GENDER DETECTION USING FACIAL IMAGE

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Neeharika Meka	**-**	**AP21110011417**
Durga Surya Teja Indigimilli	**-**	**AP21110011370**
Jyothi Madem	**-**	**AP21110011357**

Under the Guidance of

Dr. Naveen Kumar Mahamkali

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

May, 2024

Certificate

Date: 12-May-24

This is to certify that the work present in this Project entitled "**GENDER DETECTION USING FACIAL IMAGE**" has been carried out by **Neeharika, Surya Teja and Jyothi** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. Naveen Kumar Mahamkali,

Designation,

Affiliation.

Acknowledgements

We are deeply indebted to our mentor Dr Naveen Kumar Mahamkali, for his unwavering support and for the valuable mentorship received during this research project. We extend our thankfulness to my fellow colleagues at SRM University AP for their valuable input. Lastly, we express our heartfelt appreciation to all who contributed to the fulfilment of this research project. We are immensely grateful to our project mentor for his expertise, patience and willingness to share knowledge.

We would also like to acknowledge the entire team and for their collaboration, stimulating discussions made this experience more rewarding. This research would not have been possible without the collective effort and encouragement of every individual in the group. Any shortcomings in this report are our responsibility alone.

Table of Contents

Abstract

The main aim of this research project is to detect gender of the individuals. Gender recognition from images is crucial in many industries, including marketing, security, and human-computer interface. In this work, we provide a comprehensive method for gender recognition using machine learning techniques. In this example of supervised learning, the system is trained on a set of faces, both male and female, and then fresh data is classified. Other than male and female, gender identities are not taken into consideration in this research. We suggest applying the Haar-cascade classifier for face detection and the SVM classifier to tackle the gender recognition issue. Because SVM is the best classifier for binary classes, it is employed in this research. The Haar-cascade classifier is employed in real-time applications due to its rapid inference speed, ease of implementation, and low processing power consumption. a collection of facial picture datasets with associated gender designations. This can be used to test or detect gender on both live and static photos, using static images for training. Following training, a test image's output will be a picture with text indicating whether it is male or female.

Keywords: Gender detection, Binary classification, Facial image, SVM, Haar-cascade classifier, Live image, Static image.

Abbreviations

SVM	Support Vector Machine
ML	Machine Learning
KNN	K-Nearest Neighbors
ROI	Region Of Interest

List of Tables

List of Figures

Introduction

Gender recognition from facial images is a significant area of research with widespread applications across various domains such as marketing, security, and human-computer interaction. The ability to accurately determine gender from images plays a crucial role in numerous real-world scenarios. In this research project, we present a comprehensive methodology for gender recognition utilizing ML techniques.

The primary objective of this study is to develop a robust system capable of accurately detecting the gender of individuals depicted in facial images. Unlike some prior works that explore a broader spectrum of gender identities, our focus is specifically on binary gender classification, distinguishing between male and female categories.

To achieve this goal, we leverage two key components: the Haar-cascade classifier for face detection and Support Vector Machine (SVM), Random Forest, and K Nearest Neighbors (KNN) for gender recognition. The Haar-cascade classifier is chosen for its effectiveness in real-time applications, offering rapid inference speed, straightforward implementation, and minimal computational requirements. It serves as the initial step in our pipeline, enabling the detection of faces within images swiftly and accurately.

Furthermore, the system is made to be capable of handling both static images and live gender detection. This adaptability enhances its applicability in real-world settings, where the need for rapid and accurate gender recognition is paramount.

Applications of Gender Detection

Applications of this gender recognition system span a wide range of fields.

- **Marketing and Advertising:** Gender detection can help marketers and advertisers tailor their campaigns to specific demographics. By understanding the gender composition of their target audience, they can create more effective and personalized advertisements.
- **Retail and E-commerce:** Online retailers can use gender detection to provide personalized shopping experiences. For example, they can recommend products based on the gender of the shopper or customize the layout of their website accordingly.
- **Healthcare:** Gender detection technology can assist healthcare professionals in identifying the gender of patients, which can be useful for providing personalized care. For instance, certain medical conditions or treatments may vary depending on gender.
- **Security and Surveillance:** Gender detection can be integrated into security systems for surveillance purposes. It can help in identifying individuals in crowded places such as airports or train stations, aiding in security measures.
- **Human Resources:** Gender detection technology can be utilized in recruitment processes to ensure diversity and inclusivity. It can help in anonymizing resumes to reduce unconscious bias during the initial stages of candidate selection.
- **Education:** In educational settings, gender detection can be used to understand gender distribution in classrooms or online learning platforms. This information can be used to tailor teaching methods and resources to cater to the needs of diverse student populations.
- **Entertainment and Gaming:** Gender detection can enhance user experiences in entertainment and gaming applications. For example, it can be used to create personalized recommendations for movies, music, or video games based on the user's gender preferences.

- **Social Sciences and Research:** Researchers can use gender detection technology to analyze large datasets for studying gender-related trends and patterns in various fields such as sociology, psychology, and economics.
- **Customer Service:** Gender detection can be integrated into customer service systems to provide more personalized interactions. It can help customer service representatives address customers by their preferred gender pronouns and anticipate their needs more accurately.
- **Public Safety:** Law enforcement agencies can use gender detection technology for identifying suspects or missing persons in criminal investigations or search and rescue operations.

These applications offer numerous benefits, it's important to consider and mitigate potential ethical and privacy concerns associated with gender detection technology, such as data privacy, accuracy, and biases.

