CHAPTER 3

Pose-Based Boundary Energy Image

As already discussed in the Literature Survey chapter (i.e., Chapter 2), among the traditional appearance-based gait features, pose-based templates have become more popular due to their inherent capability of preserving the gait kinematics at a high resolution. Existing pose-based gait features are extensions of popular gait templates and these basically encode information from image frames containing the complete binary silhouette corresponding to the different fractions of a gait cycle. The complete binary silhouette has redundant areas of pixels and it appears that more discriminative gait features can be obtained by extracting features from the boundary (i.e., contour) of the silhouette, since pixels within a silhouette do not provide substantial gait information. In this chapter, we first propose a new gait template termed Boundary Energy image (BEI) by considering the silhouette boundary pixels only and next extend it to compute the corresponding Pose-based *BEI* feature. The main contributions of the work are as follows:

- Proposing a new basic gait template based on boundary silhouette information
- Extending the above work by developing a pose-based gait feature termed

Pose-based Boundary Energy Image by extracting features at the granularity of key poses.

- To the best of our knowledge, ours is the first pose-based gait recognition approach that is quite efficient and computes effective gait features from silhouette contours, which has been seen to outperform state-of-the-art Deep Learning-based gait recognition approaches as well.

3.1 Construction of the Boundary Energy Image

Figure 3.1 presents a block diagram of the overall gait recognition approach. With reference to the block diagram, first silhouette extraction, i.e., background subtraction is done for each frame followed by silhouette normalization in which the bounding box containing each silhouette is normalized to a fixed height and width. Subsequently, a complete gait cycle is detected and the Boundary Energy Image (BEI) feature is constructed by aggregating the silhouette contour information over this cycle. The dimension of the aggregated feature, thus obtained, is large and it is reduced by applying Principal Component Analysis (PCA). Finally, classification is carried out in the reduced sub-space to identify a subject as one among the gallery subjects. Each step in the block diagram is explained in detail in the following sub-sections.

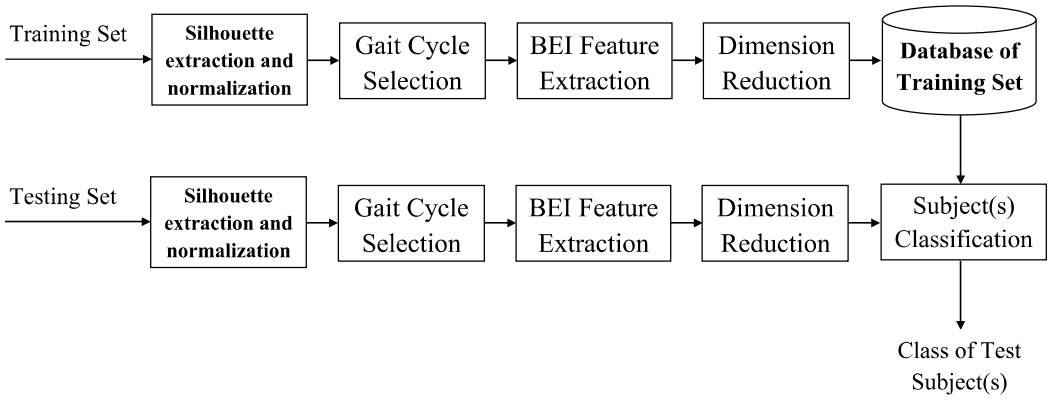

Figure 3.1: Block diagram for gait recognition using BEI feature

3.1.1 Gait Cycle Extraction

Given an input RGB sequence, we follow standard background subtraction and pre-processing techniques [1, 31, 57] to obtain a normalized silhouette corresponding to each frame. We compute the number of object pixels in each normalized frame and plot the sequence of numbers against the frame index as a continuous black curve in Figure 3.2. Although this curve reveals the periodic pattern of walking of an individual, it is not very smooth. Next, Loess algorithm [111] is applied to smooth the curve (shown as dashed line), and the frame indices at which peaks are observed in the smoothed curve are noted. For any given sequence, the double support pose and the mid-stance poses, in general, contain the maximum and minimum number of foreground pixels, respectively. Hence, in Fig. 3.2, the first peak (corresponding to *Frame 10*) refers to the first double support stance in a gait sequence, the second peak (corresponding to *Frame 22*) refers to the second double support stance that marks the end of a half cycle starting from *Frame 10*, and the third peak (corresponding to *Frame 35*) refers to the third double support stance, representing the end of a complete cycle starting from *Frame 10*. Since, human walking is symmetric in both the half cycles, gait features can be conveniently derived from either a complete cycle or only a half cycle. In all our experiments, we construct the gait features from sequences corresponding to half gait cycle only. From Figure 3.2, we can see that there are five peak points corresponding to the five double support poses of the sequence, and the four half-gait cycles corresponding to these peak points are shown in Figure 3.3.

