LEVERAGING ENSEMBLE LEARNING FOR CLASSIFICATION OF OLD PEOPLE AND KIDS

A Thesis Submitted
In Partial Fulfillment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

SIGNAL PROCESSING AND DIGITAL DESIGN

Submitted by:

KHUSNUL KHOTIMAH

(Roll No. 2K22/SPD/10)

Under the supervision of

DR AVINASH RATRE

Department of Electronics and Communication Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Bawana Road, Delhi-110042, India

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I, Khusnul Khotimah, Roll No 2K22/SPD/10, student of M.Tech. (Signal Processing and Digital Design), hereby certify that the work which is being presented in "Leveraging Ensemble Learning for Classification of Old People and Kids" which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not been previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or any other similar title or recognition.

Place: Delhi
Date: May, 2024 **(KHUSNUL KHOTIMAH)**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

I hereby certify that the Project Dissertation titled "Leveraging Ensemble Learning for Classification of Old People and Kids" which is submitted by Khusnul Khotimah, Roll No 2K22/SPD/10, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **DR. AVINASH RATRE**

Date: May, 2024 **(SUPERVISOR)**

ABSTRACT

This study evaluates the effectiveness of ensemble learning methods for classifying older people and kids, with a focus on Random Forest (RF) and AdaBoost with Decision Tree (DT) Estimator. Both age groups' set of labelled images will make up the training data and the evaluation dataset. The images will be utilized for training the models to identify features that set old people and kids apart.

By merging several weak learners into a strong classifier, this study hopes to improve classification accuracy and robustness by leveraging the potential of ensemble approaches. While the AdaBoost method focuses on iteratively modifying the weights of misclassified instances to enhance model performance, the Random Forest approach builds a forest of decision trees and combines their predictions to achieve accurate classification.

The first phase of this investigation entailed building a large dataset including images of both kids and older individuals. "Old People" and "Kids" were two separate classes created from the information. In order to ensure the dataset's inclusivity and representativeness, a variety of sources were used to gather a wide range of images.

The dataset was subsequently divided into training and validation subsets using the proper technique to make sure the model could generalize successfully to new data. Preprocessing techniques were used to standardize the images once the dataset had been prepared. This required scaling the images to a constant resolution and normalizing the pixel intensities of the images. These preprocessing procedures are essential for creating a standardized input format and improving model performance.

Following training, a variety of evaluation metrics, including accuracy, precision, recall, and F1-score, will be used to evaluate the models' performance. These metrics provide insight on the models' ability differentiate between old people and kids. The evaluation's findings will then be compared to determine which algorithm is better at identifying old people and kids. Which method is better suited for this particular task can be determined with the aid of this comparison.

The evaluation metrics from the previous project are compared with those from Support Vector Machines (SVM) and Convolutional Neural Networks (CNN), including accuracy, precision, recall, and F1-score. The results demonstrate the

effectiveness of ensemble learning approaches, with Random Forest performing the best overall, demonstrating superior performance across all metrics. AdaBoost with Decision Tree Estimator also performed well and was competitive with Random Forest.

These results demonstrate ensemble learning's potential for accurate classification tasks and offer insightful information for further study and use in a variety of fields.

AKNOWLEDGMENT

First and foremost, I express my deepest gratitude to the Almighty for bestowing upon me the strength, guidance, and opportunities to embark on this academic pursuit. I wish to express my sincere thanks and heartfelt gratitude to Dr. Avinash Ratre for his indispensable guidance, valuable time, crucial suggestions, corrections, and encouragement that accounts for the successful completion of this project. He has always been a source of inspiration. I also extend my gratitude to Prof. O.P Verma (HOD, Dept. of ECE) for giving me this opportunity.

I also want to express our gratitude to all the teaching and non-teaching personnel at Delhi Technological University's Electronics and Communication Department for their direct and/or indirect input assistance during the project.

I am immensely grateful to the Indian Council for Cultural Relations (ICCR) for their generous scholarship support, which has enabled me to pursue my studies and fulfill my academic aspirations.

Finally, my heartfelt thanks extend to my family for their unwavering love, encouragement, and sacrifices throughout this endeavor. Their steadfast support has been my source of inspiration and motivation, driving me to overcome challenges and strive for excellence. I am truly blessed to have their unwavering support and guidance on this remarkable journey.

(KHUSNUL KHOTIMAH)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ML	Machine Learning
RF	Random Forest
AdaBoost	Adaptive Boosting
DT	Decision Tree
SVM	Support Vector Machine
CNN	Convolutional Neural Network

CHAPTER 1
INTRODUCTION

Because the human face is considered as a particularly rich source of information, recognizing facial features has recently become one of the most exciting challenges in pattern recognition. Age is face characteristics in particular that have a wide range of uses [1]. Age estimate is a significant issue in the fields of computer vision and pattern recognition, which have attracted a lot of attention recently. Age estimate aims to determine a person's chronological age from their appearance [2]. In several fields, such as ad serving, visual monitoring, ageing trend analysis, demographics analysis, and content-based research, age categorization from facial images is significant. Research in this area has produced impressive results. Age categorization in real-world applications encounter a number of difficulties, such differences in facial expressions, low-resolution images, lighting, multi-ethnicity, and various ageing patterns among people. Age categorization is still an important subject for study despite advances in computer vision, requiring the creation of precise algorithms that only use face images [3].

Any living thing that wants to do an action has to be competent at it. People are able to differentiate one thing from another with easily in their daily tasks, starting in the morning. In order to make a computer as intelligent as a person, several methods have been devised. But no one categorization strategy has been shown to have the greatest accuracy across all data set types. The application of machine learning technologies has increased dramatically in several scientific domains in recent years. One definition of machine learning (ML) is a type of artificial intelligence (AI) made up of programmable machines that are accessible and can be educated without really being programmed. Machine learning algorithms are used to obtain useful information from datasets, identify hidden patterns, comprehend data structure, and generate predictions and automate decision-making. Machine learning algorithms fall into several categories and are applied to categorization tasks. Some popular techniques for implementing classification include ensemble learning, decision trees, and classification based on perceptron, support vectors, and statistical learning techniques. Several kinds of data categorization issues were resolved with the aid of machine learning methods. A machine learning approach known as ensemble modelling combines several classifiers to increase precision and lower variance in decision-making. Research suggests that combining machine learning algorithms with ensemble modelling approaches may create a robust classifier to produce more accurate predictions with the least amount of classification error risk [4][5].

Two popular ensemble learning algorithms, Random Forest and AdaBoost, are well-known for the ability to integrate several weak learners, usually decision trees, to create a more powerful prediction model. Ensemble approaches are a useful tool for reducing overfitting and enhancing generalization performance because they use the variability of individual learners and aggregate their predictions [5].

This project's main goal is to determine how effectively the Random Forest and AdaBoost with Decision Tree estimator perform while classifying old people and kids based on visual characteristics from images. The effectiveness of ensemble models will also be compared to other traditional methods like Support Vector Machine (SVM) and Convolutional Neural Network (CNN), which have been used in previously for comparable classification tasks.

The efficacy of the proposed ensemble learning techniques will be evaluated using standard evaluation metrics, including accuracy, precision, recall and F1-score. These measures enable a thorough comparison between several models by offering full insights into classification performance. We seek to offer important insights into the effectiveness of ensemble techniques for image classification tasks involving kids and older people by comparing the performance of ensemble learning methods with traditional classifiers like SVM and deep learning-based approaches like CNN.

CHAPTER 2

LITERATURE REVIEW

2.1 Machine Learning

Artificial intelligence (AI) in the form of machine learning teaches computers to think like people by building on and learning from past experiences. It analyses data and detects trends with the least amount of human interaction. In the field of machine learning, several computer algorithms are available to address certain problems, and these algorithms continue to become better over time as long as the dataset remains high-quality. These techniques create a foundation model and then add a training dataset to it. Machine learning algorithms extract useful information from datasets, find hidden patterns, comprehend data structure, and generate predictions and automate decision-making [5][6].