Literature Survey

Gender detection plays a pivotal role in various applications, including security systems, marketing, and human-computer interaction. Accurately discerning the gender of individuals enables tailored services and targeted interventions in these domains[1][2][3]. In security systems, gender detection aids surveillance and monitoring efforts by allowing for more nuanced threat assessments based on demographic information. For instance, in airports and border control checkpoints, gender detection can assist in identifying individuals who may require additional scrutiny based on security protocols. Similarly, in marketing, gender detection helps businesses tailor advertising campaigns to specific demographics, enhancing their effectiveness. Moreover, in human-computer interaction, gender detection enables personalized user experiences by adapting interfaces and services to users' gender-related characteristics and preferences.

Recent research in gender detection has predominantly focused on employing traditional machine learning algorithms such as Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbors (KNN) for gender classification tasks[4]. These algorithms offer robust performance and scalability, making them suitable for analyzing facial features and predicting gender labels[5]. Feature extraction techniques like Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Principal Component Analysis (PCA) are commonly used to represent facial images effectively, capturing discriminative information for gender classification.

However, challenges persist in gender detection due to dataset biases, class imbalances, and limitations in model generalization[6]. Many datasets used in gender detection research suffer from class imbalance, where one gender is overrepresented compared to the other[7]. This imbalance can skew model performance metrics and lead to biased predictions, especially for the minority gender. Additionally, datasets often lack diversity in terms of facial expressions, poses, lighting conditions, and occlusions, posing challenges for models to generalize across different scenarios.

To address these challenges, researchers have proposed various strategies, including data augmentation techniques, class-balanced sampling methods, and transfer learning approaches[8]. Moreover, advancements in deep learning techniques, particularly Convolutional Neural Networks, have shown promise in automatically learning discriminative features from raw pixel data, potentially overcoming the limitations of handcrafted feature extraction methods[5].

In our gender detection project, we contribute to the existing literature by integrating live gender detection capabilities with traditional machine learning algorithms such as SVM, Random Forest, and KNN[9]. Additionally, we explore hyperparameter tuning techniques to optimize model performance metrics such as accuracy, precision, recall, and F1-score. By combining real-time detection with algorithmic improvements, we aim to develop more accurate and robust gender detection systems capable of handling diverse scenarios and datasets. This holistic approach aligns with the broader goal of advancing gender detection technology for practical applications in security, marketing, and human-computer interaction.

Methodology

4.1 Data Collection and Preprocessing

4.1.1 Data Acquisition

- The dataset comprises images of male and female faces.
- These images are organized into two separate folders: one for female faces and another for male faces.
- Each image consists of a single face with minimal background noise.

4.1.2 Data Preprocessing

- Before extracting features, the images undergo preprocessing to ensure uniformity and enhance feature extraction.
- This includes resizing the images to a standard size to ensure consistency across the dataset.
- The OpenCV library is utilized for image resizing and conversion to grayscale, which simplifies subsequent processing steps.
- The function iterates over each face in the faces list.
- Each face image is resized to a fixed size of 100x100 pixels using cv2.resize(). This step ensures that all facial images have the same dimensions, which is often necessary for machine learning algorithms to process them efficiently.
- The resized face image is flattened into a 1-dimensional array using .flatten(). This converts the 2-dimensional image array into a 1-dimensional vector, which is a common format for inputting data into machine learning models.
- The flattened face image is appended to the list X.
- The corresponding label is appended to the list y. This associates each facial feature with its corresponding label.
- Finally, the function returns two lists: X, which contains the flattened facial features, and y, which contains the corresponding labels.

- Labels are assigned to indicate the gender associated with each image (0 for female, 1 for male).

4.2 Feature Extraction

- Cascade Classifier Initialization: The code initializes a cascade classifier using the Haar cascade XML file haarcascade_frontalface_default.xml, which is provided by OpenCV.
- Image Processing and Detection: The function extract_face_features iterates through the images in the specified folder (folder_path) and reads each image using OpenCV. It then converts the image to grayscale and applies face detection using the cascade classifier.
- Face Cropping and Resizing: For each detected face, the code extracts the region of interest (ROI) from the grayscale image corresponding to the detected face's bounding box. It then resizes the cropped face region to a specified target size (100x100 pixels in this case).
- Feature Storage: The resized face images are stored in a list (features) for further processing.
- Overall, this feature extraction process prepares the data by isolating and standardizing the facial regions, which are then used as input for training the machine learning models.
- resize_image() function is called. It resizes the image to a width of 800 pixels. This step is crucial for reducing computational complexity and ensuring consistent processing.
- The resized image is converted to grayscale using cv2.cvtColor().
- The features being extracted here are the facial regions detected by the face detection algorithm (face_cascade.detectMultiScale()) and then resized to the specified target_size.
- face_cascade.detectMultiScale() is used to detect faces in the grayscale image. It returns a list of rectangles representing the bounding boxes of detected faces.
- For each detected face, the coordinates (x, y) of the top-left corner, as well as the width (w) and height (h) of the bounding box, are extracted.

- The region of interest (ROI) corresponding to each detected face is cropped from the grayscale image using array slicing.
- The cropped face ROI is resized to the specified target_size using cv2.resize().
- The resized face ROI is added to the features list.
- Finally, the list of extracted facial features (features) is returned.

4.3 Model Training and Evaluation

4.3.1 Models usage to classify Gender

- Three different machine learning models are trained for gender classification: Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN).
- The SVM model employs a linear kernel, while Random Forest utilizes an ensemble of decision trees.
- For KNN, hyperparameter tuning is performed using grid search to determine the optimal number of neighbors.
- Based on the best k value we get KNN works on it to classify male and female.
- The models are trained on the preprocessed dataset consisting of Haarcascade face features and corresponding gender labels.

