**A blue and black logo

Description automatically generated**

**Face Recognition Using Dynamic Time Wrapping Algorithm**

**Student Name 1: Muhammad Haris**

**Student Name 2: Yatish Sharma**

**Student ID 1: 1440274**

**Student ID 2: 1457597**

**Course: Computational Intelligence - SoSe 23**

**Instructor: Prof. Andreas Pech**

**Submission Date: 31/09/2023**

**Table of Contents**

1. **Summary**
2. **Literature Review**
3. **Methodology**
   1. **Facial Landmark Technique**
   2. **Profile matching using Dynamic Time Warping**
4. **Results**
5. **Conclusion**

**1. Summary**

Most of the existing profile recognition algorithms rely on accurate fiducial point identification and associations among these fiducial points being established. Unfortunately, several characteristics such as a flat chin, projecting lips, a concave nose, etc. make it challenging and unreliable to detect these spots. Additionally, even for the same person, the quantity and location of fiducial sites fluctuate as expression does. This work presents a shape predictor using landmark matching method that makes use of profile data rather than requiring the extraction of all fiducial points. A dynamic time warping method is applied to match the face profile portion from nasion to throat based on the detect landmark with shape detection. Experiments are performed on profile face image databases.

**2. Literature Review**

The facial profile, which offers a supplemental structure of the face not seen in the frontal view, is a crucial component for the recognition of faces. Although it has less inherent discriminating capability than frontal pictures, it is more reliable and very simple to assess.

Several methods for automatic person identification using facial profile photos have been put out in the previous ten years. Most of these algorithms rely on the accurate identification of each fiducial point as well as the identification of the connections between these fiducial points.

Tangency-based procedures presumptively find a reference point that is tangent to the face profile at one of the fiducials by selecting the proper reference point. Profile traces manually recorded from pictures of 256 male faces. The nineth fiducial is obtained by rotating a point from the chin about the pronasale until it touches the profile above the pronasale. They find eight separate fiducials on the profiles. Later, use a 17-dimensional feature vector to raise the number of fiducials from nine to eleven and obtain 96% recognition accuracy for 112 participants. For profiles with protruding lips, the biggest issue with tangency-based approaches is the lack of a line that is bitangent to the pronasale and chin.

Some of these drawbacks are solved by Landmark detection approaches, which are also invariant to rotation, translation, and uniform scaling. Scale space approaches are used to analyze the face's profile and derive eight fiducials. This method implies that there would be nine zero-crossings on the profile, although moustaches and the hairline on the forehead could render this assumption inaccurate. Based on the finding that the profile alternates between convex and concave curvature, and that the location of maximal absolute curvature in each segment corresponds to a fiducial, extract nine fiducials. For face recognition, face profiles were generated from 3D range photos. A 3D range image's pronasale and nasion are extracted using Gaussian curvature. The method defines multiple face descriptors using high level sets of relationships between depth and curvature.

**3. Methodology**

Even though there are numerous techniques for finding fiducial points, no method has yet been able to consistently extract all the prominent fiducials for every face. This should not be surprising given the wide range of regular face profiles. Although the five fiducials that Galton employed in 1910 are typically included in the set of fiducials, the fiducials that researchers have isolated over the years have varied. A feature vector approach based on the same fiducial points of many face profiles will fail when a particular profile is too demanding for all fiducials to be consistently recovered.

Our project is around face recognition using dynamic wrapping algorithms, first we collect the training dataset of images. We computer the facial landmarks of the dataset images and collect the facial landmarks of training dataset in model database. The facial profile region from the nose to the throat is then compared using a dynamic time warping technique depending on the landmark value. Because dynamic time warping is a far more reliable distance measure for time series than Euclidean distance, it was chosen as the matching approach because it allows for the matching of comparable shapes even when they are out of phase with respect to time. The whole project report is shown in Figure 1.

A close-up of a white circle

Description automatically generated

**Figure 1. Technical approach**

**3.1 Facial Landmark Technique**

We are using facial landmark techniques for extraction of fiducial point of the face. The brows, eyes, jaw, nose, and mouth, among other prominent areas or facial components of the person's face, are utilized as facial landmarks to localize and portray the person's face. A method called facial landmarks may be used for tasks like face alignment, head posture estimation, face swapping, blink detection, and tiredness detection, among others. Our primary objective in this context of facial landmarks is to identify facial structures on the subject's face using a technique called shape prediction. Facial Landmark detection of images is shown.

A person with facial hair and beard

Description automatically generatedA person smiling with red dots

Description automatically generated

A person with red dots on his face

Description automatically generated

**Figure 2. Glimpse of Facial Landmarks on Celebrity Dataset**

The function used for detection of landmarks with the help dlib library is shown in figure 2.

# Function to detect facial landmarks

def detect\_landmarks(image\_path):

    image = cv2.imread(image\_path)

    gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    # Load the pre-trained facial landmark model

    predictor = dlib.shape\_predictor("/content/shape\_predictor\_81\_face\_landmarks.dat")

    detector = dlib.get\_frontal\_face\_detector()

    # Detect faces in the image

    faces = detector(gray)

    # Assume only one face is present in the image

    if len(faces) == 0:

        print("No faces detected.")

        return None

    # Get the facial landmarks for the first face

    shape = predictor(gray, faces[0])

    landmarks = []

    for i in range(68):

        x, y = shape.part(i).x, shape.part(i).y

        landmarks.append([x, y])

    return np.array(landmarks)

**Figure 3. Function used to detect Facial landmarks**.

**3.2 Profile matching using Dynamic Time Warping**

For recognition of face, we are using Dynamic Wrapping algorithm using DTW distance matrix approach. An approach for comparing two sequences that may differ in length or duration is called Dynamic Time Warping (DTW). Aligning the two sequences and determining a distance function between them allows for the comparison of similarity. The absolute values of curvature, which are utilized to describe the forms of facial contours, are used to compute the similarity of two profiles, and we select the dynamic time-based warping as the matching technique. An approach to determine the best score and alignment between two strings is called dynamic time warping. This approach allows for the matching of comparable shapes even when they are out of phase in the time axis, making it a considerably more reliable distance measurement for time series than Euclidean distance.

