Project : Image Classification using artificial neural networks

Google drive folder : https://rb.gy/stuvws

Introduction and Overview:

1 - Project idea and overview.

- Image classification is a fundamental task in computer vision, where the goal is to automatically categorize images into predefined classes or labels. In this project, we'll build an image classification system using Artificial Neural Networks (ANNs), a type of deep learning model, to classify images into different categories. The project will involve preprocessing image data, building and training the ANN model, and evaluating its performance on a test dataset.

2 - Similar applications in the market

Image Classification-App on Google Play

This app based on Artificial Intelligence technology to classify the object in the image

3- Articles

1. "ImageNet Classification with Artificial Neural Networks"

- **Authors:** Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton
- **Published in:** Advances in Neural Information Processing Systems (NIPS), 2012.
- **Link:** https://proceedings.neurips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Summary:
This seminal work introduces the utilization of Artificial Neural Networks (ANNs) for large-scale image classification. The authors present the AlexNet architecture, a deep neural network consisting of fully connected layers. The paper demonstrates the effectiveness of ANNs in image recognition tasks and highlights the importance of deep learning techniques for achieving state-of-the-art results.

---

2. "Going Deeper with Artificial Neural Networks"

- **Authors:** Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich
- **Published in:** IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
- **Link:** Going deeper with convolutions | IEEE Conference Publication | IEEE Xplore

Summary:
This paper introduces the concept of increasing the depth and width of Artificial Neural Networks (ANNs) for improved accuracy in image classification. The authors present innovative architectures for deep networks, focusing

on the depth of fully connected layers. The paper underscores the significance of expanding the capacity of ANNs for capturing intricate features in images.

---

3. "Very Deep Artificial Neural Networks for Large-Scale Image Recognition"

- **Authors:** Karen Simonyan, Andrew Zisserman
- **Published in:** International Conference on Learning Representations (ICLR), 2015.
- **Link:**https://www.researchgate.net/publication/265385906_Very_Deep_Convolutional_Networks_for_Large-Scale_Image_Recognition

Summary:
This research introduces the VGG (Visual Geometry Group) network architecture, emphasizing the simplicity and depth of Artificial Neural Networks (ANNs). The authors systematically explore the impact of network depth on image classification performance, using fully connected layers. The paper provides insights into the advantages of deeper ANNs for image recognition tasks.

---

4. "Image Classification with Artificial Neural Networks: Enhancing the Learning Process"

- **Authors:** Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi
- **Published in:** IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- **Link:** Deep Learning based Feature Extraction for Texture Classification - ScienceDirect

Summary:
Building upon the principles of Artificial Neural Networks (ANNs), this paper presents an improved methodology for image classification. The authors introduce enhancements to the learning process, focusing on fully connected layers. Techniques such as batch normalization contribute to improved training convergence and overall accuracy in image recognition tasks.

---

5. "Rethinking the Design of Artificial Neural Networks for Computer Vision"

- **Authors:** Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna
- **Published in:** Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- **Link:**https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf

Summary:
This paper revisits the design principles of Artificial Neural Networks (ANNs) for computer vision tasks. The authors present a refined architecture, Inception-v3, focusing on factorizing fully connected layers and reducing the number of parameters. The result is an efficient ANN model that achieves state-of-the-art performance in image classification.

Proposed Solution & Dataset:

1 - Main Functionalities

- The main functionality of our project , the model takes an image as input and feed it to the model we trained , then the model processes the image and outputs the prediction using the extracted features during the training

phase in our model . After the predicted output the accuracy metric is evaluated by comparing between the predictions and the true labels.

2 - Dataset

Overview:

- The **German Traffic Sign Benchmark** is a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011. We cordially invite researchers from relevant fields to participate: The competition is designed to allow for participation without special domain knowledge

Dataset Name: GTSRB - German Traffic Sign Recognition Benchmark

Context

This dataset contains images of the following food items:.

- Single-image, multi-class classification problem
- More than 40 classes
- More than 50,000 images in total
- Large, lifelike database

Contents

- This dataset is split into two sets:

1. train (80%)
2. test (20%)

Dataset Source:

- https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

Applied Algorithms:

We used the sequential model aka fully connected layers which are linearly stacked layers , This means that the output of one layer serves as the input to the next layer forming a chain like network.

