ANALYSIS AND RESULTS

Comparative Study between Shuffle Net Algorithm, MTCNN, Shufflenet and MTCNN combined Algorithm for Face detection:

A comparative study between ShuffleNet Algorithm, MTCNN, ShuffleNet and MTCNN combined Algorithm for Face Detection is an interesting topic. Here is a brief overview of each algorithm and their respective strengths and weaknesses:

ShuffleNet Algorithm:

ShuffleNet is a lightweight neural network architecture designed for mobile and embedded devices. It uses a group convolutional operation to reduce computation and memory requirements, while maintaining high accuracy. ShuffleNet can be trained to perform face detection, but it may not be as accurate as other more specialized algorithms.

MTCNN:

MTCNN is a specialized face detection algorithm that uses a multi-stage neural network to detect and locate faces in an image. It is known for its high accuracy and ability to handle faces of varying sizes, orientations, and lighting conditions.

ShuffleNet and MTCNN Combined Algorithm:

The ShuffleNet and MTCNN combined algorithm uses a combination of the two algorithms to achieve high accuracy while maintaining a low computational cost. ShuffleNet is used to pre-process the input image and reduce the computational load on MTCNN. MTCNN is then used to perform accurate face detection on the pre-processed image.

Comparing these algorithms, we can see that MTCNN is the most accurate and specialized face detection algorithm. However, it is computationally expensive, making it unsuitable for mobile and embedded devices. ShuffleNet is lightweight and efficient but may not be as accurate as MTCNN. The combined algorithm offers a good balance between accuracy and computational efficiency, making it suitable for mobile and embedded devices.

In summary, the choice of algorithm will depend on the specific requirements of the application. If accuracy is the top priority, MTCNN is the best choice. If computational efficiency is more important, ShuffleNet or the combined algorithm may be a better choice.

Table for Number of faces detected in Shufflenet vs MTCNN vs ShuffleNet and MTCNN Combined Algorithm:

Name of the video	**Shufflenet**	**MTCNN**	**Shufflenet + MTCNN**
	Number of faces detected	Number of faces detected	Number of faces detected
American Beauty - 00170	111	222	222
American Beauty - 00222	58	244	244
American Beauty - 00443	164	350	350
American Beauty - 00951	300	248	248
American Beauty - 01597	562	1342	1342
As Good As It Gets - 01766	279	830	830
As Good As It Gets - 01935	149	454	454
Big Fish - 00674	439	1621	1621
Big Lebowski, The - 00818	128	378	378
Casablanca - 03025	62	202	202

Table 1: Number of faces detected in Shufflenet, MTCNN and Shufflenet and MTCNN combined Algorithm.

Graph:

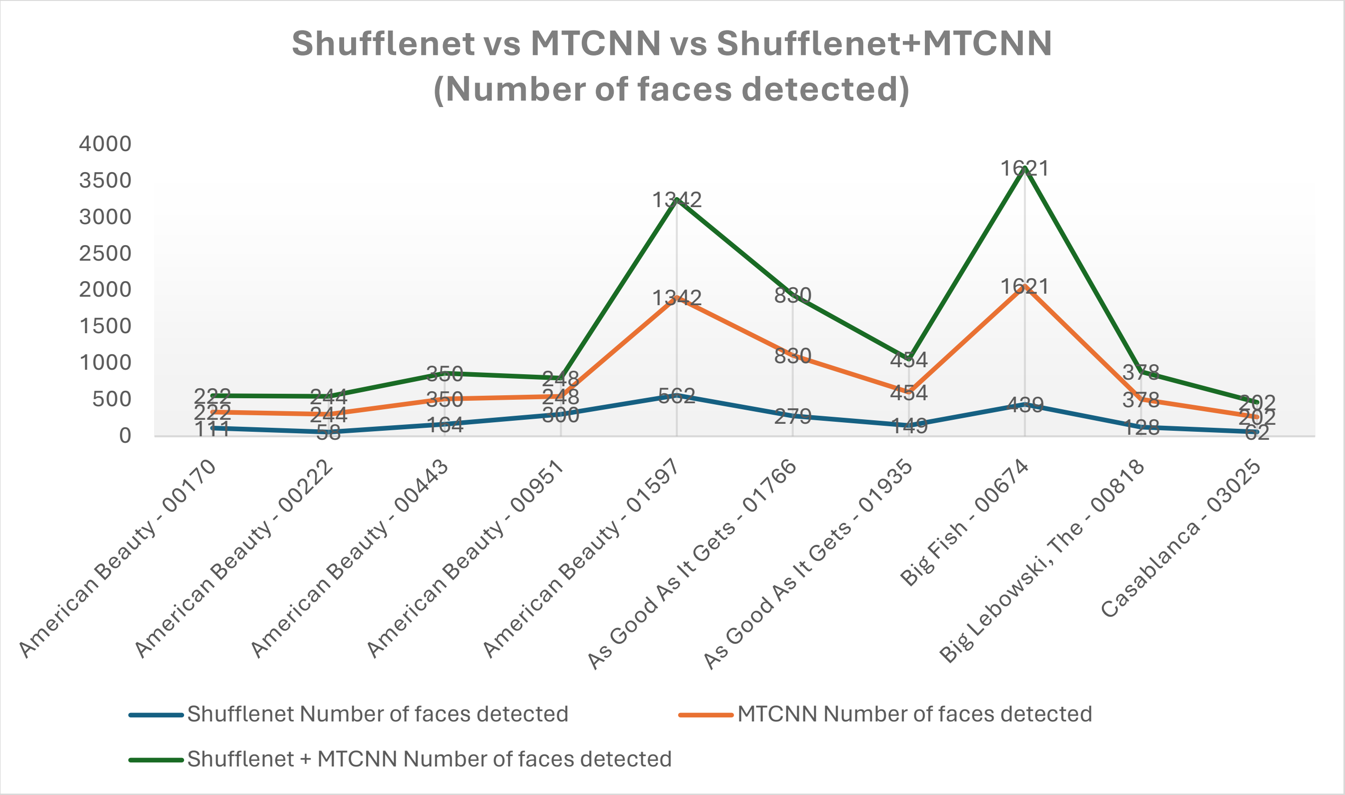


Figure1: Graph for Comparative Study between Shuffle Net Algorithm, MTCNN, Shufflenet and MTCNN combined Algorithm for Face detection in data set of 10 videos on number of faces detected.

Description: - Here, it represents number of faces detected in Shufflenet, MTCNN and combined Algorithm for shufflenet and MTCNN.

Table for Execution time (in sec) in Shufflenet vs MTCNN vs ShuffleNet and MTCNN Combined Algorithm:

Name of the video	**Shufflenet**	**MTCNN**	**Shufflenet+MTCNN**
	Execution time (in seconds)	Execution time (in seconds)	Execution time (in seconds)
American Beauty - 00170	12.083	14.501	13.823
American Beauty - 00222	9.534	12.689	11.56
American Beauty - 00443	71.818	72.739	73.026
American Beauty - 00951	164.069	166.422	165.968
American Beauty - 01597	136.176	137.285	137.185
As Good As It Gets - 01766	47.77	53.272	51.357
As Good As It Gets - 01935	23.334	25.554	54.445
Big Fish - 00674	37.99	52.685	51.164
Big Lebowski, The - 00818	23.307	24.953	23.307
Casablanca - 03025	12.931	14.865	13.396

Table 2: Execution time in Shufflenet, MTCNN and Shufflenet and MTCNN combined Algorithm.

Graph:

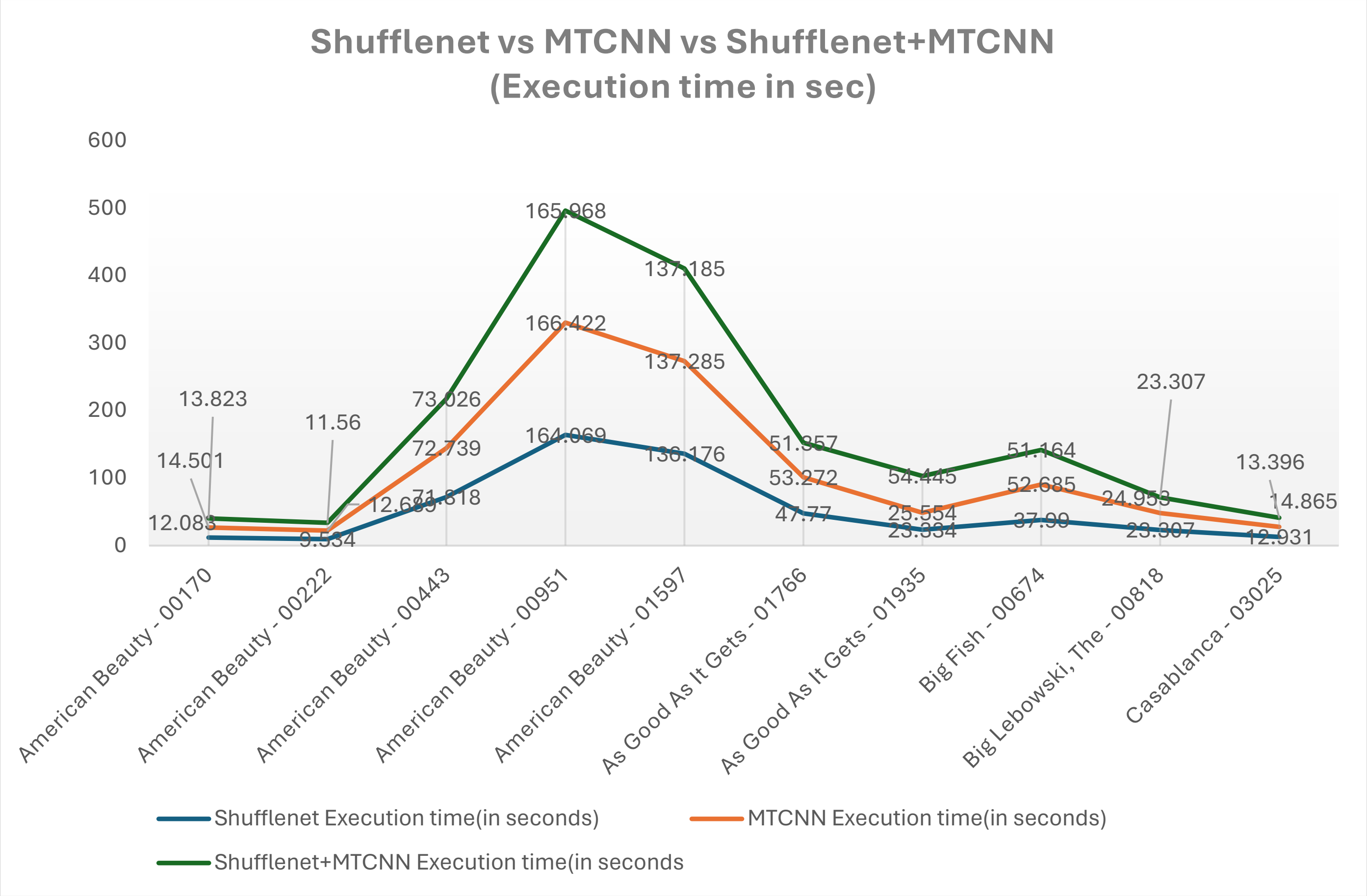


Figure 2: Graph for Comparative Study between Shuffle Net Algorithm, MTCNN, Shufflenet and MTCNN combined Algorithm for Face detection in data set of 10 videos on number of faces detected.

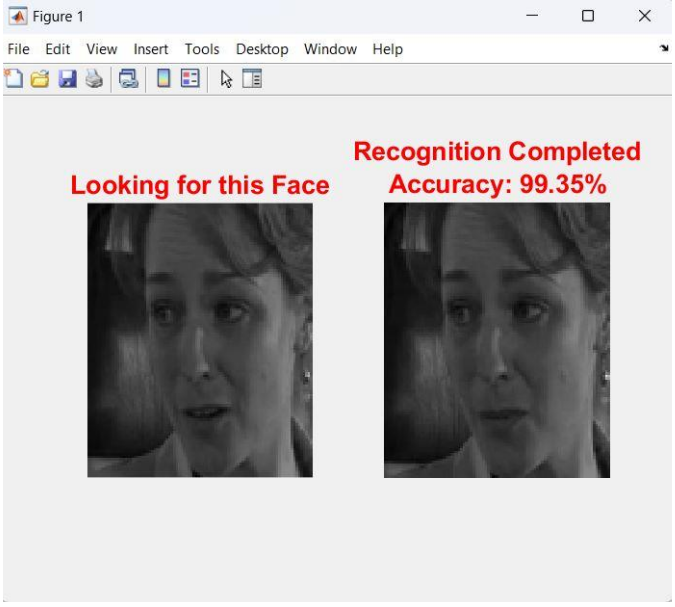
Description: - Here, it represents number of faces detected in Shufflenet, MTCNN and combined Algorithm for shufflenet and MTCNN.

Explanation of the output of eigenface recognition, which is a technique for facial recognition using principal component analysis (PCA):

The first code loads a set of 100 facial images from a database and returns them as a 2D matrix of pixel values. The matrix has a size of 10304 x 100, where each column represents a different facial image. The images are grayscale and resized to a fixed size of 112x92 pixels. The function uses the Persistent variable to store the loaded images in memory so that they can be accessed faster if the function is called again in the future. The output of this algorithm is the matrix of facial images, which can be used as input to the eigenface recognition algorithm or other facial recognition methods.

The second algorithm performs the eigenface recognition process on a set of facial images loaded from a database. The output of this algorithm is a pair of images displayed side by side. The left image shows the target face that the algorithm is trying to recognize, and the right image shows the closest matching face found in the database along with the accuracy of the recognition in percentage.

Output:



Looking for this Face

Recognition Completed
Accuracy: 99.35%

Figure 3: Output of face Recognition using eigenface.