3.1.2 Boundary Images and the Boundary Energy Image

Each normalized silhouette frame is a binary image where the object pixels are colored white and the background is colored black. We apply morphological dilation operation on each binary silhouette using a disk structuring element and next compute an absolute difference between the dilated image and the corresponding normalized binary silhouette to obtain the boundary image. Let there be N frames in a half gait cycle of a subject and B_i represent the boundary image corresponding to the frame F_i. The *Boundary Energy Image (BEI)* feature is computed by averaging the boundary images corresponding to a half gait cycle. If $B_i(x, y)$

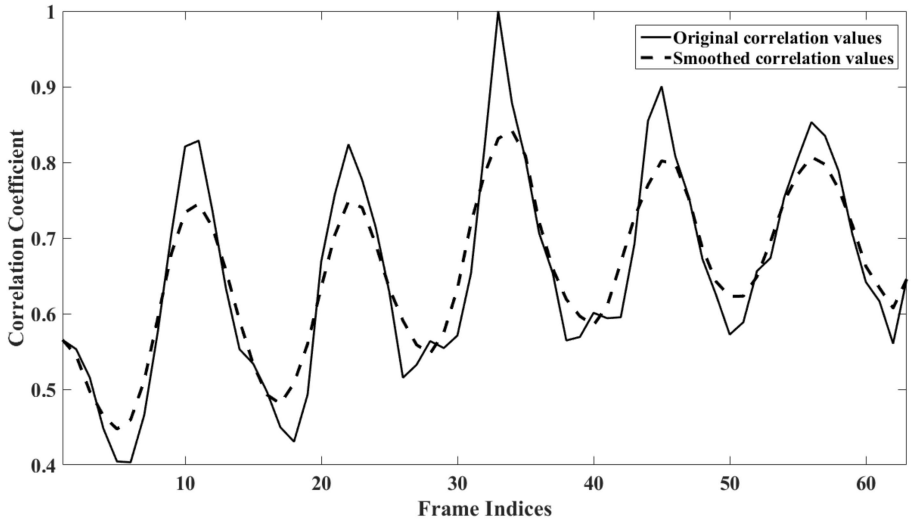

Figure 3.2: Plot of number of object pixels against frame index before and after smoothing

denotes the pixel intensity at pixel coordinate (x,y) in B_i, $i = 1, 2, ..., N$, then:

$$BEI(x, y) = \frac{1}{N} \sum_{i=1}^{N} B_i(x, y). \tag{3.1}$$

Figure 3.4 shows samples of boundary images corresponding to half gait cycles of walking sequences of two different subjects, and the rightmost image in each row is the corresponding *BEI* feature. Visual observation shows that *BEI* captures accurate silhouette shape and also encodes the shape variation over a gait cycle. Once *BEI* is computed, its dimension is reduced by applying Principal Component Analysis (PCA), and finally Diagonal Linear Discriminant Analysis (DLDA) is used for classification of an unknown subject into one of the gallery samples.

3.1.3 Experiments and Results for Evaluating the Effectiveness of the *BEI* Feature

This sub-section presents a thorough evaluation of the proposed *BEI* feature and a comparative study with other popular gait templates. We carry out all experi-