Detailed description

Data Loading and Splitting

- Initially we imported the packages and libraries that we would need through out the project
- Then we loaded the dataset , split it into training set and testing set , we transformed the images in an array shape and we resized the images loaded

Data Preprocessing

- Then we reshaped the data using the .reshape function to make our array shape a suitable consistent input for our neural network
- Also we Convert the image to a NumPy array, Add an extra dimension to the array

- Additionally we normalized our data to bring all features to similar scale to prevent certain features from dominant others , we did this to improve convergence in our neural network and to normalize the data between 0 to 1

Model Architecture

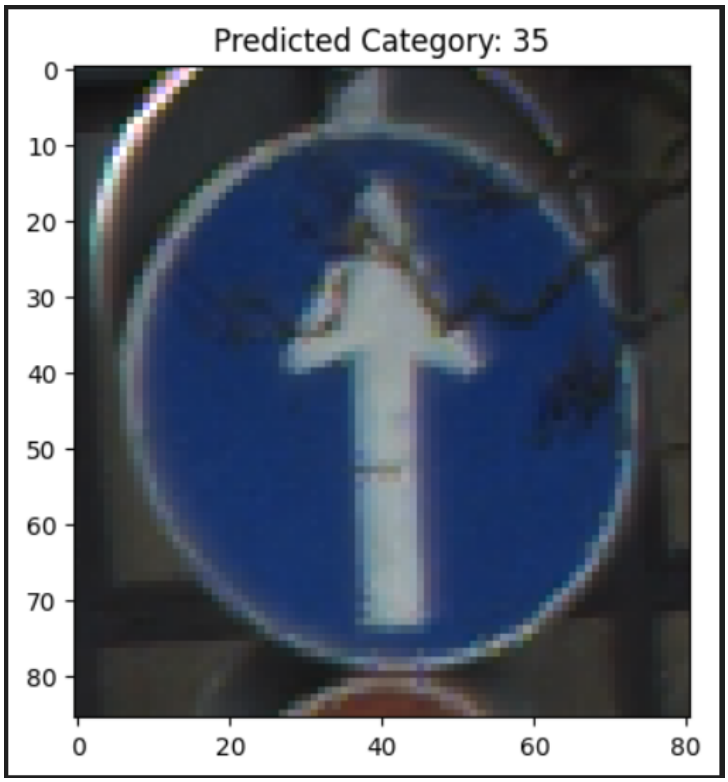
- We used the sequential model aka fully connected layers which are linearly stacked layers , This means that the output of one layer serves as the input to the next layer forming a chain like network.
- The first layer in our network is the flatten layer to transform our image's multidimensional arrays into a one dimensional array to make it a suitable format for dense layer later on
- The second layer is the dense layer which is a fully connected layers , each layer takes input from the previous one associated with weights and biases (those weights and biases are learned during the whole dense layers training processes)
- In the second layer we used the Relu activation function (which is simple and efficient for various neural networks) to facilitate efficient training by avoiding obstacles like vanishing gradient and to handle biases better.
- The third layer is the batch normalization layer , we used it to accelerate the training process by normalizing each input for each layer .
- The forth layer is another dense layer to make us obtain more abstract high-level features that can be better for the model in the automated feature extraction and to make accurate predictions , dense layers is so important through out the process as it makes the model better understand more complex patterns in data and extract abstract features .
- The fifth layer is a drop out layer which is a regularization step used to prevent overfitting and not being biased to training data and also this layer introduces noise and random fluctuations this step reduces overfitting by randomly setting a fraction of input units to zero during training
- The sixth layer is a another dense layer , this layer is our output layer . the number of neurons in this layer must be the same number of our classes as each neuron corresponds to a class , we used the softmax activation function which is commonly used in multi-class classification problems as it converts the raw model outputs (logits) into probabilities by normalizing output to sum to one in order the indicate the probability of input belonging to a certain predicted class (one neuron of the ten neurons)
- Finally the model compilation step to configure the training process by specifying the optimizer which is SGD (stochastic gradient descent) that optimizes the weights during training , this optimizer introduces noise to help escape local minima (parameters optimization) .we chose the sparse category entropy loss function which is suitable with the output layer's activation function softmax to quantify the difference between the model's predictions and the true labels during training . we chose accuracy as a performance metric for our model , which measures the proportion of correctly classified samples.