The objective of computer vision and machine learning is to provide computers the capacity to collect and comprehend data and make decisions based on past and current outcomes. The Internet of Things, Industrial Internet of Things, and human-computer interfaces all depend on computer vision. Complex human activities in multimedia streams are recognized and tracked using computer vision and machine learning algorithms. There are three forms of learning for the data analysis and prediction task: semi-supervised, unsupervised, and supervised [6].

Various methods of learning depend on the way algorithms extract more complex information from the raw input by using multiple layers. A number of methods for learning are shown in Figure 2.1. These learning approaches are becoming state-of-the-art in computer vision applications such as association rules, object identification, text recognition from an image, and image segmentation [6].

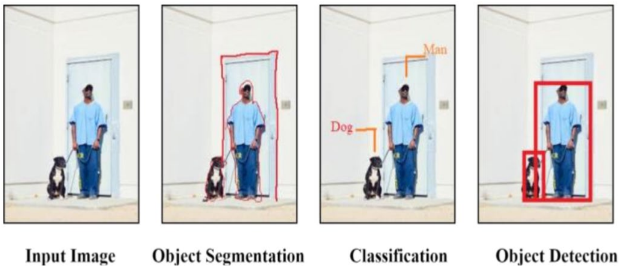


Fig 2.1 Segmentation, Classification and Object Detection

2.2 Support Vector Machine

The Support Vector Machine (SVM) technique is a supervised learning method that could be used for regression analysis and classification/prediction. It is trained as a function approximator and can be applied to comprehend the system's structure. Non-linearly separable and linearly separable data may both be trained into an SVM. Known as support vectors, the SVM finds the most significant samples. The samples that are closest to the decision surface being generated are the support vectors, which come from both of the classes that are being examined [4][7].

As can be seen in Fig. 2.2., the goal of SVM is to yield a single hyperplane decision surface that is equally spaced between two decision borders and has the largest margin possible to split the classes linearly. Finding data sets when there are insufficient training data and where using a lot of statistics normally cannot guarantee the best answer is the aim of assisting vector machine learning [4][7].

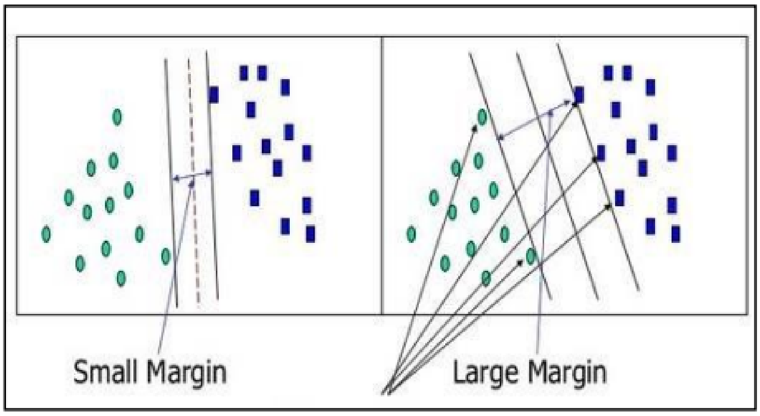


Fig 2.2 Margin and Support Vector

Using a designated training set of data (administered learning), a Support Vector Machine (SVM) is a discriminative classifier that is formally defined by an isolated hyper plane. The ideal hyper plane that results from this computation is used to classify new models. This hyperplane in two dimensions is a line that divides a plane into both sides, which is referred to as the division of each class on each side [8].

The shortest line that connects the convex hulls of the two classes (dotted line) is orthogonal to the ideal hyper plane, and it intersects it halfway. A threshold b and a weight vector w exist such that [8],

$$yi \cdot \big((w \cdot xi) + b\big) > 0 \tag{1}$$

w and b are rescaled so that the point or points nearest to the hyperplane fulfil,

$$|(w \cdot xi) + b| = 1 \tag{2}$$

we get the hyperplane's shape (w,b) with,

$$yi \cdot \big((w \cdot xi) + b\big) \geq 1 \tag{3}$$

The margin is equal to $\frac{2}{\|w\|}$ when measured perpendicular to the hyperplane. Therefore, we must minimize $\|w\|$ in order to maximize the margin, according to eq (3). The least limit is found in the ideal hyperplane, which is defined as the one with the greatest edge of partition between the two classes that is displayed in Fig 2.3 [8].

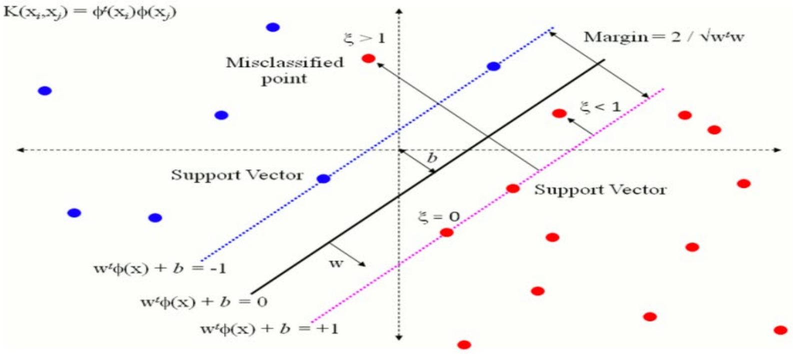


Fig 2.3 Linearly Separable Samples

SVM's primary goal is accurate classification of unseen data. Numerous fields have numerous applications for it. Applications for SVM are [4]:

- Text identification: In the fields of image processing and machine learning, character identification is growing as a challenging and fascinating subject. SVM is one of the most often utilized techniques in many types of pattern recognition. It is never an easy task for a machine to identify letters, numbers, figures, or humans. Numerous methods in this field were also put forth.
- Face identification and recognition: A square bordering might be generated around a face once certain areas of the image are identified as faces and non-faces using an SVM classifier.
- Disease diagnosis: By providing a fast and accurate means of diagnosing

diseases, SVM Classifiers have the potential to significantly impact the medical industry. Since identifying diseases is the most important function in the field of health care. Many lives are saved when an infection is discovered early.

2.3 Convolutional Neural Network

Deep CNNs have been effectively used in a variety of applications, including facial keypoint detection, speech recognition, face parsing, action categorization, and human posture estimation. The adaptability and efficiency of deep CNNs in the area of computer vision and pattern recognition are demonstrated by these applications. The tasks of age and gender categorization from unrestricted images are, however, a relatively new and untested use for them [9].

Deep neural networks, including CNNs, can process input that has had minimal preprocessing and can discover the best network configurations through training processes, in contrast to conventional machine learning techniques that rely on manual feature specification. As a unique kind of feed-forward network, CNNs are excellent at analyzing visual data and have demonstrated outstanding performance in tasks like image segmentation and classification [10]. As shown in Fig 2.4.

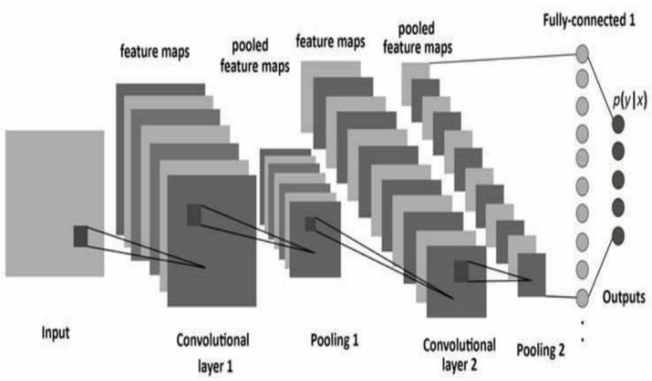


Fig 2.4 The architecture of Convolutional Neural Network

CNNs are one of the most significant neural network algorithms used in pattern recognition and machine learning. It is generally known that their dominance in terms of accuracy and quickness. The two main parts of CNNs, which are created expressly for image analysis, are feature extraction and classification. Convolutional layers and pooling processes are used in the feature extraction process to extract relevant properties from the input data. In the following classification component, which consists of completely connected layers, predicted objects in the image are given probabilities, acting as a classifier [1].

