Rapid Visual Multi-Face Detection

1st Sankalp Bhamare
Electrical Engineering
200110096@iitb.ac.in

2nd Vansh Kapoor
Electrical Engineering
200100164@iitb.ac.in

3rd Ankur Verma
Electrical Engineering
20d070014@iitb.ac.in

***Abstract*—This paper presents two machine learning approaches for visual multi-face detection, capable of processing images quickly and achieving high detection rates. The first approach is a Haar features-based Adaboost Cascade Classifier, which combines classical image processing concepts with machine learning techniques. It utilizes the modern technique of ”Integral Image Representation” for rapid computation of Haar features, subsequently used by Adaboost for classifier learning. We optimize a cascade of such strong classifiers for faster face detection. Our second approach employs classical Convolutional Neural Networks to detect faces through a moving window running across the entire image. We also conduct a comparison of the speed of both methods.**

***Index Terms*—Adaboost, Convolutional Neural Network, Haar features, Integral Image.**

I. Introduction

A. Motivation and Proposal

Our project's primary objective is to implement a technique for visual multi-face detection that not only yields accurate results but also exhibits minimal computational complexity. There is a ever growing demand of quick visual multi-object detection in various fields such as

- **Security and surveillance**:Rapid face detection has the potential to be used in real-time for identifying individuals, thereby enhancing public safety and preventing criminal activities. It is also useful in photo detection software for face enhancement and manipulation.
- **Finance and Marketing**: Businesses can use rapid face detection to analyze customer reactions to products and advertisements, which can provide valuable insights to improve their marketing strategies
- **Gaming and Tracking**: Real-time facial expression recognition and tracking can be achieved using rapid face detection, making it useful in applications such as gaming, virtual reality, and augmented reality. photo-editing software
- **Medical Image Processing**: Doctors and medical professionals can use rapid face detection to identify and diagnose medical conditions that present in the face

While modern machine learning methods are renowned for providing exceptional accuracy, they often require substantial computational time. By synergizing classical image processing concepts that entail lower computational complexity with modern machine learning techniques that ensure higher accuracy, we aim to meet our project's demands with success

Identify applicable funding agency here. If none, delete this.

II. Background Research

We conducted an extensive literature review that encompassed [1], [2], and [3]. In [1], the Haar feature extractor was employed, followed by Adaboost for classifier learning and a cascade of strong classifiers for prediction. Given that Haar features were utilized in the first method, we invested a substantial amount of time comprehending their implementation in the algorithm.
Our effort proved to be quite beneficial, given that the object detection accuracy is heavily influenced by the size and type of Haar filters utilized. Additionally, we thoroughly explored Adaboost algorithm and cascade classifiers, along with their variations. For Method-II, we incorporated our unique ideas while leveraging the core concepts of [2].
To gain further insight into modern implementations of multi-object detection, we scrutinized [3], which employs the *Single Shot Detector* that accomplishes object detection in a single forward pass of the neural network, thereby enabling faster and more efficient detection than traditional object detection techniques that necessitate multiple passes.

III. Innovative Modifications Implemented

- We integrated our Python code with a **live webcam**, enabling it to detect faces and display them with bounding boxes.
- After analyzing various resources and research papers, we observed that most of them focused on adding a new algorithm or modification to a popular machine learning method, without incorporating any classical image processing techniques. However, in our first method, we implemented an **image sharpening** technique which significantly enhanced the classification accuracy.
- The sharpening process improved the efficacy of Haar filters, as they rely on the gradient of an image subpart. Additionally, we also experimented with **histogram equalization**, but its impact on accuracy was inconclusive.
- For Method-II, we drew inspiration from the algorithm mentioned in [2] and made certain modifications by incorporating classical image processing techniques in a similar manner.

IV. Methods

A. Haar Features-Based Adaboost Cascade Classifier

1) Algorithm

- **Integral Image**: The computation of the sum of all the pixels above and to the left of a particular pixel, including the pixel itself, results in the Integral Image at that point. Integrating an image in this manner facilitates fast computation of the pixel sum in any bounding box, thereby expediting the Haar kernel convolution process.
- **Haar Feature Extractor**: Haar features play a vital role in face detection as they detect the gradient of a specific part of an image. These features are particularly useful for detecting important facial landmarks such as the region between the eyes and cheeks, cheeks and nose, and eyebrows and skin.