Model Training

- We fit the model with our the training set to start the training phase .

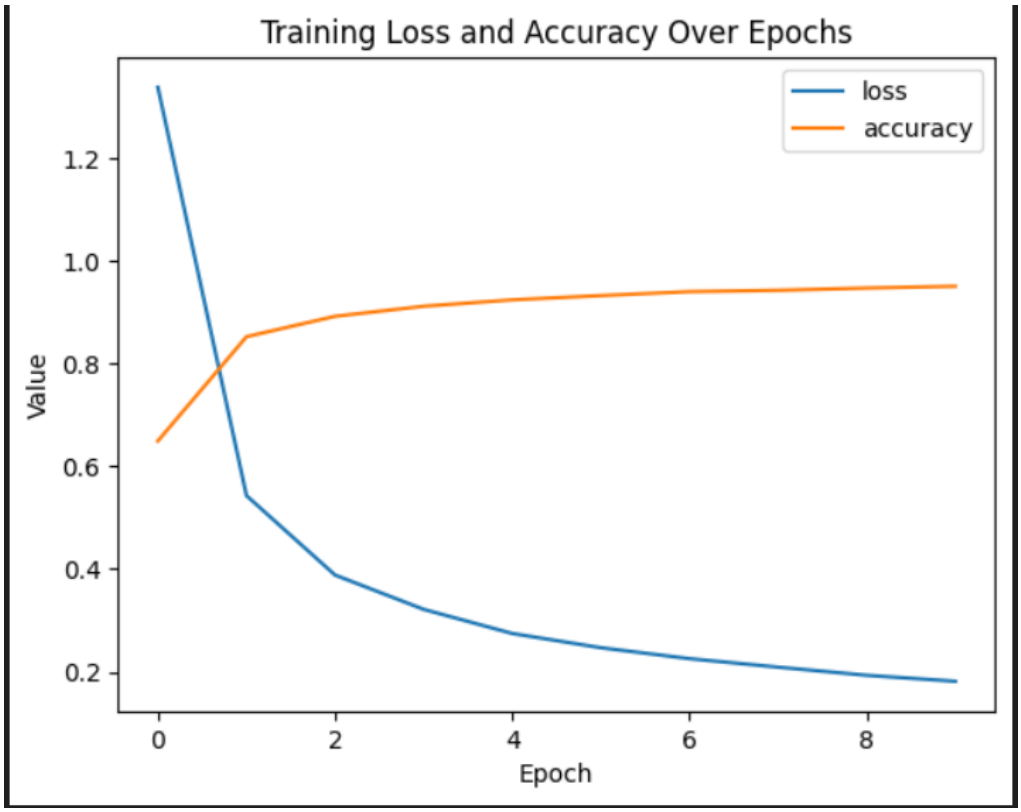
Experiments & Results:

Experiments

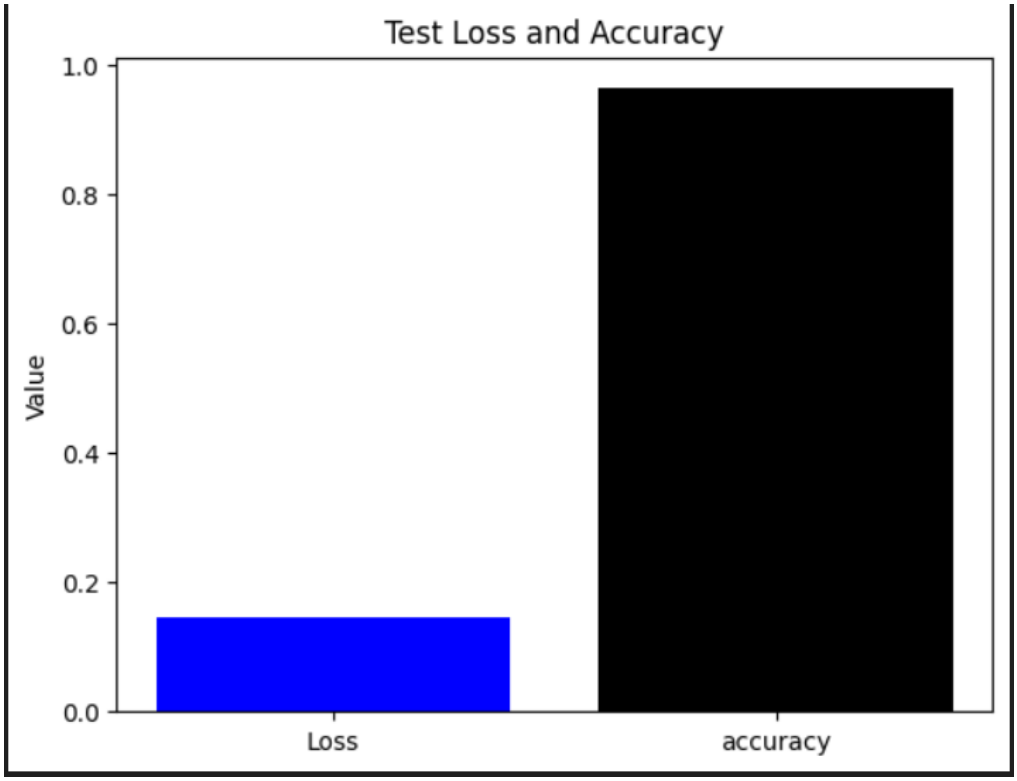


Results

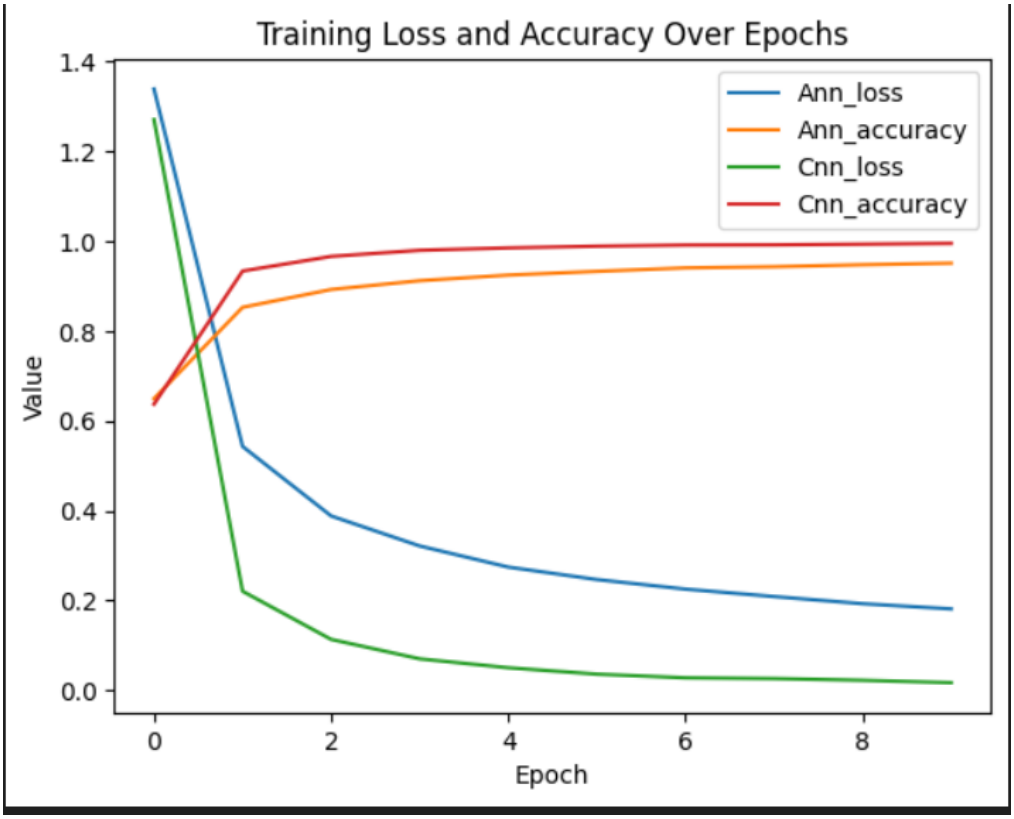
Plotting loss and accuracy



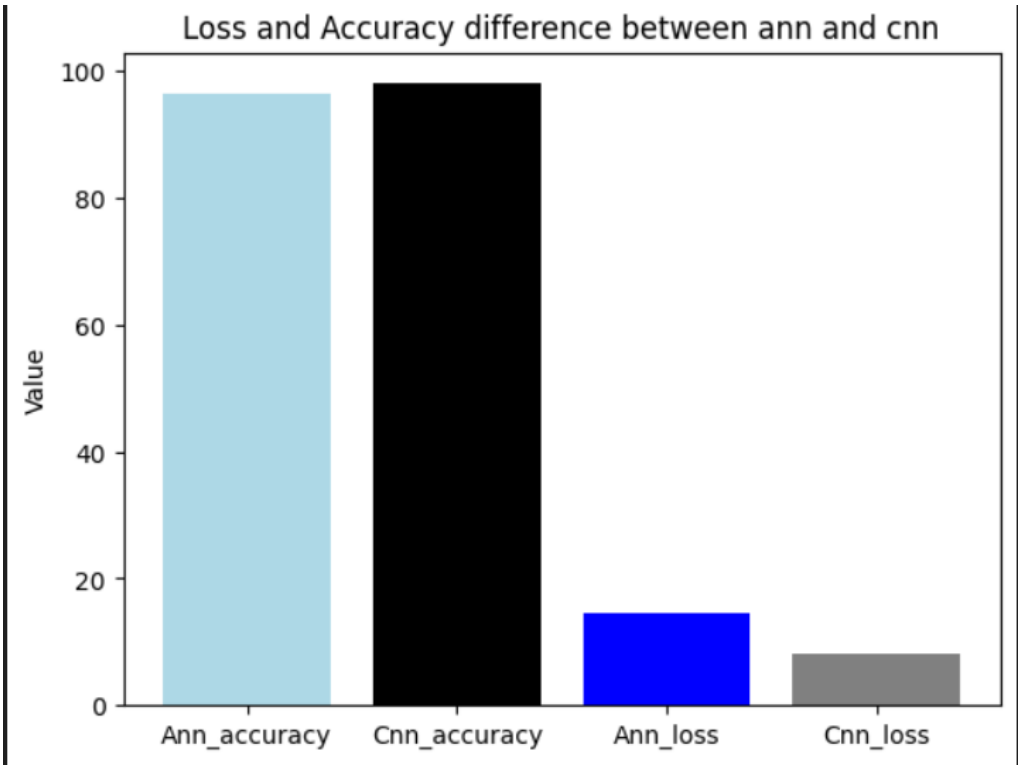
Bar Plot between loss and accuracy of test set



Plotting accuracy and loss difference between ANN and CNN



Bar plot between accuracy and loss difference between ANN and CNN



Analysis, Discussion, and Future Work:

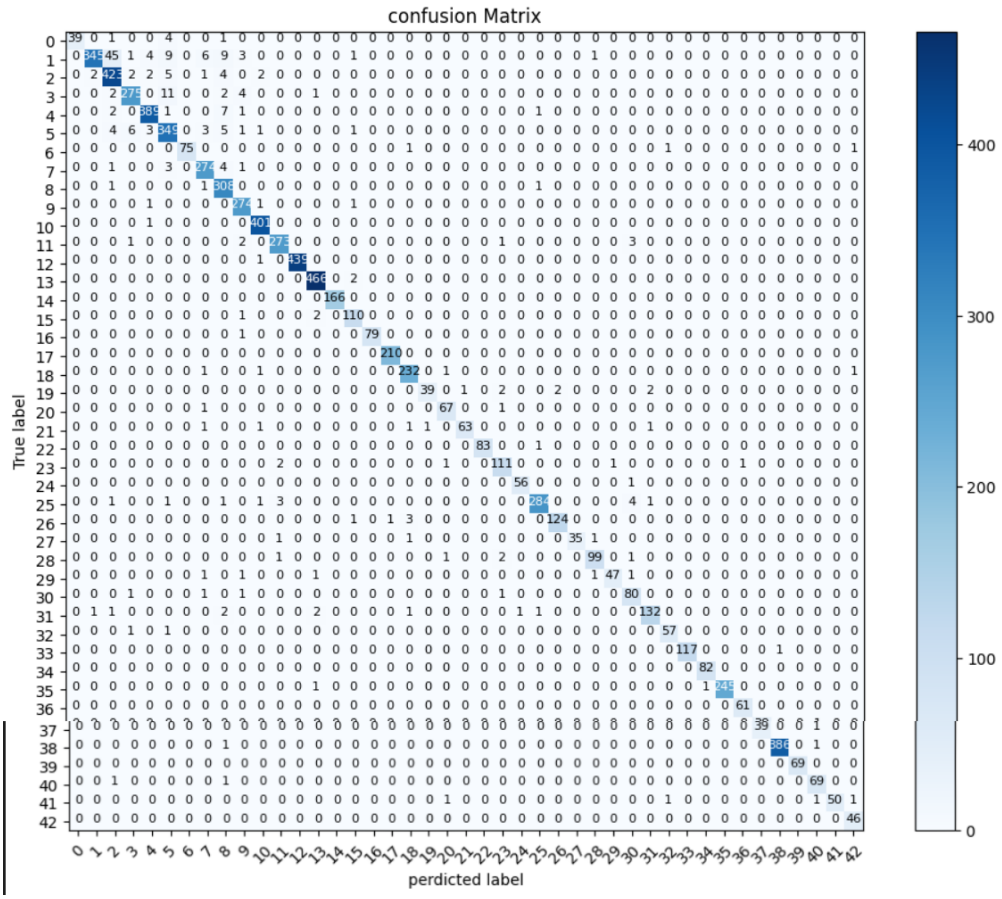
1-Analysis of the results, what are the insights?

- **Accuracy and Performance:**

  model accuracy 96%

- **confusion matrix:**

1. **Diagonal**: The numbers on the diagonal represent the correctly classified images.
2. **Off-diagonal**: The numbers off the diagonal represent the misclassified images.



2-What are the advantages / disadvantages?

Advantage:

Model can predict the image class by generating the image labeled by it's predicted class .

Disadvantage:

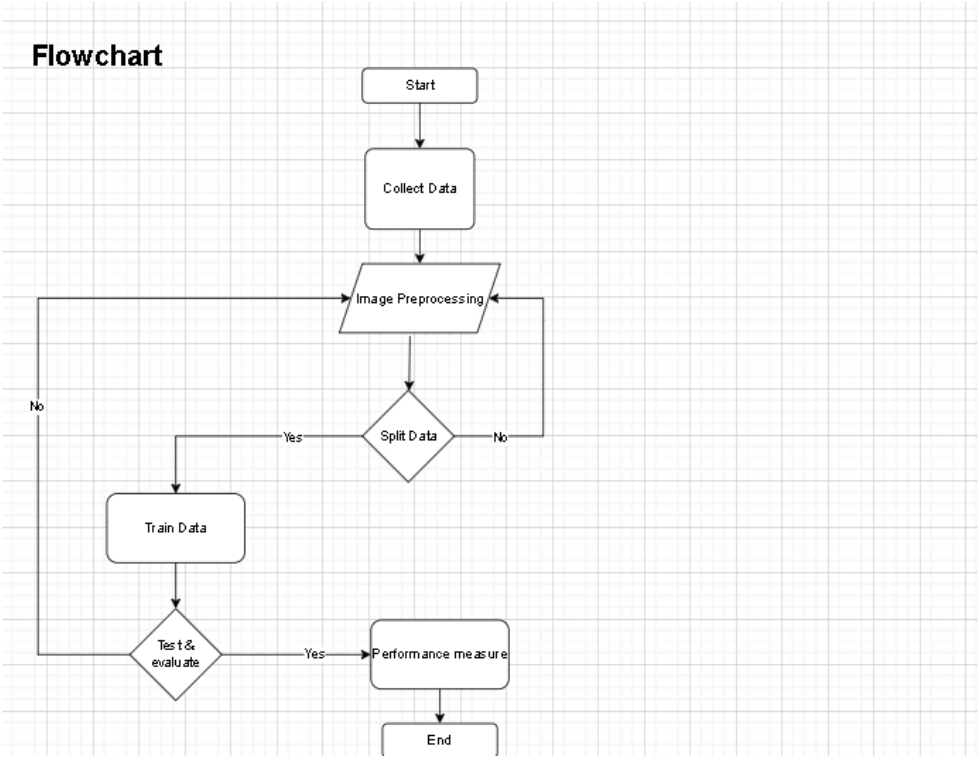
Computational Complexity: Training deep neural networks can be computationally expensive, especially for our large dataset .

