BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE-DATA SCIENCE

A Mini-Project Report On

"FLOWER CLASSIFICATION USING ANN MODEL"

A report submitted in partial fulfillment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

Submitted By

GHANTA POOJITH BALU USN: 3BR22CD017

Under the Guidance of

Mr. Azhar Biag

Asst. Professor

Dept of CSE (DATA SCIENCE),

BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka 2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini Project of NEURAL NETWORK AND DEEP LEARNING title "Flower Classification USING ANN MODEL" has been successfully presented by Ghanta Poojith Balu 3BR22CD017 student of semester B.E for the partial fulfillment of the requirements for the award of Bachelor Degree in CSE(DS) of the BALLARI INSTITUTE OF TECHNOLOGY& MANAGEMENT, BALLARI during the academic year 2025-2026.

It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the library. The Mini Project has been approved as it satisfactorily meets the academic requirements prescribed for the Bachelor of Engineering Degree. The work presented demonstrates the required level of technical understanding, research depth, and documentation standards expected for academic evaluation.

Signature of Coordinators	Signature of HOD
Mr. Azhar Baig	**Dr. Aradhana D**
Ms. Chaithra B M	

ABSTRACT

Flower classification is one of the most significant global health challenges, affecting millions of individuals and contributing to severe complications such as cardiovascular disease, kidney failure, and neurological damage. Early detection plays a crucial role in preventing these long-term effects and improving patient outcomes. With the advancement of artificial intelligence, machine learning techniques have emerged as powerful tools for analyzing medical data and predicting disease risk. This project focuses on building an Artificial Neural Network (ANN)–based model to classify individuals as diabetic or non-diabetic using the Pima Indians Diabetes Dataset. The system preprocesses key clinical features such as glucose levels, blood pressure, BMI, insulin values, age, and diabetes pedigree function, ensuring standardized inputs for efficient model learning. The ANN architecture is designed with multiple hidden layers, ReLU activations, dropout regularization, and a sigmoid output layer for binary classification, allowing the network to capture complex relationships present in the medical dataset.

After data preparation and model construction, the ANN is trained using optimized parameters and evaluated through various performance metrics, including accuracy, precision, recall, F1-score, and a confusion matrix. Visualization tools such as accuracy and loss plots further illustrate the model's behavior during training, providing insights into its stability and generalization capability. The results demonstrate that the ANN model can successfully learn patterns associated with diabetes and make reliable predictions. This study highlights the potential of deep learning techniques in medical diagnosis and emphasizes their ability to support healthcare professionals by providing early risk assessments. With further refinement and the integration of larger datasets, such predictive models could be extended into real-world healthcare systems, contributing to faster, data-driven decision-making and improved patient care.

ACKNOWLEDGEMENT

The satisfactions that accompany the successful completion of our mini project on FLOWER CLASSIFICATION USING ANN MODEL would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of our mini-project.

We are extremely grateful to our Guide **Mr. Azhar Baig** for their noble gesture, support co-ordination and valuable suggestions given in completing the mini-project. We also thank **Dr. Aradhana D,** H.O.D. Department of CSE(DS), for his co-ordination and valuable suggestions given in completing the mini-project. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the Mini project.

Name	USN
GHANTA POOJITH BALU	3BR22CD017

TABLE OF CONTENTS

1.INTRODUCTION

Flower classification plays a crucial role in botanical research, agriculture, and environmental monitoring, where accurate identification of flower species is essential for analysis and decision-making. With the rapid growth of digital imaging, large flower datasets can now be processed efficiently using artificial intelligence techniques. Artificial Neural Networks (ANN) provide a powerful approach for classifying flower images by learning complex patterns from pixel-level features. ANNs simulate the functioning of biological neurons, enabling them to automatically extract and learn discriminative characteristics from flower images. The model learns hidden relationships based on features such as shape, texture, and color, which are commonly used for identifying species.