4.3.2 Model Evaluation

- Split the dataset into training and testing sets using the train_test_split function.
- All the models are split into train and test set at 80% and 20% of the datasets respectively.
- After training, the performance of each model is evaluated using various metrics, including accuracy, precision, recall, and F1-score.
- These metrics provide insights into the effectiveness of each model in correctly classifying male and female faces.

4.4 Testing Image and live detection

4.4.1 Test Image Processing

- The trained models are applied to process test images to predict the gender of individuals depicted in these images.
- Test images are loaded, and faces are detected using the Haar cascade classifier.
- For each detected face, cropped features from the face_cascade.detectMultiScale() are taken and provided as input to the trained models for gender prediction.
- Predicted genders are overlaid on the test images, allowing visual confirmation of the model's performance.
- The code loads a dataset of male and female facial images. It uses the Haar cascade algorithm to detect faces in each image. After detecting faces, it extracts facial features and preprocesses them for gender classification.
- The models predict the gender (male or female) for each detected face. For each predicted gender, the code draws a rectangle around the face and labels it with the predicted gender.
- Finally, it displays the processed image with gender labels.

4.4.2 Integration with Live Video Feed

- Implement real-time gender detection by processing live video streams from a camera.
- The final stage involves real-time gender detection using a webcam feed.
- The webcam captures frames, which are processed in real-time to detect faces using the Haar cascade classifier.
- Cropped features are extracted from the detected faces, and the trained models predict the gender of individuals.

- Predicted genders are displayed on the video feed, enabling live gender classification of individuals in front of the camera. Overlay the predicted gender labels on the video frames for visualization.
- It follows similar steps as static image gender detection to preprocess the facial features and classify gender using the SVM, Random Forest and KNN model.
- For each predicted gender, the code draws a rectangle around the face and labels it with the predicted gender.
- It continuously updates and displays the webcam feed with real-time gender predictions.

4.5 Steps to follow while running the cells

- After training the Model, the test image popped out with gender detection label must be closed to run the next cell. Otherwise, the next cell cannot be executed.
- After checking with live detection, ‘q’ should be pressed anywhere on the live detection window to stop continuous capture of live gender detection. Otherwise, it continuously detects the gender.

Flow Chart of Implementing Gender Detection

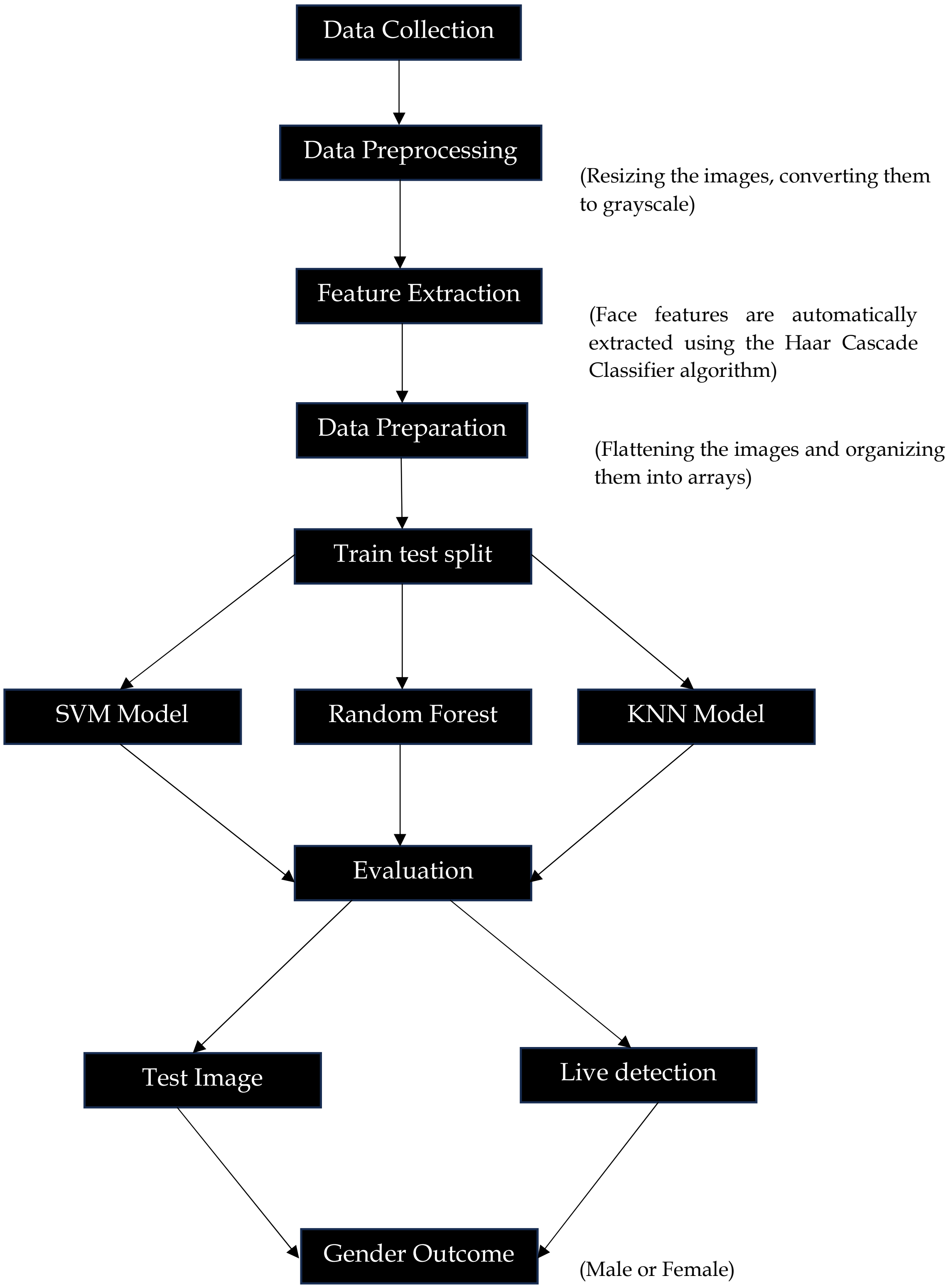

Figure 1. Gender Detection Flow Chart

Discussion

Based on the evaluation metrics we got, the algorithms can be ranked in descending order based on their performance – evaluation metrics:

Random Forest > KNN > SVM

Random Forest Model:

Reasons:

- The Random Forest model achieved the highest accuracy among all the models, indicating its superior overall performance in correctly classifying male and female instances.
- It exhibited the highest recall score, suggesting that it can identify a greater percentage of all actual male and female instances compared to the other models.
- With a balanced F1-score, the Random Forest model demonstrates a harmonious combination of precision and recall, making it the top-performing algorithm in this scenario.

Support Vector Machine (SVM) Model:

Reasons:

- While the SVM model achieved a slightly lower accuracy compared to the Random Forest model, it still performed exceptionally well in terms of precision.
- With a high precision score, the SVM model demonstrates a strong ability to correctly classify male and female instances without many false positives.
- Despite having a lower recall score than the Random Forest model, the SVM model still maintains a respectable F1-score, indicating a balanced performance between precision and recall.

K-Nearest Neighbors (KNN) Model:

Reasons:

- The KNN model achieved a slightly lower accuracy compared to both the Random Forest and SVM models but still performed well overall.
- It exhibited a high recall score, indicating its ability to correctly identify a large percentage of all actual male and female instances.
- While its precision score is slightly lower than that of the SVM model, the KNN model's balanced F1-score reflects its effectiveness in combining precision and recall.

In summary, based on the provided evaluation metrics, the Random Forest model performed the best overall, followed by the SVM model and then the KNN model.

Outputs for SVM:

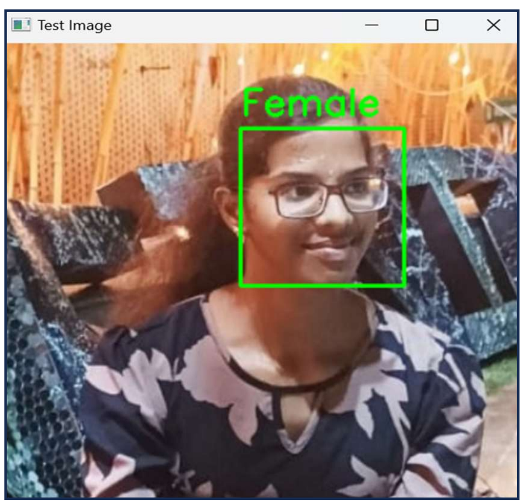

Figure 2. Output for static test image

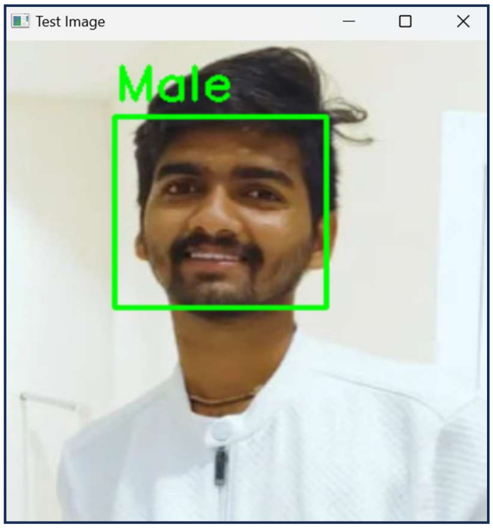

Figure 3. Output for static test image

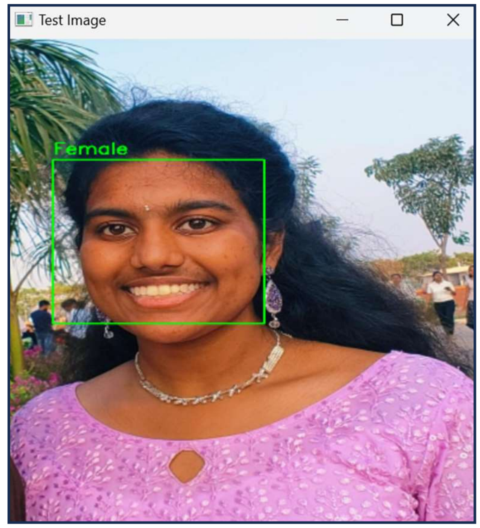

Figure 4. Output for static test image

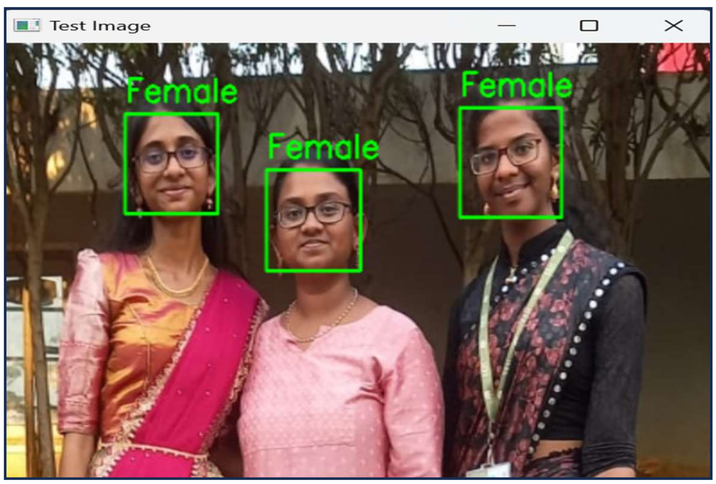

Figure 5. Output for static test image (Group)