3-Why did the algorithm behave in such a way? What might be the future modifications you'd like to try when solving this problem?

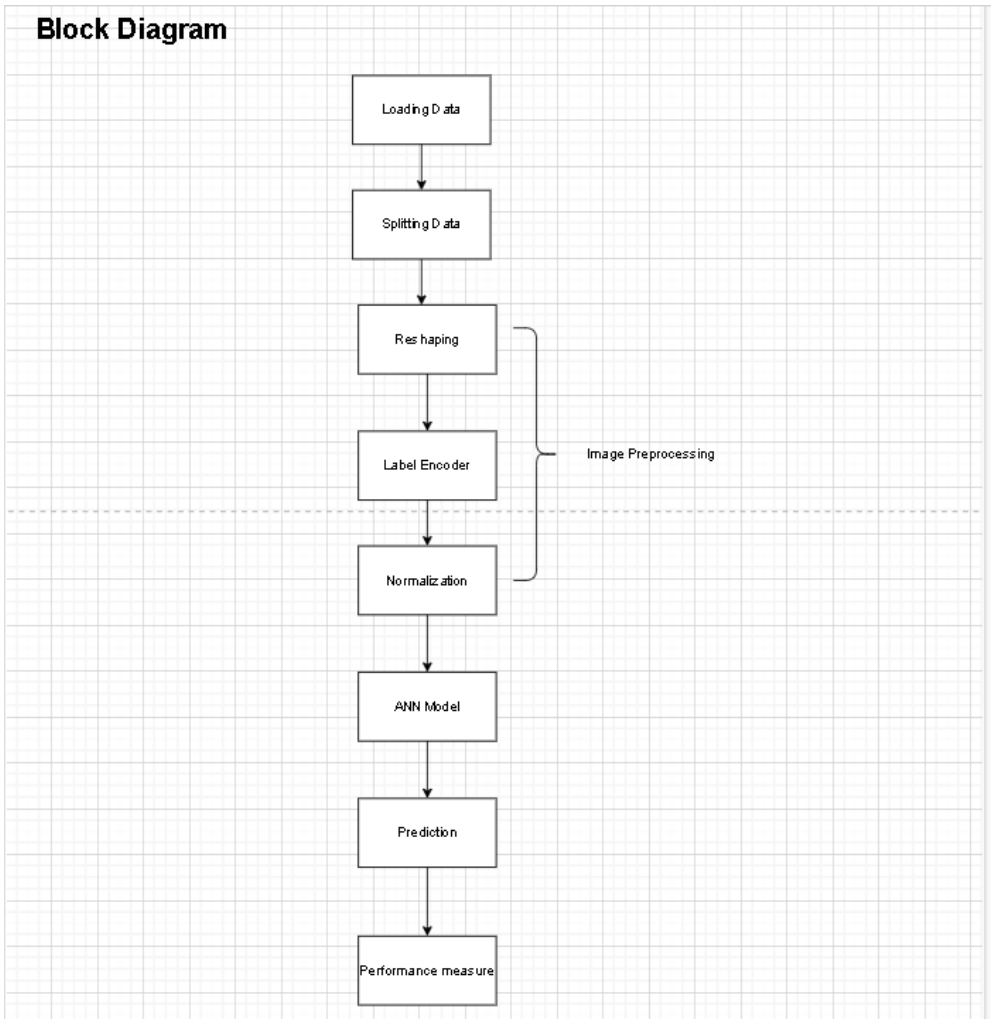
The model predicted labels with accuracy 96 , this is obtained by the applying suitable preprocessing steps to our dataset and the architecture in the model designing process . The future modification that will provide us with higher accuracies is using convolutional layers in our model .