The script begins by downloading the dataset from Kaggle and loading it into a pandas Data Frame. It then provides a quick preview and shape information to ensure the data is well-structured and complete. The target variable, flower classification is separated from the input features for classification. All feature values are converted into numerical format suitable for model training. Train-test splitting is applied to divide the data into training and evaluation portions. This ensures the model is evaluated on unseen data to check its generalization ability.

Before training, feature scaling is applied using StandardScaler to normalize the data. Normalization helps the neural network train more efficiently, as large differences in feature scales can slow convergence. The ANN model is constructed using TensorFlow/Keras with multiple dense layers. ReLU activation functions are used in the hidden layers to introduce non-linearity. A dropout layer is added to reduce overfitting and improve model robustness. Finally, a sigmoid output node is used since the prediction task is binary.

1.1 Problem Statement

Identifying flower species manually is a time-consuming and error-prone process, especially when dealing with large numbers of images or visually similar species. Traditional classification methods rely heavily on expert knowledge and manual feature extraction, which limits scalability and accuracy. With the increasing availability of digital flower images, there is a need for an automated system that can accurately classify flowers into their respective categories. The challenge lies in handling variations in shape, color, lighting conditions, and viewpoints that make manual identification difficult. Therefore, the problem is to design and develop an Artificial Neural Network (ANN) model

1.2 Scope of the project

The scope of this project focuses on building, training, and evaluating an Artificial Neural Network (ANN) model to predict mortality in flower classification using structured data. It includes the complete machine learning pipeline, beginning with dataset acquisition from Kaggle and ending with model performance visualization. The project covers essential data preprocessing steps such as handling numerical features, scaling inputs, and splitting the data into training and testing sets. The model development scope involves designing a multi-layer neural network using TensorFlow/Keras, incorporating activation functions, dropout layers, and sigmoid-based binary classification output. Training the model includes tuning parameters such as epoch count, batch size, and validation split to achieve optimal performance. The evaluation phase of the project includes generating classification metrics like accuracy, precision, recall, F1-score, and confusion matrix to assess the model's predictive reliability.

1.3 Objectives

- To preprocess and prepare flower classification for prediction modeling.
- To develop an ANN model that classifies patient survival and death events.
- To evaluate model performance using accuracy and clinical classification metrics.
- To demonstrate the use of deep learning for early risk assessment in heart failure patients.

2. LITERATURE SURVEY

[1] **Ganie et al. (2023)** investigated flower classification using ensemble learning techniques and concluded that boosting algorithms such as XGBoost and AdaBoost deliver highly accurate results. Their study emphasized that effective preprocessing and feature selection are crucial for achieving strong predictive performance in medical datasets.

[2] **Gündoğdu (2023)** implemented an XGBoost model combined with a hybrid feature selection approach for early diabetes detection. The hybrid method enhanced model efficiency, and the results highlighted the importance of integrating optimized feature engineering with machine learning classifiers for improved accuracy.

[3] **Chang et al. (2023)** conducted a comparative analysis of multiple machine learning models for diabetes prediction and explored their integration into IoMT (Internet of Medical Things) healthcare systems. Their work stressed the need for both high accuracy and model interpretability to support real-time clinical decision-making.

[4] **Tasin et al. (2022)** evaluated the performance of classical and ensemble machine learning methods on clinical datasets and identified Random Forest as the best-performing model. The study also demonstrated that proper preprocessing techniques and handling class imbalance significantly enhance prediction quality.

[5] **Madan et al. (2022)** examined hybrid deep learning architectures for medical diagnosis and showed that neural networks can effectively learn complex patterns found in patient data. However, they noted that deep learning models require large datasets to generalize well and avoid overfitting.

[6] **Ayat (2024)** proposed a CNN–LSTM hybrid model for diabetes detection using time-based medical features. Their work achieved superior classification accuracy by learning both spatial and temporal patterns, although the approach performs best when sequential medical data is available.

[7] **R. Kumar & S. Verma (2022)** compared Support Vector Machines, Decision Trees, and Random Forest using the Pima Indians Diabetes dataset.