Additionally, CNNs are constructed up of many layer types, such as convolutional, pooling, ReLU, and fully-connected (FC) layers. The input image is subjected to filters by the convolutional layer, which creates a feature map that emphasizes important patterns [1].

The feature map's size is decreased by the pooling layer, improving computational efficiency. The network design has several iterations of these layers. After flattening the input and determining the probabilities for each feature, FC layers receive the outputs of the compact feature maps. The Rectified Linear Units (ReLU) layer introduces nonlinearity, increasing the ability of the network to recognize complicated linkages [1].

Layer types commonly used in Convolutional Neural Networks (CNNs) include convolutional layers, pooling layers, Rectified Linear Unit (ReLU) layers, and fully-connected (FC) layers [9].

1. Convolutional Layers:
   The foundational units of CNNs are convolutional layers. Convolutions are carried out by applying a set of learnable filters (also known as kernels) to the input data. These filters aid in collecting spatial hierarchies and extracting specific features from the data. Layers using convolutional neural networks can recognize patterns in edges, textures, and shapes. Each filter generates a feature map that shows the learnt feature's presence or absence in various spatial areas.
2. Pooling Layers:
   The spatial dimensions (width and height) of the feature maps produced by convolutional layers are decreased by pooling layers. They employ methods like maximum pooling or average pooling to aggregate local data. While average pooling determines the average value, max pooling takes the highest value obtained inside an area. Pooling increases computational efficiency by

reducing the number of parameters and improving the network's robustness to changes in the input.

3. Rectified Linear Unit (ReLU) Layers:
   ReLU layers produce the network nonlinearity. They implement the ReLU activation function, which maintains the positive values while setting the negative values to zero. ReLU layers allow in the implementation of non-linear transformations to identify complex patterns in the data. Due to its ease of use and capacity to address the vanishing gradient issue, this activation function has gained acceptance.
4. Fully-Connected (FC) Layers:
   All of the neurons in the current layer are connected to all of the neurons in the preceding layer through fully connected layers. These layers transform the high-level characteristics that the preceding layers obtained to the appropriate output classes after processing them. In a CNN design, FC layers frequently come after the convolutional and pooling layers. A softmax activation function is often used in the last FC layer to generate class probabilities.

Researchers may now investigate age and gender categorization from unconstrained images by using the ability of deep CNNs, providing up new opportunities in areas like ad serving, demographics analysis, and content-based research. Deep CNNs may improve the accuracy and robustness of the outcomes from these challenges, advancing pattern recognition and computer vision techniques [11].

2.4 Ensemble Learning

In order to improve accuracy and decrease the model's variance, many models are trained on the same dataset using the ensemble learning approach. Ensemble models predicted leveraging the combined output of the basic classifiers, combining the capabilities of individual machine learning algorithms to increase accuracy [5][12].

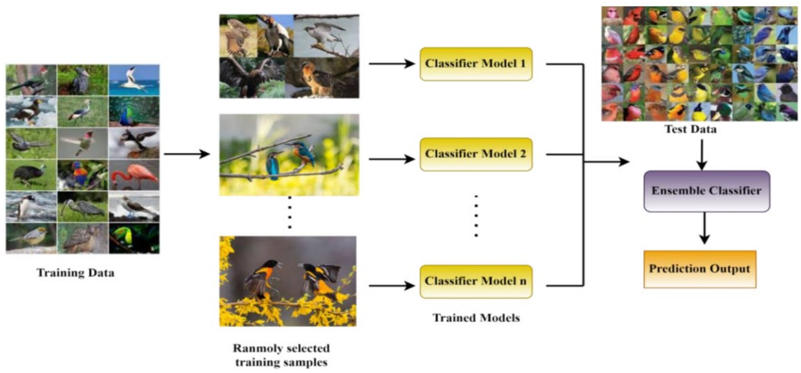


Fig 2.5 Framework for Ensemble Learning

Ensemble learning uses techniques that expand and combine models to provide better outcomes. Base models, which offer higher prediction accuracy than a single trained model, are several models that are utilized as inputs for ensemble approaches in this learning. Ensemble Learning employs three distinct strategies [6]:

- Boosting: A weight in boosting is assigned to every training pair. Multiple classifiers can be learned in sequence using iterative learning. Learning results in weight modifications for each classifier. The accuracy of each classifier determines its weight, which is then combined to create the final boosted classifier.
- Bagging/Bootstrap Aggregating: Using this ensemble approach, a learning scheme's model ensemble is created, with each model providing an equally weighted prediction.
- Stacking: There are two categories of input data: training and testing. The training dataset is used as input to build a meta classifier after being trained using several classifiers. The final trained model is the output of the meta classifier. After that, the testing dataset will be used to evaluate the classifier's (meta) prediction accuracy.

Advantages of Ensemble learning [6]:

- Ensemble techniques are more accurate predictors than individual models when compared to the majority of other ML types.
- Ensemble techniques are quite helpful when a dataset comprises both linear and non-linear data types; many models may be connected to manage this sort of data.

- A model ensemble is always more stable and less noisy.
- With ensemble techniques, bias and variance may be reduced
- The model is usually neither underfitted nor overfitted.

Disadvantages of Ensemble learning [6]:

- It is challenging to learn ensemble learning, and a bad choice might lead to a model that predicts less accurately than an individual model.
- The ensemble model requires a lot of time and storage.

2.5 Decision Tree

Quinlan (1987) was the one who first invented decision tree algorithms. Due to several advantages over other algorithms, the decision tree classification technique known as the Decision Tree technique is widely used. When processing tiny data, decision trees are a very easy way to comprehend and may also make simple judgements from complicated ones by converting them into simple ones. This method maintains the quality of the results obtained by utilizing the criteria for each node Another advantages, such as the potential for over-fitting of the target sample, biassed or unstable tree generation, etc. As a result, some actions, including determining the tree's maximum depth, applying an ensemble model, or balancing the dataset before utilizing the tree model, can be implemented [13][14]. Fig 2.6 illustrate a structure of Decision Tree.

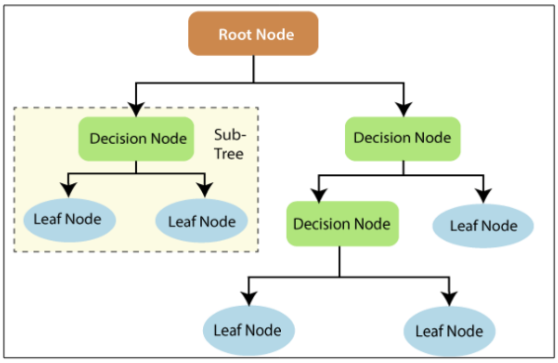


Fig. 2.6 Structure of Decision Tree

One of the effective techniques that is frequently applied in many domains, including machine learning, image processing, and pattern recognition, is the decision tree. The approach builds a multiway tree by greedily determining the categorical feature that will provide the most information gain for categorical targets for each node. After reaching their maximum size, trees are typically pruned in order to enhance their capacity to generalize to new data. A non-parametric supervised learning technique for classification is the decision tree. This is a standard classification technique that utilizes decision trees to train the model and the inductive process to provide legible rules before analyzing the fresh sample data. In general, the fitter the model and the more complicated the decision rules, the deeper the decision tree [13][15].

There are two ways for building a decision tree. Building a decision tree is the initial phase, which involves creating a decision tree using a training sample set. The decision tree's pruning is the second phase [13].