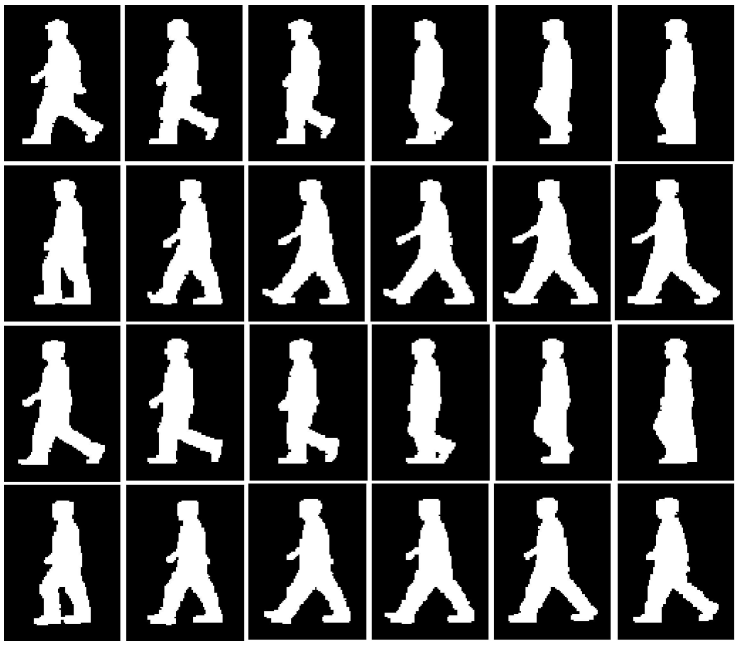

Figure 3.3: Silhouette frames after cycle selection

ments on a system having 64 GB RAM, one i9-18 core processor, along with three GPUs: one Titan Xp with 12 GB RAM, 12 GB frame-buffer memory and 256 MB BAR1 memory and, two GeForce GTX 1080 Ti with 11 GB RAM, 11 GB frame-buffer memory and 256 MB of BAR1 memory. Two different gait data sets have been used for our experiments, namely the CASIA B [2] and the TUMGAID [3] data and a brief description of each data set has already been provided in Section 2.4 of Chapter 2. For experimenting with each of the TUMGAID and the CASIA B data, we consider three different combinations of four normal walking sequences from all subjects to form the training set and the remaining sequences are considered as the test set.

As already explained in Section 3.1.2, to compute the boundary image we apply morphological dilation operation on each normalized binary silhouette frames and next compute an absolute difference between the dilated image and the original

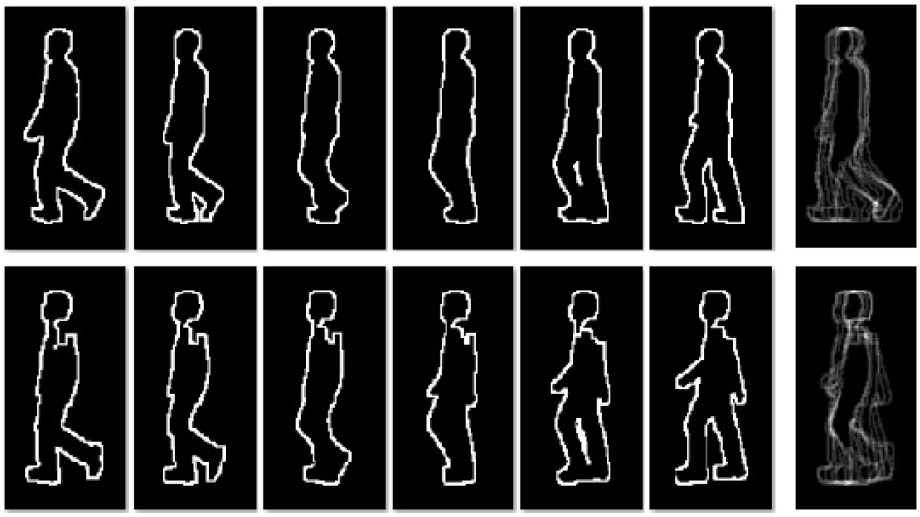

Figure 3.4: Boundary images corresponding to half gait cycles of two different subjects and the corresponding BEI feature

Table 3.1: Table showing the percentage accuracy of our method on the different test sets in CASIA B data for different sizes of the structuring element

