Diamond Image Classification

PATTERN RECOGNITION COURSE PROJECT

TEAM DETAILS:
Eswar Yalakanti, S20210020332
Adi Narayana M, S20210020292
Akhil Sai Sri Ram, S20210020316

I. INTRODUCTION

Classifying diamonds based on their visual features is a crucial and complex task in geometry and jewelry assessment. now we suggest creating a machine learning model to automate the categorization of diamonds using images and important characteristics like shape, cut, colour, carat, symmetry, fluorescence, and clarity. the dataset includes a variety of diamonds with different shapes, each falling into one of seven predefined categories: cushion, emerald, heart, pear, princess, oval, and round. also, we will compare various models' efficiency to know which model is giving better results. also, we will improve the model efficiency using various techniques and combining two different models and techniques.

The dataset provided for this task contains a diverse set of diamond images, each associated with key attributes like shape, cut, color, carat, symmetry, fluorescence, and clarity. The goal is to leverage this dataset to train a model capable of automatically classifying diamonds into predefined categories. The dataset includes seven classes: cushion, emerald, heart, pear, princess, round.

Classes:

Cushion: Diamonds with a square or rectangular shape and rounded corners.

Emerald: Diamonds with a rectangular shape and cut corners.

Heart: Diamonds with a distinctive heart shape.

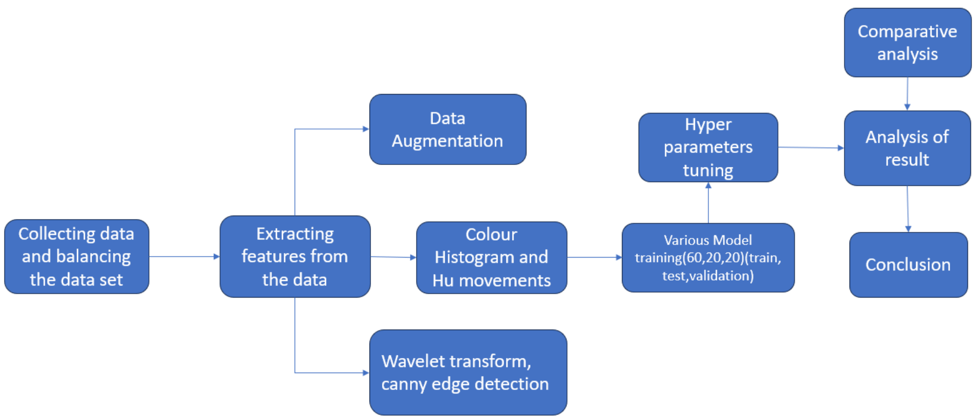
Pear: Diamonds with a teardrop or pear shape.

Princess: Diamonds with a square or rectangular shape but with pointed corners.

Round: Diamonds with a classic round shape.

Now after defining the classes now we will extract the features train the model and test it and measure the model accuracy for checking whether it is a best model or not.

II. BLOCK DIAGRAM



III.METHODOLOGY

1. Data Preprocessing:
- Load the diamond image dataset, ensuring that it includes labels for each diamond (cushion, emerald, heart, pear, princess, round).
- Inspect and clean the dataset for any missing or inconsistent data.
- Split the dataset into training and testing sets to evaluate the model's performance.

2. Image Preprocessing:
- Resize all images to a consistent size to ensure uniformity.
- Normalize pixel values to bring them within a common range (e.g., [0, 1]).
- Optionally, apply data augmentation techniques to increase the diversity of the training set (e.g., rotation, flipping, zooming).

3. Feature Extraction:

i. Edge Detection:
- Apply Canny edge detection on each image to extract edges.
- Calculate edge histogram, edge density, and edge length from the edge-detected images.

ii. Wavelet Transform:
- Apply wavelet transform on the images to extract features at different scales.
- Capture details in the form of coefficients at various resolutions.

iii. Colour Histogram:
- Convert images to a colour space (e.g., RGB, HSV).
- Compute colour histograms for each channel to represent the distribution of colours.

iv. Shape Features (Hu Moments):
- Apply contour detection to find the boundaries of the diamond shapes.
- Calculate Hu moments as shape descriptors.
- Combine all extracted features into a feature vector for each diamond image.

4. Model Selection and Training:
- Choose a suitable classification model for your task (e.g., Support Vector Machine, Random Forest, Convolutional Neural Network).
- Split the dataset into training and validation sets.
- Train the model on the training set using the extracted features.

5. Model Evaluation:
- Evaluate the model on the validation set to fine-tune hyperparameters and assess its performance.
- Use metrics such as accuracy, precision, recall, and F1-score to measure classification performance.

6. Hyperparameter Tuning:
- Optimize the model's hyperparameters to improve its performance.
- Use techniques like grid search or random search for hyperparameter tuning.

7. Testing and Final Evaluation:
- Assess the model's performance on the test set to ensure generalization to new, unseen data.
- Evaluate the model's accuracy and other relevant metrics.