The process of analyzing, fixing, and correcting the decision tree created in the earlier step is known as pruning. The decision tree generating procedure for data verification produces preliminary rules that exclude branches that might potentially impact the accuracy of the pre-balance [13].

2.6 Adaboost

The Adaboost technique is an iterative process that was first presented by Freund and Schapire in 1997. The Adaboost algorithm uses many classifiers in an iterative process. It trains several models (weak models) for the same training set, then combines these weak models to create a final, stronger model (strong model). The fundamental idea behind Adaboost is to fit a series of weak learners using boosting iterations that include distributing updated weights to each training sample on progressively altered copies of the data. The training instances whose weights were incorrectly categorized by the boosted model produced in the previous step are raised for each future iteration, whereas the weights of the training examples whose classifications were correct are decreased. The final classifier (strong model) is created by combining the results from each weak classifier using a weighted majority vote [13][16].

The way the Adaboost algorithm operates is that it generates the output labels after first classifying the training input data. After that, it compares the output with the findings and increases the weight if the output is incorrectly categorized. Once more, the same procedure is carried out, the weights are continuously updated, and misclassified data are arranged using the enhanced weights [16].

2.7 Random Forest

Leo Breiman developed the machine learning algorithm known as Random Forest (RF) Breiman [17] in 2001. It combined the extended bagging approach known as the random subspace method with the integrated learning theory of bagging. The training and testing processes run simultaneously in a random forest due to every tree in the forest is independent of the other trees. A combination of tree-structured predictors and a detector is called a random forest. The predictions of several decision trees are combined into a single model using the random forest classifier [18][19][20].

A machine learning technique called the Random Forest classifier combines several tree classifiers. Each tree classifier creates a unit vote for the most prominent class in the tree in order to categorize an input vector [21].

As seen in Fig. 2.7, Random Forest, one of the most used machine learning algorithms in classification research using various types of data, improves classification accuracy by generating several decision trees. Numerous studies have demonstrated that the Random Forest classifier is quick and resistant to overfitting, even when dealing with high data dimensionality and multi-linearity [21].

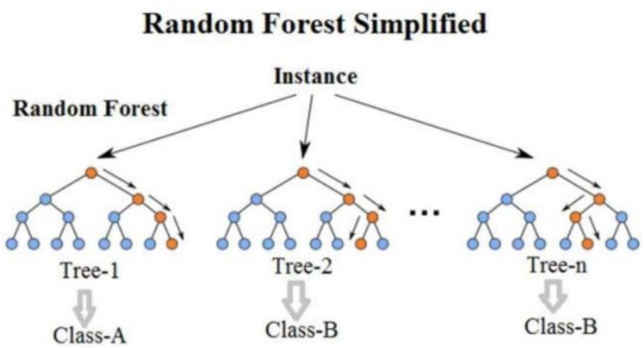


Fig. 2.7 The architecture of Random Forest Algorithm

The fundamental principles of Random Forest classification are as follows: first, n decision tree models were constructed for each of the n samples to acquire n classification results; next, n samples are randomly selected using bootstrap sampling (chosen randomly and with put-back from the training set of N training samples), each with the same sample size as the original training set; in the end, the overall classification of each record is established by a vote based on the n classification outcomes [18].

A random forest multi-way classifier is constructed up from numerous trees, each of which was produced by randomization in some way. The posterior distribution estimates across the image classes are used to identify the leaf nodes in each tree. Every internal node has a test that divides the space of data to be categorized as best it can. Sending images down each tree and aggregating the leaf distributions it reaches is how an image is classed. Two places in the training process where randomness might be introduced are when subsampling the training data to create each tree using a different subset and when choosing the node tests [22].

The scalable supervised techniques for handling high-dimensional data have been enhanced by random forests; they work very well with HSI data and may be applied to large-scale challenges. Moreover, there are very few parameters in random forests that need to be changed, which reduces the data's assumptions. Random forests do, however, have some disadvantages [19]:

- The approach needs an enormous quantity of labelled data
- It is not very good at handling the problem of an unequal number of target and background pixels, particularly when there are relatively few target samples.

CHAPTER 3

PROPOSED METHODOLOGY

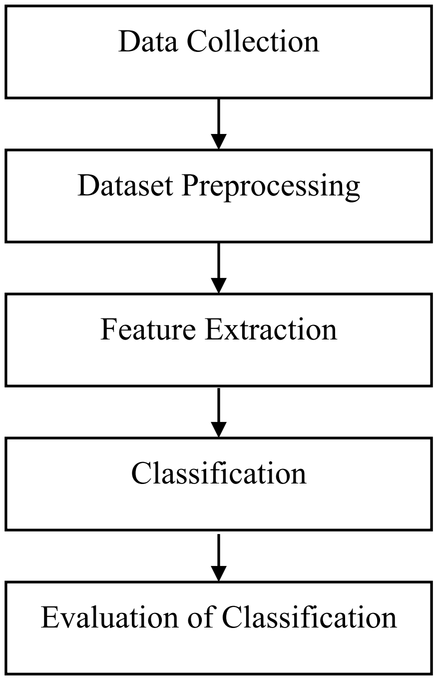


Fig. 3.1 Proposed Flow Diagram

3.1 Data Collection

The first stage in creating a machine learning model for identifying old individuals and kids is dataset collection. This stage involves searching for and collecting a dataset from several sources that includes pictures of kids and older individuals. To ensure that all classes are well represented, the dataset should preferably contain a wide range of age variances. It's necessary to make sure the dataset is varied and reflective of the target population. This involves offering illustrations of older people and kids from various ethnicities, gender, and cultural backgrounds.

3.2 Data Preprocessing

Apply a series of steps or techniques to the image dataset before using it to train or test the model. The goal of dataset preprocessing is to ensure that the dataset is in a suitable, consistent, and ready-to-use format for the training and evaluation process. Steps involved in dataset preprocessing may include defining classes and identifying the location in which the data collection is stored. It enables navigating and organizing the directory structure of the dataset.

Labels and images are loading where for feature extraction and further processing, reading pictures is essential. Labels are required in order to connect each image to the appropriate class and enable supervised learning. Furthermore, it requires labelling objects. A corresponding class label is assigned when an image is imported and processed. By mapping the retrieved characteristics to the corresponding classes, this phase makes supervised learning easier. Split into test and training sets where there are training and testing sets inside the data set. The machine learning model is trained on the training set, and its performance on unseen data is assessed on the test set. Splitting the data facilitates evaluating the model's capacity for generalization and identifies possible instances of overfitting.

3.3 Dataset and Label Creation.

With the dataset and labels created, we can use this data to train the model in a classification task, allowing the model to learn the patterns associated with age classification or class detection in the images.

3.4 Dataset Splitting

Dataset splitting refers to the process of dividing the dataset into two separate subsets, namely the training data and the validation data. The purpose of dataset splitting is to test and evaluate the performance of the trained model and determine the appropriate percentage split according to the requirements (e.g., 80% for training and 20% for validation).

3.5 HOG Feature Extraction

HOG (Histogram of Oriented Gradients) feature extraction is a technique used to transform an image into a simplified feature representation while still preserving important information related to the texture and shape of objects in the image.

In computer vision and image processing, the HOG (Histogram of Oriented Gradients) feature extraction technique is frequently used to extract useful characteristics from images. It succeeds in capturing the shape and texture of the items that are visible in the image. The HOG method works by calculating the gradient distribution in nearby image areas [23].

The following is the HOG feature extraction process [23]:

1. Gradient computation:

   Computing the image's gradient is the first step in the extraction of HOG features. The edges and limits of things may be identified using gradients, which represent variations in pixel intensity. The image's horizontal and vertical intensity gradients are estimated by the use of gradient filters, such as the Sobel operators, to calculate the gradient.

2. Cell formation:

   Each cell in the image appears small and overlaps another. Typically, each cell has a few pixels (8x8 or 16x16, for example). The image is divided into cells to capture local gradient and texture differences.