Scenario	r=1	r=2	r=3	r=4	r=5	r=6	r=7	r=8	r=9	r=10	r=11
C_1	98.79	98.79	99.59	100.0	99.59	99.59	99.59	99.59	99.59	99.59	99.59
C_4	85.88	84.27	83.87	81.04	79.43	75.80	72.98	70.56	70.16	68.54	66.53
C_5	42.74	45.56	46.77	44.35	45.96	43.54	41.53	42.33	40.72	39.91	39.91
Average	75.80	76.20	76.74	75.13	74.99	72.97	71.36	70.29	70.15	69.34	68.67

image. A disk structuring element has been used for this dilation operation. We first study the *BEI*-based gait recognition accuracy for three different training-test combinations corresponding to each data set, namely C_1, C_4, C_5 for the CASIA B data and T_1, T_4, T_5 for the TUM-GAID data (as shown in Table 2.1), for varying values of the radius (r) of the structuring element. Corresponding results are presented in Tables 3.1 and 3.2 for the CASIA B and the TUMGAID data sets for values of r ranging from 1 to 11. From the tables, we observe that the accuracy values for the different values of r are quite high for scenarios C_1 in case of the CASIA-B data and scenario T_1 in case of the TUMGAID data. These are the scenarios for which both the training and test sequences have similar covariate conditions causing the appearance of the *BEI* features extracted from the two sequences to be closely similar to each other, thereby resulting in a high recognition

Table 3.2: Table showing the percentage accuracy of our method on the different test sets in TUMGAID data for different sizes of the structuring element

Scenario	r=1	r=2	r=3	r=4	r=5	r=6	r=7	r=8	r=9	r=10	r=11
T_1	91.47	92.62	93.93	94.26	94.75	95.08	95.73	95.73	96.22	96.22	96.22
T_4	73.27	71.63	70.52	65.24	52.45	42.13	35.08	33.60	26.06	22.13	22.45
T_5	79.50	80.81	81.47	82.45	82.95	84.75	84.91	85.24	86.22	86.89	87.04
Average	81.42	81.69	81.97	80.65	76.72	73.98	71.91	71.53	69.50	68.41	68.57

accuracy. A relatively lesser accuracy is observed for scenarios C_4 and C_5 in case of the CASIA B data and scenarios T_4 and T_5 in case of the TUMGAID data, i.e., when the covariate conditions are varied during the training and test data capture. Out of these four scenarios, a substantial decrement in the recognition accuracy is observed for each of the r values corresponding to scenario C_5. This is since for this particular scenario the training set consists of normal walking sequences and the test set consists of sequences with people wearing coats and the *BEI* features generated from these two sequences have significant appearance dissimilarity. This difference is comparatively much lesser for each of the other scenarios shown in Table 2.1.

In general, it is seen from the tables $r = 3$ provides the best recognition accuracy for each scenario, and at this value of r, the accuracy given by the *BEI* feature is not much affected even if the subjects have different co-variate conditions during the training and testing phases (refer to scenarios C_4 and C_5 in Table 3.1, and T_4 and T_5 in Table 3.2).

Table 3.3: Table showing comparative performance analysis of our approach with different number of half-cycles on CASIA B data

Probe frame Name	C_1	C_4	C_5	Average
One Half-Cycle	91.53	77.82	41.12	70.29
Two Half-Cycles	97.58	81.04	43.54	74.05
Three Half-Cycles	98.38	83.28	44.53	75.39
Four Half-Cycles	98.38	84.27	47.58	76.74

Each of the following experiments also considers the radius of the structuring element as three. Next, we study the improvement of the *BEI*-based recognition

accuracy with increase in the volume of the training data for each of the different training-test scenarios specified for the CASIA B data in Table 2.1. Basically, to increase the volume of the training data, we extract more half-gait cycles from each gallery sequence and henceforth construct a bigger database of gallery features for each subject. Table 3.3 reports the recognition accuracy values for the above experiment for the following different training-test scenarios corresponding to the CASIA B data: C_1, C_4 and C_5 (refer to Table 2.1. For each test sequence, one half-cycle of gait is considered. As expected, it is observed from the table that the recognition accuracy for each of the different scenarios usually increases with increment in the training data. But the accuracy values for scenario C_5 is, in general, lower compared to both C_1 and C_4, which is also because of significant appearance differences between the corresponding training and test sequences for scenario C_5.