8. Continuous Improvement:
- Monitor the model's performance over time and retrain it with new data if necessary.

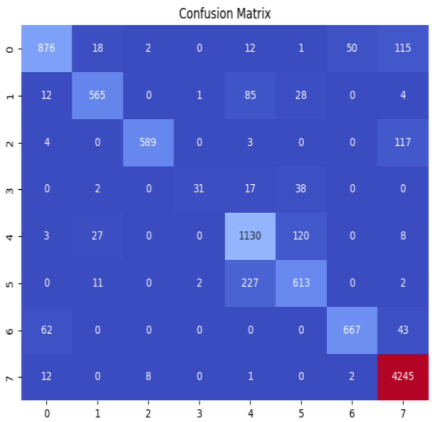
- Explore advanced techniques or larger models for potential performance gains.
- If satisfied with the model's performance, deploy it for real-world usage.
- Implement a user interface for users to input diamond images and receive classifications.

IV.RESULTS

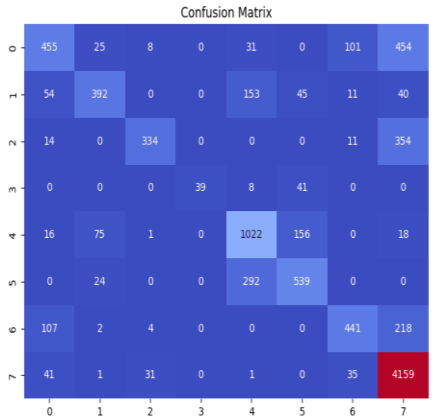
COMPARISONS OF ACCURACY FOR DIFFERENT MODELS

	Model	Accuracy (%)	F1 Score (%)
1	SVM (with images)	75	73.25
2	Random Forest	89.37	89.16
3	K-Nearest Neighbour	75.68	74.12
4	ANN	84.33	83.63
5	XG Boost	96	95.67
6	CAT Boost	86	85.99

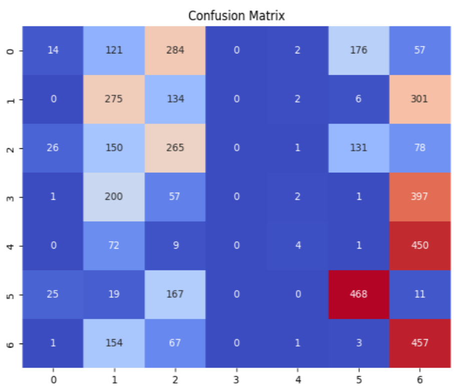
Confusion matrices for different models:



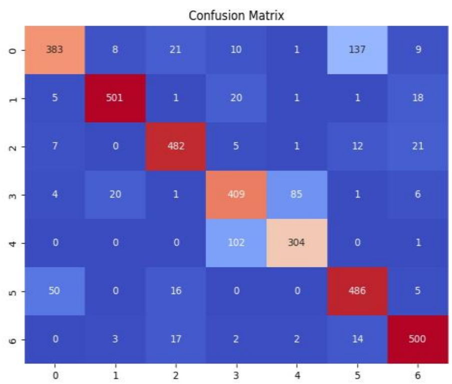
For random forest



For KNN



For SVM using direct data set

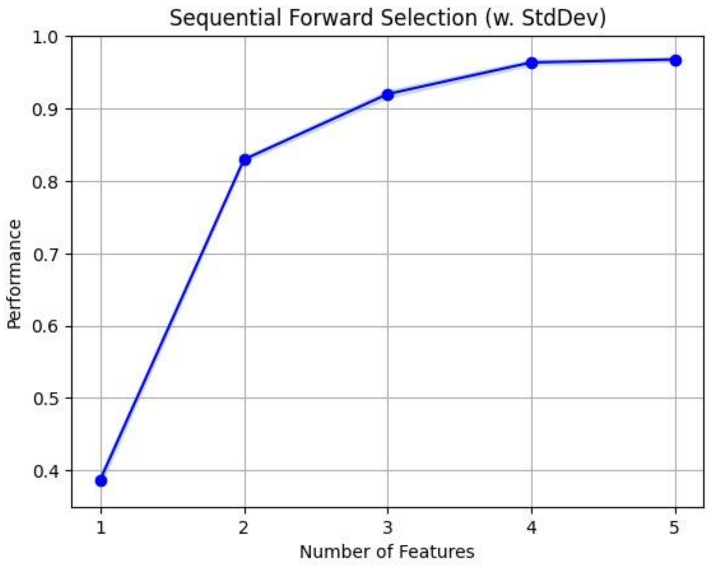


For CAT Boost

Extension work:

SFS (sequential forward selection):

As the algorithm iterates through the features, it systematically builds a feature set by adding the most informative feature in each step. The final output is a subset of features that maximizes the chosen evaluation metric, such as accuracy or F1 score. This selected feature subset is intended to enhance model generalization, improve interpretability, and potentially reduce overfitting. While SFS provides a computationally efficient approach to feature selection, it may not always guarantee the global optimal subset, as it makes locally optimal decisions at each step and gives the suitable no of features required for the classification.. After SFS the accuracy got increased to test accuracy=96.1 and train accuracy =99.



SFS (sequential forward selection)

Analysis and Conclusions:

- By combining the features extracted from wavelet transform and canny-edge detection with the shape, cut, color, carat, symmetry, fluorescence and clarity we get the best accuracy by using the XG Boost Model.
- No matter how many features are there sfs will be constant after five features that is for the data set most optimised no of features are five

REFERENCES:

:

Dataset:

https://www.kaggle.com/datasets/aayushpurswani/diamond-images-dataset

Wavelet transform:

https://pywavelets.readthedocs.io/en/latest/

For SFS:

https://analyticsindiamag.com/a-complete-guide-to-sequential-feature-selection/

Canny edge detection:

https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123