The ideal alignment of two sequences is determined by using the Needleman-Wunsch [10] global alignment method while considering their full length. We compute D(i,j), for complete sequences for two strings s[1...n] and t[1...mt, where i ranges from 1 to m and j ranges from 1 to n. D(i,j) is defined as:

*`****D ( i , j ) = min{ D [ i-1 , j-1 ] + d ( s[j] , t[i] ), D [ i-1 , j ]+ cost, D [ i , j-1 ] + cost }.***

(1)

The similarity between two places on the facial profiles is shown here by the expression d ( s[j] , t[i] ). Since the curvature of the profile represents the facial profile, d ( s[j] , t[i] ) is determined using the Euclidean distance.

***d ( s[j], t[i] ) = || s[j] – t[i] ||.***

(2)

The repercussions are specified for both vertical and horizontal gaps. It ought to be modest and serve just to regulate nondiagonal motions. In general, the fines need to be less than 1/100 of the highest value of the d ( s[j], t[i] ) . We apply the same constant penalty in our approach to both horizontal and vertical cost. The alignment's best score is D (m, n), which is the final score.

.

The function used for implementation of Dynamic Wrapping algorithm is displayed in figure 3.

# Function to calculate DTW distance

def calculate\_dtw\_distance(sequence1, sequence2):

    # Normalize the sequences

    sequence1 = sequence1 / np.linalg.norm(sequence1, axis=1, keepdims=True)

    sequence2 = sequence2 / np.linalg.norm(sequence2, axis=1, keepdims=True)

    # Initialize the DTW matrix

    dtw\_matrix = np.zeros((len(sequence1), len(sequence2)))

    # Fill the DTW matrix

    for i in range(len(sequence1)):

        for j in range(len(sequence2)):

            cost = euclidean(sequence1[i], sequence2[j])

            if i == 0 and j == 0:

                dtw\_matrix[i][j] = cost

            elif i == 0:

                dtw\_matrix[i][j] = cost + dtw\_matrix[i][j - 1]

            elif j == 0:

                dtw\_matrix[i][j] = cost + dtw\_matrix[i - 1][j]

            else:

                dtw\_matrix[i][j] = cost + min(

                    dtw\_matrix[i - 1][j],        # insertion

                    dtw\_matrix[i][j - 1],        # deletion

                    dtw\_matrix[i - 1][j - 1]     # match

                )

    return dtw\_matrix[-1][-1]

**Figure 4. Function used to calculate distance using Dynamic Time Wrapping algorithm.**

**4. Result**

We used two different images of celebrity dataset with different angles and expression, computing the distance between the images using dynamic wrapping algorithm. We took out the results of four different actors, images of them of a separate event with different angles and expression. The threshold set for distance is 5. The images DTW distance is less than 5 is recognize while the distance above 5 is consider not recognized. The threshold may vary because of the angle of the photo taken.

First result is of image of Angelina Jolie, we compare two different images of her from two separate photoshoots. The threshold was set at 5. The distance we obtained was 2.75 and the system recognized the face from the database.

A person with red dots on her face

Description automatically generatedA person with red dots on her face

Description automatically generated

**Face recognized! DTW Distance: 2.7502686009400357**

The second result is of actor Jhonny Depp, the images are of frontal view, one image is taken closer while another image is taken far off. The DTW distance obtained was 4.98.

A person with a beard and mustache

Description automatically generatedA person with a mustache and beard

Description automatically generated

**Face recognized! DTW Distance: 4.985531747388747**

The third result is of actor brat Pit, same process two photos of him at two different event and from two different angles. The threshold was set same on 5. The result we obtained was little vivid because the images are from two completely different angles. The distance between these images was 9.19. The system identifies the face to be unrecognized with 9.19 because the threshold was set on 5.

A person with a beard and mustache

Description automatically generatedA person in a tuxedo

Description automatically generated

**Face not recognized. DTW Distance: 9.19398039838107**

Then we changed the threshold to be around 10 and fixed the threshold at 10 and considered the recognition at distance less the 10. After applying the threshold to be at 10, the system detected the face to be recognized.

A person with a beard and mustache

Description automatically generatedA person in a tuxedo

Description automatically generated

**Face recognized! DTW Distance: 9.19398039838107**

We took out the result of four different actors at threshold of 10 and did a comparison shown in table 1.

|  |  |  |
| --- | --- | --- |
| **Actor Name** | **DTW Distance** | **Result** |
| **Angelina Jolie** | 2.75 | Face Recognized |
| **Brat Pitt** | 9.19 | Face Recognized |
| **Jhonny Depp** | 4.98 | Face Recognized |
| **Drake** | 7.38 | Face Recognized |

1. **Conclusion**

In This project presents a Facial Landmarks technique that tries to utilize all of the profile's available information, stored the information in the model database. Then the testing image passed from the same landmark technique and matched with the facial landmark information already placed in the database using dynamic time wrapping algorithm. The DTW distance method is used in this project for matching facial profile portion.

We can see from the testing that our Facial Landmark technique shows promise. The approach can still function adaptably even for face profiles with significant expression fluctuation, where the quantity and placement of fiducial points are visibly varied. The facial point is somewhat susceptible to noise, which reduces performance, but it may be improved in the future by employing a more precise verification procedure.