Extra Visualization to explain Sequence of the model:

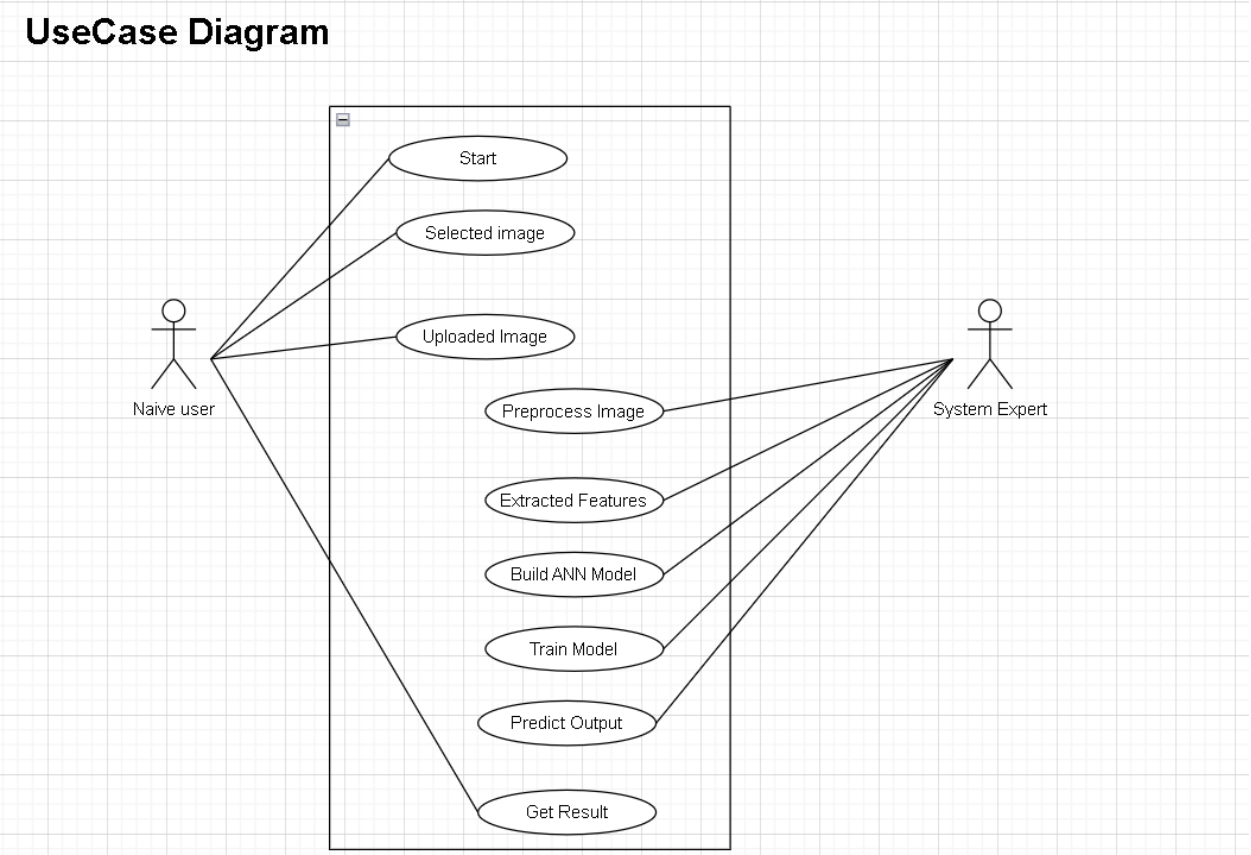
1-FlowChart



2-Block Diagram


Block Diagram
Loading Data
Splitting Data
Reshaping
Label Encoder
Normalization
Image Preprocessing
ANN Model
Prediction
Performance measure

3-UseCase



Shared Folder

https://rb.gy/stuvws

https://drive.google.com/drive/folders/1a6d1k2N-aU-jHbP3vM2m7ULdOij51Wpe

https://github.com/marwanali213/Automated-object-detection-using-artificial-neural-network-/tree/main