3.2 Pose-based Boundary Energy Image

From the previous set of experiments, the *BEI* template has been found to be quite effective for gait recognition and can be used as an effective means for gait recognition. However, it has been seen in past studies [1, 57] that pose-based features, i.e., those that compute features at fractional parts of a gait cycle, are capable of retaining a higher dynamic content that simple aggregated features. Hence, it appears that extension of our previous work on *BEI*-based recognition to pose-based *BEI* features would result in a more improved gait recognition performance, which we consider studying next. The individual steps for gait recognition using the *pose-based BEI* features are explained using the block diagram shown in Fig. 3.5. With reference to the block diagram, given a set of RGB gallery gait sequences, first preprocessing steps are done in a manner similar to that done while constructing the *BEI* feature (refer to Section 3.1). The normalized silhouette sequences corresponding to the gallery sequences are used to construct a database of key poses and also the *pose-based BEI* features for each key pose. Given a test sequence, similar preprocessing steps are carried out followed by mapping of normalized silhouette frames to the appropriate key poses. Next, *pose-based BEI* features are constructed for the different key poses and a final comparison between

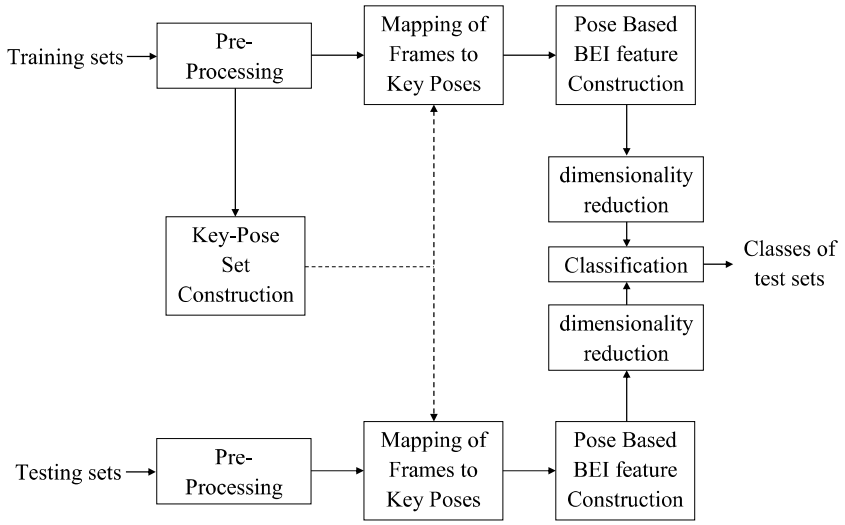

Figure 3.5: Block diagram for Pose based Boundary Energy Images (BEI) from feature construction to test sets classification

each pair of gallery and test sequences is done in the *PCA*-transformed space using an *LDA* classifier.

3.2.1 Key Pose Set Construction and Mapping of Frames to the Key Poses

Given the set of normalized gallery gait sequences, we align the sequences such that each of these start with a similar walking pose and consider half of a gait cycle from each sequence. Next, constrained K-Means algorithm is applied to determine the optimal key poses from the above half cycles using an algorithm similar to that given in [1]. The two constraints involved in the constrained K-Means algorithm are as follows:

- The only transitions allowed are from a cluster to its immediately preceding or immediately succeeding clusters.
- Following cluster assignment, the transition order of cluster indices is checked, and reassignment is done (if required) to ensure that each frame's cluster index is less than or equal to the succeeding frame's cluster index.

The optimal value for K in the constrained K-Means clustering, as determined through a rate distortion plot, using an algorithm similar to that in [1] is six. The six key poses thus obtained are the cluster centers obtained after convergence and these are shown in Fig. 3.6.

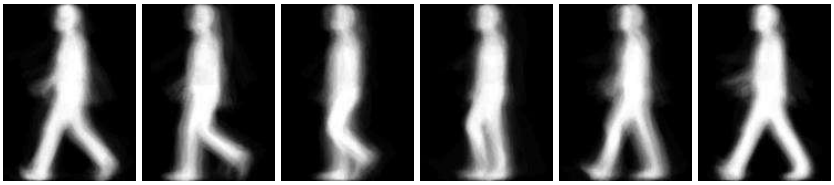

Figure 3.6: Six key poses in a gait cycle

Given an input sequence of normalized binary frames, we need to map the frames of the sequence to the appropriate key poses. It may be argued that simple Euclidean distance can be employed to classify each frame to the key pose with which it matches the most. However, such an approach is likely to ignore the temporal continuity present in the sequence of frames, causing two adjacent binary silhouette frames to get mapped to two vastly dissimilar key poses, which is not desirable. To mitigate this effect, the mapping of frames to the above key poses is done by making use of the temporal constraints imposed by a state transition model, instead of Euclidean distance only. Fig. 3.7 presents the state transition