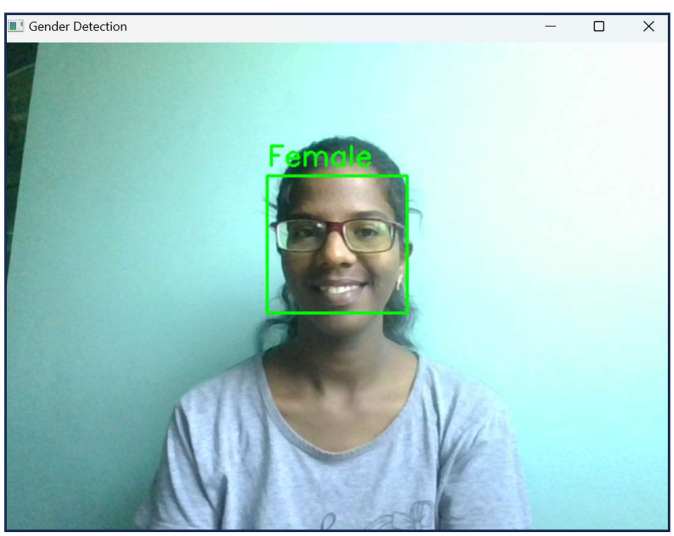

Figure 6. Output for live image

Outputs for Random Forest:

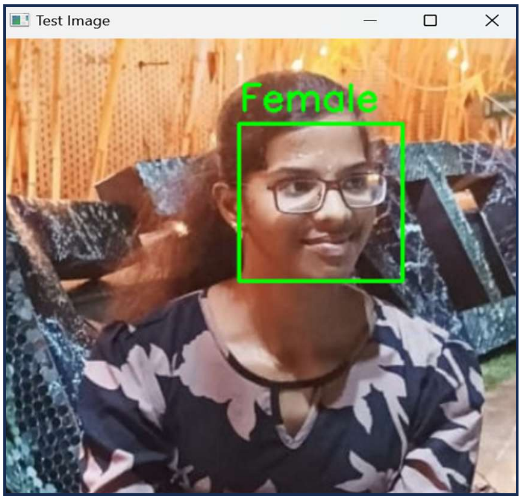

Figure 7. Output for static test image

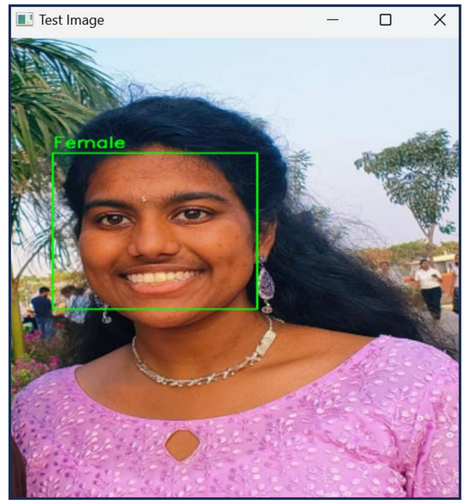

Figure 8. Output for static test image

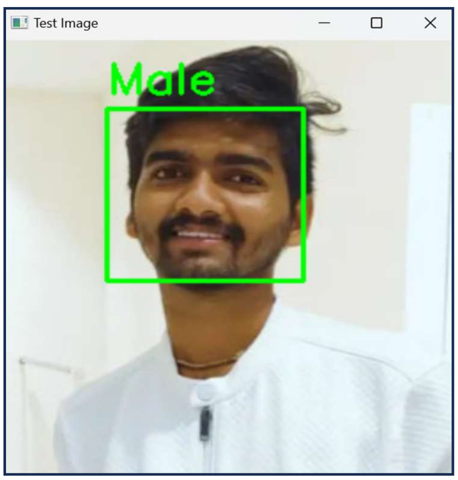

Figure 9. Output for static test image

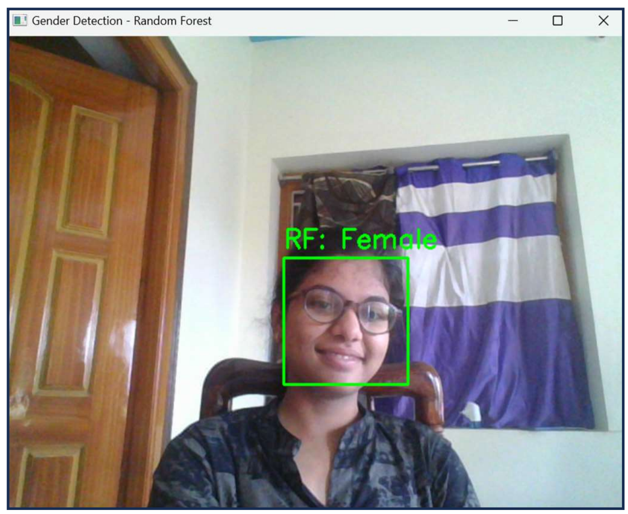

Figure 10. Output for live image

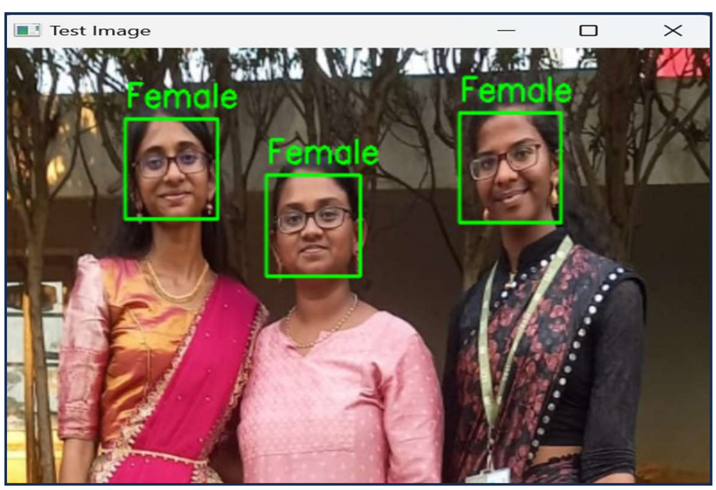

Figure 11. Output for static test image (Group)