3. Orientation Binning:

   The orientations of the gradients are quantized inside each cell into a predetermined number of orientation bins (such as 9 bins including 0 to 180 degrees). The continuous gradient orientations transform into discrete values in this stage. Each orientation bin's contribution is determined by the size of the gradients.

4. Histogram Calculation:

   The gradient magnitudes are added up into the relevant orientation bins for each cell, resulting in a histogram of the gradient orientations for that cell. The local image structure is shown by the histogram, which also summarizes the distribution of gradient orientations inside the cell.

5. Block Normalization:

   Neighboring cells are grouped into blocks to allow for local changes in illumination and contrast. To make sure that the description is robust to variations in illumination, normalization is done inside each block. L1-norm or L2-norm normalization are frequent normalization techniques.

6. Descriptor Formation:

   The HOG description for a block is created in the final stage by combining the histograms from all the cells inside that block. The descriptor records the spatial organization of local gradient data and encodes the region's shape and texture details.

3.6 Model Building

- Random Forest model building

  Random Forest is a kind of ensemble learning technique. A decision tree is a structure that resemble a flowchart, with leaf nodes standing in for class labels or results, inside nodes for features, and branches for decision rules. From the training set of data, each decision tree in the Random Forest ensemble separately learns decision rules.

  - Configuration parameters to determine the behavior and performance of the Random Forest classifier.
  - The fit method takes the training data as input and trains the Random Forest model using the provided data.
  - Bootstrap sampling. Each Random Forest decision tree is trained using a bootstrapped sample of the training set during the training process. Using replacement data, bootstrap sampling selects samples at random from the training set.
  - Random feature subset. In a decision tree, only a random subset of characteristics is taken into consideration for splitting at each split. This randomization reduces overfitting and helps in the decorrelation of the trees in the ensemble.
  - Aggregation of prediction. To arrive at the final prediction, all of the decision trees' individual predictions are combined after training.

- Adaboost model building based on Decision Tree estimator

  - Adaptive Boosting, or AdaBoost, is an ensemble learning technique that builds a robust classifier by combining several weak learners, or base estimators.

- Base estimator selection. As decision trees are simple models capable of identifying fundamental patterns in the data, they are frequently used as weak learners.
- The AdaBoost classifier is initialized by specifying the base estimator and the number of estimators (decision trees) in the ensemble.
- A preprocessing pipeline is defined to standardize the feature vectors using standard scaling.
- The training data is used to fit the preprocessing pipeline and determine each feature's mean and standard deviation.
- The preprocessed training data and corresponding labels are used to train the AdaBoost classifier.
- The decision trees are constructed in sequentially, with each new tree focusing more on the instances that the previous trees misclassified.

3.7 Evaluation of Classification

One popular visualization technique in supervised learning is the confusion matrix. In the matrix, each row denotes events in the actual class, and each column is an example of the predicted class. Actual and predictable information regarding the classification system is contained in the confusion matrix [14].

Table 1 Confusion Matrix

Actual Class	Predicted Class	
	Negative	Positive
Negative	TN	FP
Positive	FN	TP

Where, TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

Predict on the validation data using the trained model and evaluate the performance using evaluation metrics such as accuracy, precision, recall, and F1-score. Analyze the evaluation results and identify areas that need improvement or optimization.

- Accuracy. Accuracy is defined as the ratio of correct estimations to total appraisals. It resolves errors in the data set and the quality of the data. The unit of measurement is the percentage (%) [20]. The expression of accuracy is given in equation (4).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4}$$

- Precision. Finding the percentage of results that the classifier indicates as belonging to a particular class actually belongs to that class is known as precision [24]. The expression of precision is given in equation (5).

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

- Recall. Recall or sensitivity is defined as the ratio of the appraisal's true positives to the total number of true positives and false negatives. It is an indicator of well-perceived positive attributes. With relation to rate (%), it is approximated [20]. The expression of recall is given in equation (6).

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

- F1-Score. The harmonic mean of recall and accuracy is the F1-score [24]. The expression of F1-score is given in equation (7).

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{7}$$

3.8 Visualization of The Training Set Results

An essential phase in the construction of the model is the visualization of the training set results. Gaining insights that can guide future modifications to the model or training procedure, as well as understanding the behavior of the model during training, are all improved by it. Visualization of the training set results is obtained from loading images from the dataset directory, organizes them by class, and then plots the first 10 images from each class. The resulting visualization shows a grid of images where each row corresponds to a different class, and each column represents an individual image within that class.

3.9 Visualization of The Test Set Results

Visualization of the test set result involves displaying the images from the test set along with their corresponding labels in order to evaluate the performance of a machine learning model. Visualization of the test set results is obtained from plotting the first 10 images from the test set along with their corresponding class labels and providing a qualitative understanding of the dataset and model predictions.

3.10 Model Comparison

Several evaluation standards, including accuracy, precision, recall, and F1 score, are used in the model comparison process to evaluate the performance of different models.

These metrics provide insight into how effectively the models manage issues like false positives and false negatives and accurately identify instances across various classes. We are able to determine which model performs better overall in terms of classification accuracy and resilience by comparing these metrics among a variety of models. By assisting in the process of choosing the best model for a particular job or dataset, this comparative analysis improves well-informed decision-making.

CHAPTER 4

ANALYSIS RESULT

Based on the analysis of the results, the collection of 100 datasets from various sources which includes old people and kids, each of which consists of 50 data. The dataset obtained involves presenting illustrations of parents and children from various ethnic, gender, and cultural backgrounds and contains wide variations.

Following that, the dataset was split into 20% for validation and 80% for training. The training set comprised 80 data points used to train the model, while the remaining 20 data points were used to evaluate the model's performance. This dataset split is crucial to avoid overfitting and ensure good generalization to unseen data.

```
Total training data: 80
Total validation data: 20
```

A confusion matrix is then computed as an essential component in the model evaluation stage to evaluate how effectively the classification model performs. The accuracy to which the model can distinguish between two classes—the old people and the kids—is shown in detail in this confusion matrix. The confusion matrix of Random Forest model is represented in Fig 4.1.

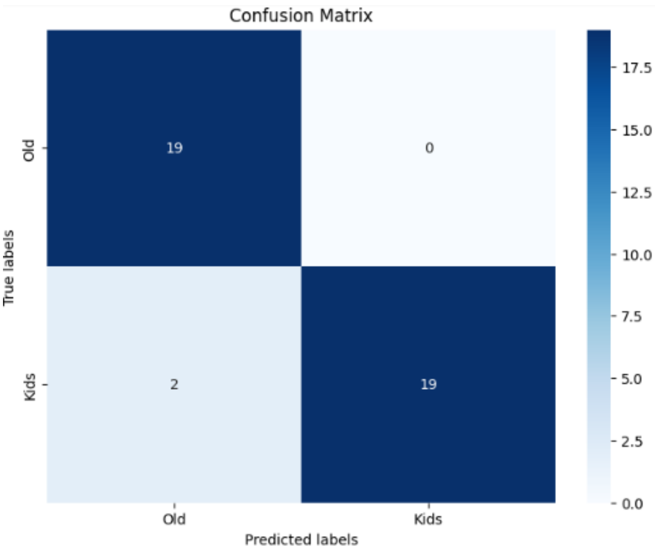


Fig. 4.1 Confusion Matrix of Random Forest Model

The confusion matrix is a table of data that provides an overview of how efficiently a classification system distinguishes between old people and kids. There are four component parts of the confusion matrix of Random Forest model:

1. True Positive (TP): 19

   The quantity of instances (kids) that are truly positive and that the Random Forest model accurately predicts as positive.

2. True Negative (TN): 19

   The quantity of instances (old people) that are truly negative and are the Random Forest model accurately predict as negative.