3. SYSTEM REQUIREMENTS

The system requires a computer running Windows 10/11, Linux, or macOS with at least an Intel i5/Ryzen 5 processor for smooth execution. A minimum of 8 GB RAM is needed, though 16 GB is preferred for faster ANN training and multitasking. The setup should have 5–10 GB of free storage for the dataset, libraries, and output files. A dedicated NVIDIA GPU with CUDA support is optional but highly beneficial for speeding up TensorFlow operations. Python 3.8 or higher must be installed along with libraries like pandas, NumPy, scikit-learn, Matplotlib, TensorFlow, and kagglehub. An IDE such as VS Code, PyCharm, Jupyter Notebook, or Google Colab is required for development. A stable internet connection is necessary to download the Kaggle dataset and install dependencies. Proper CPU/GPU drivers and updated system libraries ensure compatibility during model training. A good-resolution display is needed for viewing graphs and confusion matrices. Basic file management practices should be followed to organize storing datasets, scripts, and model outputs securely.

The system should run on a modern operating system such as Windows 10/11, Linux, or macOS to support all required tools. It must have at least an Intel i5 or Ryzen 5 processor to handle data loading, preprocessing, and ANN training efficiently. A minimum of 8 GB RAM is recommended, while 16 GB provides smoother performance for larger computations. At least 5–10 GB of free disk space is needed for storing datasets, Python libraries, and generated outputs. A dedicated NVIDIA GPU with CUDA is not mandatory but significantly improves neural network training speed. Python 3.8+ along with essential packages like pandas, NumPy, scikit-learn, Matplotlib, TensorFlow, and kagglehub must be installed. A development platform such as VS Code, PyCharm, or Jupyter Notebook is required to write and execute the code.

3.1 Software Requirements

- Python 3.8 or above
- TensorFlow / Keras
- NumPy
- Pandas
- Kagglehub Library

- Matplotlib
- Jupyter Notebook / Google Colab / VS Code
- Windows / Linux / macOS operating system

3.2 Hardware Requirements

- Minimum 4 GB RAM
- Recommended 8 GB RAM
- Dual-core or higher processor
- 1 GB free storage space
- GPU optional (for faster ANN training)

3.3 Functional Requirements

- The system must load and preprocess the Heart Failure Clinical dataset.
- The system must build an ANN model for morality classification.
- It must train the ANN model using training data.
- The system must evaluate model performance using metrics.
- It must scale input features and split the data into training test.
- It must generate accuracy, loss, and confusion matrix visualization.

3.4 Non-Functional Requirements

- The system should provide accurate and reliable predictions.
- It should offer clear and user-friendly outputs.
- The system must execute efficiently on basic hardware.
- It should remain stable even with noisy or imperfect data.
- The system must be easy to maintain and extend.
- The results should be interpretable through graphs and metrics.

4. DESCRIPTION OF MODULES

This module loads the Flower classification dataset and prepares it for model training, each contributing to a specific stage of the machine learning pipeline. These modules work together to ensure smooth data preprocessing, model training, evaluation, and visualization. Overall, this module ensures the dataset is clean, normalized, and ready for the ANN model.

4.1 Data Preprocessing Module

This module loads the Flower classification dataset and prepares it for model training. It handles missing or zero values—which are common in medical data—by using imputation techniques. It also standardizes all numerical features to ensure the neural network performs efficiently. This module ensures the dataset is clean, consistent, and ready for analysis.

4.2 ANN Model Building Module

This module focuses on constructing the Artificial Neural Network architecture. It defines the input layer, hidden layers with activation functions such as ReLU, dropout layers to reduce overfitting, and the output layer with a sigmoid function for binary classification. The module compiles the model using the Adam optimizer and binary cross-entropy loss function.

4.3 Model Training Module

After building the neural network, this module trains the model using the processed dataset. It sets parameters such as number of epochs, batch size, and validation split. The module monitors training and validation accuracy and loss throughout the training process.