Outputs for KNN:

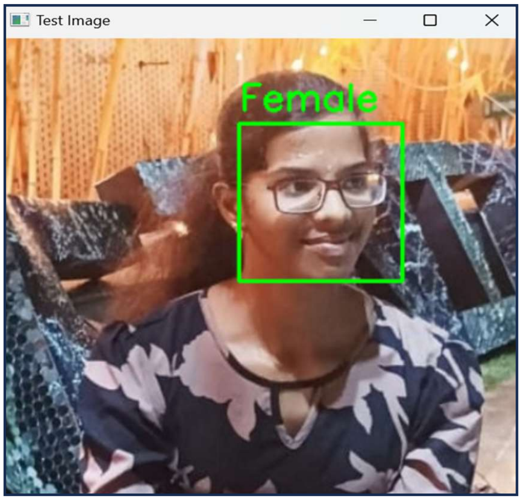

Figure 12. Output for static test

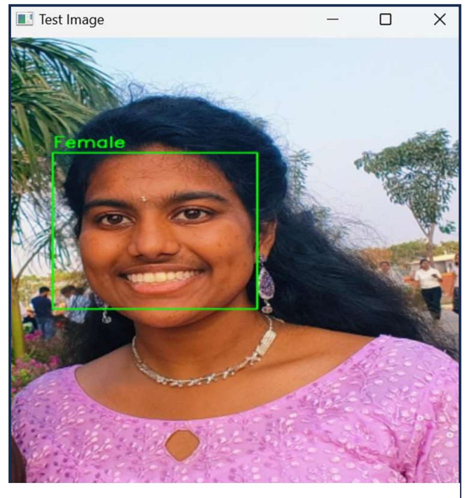

Figure 13. Output for static test

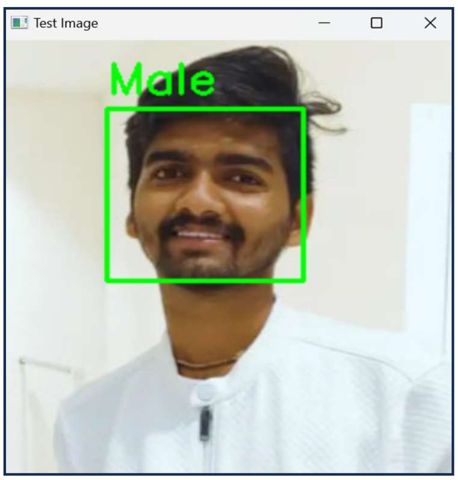

Figure 14. Output for static test

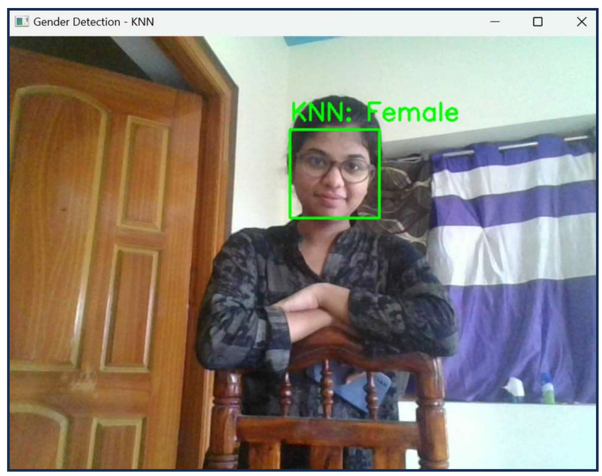

Figure 15. Output for live image

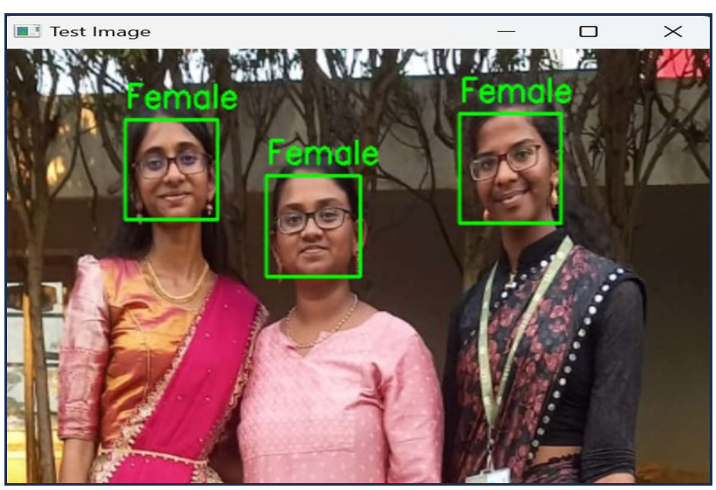

Figure 16. Output for static test image (Group)

Model Evaluations for SVM

Evaluation Metrics	Percentage
Accuracy	0.9702970297029703
Precision	0.9828926905132193
Recall	0.9619482496194824
F1-score	0.9723076923076923