3. False Positive (FP): 0

   The quantity of instances that the Random Forest model incorrectly predicts as positive while they are actually negative.

4. False Negative (FN): 2

   The quantity of instances that the Random Forest model incorrectly predicts as negative while they are actually positive.

The confusion matrix of AdaBoost Classifier model (Decision Tree estimator) is represented in Fig. 4.2.

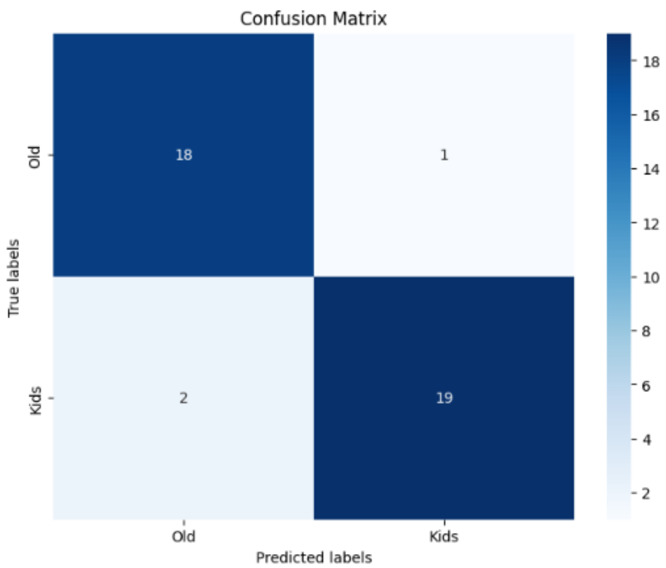


Fig. 4.2 Confusion Matrix of AdaBoost Classifier Model
(Decision Tree Estimator)

There are four component parts of the confusion matrix of AdaBoost Classifier model (Decision Tree estimator):

1. True Positive (TP): 19

   The amount of truly positive findings (kids) that the AdaBoost classifier model, which uses a Decision Tree estimator, accurately predicts as positive.
2. True Negative (TN): 18

   The amount of truly negative findings (old people) that the AdaBoost classifier model, which uses a Decision Tree estimator, accurately predicts as negative.
3. False Positive (FP): 1

   The amount of events that the AdaBoost classifier model, which uses a Decision Tree estimator, inaccurately predicts as positive, whereas in actuality the result is negative.
4. False Negative (FN): 2

   The amount of events that the AdaBoost classifier model, which uses a Decision Tree estimator, inaccurately predicts as negative, whereas in actuality the result is positive.

The confusion matrix of SVM model is represented in Fig 4.3.

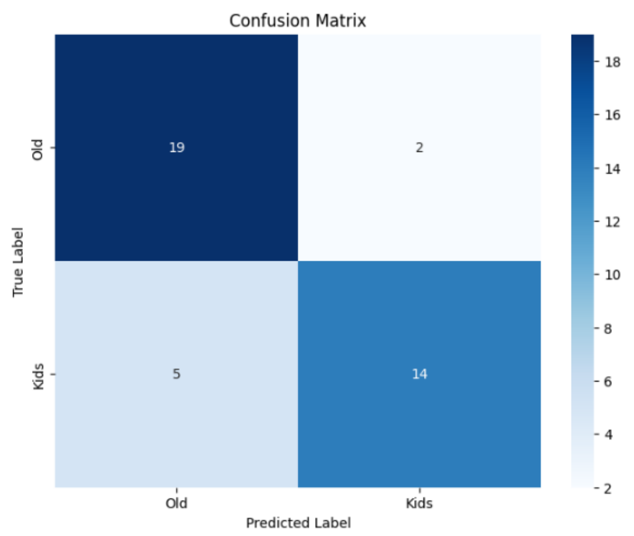


Fig. 4.3 Confusion Matrix of SVM Model

There are four component parts of the confusion matrix of SVM Model:

1. True Positive (TP): 14

   The quantity of instances (kids) that are truly positive and that the SVM model accurately predicts as positive.

2. True Negative (TN): 19

   The quantity of instances (old people) that are truly negative and are the SVM model accurately predict as negative.

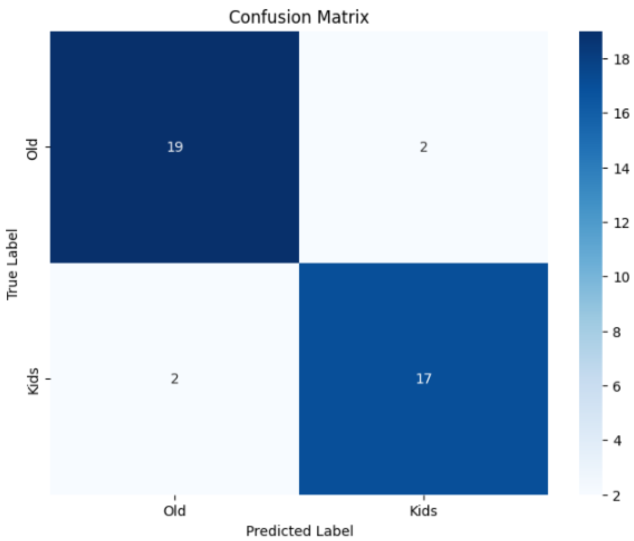
3. False Positive (FP): 2

   The quantity of instances that the SVM model incorrectly predicts as positive while they are actually negative.

4. False Negative (FN): 5

   The quantity of instances that the SVM model incorrectly predicts as negative while they are actually positive.

The confusion matrix of CNN model is represented in Fig 4.4.



4.4. Confusion Matrix of CNN Model

There are four component parts of the confusion matrix of CNN model:

1. True Positive (TP): 17

   The amount of truly positive findings (kids) that uses the CNN model, accurately predicts as positive.

2. True Negative (TN): 19

   The amount of truly negative findings (old people) that uses the CNN model, accurately predicts as negative.

3. False Positive (FP): 2

   The amount of events that uses CNN model, inaccurately predicts as positive, whereas in actuality the result is negative.

4. False Negative (FN): 2

   The amount of events that uses the CNN model, inaccurately predicts as negative, whereas in actuality the result is positive.

Next, evaluation parameters including accuracy, precision, recall, and F1-score values were used to the confusion matrix data to perform a further analysis. The evaluation results were based on the model's predictions on the validation data. The validation of Random Forest model is presented in Tab 2.

Table 2 Evaluation Metrics of Random Forest Model

Evaluation Metrics	*Evaluation Result*
Accuracy	0.95
Precision	0.9547619047619047
Recall	0.95
F1-Score	0.95

The evaluation metrics provide insights into the performance of the Random Forest model for the given classification task:

- Accuracy: 0.95

  The total accuracy of the model's predictions across all classes is measured by accuracy. The accuracy of the Random Forest classifier is 95%, indicating that 95% of the predictions made by the model are correct.
- Precision: 0.9547619047619047

  The precision of a model is defined as the ratio of true positive predictions to all positive predictions. It measures how well the model avoids producing false positive results. the precision of the Random Forest classifier is approximately 95.48%, indicating that when the model predicts a class, it is correct about 95.48% of the time.

- Recall: 0.95

  The proportion of true positive predictions among the actual positive cases in the dataset is measured by recall, which is sometimes referred to as sensitivity. It measures the ability of the model to find every relevant instance of a particular class. the recall of the Random Forest classifier is 95%, indicating that the model correctly identifies 95% of the actual positive instances.
- F1-score: 0.95

  The harmonic mean of recall and accuracy is known as the F1-score. It finds a balance between recall and accuracy, and it's frequently used as an independently performance measurement for models. The F1-score of the Random Forest classifier is 95%, indicating a balanced performance between precision and recall.

The validation metrics of AdaBoost Classifier model (Decision Tree Estimator) is presented in Tab 3.