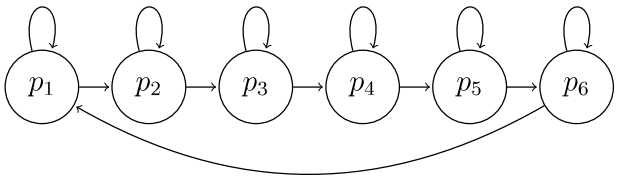

Figure 3.7: State transition diagram as [1] with five states and each state mapped with a key-pose

model used in this work with six states p_1, p_2, ..., p_6. With reference to the figure, each state in the figure corresponds to a key pose and from each of the states p_1, p_2, ... p_5, transitions are allowed to the same state or to the immediately next state. From the final state p_K (i.e., p_6), transitions are allowed to the same state

or to the first state p_1. The transitions are shown as directed edges in the figure. For example, if a frame F_i gets mapped to key pose p_k, then the immediately next frame, i.e., frame F_{i+1} would get mapped to key pose p_{k+1} if $k < K$, or to the first key pose p_1 if $k = K$. This mapping process helps to retain the temporal continuity in the assigned key pose labels of the adjacent binary silhouette frames which is practically not possible using Euclidean distance alone due to the presence of noise in the binary silhouette images. in other words, the above state transition model always constrains a pair of consecutive silhouette frames to get mapped to either the same key pose or consecutive key poses (in a cyclical sense), thereby preserving the temporal ordering of walking.

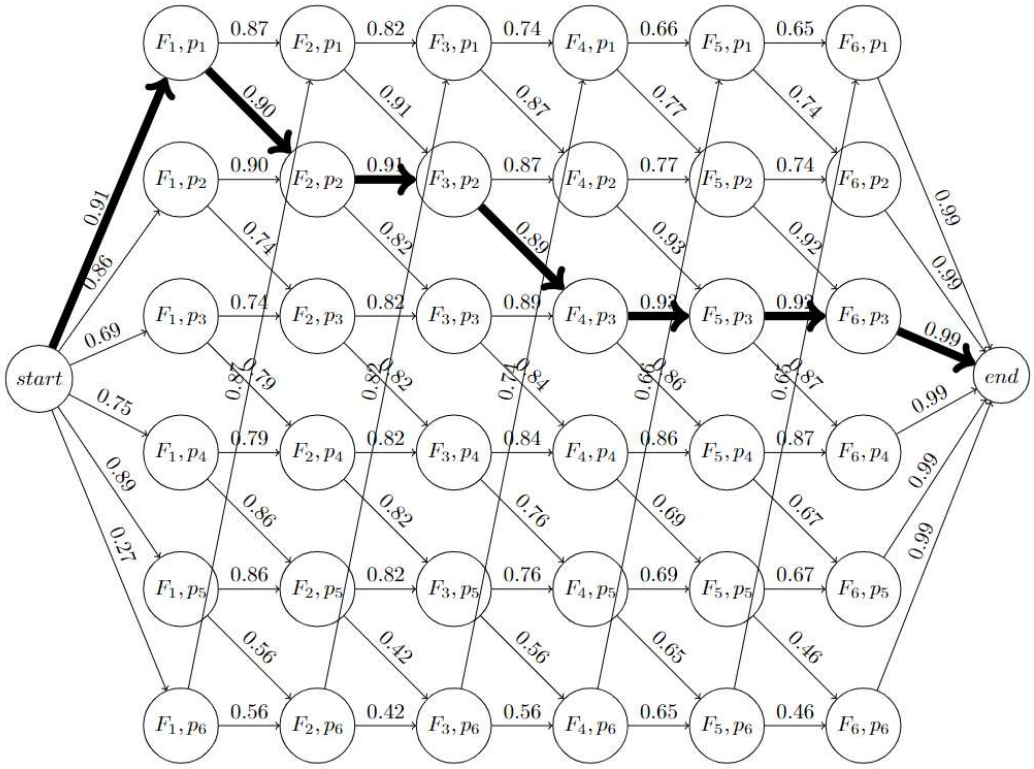

Figure 3.8: Directed acyclic graph following the constraints state transition diagram of Fig. 3.7 considering five key poses p_1, p_2, ..., p_5