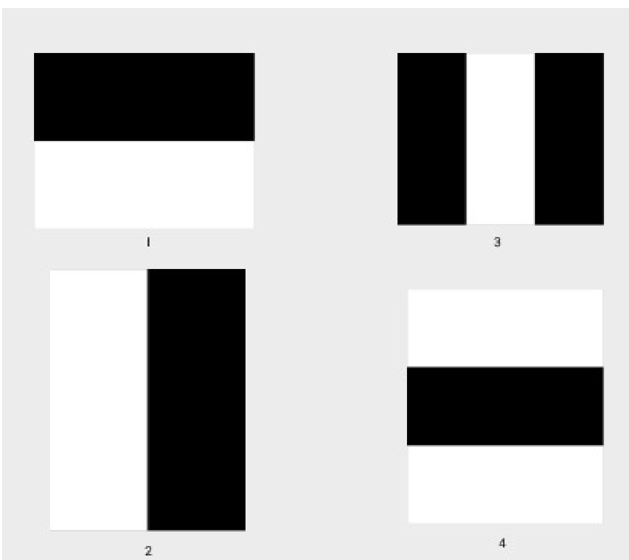

Fig. 1. The image displays four types of Haar features used for face detection. Type 1 detects the region between the eyes (darker) and cheeks (lighter), Type 2 detects the region between cheeks (lighter) and nose (darker), Type 3 detects the nose (darker) with cheeks (lighter) on both sides, and Type 4 detects eyebrows (darker) with skin (lighter) on both sides.

- **Adaboost** Adaptive Boosting, also known as Adaboost, is a classification and regression analysis algorithm in machine learning. It involves combining several "weak" classifiers to form a "strong" classifier. The final "strong" classifier is obtained as a weighted sum of the weak classifiers. During the classification phase, the strong classifier evaluates each weak classifier and integrates their results to arrive at a final decision. The algorithm is mentioned below

> - Given example images (x_1,y_1), $\cdots$, (x_n,y_n) where y_i = 0, 1 for negative and positive examples respectively
> - Initialize weights $w_{1,i} = \frac{1}{2m},\frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
> - For $t = 1, \cdots , T$:
>   1) Normalize the weights,
> $$w_{t,i} \Leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
>   so that w_t is a probability distribution.
>   2) For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i|h_j(x_i) - y_i|$.
>   3) Choose the classifier, h_t, with the lowest error ϵ_t.
>   4) Update the weights:
> $$w_{t+1,i} = w_{t,i}\beta_t^{1-\epsilon_i}$$
>   where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
> - The final strong classifier is:
> $$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
> where $\alpha_t = \log \frac{1}{\beta_t}$

- **Cascade Classifier** The cascade classifier framework involves integrating a strong classifier obtained from training with other strong classifiers. An optimal number of cascade classifiers that best fits the training data is then obtained through training the data.

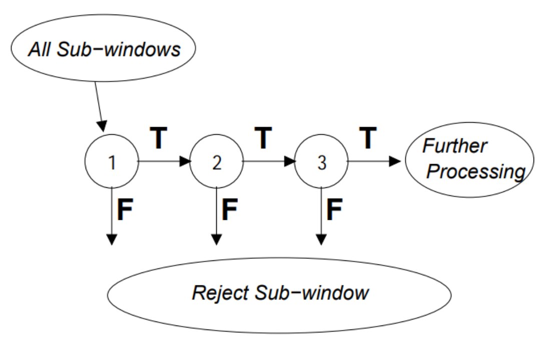

Fig. 2. Operation of Cascade Classifiers

B. Object Detection Using Classical Convolutional Neural Network

Method-II implements object detection by training a convolutional neural network with 2 hidden convolutional layers and 1 fully connected layer.