Table 1. Evaluation metrics for SVM

Model Evaluations for Random Forest

Evaluation Metrics	Percentage
Accuracy	0.981023102310231
Precision	0.975975975975976
Recall	0.989345509893455
F1-score	0.982615268329554

Table 2. Evaluation metrics for Random Forest

Model Evaluations for KNN

Evaluation Metrics	Percentage
Accuracy	0.9711221122112211
Precision	0.9655688622754491
Recall	0.9817351598173516
F1-score	0.9735849056603774

Table 3. Evaluation metrics for KNN

Description of Outcomes

- All the three algorithms worked very fine.
- Overall, the choice of the "best" model may depend on the specific requirements of the application.
- In most of the cases, Random Forest is getting with high accuracy, precision, recall and f1-score.
- If minimizing false positives (precision) is crucial, SVM might be preferred.
- If minimizing false negatives (recall) is more important, Random Forest could be the better choice.
- If a balance between precision and recall is desired, KNN seems to perform slightly better.
- Additionally, considerations such as computational complexity and scalability also influence the choice of the model.
- Here for KNN, we have found the best k value using GridSearch to implement gender detection using KNN Algorithm.
- And the Best K value we got is 1.
- So KNN checks for only 1 nearest neighbour to detect the gender of a person or group.

Observations (Comparing SVM, Random Forest and KNN)