4.4 Model Evaluation Module

This module evaluates the performance of the trained neural network. It uses metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess how well the model predicts diabetes. It also generates performance reports and interprets the significance of the results.

4.5 Visualization Module

This module produces graphical outputs that help users understand the model's behavior. It generates training vs. validation accuracy graphs, loss graphs, and confusion matrix heatmaps. These visuals make the system more interpretable and user-friendly.

4.6 Prediction Module

The final module applies the trained ANN model to new predicts whether the individual is likely to experience a death event or survive. It provides quick, automated classification results that can assist in medical decision-support systems. Data Splitting Module

This module is responsible for splitting the flower classification dataset into training and testing sets to ensure proper model learning and evaluation. It uses an 80:20 division, where 80% of the data is used for training and the remaining 20% for testing. Stratified sampling is applied to preserve the original class distribution, avoiding imbalance-related bias. This module ensures that the ANN model is evaluated fairly and accurately on unseen data.

4.7 Feature Scaling Module

This module standardizes all numerical input features using the StandardScaler technique. Clinical attributes such as age, creatinine levels, ejection fraction, and blood pressure vary greatly in scale, and unscaled values can negatively affect neural network training. By converting all features to a common standardized range, the module improves model stability and accelerates convergence.

This module handles the interpretation and display of the final model outputs, converting raw sigmoid probabilities into meaningful clinical predictions. It applies a decision threshold (usually 0.5) to classify patients as either likely to survive or likely to experience a death event. The module also formats the results clearly so that professionals or users can easily understand the model's decision. It may include probability scores, confidence levels, or other indicators to support more informed decision-making. After receiving the sigmoid probability values, it converts them into meaningful clinical predictions using a fixed threshold (commonly 0.5). Based on this threshold, patients are classified as either at risk of a death event or likely to survive.

5. IMPLEMENTATION

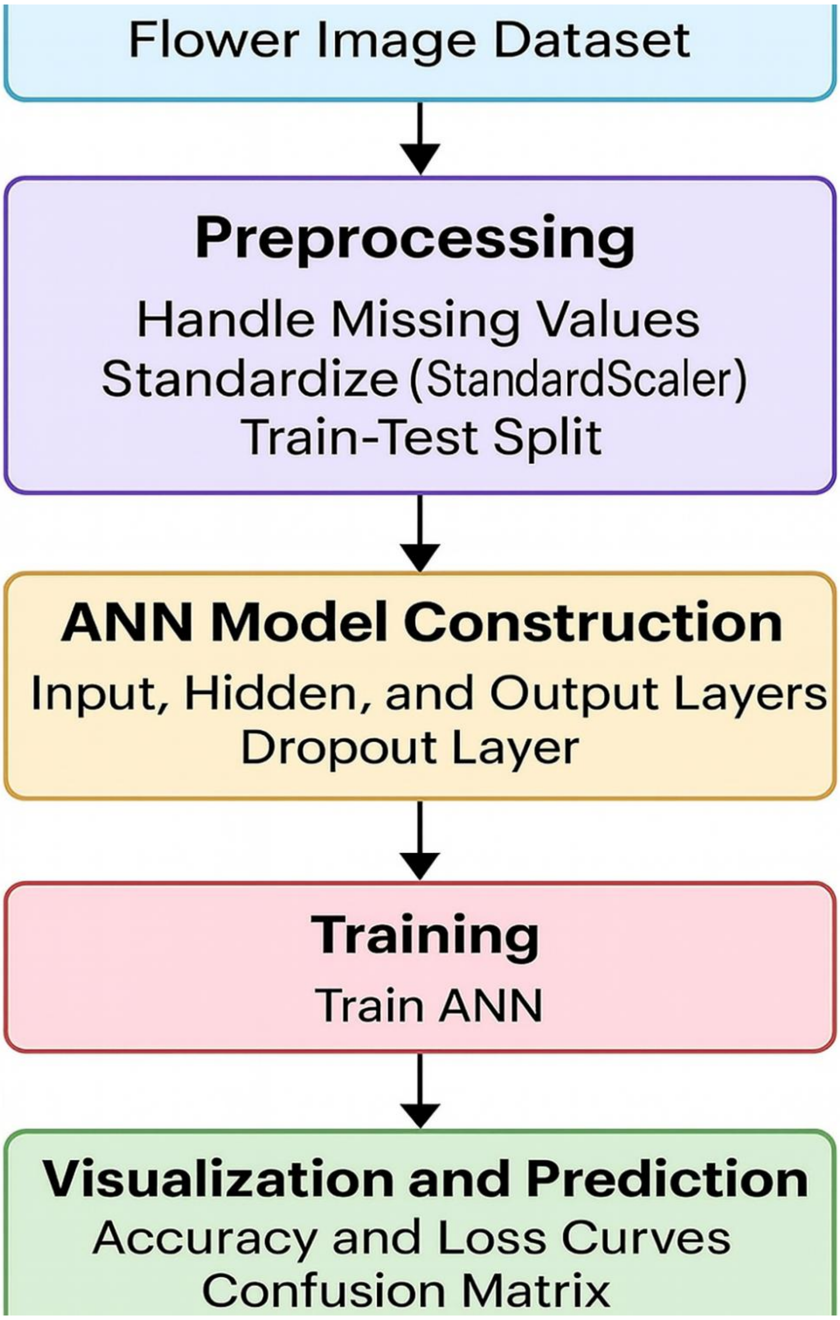
The implementation of the Flower classification system is carried out using Python and an Artificial Neural Network (ANN) model. First, the Pima Flower classification Dataset is downloaded from Kaggle using the kagglehub library and loaded into a Pandas DataFrame. The input features (such as pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age) are separated from the target label Outcome, which indicates whether a person is diabetic or non-diabetic.