Table 3 Evaluation Metrics of AdaBoost Classifier Model (Decision Tree Estimator)

Evaluation Metrics	*Evaluation Result*
Accuracy	0.925
Precision	0.9262499999999999
Recall	0.925
F1-Score	0.925046904315197

The evaluation results indicate the performance of the trained model on the validation data. The evaluation metrics:

- Accuracy: 0.925

  The accuracy of the classifier is 0.925, which means that it correctly predicted 92.5% of the instances in the test set.
- Precision: 0.9262499999999999

  The proportion of true positive predictions among all positive predictions is shown by the classifier's precision, which is at 0.926. Stated otherwise, the accuracy of the classifier's predictions for positive classes is around 92.6%.

- Recall: 0.925

  The percentage of true positive results that the classifier properly found is shown by its recall, which stands at 0.925. In this instance, the classifier correctly classified about 92.5% of the actual positive instances.
- F1-score: 0.925046904315197

  Precision and recall are harmonic means, and the F1-score finds a balance between both. The classifier's F1-score of 0.925 shows that recall and precision are well-balanced.

The validation metrics of SVM model is presented in Tab 4.

Table 4 Evaluation Metrics of SVM Model

Evaluation Metrics	*Evaluation Result*
Accuracy	0.825
Precision	0.875
Recall	0.7368421052631579
F1-Score	0.7999999999

The evaluation metrics provide insights into the performance of the Support Vector Machine (SVM) model for the given classification task:

- Accuracy: 0.825

  This indicates the overall correctness of the model's predictions. An accuracy of 0.825 suggests that approximately 82.5% of the predictions made by the SVM model are correct.
- Precision: 0.875

  Precision measures the accuracy of the positive predictions made by the model. A precision of 0.875 indicates that when the SVM predicts a positive class, it is correct about 87.5% of the time.

- Recall: 0.7368421052631579
  Recall, also known as sensitivity or true positive rate, represents the ability of the model to capture all positive instances. A recall of 0.7368 suggests that the model correctly identifies approximately 73.68% of all actual positive instances.
- F1-score: 0.8
  The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. An F1-score of 0.8 indicates a good balance between precision and recall for the SVM model.

The validation metrics of CNN model is presented in Tab 5.

Table 5 Evaluation Metrics of CNN Model

Evaluation Metrics	*Evaluation Result*
Accuracy	0.9
Precision	0.8947368421052632
Recall	0.8947368421052632
F1-Score	0.8947368421052632

The evaluation results indicate the performance of the trained model on the validation data. The evaluation metrics:

- Accuracy: 0.9
  Accuracy is the ratio of correctly predicted samples to the total number of samples. In this case, the model achieved an accuracy of 0.9, indicating that it correctly predicted 90% of the samples in the validation data.
- Precision: 0.8947368421052632
  The ratio of actual positives to the total of true positives and false positives is known as precision. It evaluates how well the model can categorize positive samples. The model has high precision for both classes (Old People and Kids), as seen by the weighted average precision of 0.8947368421052632.

- Recall: 0.8947368421052632

  The proportion of actual positives to the total of true positives and false negatives is known as recall. It measures how well the model can locate positive samples. The model accurately detected approximately 89.4% of the positive samples, as indicated by the recall score of 0.8947368421052632.

- F1-score: 0.84769820971867

  The harmonic mean of recall and precision is known as the F1-score. It offers a combination between recall and precision. The model's F1-score of 0.84769820971867 shows that recall and precision are generally well-balanced.

Visualization of the training set results is obtained from loading images from the dataset directory, organizes them by class, and then plots the first 10 images from each class. The resulting visualization shows a grid of images where each row corresponds to a different class, and each column represents an individual image within that class.

Fig. 4.5 Visualization of Old People Class Training Set Images

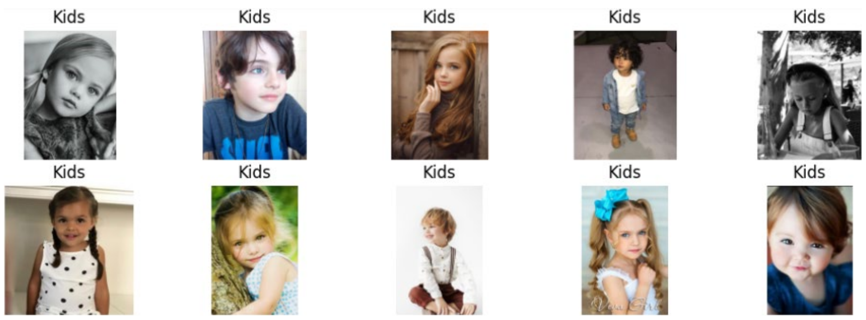


Fig. 4.6 Visualization of Kids Class Training Set Images

Visualization of training set results allows to inspect and verify that the training data is loaded correctly and to get a sense of what the images look like in each class. Visualization of the training set results for old people and kids class are presented in

Fig 4.5 and Fig 4.6, respectively.

Visualization of the test set results is obtained from plotting the first 10 images from the test set along with their corresponding class labels and providing a qualitative understanding of the dataset and model predictions. It's a helpful step for inspecting the quality of the data and verifying that the images are correctly labeled. Visualization of the test set results for old people and kids class are presented in Fig 4.7 and Fig 4.8, respectively.

Fig. 4.7 Visualization of Old People Class Test Set Images

Fig. 4.8 Visualization of Kids Class Test Set Images

The analysis results were visualized through a table and a graph of the evaluation metrics. The evaluation metrics table and graph of RF, AdaBoost (DT), SVM and CNN model, including accuracy, precision, recall, and F1-score, are presented in Tab 6 and Fig 4.9, respectively.

Table 6 Evaluation Matrices of RF, AdaBoost (DT), SVM and CNN Models

Classifier Models	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Random Forest	95%	95.47%	95%	95%
AdaBoost (DT)	92.50%	92.62%	92.50%	92.60%
SVM	82.50%	87.50%	73.68%	79.99%
CNN	90%	89.47%	89.47%	89.47%

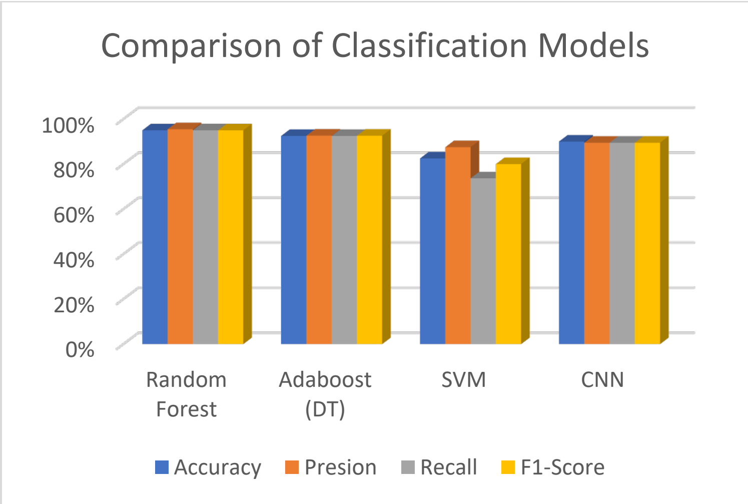


Fig 4.9 Evaluation Metrices of RF, AdaBoost (DT), SVM and CNN

This graph provides a clearer understanding of the model's performance based on the calculated evaluation metrics, allowing for better comparison and comprehension of the model's performance.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The following conclusions could be drawn in considering the findings from the comparison of different models including random forest, Adaboost (DT), SVM and CNN for the classification of old people and kids:

1. Accuracy: With an accuracy of 95%, Random Forest proved the most accurate, closely followed by Adaboost (Decision Trees), at 92.50%. CNN and SVM had somewhat lower accuracy, at 90% and 82.50%, respectively.
2. Precision: The highest precision resulted by Random Forest (95.47%), followed by Adaboost based on Decision Tree (92.62%). The precision scores obtained by SVM and CNN are 87.50% and 89.47%, respectively.
3. Recall: Strong recall rates of 95% and 92.50%, respectively, were displayed by Random Forest and Adaboost (Decision Tree), demonstrating their ability to identify relevant instances. CNN outperformed SVM with a recall rate of 89.47%, while SVM's recall rate was lower at 73.68%.
4. F1-score: With an F1 score of 95%, Random Forest achieved the highest, demonstrating a balance between recall and precision. Adaboost, which uses a decision tree as its basis, came in second with an F1 score of 92.60%. With an F1 score of 79.99%, SVM demonstrated a trade-off between recall and precision. CNN received an 89.47% F1 score.