The problem of mapping frames to key poses via state transition model is mapped to the problem of finding the longest path in a weighted Directed Acyclic Graph

(DAG), as explained next. Let F_1, F_2, ..., F_N denote the N frames present in an input gait sequence, and p_1, p_2, ..., p_K denote the set of K key poses. In the DAG, a node is present corresponding to each pair of frame and key pose, resulting in a total of $N \times K$ nodes. Let us represent each node as a tuple $\{F_i,p_k\}$, where i= 1,2,.., N and k = 1,2,...,K. A directed edge is added from $\{F_i,p_{k_1}\}$ to $\{F_{i+1},p_{k_2}\}$ only if there exists an allowable transition from state p_{k_1} to state p_{k_2} ($1 \leq k_1,k_2 \leq K+1$), as per the state transition model in Fig. 3.7. The weight associated with any incoming edge to node $\{F_i,p_k\}$ is the correlation score (M_{ik}) between the encoded representations of F_i and p_k, and the edges in the graph are unweighted. Since a single frame of a sequence cannot get mapped into multiple key pose states, there does not exist any edge in the DAG from vertex $\{F_i,p_k\}$ to any of $\{F_i,p_j\}$, where j=1,2,...,K. Fig. 3.8 shows a sample DAG for mapping six input binary silhouette frames to the six key poses derived previously. The nodes labeled as '*start*' and '*end*' are two dummy nodes added to the graph with the aim of finding the longest path from the '*start*' node to the '*end*' node. All the incoming edges to the '*end*' node can be assigned any similar value and here we have set this value to 0.99. Finally, the longest path in the DAG is computed using Dijkstra's algorithm which is shown with bold lines. From the given example, the following mapping is established: frame F_1 gets mapped to key pose p_1, F_2 gets mapped to key pose p_2, F_3 gets mapped to key pose p_2, F_4 gets mapped to key pose p_3, F_5 gets mapped to key pose p_3, and F_6 gets mapped to key pose p_3. Fig. 3.9 shows a normalized binary silhouette sequence and the key poses to which each frame of the sequence got mapped using the above algorithm and the boundary images of the mapped silhouette frames are shown in Fig. 3.10.

3.2.2 Feature Construction

Suppose, the state transition model maps t consecutive frames among the N input frames, namely, frames F_{i_1}, F_{i_2}, ...,F_{i_t} to key pose p_k. To compute the Boundary Energy Image (BEI_k) for key pose p_k, we aggregate the corresponding boundary

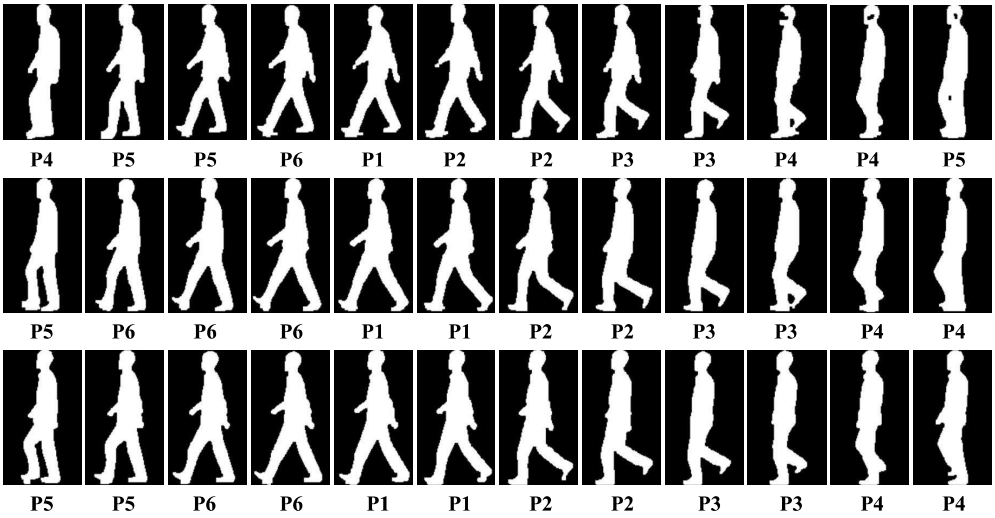

Figure 3.9: A normalized binary sequence and the key poses to which each frame of the sequence got mapped to

images B_{i_1}, B_{i_2}, ...,B_{i_t} through averaging. Mathematically,

$$BEI_k = \frac{1}{t}\sum_{j=1}^{t} B_{i_j}. \tag{3.2}$$

The final pose-based *BEI* features of a subject is a collection of the above features for the K different key poses. If P_{BEI} denote the Pose-based *BEI* feature, then P_{BEI} =$\{BEI_1\ BEI_2\ ...\ BEI_K\}$. The pose-based *BEI* features corresponding to the boundary image sequence shown in Fig. 3.10 are shown in Fig. 3.11.