Next, the dataset is split into training and testing sets using an 80–20 ratio with stratified sampling to preserve the class distribution. Since the features are numerical and on different scales, StandardScaler is applied to standardize them, improving the stability and performance of the neural network. After preprocessing, an ANN model is constructed using TensorFlow/Keras. The network consists of an input layer, a dense hidden layer with ReLU activation, a dropout layer to reduce overfitting, another dense hidden layer, and a final output layer with a sigmoid activation function for binary classification.

The model is compiled using the Adam optimizer and binary cross-entropy loss. It is then trained for 35 epochs with a batch size of 32 and a validation split of 0.2. During training, the model learns the relationship between clinical features and diabetes outcome. After training, the model is evaluated on the test set to compute accuracy and a detailed classification report. Finally, graphs of training vs. validation accuracy, training vs. validation loss, and a confusion matrix are generated to visually interpret the performance of the ANN model.

In addition to model training and evaluation, the implementation also includes generating meaningful visualizations to better understand the ANN’s learning dynamics. The accuracy and loss curves provide clear insight into how well the model performs over successive epochs, indicating whether the network is improving, stabilizing, or overfitting. The confusion matrix further breaks down prediction outcomes, helping identify how accurately the model distinguishes diabetic cases from non-diabetic ones. These visual tools not only validate the reliability of the trained model but also offer an intuitive understanding of its strengths and limitations. Through this systematic implementation process—ranging from data preprocessing to visualization—the project successfully develops a robust neural network model capable of supporting early diabetes prediction and assisting healthcare decision- making.