The first hidden layer consisted of 32 (3,3) kernels with ReLu

activation, followed by a maxpool layer of size (2,2). The second convolutional layer used 64 (3,3) kernels with ReLu activation and max pooling.
The output of this layer was then flattened and input to the fully connected layer, which provided the final classification of face or non-face. A batch size of 16 was found to provide the best accuracy during training.

V. Dataset

The model was trained on a data set which consisted of blurred images with faces and without faces. After training we performed object detection through a moving window running across the entire image. We also conduct a comparison of the speed of both methods

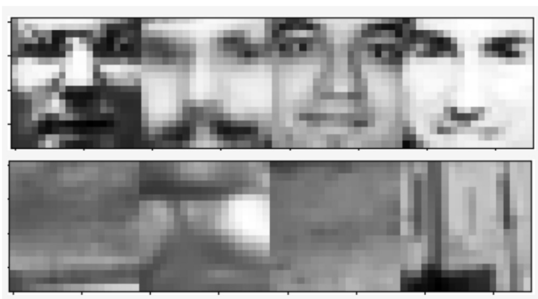

Fig. 3. The training dataset contains two rows, where the first row includes images of faces, and the second row comprises images without any faces.

VI. Results

While Method-I produced excellent results for object detection, similar to those reported in [1], some faces were still misclassified or not detected. However, we were able to achieve outstanding results with Method-I, achieving 100% training accuracy and 96% test accuracy.
On the other hand, in method-II, we achieve a training accuracy of 99.6% and a test accuracy of 98%.

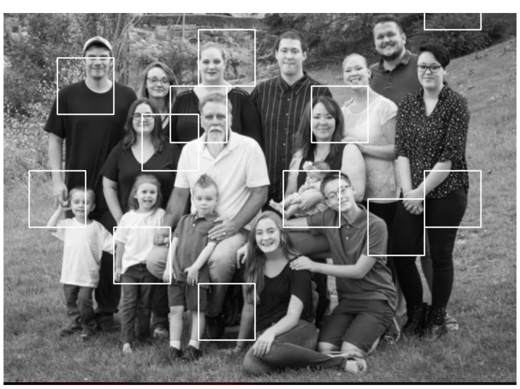

Fig. 4. Multiple Face Detection Using Method-I

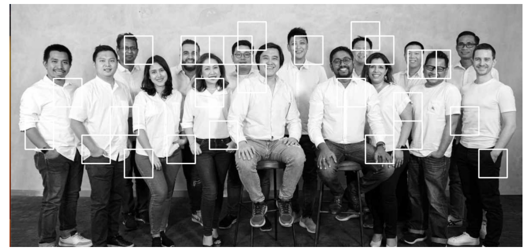

Fig. 5. Multiple Face Detection Using Method-I

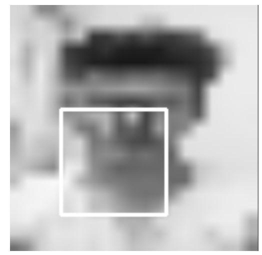

Fig. 6. Face Detection of a Blurr Image Using Method-II

Though both Method-I and Method-II gave similar training and validation scores, Method-I performed better than Method-I in terms of both computation speed and test accuracy

VII. Contribution

- **Project Formulation**: Vansh
- **Data Collection and Processing**: Sankalp
- **Method-I Implementation**: Sankalp and Vansh
- **Method-II Implemetation**: Ankur and Vansh
- **Video Presentation**: Sankalp and Ankur
- **Report**: Vansh

References

[1] P. Viola and M. Jones, ”Rapid object detection using a boosted cascade of simple features,” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[2] H. A. Rowley, S. Baluja and T. Kanade, ”Neural network-based face detection,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, Jan. 1998, doi: 10.1109/34.655647.

[3] B. Ye, Y. Shi, H. Li, L. Li and S. Tong, ”Face SSD: A Real-time Face Detector based on SSD,” 2021 40th Chinese Control Conference (CCC), Shanghai, China, 2021, pp. 8445-8450, doi: 10.23919/CCC52363.2021.95502

[4] MIT CBCL Face Database: http://cbcl.mit.edu/software-datasets/FaceData2.html