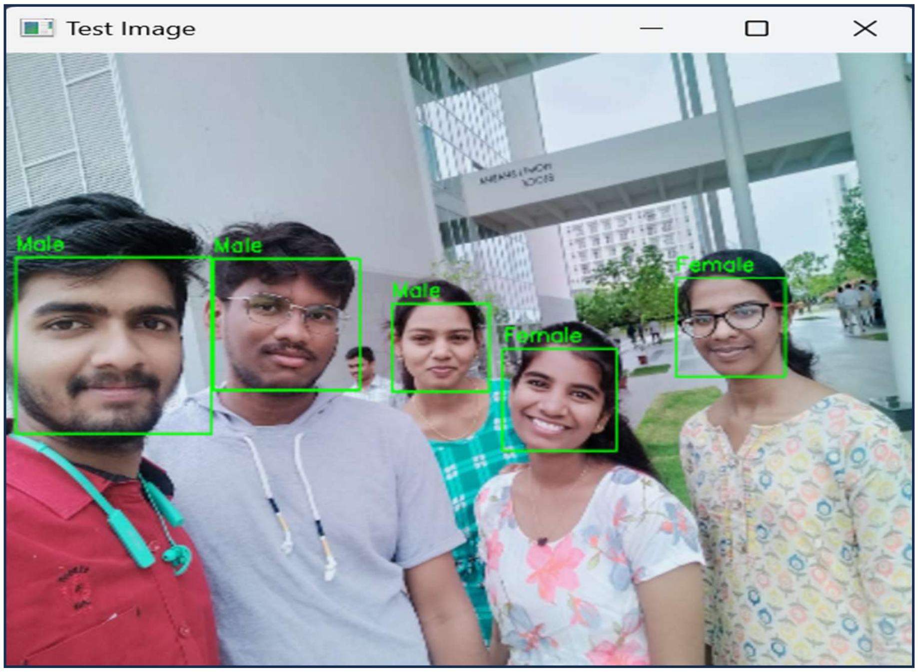

Figure 17. Gender Detection using SVM

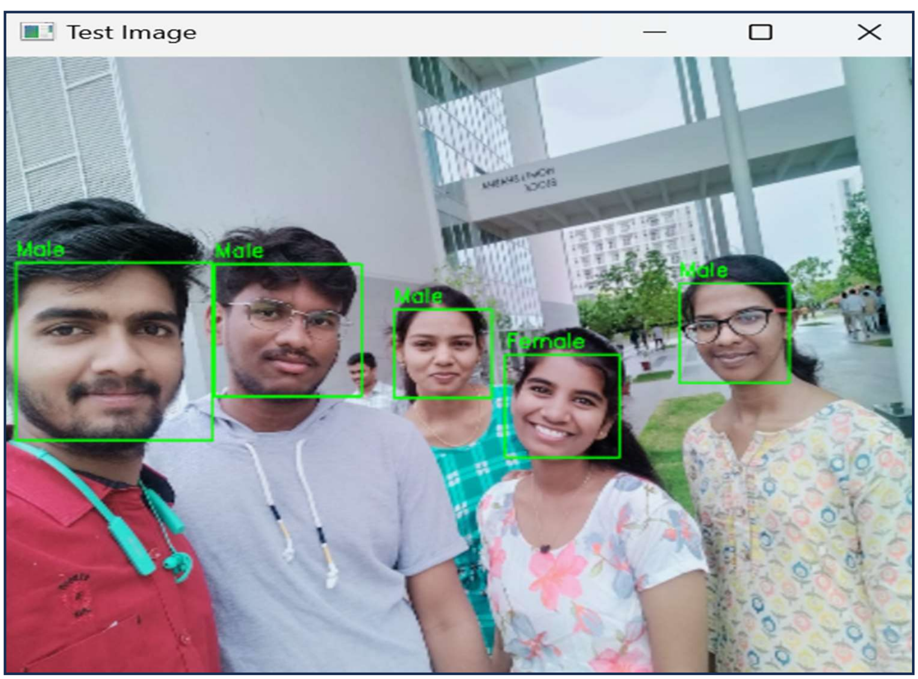

Figure 18. Gender Detection using Random Forest

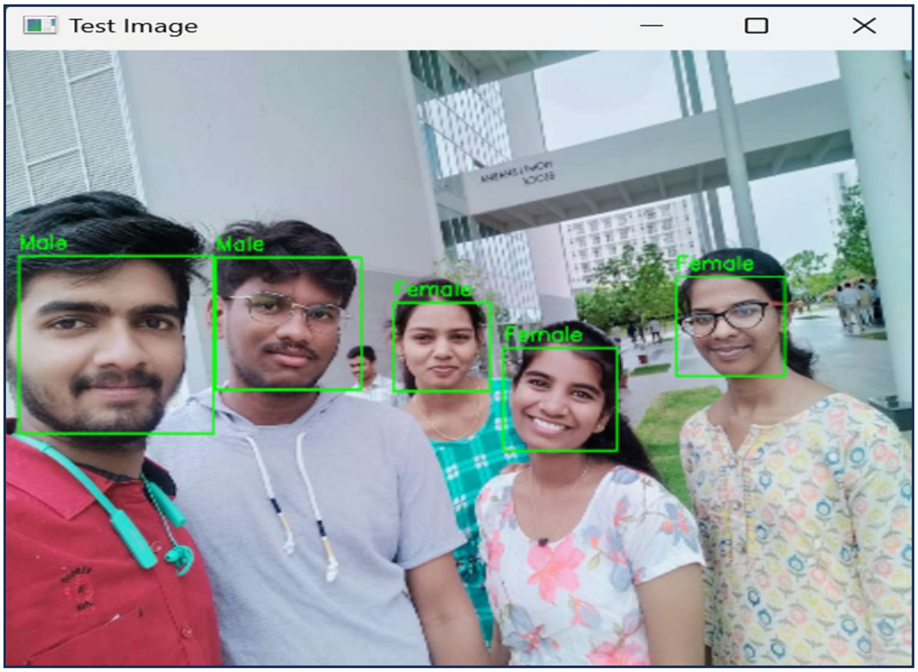

Figure 19. Gender Detection using KNN

We can observe that same group image is tested with all the three algorithms.

- Though Random Forest is having slightly higher in the evaluation metrics compared to KNN and SVM, this image is getting correctly classified only when KNN algorithm is used.
- Mostly Random Forest is detecting some females also as males in a group image compared to SVM whereas KNN is working good in the case of group images.
- KNN is working good for group images in all the cases while SVM works better than Random Forest. The k value we got using GridSearch algorithm is 1.
- If the optimal value of k is chosen through hyperparameter tuning, KNN might perform better for the specific group image compared to Random Forest, which has its own set of hyperparameters that need to be tuned.

Simply, For group images best model in descending order

KNN > SVM > Random Forest

Limitations

- Sensitive to Lighting Conditions: The performance of face detection algorithms can be affected by variations in lighting conditions. Illumination changes, shadows, and reflections can degrade the accuracy of gender detection.
- Haar Cascade and many machine learning-based gender detection algorithms rely on detecting facial features in a specific orientation, typically frontal or slightly angled faces. When images are completely rotated (>= 90 degrees), these algorithms may fail to detect faces accurately, leading to erroneous gender predictions.
- The project is made to detect only male and female genders other than neutral genders is a significant constraint.

Concluding Remarks

Successful Implementation: The project successfully implements machine learning models for gender detection from facial images, showcasing the practical application of computer vision and machine learning techniques.

Model Comparison: Through rigorous evaluation, the project compares the performance of three different models: Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN). This comparison provides valuable insights into the strengths and weaknesses of each approach.

Real-time Application: The inclusion of live gender detection functionality demonstrates the potential real-world application of the developed models. This feature opens up possibilities for integrating the solution into various systems, such as security systems, user profiling, or customer analytics.

Future Directions: There are several avenues for future exploration and improvement. This includes experimenting with different feature extraction techniques, exploring ensemble methods, incorporating deep learning approaches for feature learning, and enhancing the system's robustness to variations in lighting, pose, and facial expressions.

The project lays a solid foundation for gender detection from facial images, offering valuable insights, practical implementations, and avenues for further research and development.

Future Work

- **User Interface Enhancements:** Building a user-friendly interface for the gender detection system, allowing users to interact with the system easily and providing visual feedback on the classification results.
- **Inclusion of Other Gender Identities:** Expanding the scope of the gender recognition system to include a broader spectrum of gender identities beyond binary classification (male/female).
- **Handling Occlusions and Variations:** Developing methods to handle facial occlusions, variations in lighting conditions, facial expressions, and hairstyles to improve the model's robustness in real-world scenarios.
- **Privacy and Ethical Considerations:** Considering privacy and ethical implications related to facial recognition technology. Implement mechanisms to ensure data privacy and mitigate potential biases in the model.

References

1. Dantcheva, A., Velardo, C., Dugelay, J. L. (2012). Gender detection from unconstrained and articulated human body.
2. Jung, H., Kim, J., Lee, K. (2020). Real-time gender classification using convolutional neural networks from facial images.
3. Chen, Z., Ge, H., Liu, L., Xu, C. (2015). Gender recognition by combining facial, body and gait features.
4. Shao, M., Liu, Z., Xu, X., Song, B., Shen, Y. (2019). Gender classification from facial images using machine learning techniques.
5. Lei, Y., Shan, S., Gao, W., Li, S. Z. (2017). Learning discriminant face descriptor.
6. Gangaprasad, S., Govindan, V. K. (2021). Gender classification from facial images using ensemble methods.
7. Huang, G. B., Ramesh, M., Berg, T., Learned-Miller, E. (2020). Labeled faces in the wild: A survey.
8. Liu, W., Wen, Y., Yu, Z., Li, M. (2018). Feature learning based deep supervised hashing with pairwise labels.
9. Hao, Q., Zhang, L., Zhao, B. (2022). Gender classification based on facial images using machine learning techniques.