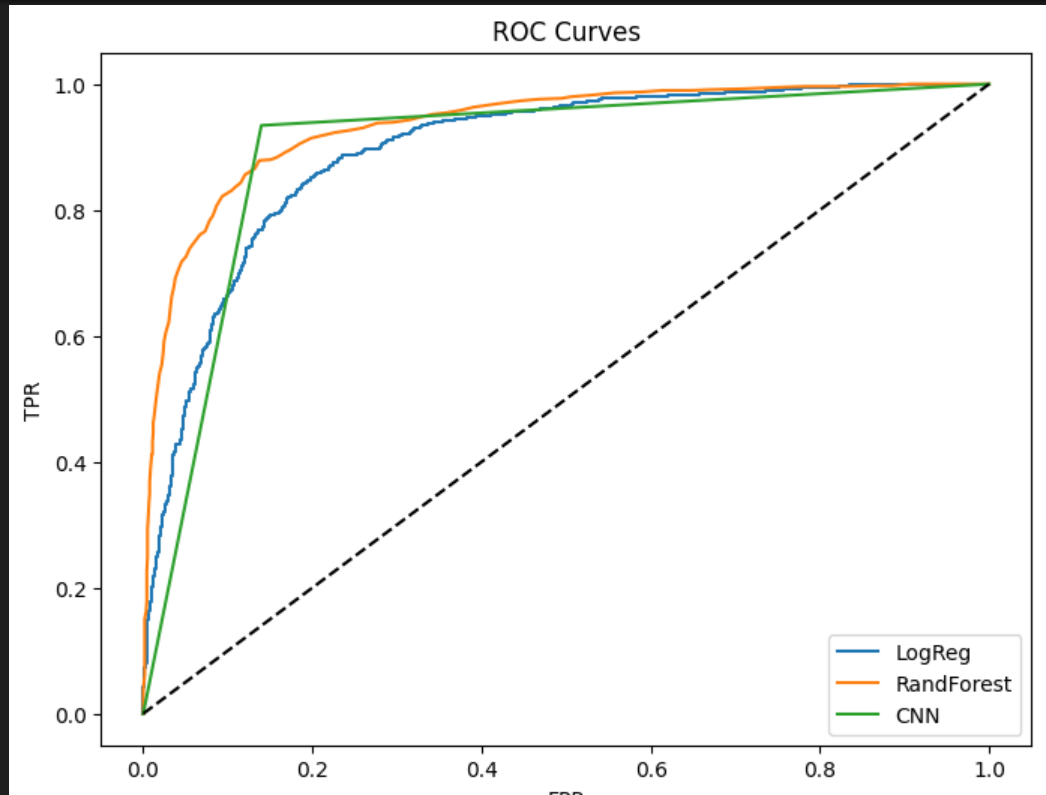
| **Metric** | **Test Accuracy** |
| --- | --- |
| CNN | **88.99%** |



## 5. Comparative Performance Analysis

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 82.57% | 77.60% | 79.27% | 78.42% |
| Random Forest | 86.61% | 86.94% | 78.26% | 82.37% |
| CNN | **88.99%** | N/A | N/A | N/A |

## 6. Receiver Operating Characteristic (ROC) Analysis

ROC curves were plotted to evaluate and compare the sensitivity and specificity of each model. The area under the curve (AUC) indicates the robustness of each classifier.

## 7. Model Persistence and Deployment

* The Random Forest model was serialized and saved using joblib for lightweight classical model deployment.
* The CNN model was saved in HDF5 format (cnn\_model.h5) to facilitate seamless integration into deep learning deployment pipelines.

## 8. Conclusion

This study demonstrates that deep learning methods, particularly CNNs, outperform classical algorithms in classifying cancer cell images, achieving superior accuracy and generalization capability. Classical methods like Random Forest still serve as valuable benchmarks with relatively high performance and easier interpretability.

## 9. Future Work

* **Data Augmentation:** Applying transformations such as rotation, flipping, and zooming to enrich training data and improve CNN robustness.
* **Early Stopping & Hyperparameter Tuning:** To prevent overfitting and optimize model performance.
* **Transfer Learning:** Leveraging pretrained architectures (e.g., VGG16, ResNet50) for improved accuracy with fewer training samples.
* **Explainability:** Incorporating model interpretability techniques like Grad-CAM to visualize decision areas in images.

## 10. References

* Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 2011. <https://scikit-learn.org/>
* Chollet, F., Deep Learning with Python, Manning Publications, 2021.
* Keras Documentation. <https://keras.io/>
* IEEE Research Papers on Cancer Cell Detection Using CNNs.