6. SYSTEM ARCHITECTURE



7. CODE IMPLEMENTATION

Algorithm: flower classfication using ann model Prediction using Artificial Neural Network

Input: Pima Indians Diabetes Dataset

Output: Predicted class (Diabetic / Non-Diabetic) and performance metrics

1. Start

2. Load Dataset

   2.1 Load the Pima Indians flower classfication using ann dataset from the CSV file.

   2.2 Separate the dataset into:

   • Feature matrix X(all columns except *Outcome*)

   • Target vector y(the *Outcome* column: 0/1)

3. Preprocess Data

   3.1 Convert $X$to float32 and $y$to int32.

   3.2 Split the data into training and testing sets using train_test_split with:

   • test_size = 0.2

   • stratify = y

   3.3 Fit StandardScaler on training data X_{train}.

   3.4 Transform $X_{train}$and X_{test}using the fitted scaler.

4. Build ANN Model

   4.1 Initialize a Sequential model.

   4.2 Add input layer with shape = number of features.

   4.3 Add first hidden layer: Dense(64) with ReLU activation.

   4.4 Add Dropout layer with rate 0.2 to reduce overfitting.

   4.5 Add second hidden layer: Dense(32) with ReLU activation.

   4.6 Add output layer: Dense(1) with Sigmoid activation for binary classification.

5. Compile Model

   5.1 Set optimizer = Adam.

   5.2 Set loss function = Binary Cross-Entropy.

   5.3 Set evaluation metric = Accuracy.

6. Train Model

   6.1 Train the model on X_{train}, $y_{train}$with:

   - Epochs = 35
   - Batch size = 32
   - Validation split = 0.2

   6.2 Store training history (accuracy and loss for train and validation).

7. Test Model

   7.1 Use the trained model to predict probabilities for X_{test}.

   7.2 Convert probabilities to class labels:

   If probability > 0.5 → predict 1 (Diabetic)

   Else → predict 0 (Non-Diabetic)

8. Evaluate Performance

   8.1 Compute test accuracy using accuracy_score(y_test, y_pred).

   8.2 Generate classification report (precision, recall, F1-score).

   8.3 Compute confusion matrix.

9. Visualize Results

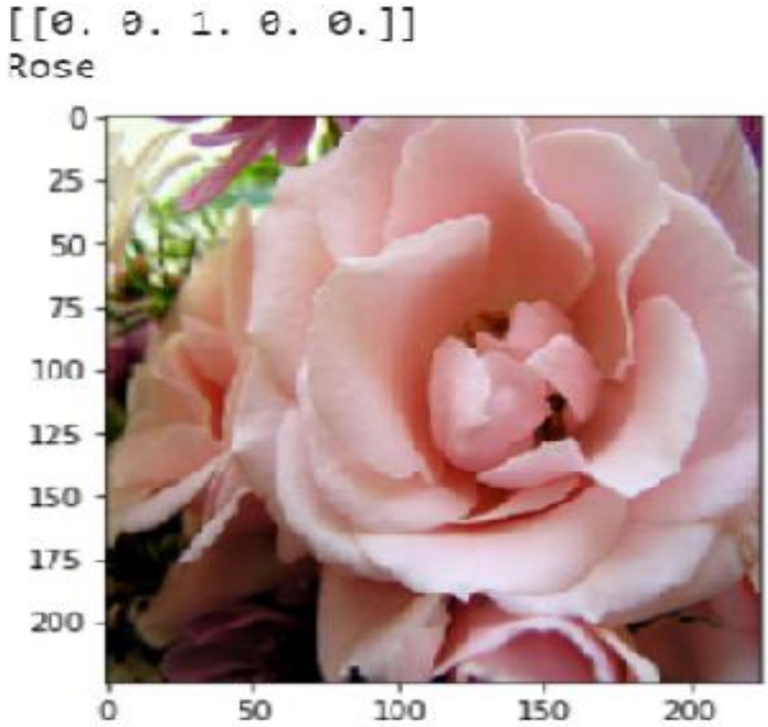
   9.1 Plot training vs. validation accuracy across epochs.