Based on these parameters, it can be concluded that in terms of accuracy, precision, recall, and F1-score, the different models including Random Forest, Adaboost (Decision Tree), SVM, and CNN, performed an effective job of classifying old people and kids. On the other hand, it seems that Random Forest proves to be the most effective model for the given classification task, followed closely by Adaboost (Decision Tree), which performed admirably, showing competitiveness with Random Forest. while SVM exhibited comparatively lower performance, particularly in recall and F1 score, suggesting room for improvement, and CNN performed well but was slightly behind Random Forest and Adaboost (Decision Tree) in terms of accuracy and

F1 score.

5.2 FUTURE SCOPE

Several actions may be made in the future to enhance the model's performance in the task of detecting old people and kids. To ensure the representativeness and diversity of the data, it is important to begin by collecting wide and different data sets. Improve the dataset that the models' performance may be improved by consistently growing and varying the dataset. Increasing the number of old people and kids images collected from various sources, demographics, and backgrounds can assist enhance the models' generalization and accuracy.

Using improved feature engineering approaches may help extract more insightful representations from the data and improve classification accuracy. Further performance gains might result from experimenting with different ensemble techniques like model stacking or gradient boosting machines. Another optimization method is hyperparameter tuning, where methods such as grid search or random search could be helpful in adjusting model parameters for optimal performance.

To further improve the efficacy and applicability of ensemble learning in classifying older people and kids, further research should be conducted in the following areas: augmenting the dataset with synthetic samples, addressing class imbalance, improving model interpretability, deploying models for real-time inference, and evaluating cross-domain generalization. There is a great deal of potential for these efforts to make significant improvements to the field of machine learning.

REFERENCES

1. Benkaddour, M. K., Lahlali, S., & Trabelsi, M. (2021). Human Age and Gender Classification using Convolutional Neural Network. In *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being, IHSH 2020* (pp. 215–220). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IHSH51661.2021.9378708
2. Zhu, Y., Li, Y., Mu, G., & Guo, G. (2016). A Study on Apparent Age Estimation. In Proceedings of the IEEE International Conference on Computer Vision (Vol. 2015-February, pp. 267–273). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICCVW.2015.43
3. Hassan, K. R., & Ali, I. H. (2020). Age and Gender Classification using Multiple Convolutional Neural Network. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928). IOP Publishing Ltd. https://doi.org/10.1088/1757-899X/928/3/032039
4. Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine Learning Applications based on SVM Classification: A Review. *Qubahan Academic Journal*, *1*(2), 81–90. https://doi.org/10.48161/qaj.v1n2a50
5. Latif, S., Fang, X. W., Arshid, K., Almuhaimeed, A., Imran, A., & Alghamdi, M. (2023). Analysis of Birth Data using Ensemble Modeling Techniques. *Applied Artificial Intelligence*, *37*(1). https://doi.org/10.1080/08839514.2022.2158273
6. Mahadevkar, S. V., Khemani, B., Patil, S., Kotecha, K., Vora, D. R., Abraham, A., & Gabralla, L. A. (2022). A Review on Machine Learning Styles in Computer Vision - Techniques and Future Directions. *IEEE Access*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2022.3209825
7. Shavers, C., Li, R., & Lebby, G. (2006). An SVM-based approach to face detection. In *Proceedings of the Annual Southeastern Symposium on System Theory* (Vol. 2006, pp. 362–366). https://doi.org/10.1109/ssst.2006.1619082
8. Riyazuddin, Y. Md., Basha, S. M., … Banu, S. N. (2020). Effective Usage of Support Vector Machine in Face Detection. *International Journal of Engineering and Advanced Technology*, *9*(3), 1336–1340.

https://doi.org/10.35940/ijeat.c5406.029320

9. Fei-Fei, L., Karpathy, A., & Johnson, J. (2018). Convolutional Neural Networks (CNNs). In Stanford CS231n: Convolutional Neural Networks for Visual Recognition (Version Spring 2018). Retrieved from http://cs231n.github.io/convolutional-networks/
10. Abirami, B., Subashini, T. S., & Mahavaishnavi, V. (2020). Gender and age prediction from real time facial images using CNN. In *Materials Today: Proceedings* (Vol. 33, pp. 4708–4712). Elsevier Ltd. https://doi.org/10.1016/j.matpr.2020.08.350
11. Levi, G., & Hassncer, T. (2015). Age and gender classification using convolutional neural networks. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (Vol. 2015-October, pp. 34–42). IEEE Computer Society. https://doi.org/10.1109/CVPRW.2015.7301352
12. Bhuiyan, M., & Islam, M. S. (2023). A new ensemble learning approach to detect malaria from microscopic red blood cell images. *Sensors International*, *4*. https://doi.org/10.1016/j.sintl.2022.100209
13. Ye, Z., Guo, S., Chen, D., Wang, H., & Li, S. (2021). Drilling formation perception by supervised learning: Model evaluation and parameter analysis. *Journal of Natural Gas Science and Engineering*, *90*. https://doi.org/10.1016/j.jngse.2021.103923
14. Riansyah, M., Suwilo, S., & Zarlis, M. (2023). Improved Accuracy In Data Mining Decision Tree Classification Using Adaptive Boosting (Adaboost). *SinkrOn*, *8*(2), 617–622. https://doi.org/10.33395/sinkron.v8i2.12055
15. Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165
16. Ahmad, I., Haq, Q. E. U., Imran, M., Alassafi, M. O., & Alghamdi, R. A. (2022). An Efficient Network Intrusion Detection and Classification System. *Mathematics*, *10*(3). https://doi.org/10.3390/math10030530
17. Breiman L. Random forests. Mach Learn (2001) 45:5–32. doi: 10.1023/ A:1010933404324
18. Li, J., Shi, J., Chen, J., Du, Z., & Huang, L. (2023). Self-attention random forest for

breast cancer image classification. *Frontiers in Oncology*, *13*. https://doi.org/10.3389/fonc.2023.1043463

19. Dong, Y., Du, B., & Zhang, L. (2015). Target detection based on random forest metric learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(4), 1830–1838. https://doi.org/10.1109/JSTARS.2015.2416255
20. Thayumanavan, M., & Ramasamy, A. (2021). An efficient approach for brain tumor detection and segmentation in MR brain images using random forest classifier. *Concurrent Engineering Research and Applications*, *29*(3), 266–274. https://doi.org/10.1177/1063293X211010542
21. Avci, C., Budak, M., Yagmur, N., & Balcik, F. B. (2023). Comparison between random forest and support vector machine algorithms for LULC classification. *International Journal of Engineering and Geosciences*, *8*(1), 1–10. https://doi.org/10.26833/ijeg.987605
22. Bosch, A., Zisserman, A., & Muñoz, X. (2007). Image classification using random forests and ferns. In *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2007.4409066
23. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005* (Vol. I, pp. 886–893). https://doi.org/10.1109/CVPR.2005.177
24. Płachta, M., Krzemień, M., Szczypiorski, K., & Janicki, A. (2022). Detection of Image Steganography Using Deep Learning and Ensemble Classifiers. *Electronics (Switzerland)*, *11*(10). https://doi.org/10.3390/electronics11101565