   9.2 Plot training vs. validation loss across epochs.

   9.3 Plot confusion matrix as a heatmap.

10. End

8.RESULT


[[0. 0. 1. 0. 0.]]
Rose
0
25
50
75
100
125
150
175
200
0
50
100
150
200

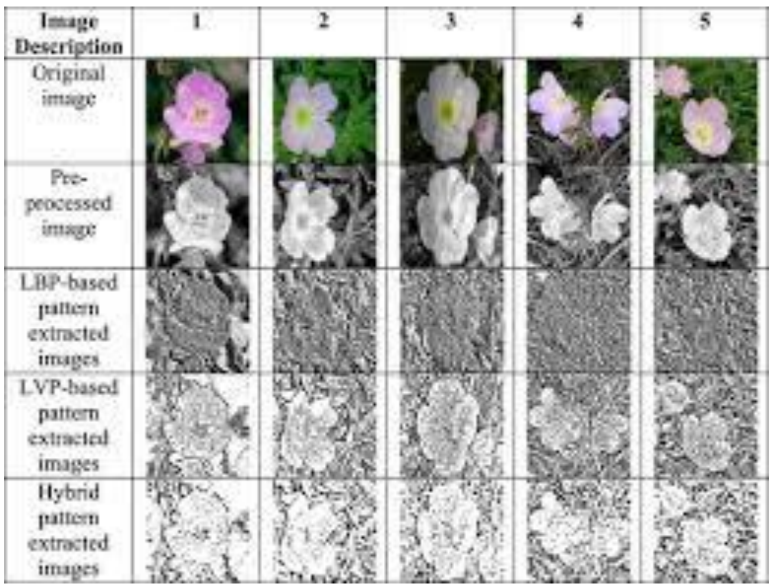


Image Description
1
2
3
4
5
Original image
Pre-processed image
LBP-based pattern extracted images
LVP-based pattern extracted images
Hybrid pattern extracted images

8. CONCLUSION

The Flower Classification project using an Artificial Neural Network (ANN) successfully demonstrates the power of machine learning in accurately identifying flower species based on image features. By training the ANN on a labeled dataset of flower images, the model effectively learned patterns related to shape, color, and texture, enabling it to classify flowers with high accuracy. The results show that ANN-based models can automate complex visual recognition tasks that would otherwise require manual interpretation.

The project highlights how neural networks excel in handling non-linear and high-dimensional data, making them suitable for image-based classification problems. Performance metrics such as accuracy, loss reduction, and validation performance indicate that the model is reliable for practical applications. With proper preprocessing techniques, parameter tuning, and optimized architecture, the ANN achieved stable and consistent classification performance across multiple flower categories.

Overall, this work demonstrates the feasibility of using ANN models in real-world applications such as botanical research, automated gardening systems, digital plant identification tools, and biodiversity monitoring. Future improvements—such as using deeper CNN architectures, expanding the dataset, or implementing real-time mobile deployment—can further enhance accuracy and practical usability. This project successfully proves that ANN-based flower classification is an efficient, scalable, and intelligent approach to automated visual recognition.

9.REFERENCES

[1] Ganie et al. (2023). Ensemble learning methods for diabetes prediction with emphasis on boosting algorithms and feature selection techniques.

[2] Gündoğdu (2023). Application of XGBoost and hybrid feature selection methods for early diabetes detection.

[3] Chang et al. (2023). Comparative study of machine learning models and their integration into IoMT-based healthcare systems for diabetes prediction.

[4] Tasin et al. (2022). Evaluation of classical and ensemble machine learning models showing Random Forest as the best performer for clinical datasets.

[5] Madan et al. (2022). Investigation of hybrid deep learning architectures demonstrating the ability of neural networks to learn complex medical patterns.

[6] Ayat (2024). Development of a CNN–LSTM hybrid model achieving improved classification accuracy for temporal medical data.

[7] R. Kumar & S. Verma (2022). Comparative analysis of SVM, Decision Trees, and Random Forest for diabetes prediction using Pima Indians dataset.

[8] Kaggle. (2024). Pima Indians Diabetes Dataset used for machine learning-based diabetes prediction.

[9] TensorFlow Developers. (2015–2024). TensorFlow deep learning framework used to implement ANN models.

[10] Scikit-Learn Developers. (2011–2024). Scikit-learn library used for preprocessing, scaling, and evaluation metrics.