3.2.3 Recognition Using the Pose-Based BEI Feature

Once pose-based *BEI* is constructed, it is used in the subsequent comparison phase. Consider a gallery of M subjects, and further let the feature computed for these gallery subjects be represented by P^1_{BEI}, P^2_{BEI}, ..., P^M_{BEI}, respectively. Since the dimension of the pose-based *BEI* feature is high, we apply PCA to reduce the feature dimension by retaining 95% variance. Let these reduced features for the M subjects be denoted by $\mathcal{P}^1_{BEI}$, $\mathcal{P}^2_{BEI}$, ..., $\mathcal{P}^M_{BEI}$, respectively. Next, consider an input test sequence, and the PCA-reduced pose-based BEI feature obtained from

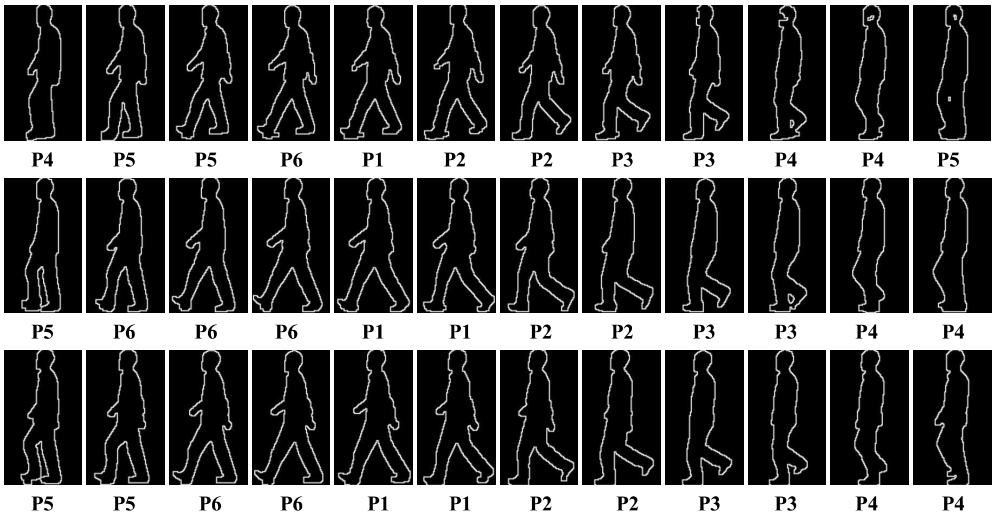

Figure 3.10: A normalized boundary image frames of sequence present in Fig. 3.9

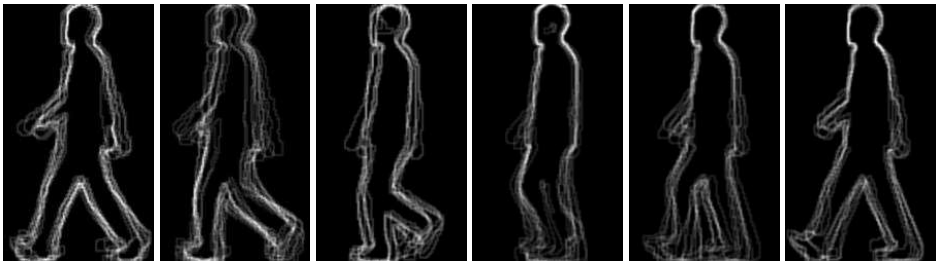

Figure 3.11: Pose-based *BEI* with six key poses

this sequence be given by $\mathcal{P}^T_{BEI}$. The feature $\mathcal{P}^T_{BEI}$ is compared with each of the features derived from the M subjects, i.e., with $\mathcal{P}^1_{BEI}$, $\mathcal{P}^2_{BEI}$, ..., $\mathcal{P}^M_{BEI}$, and next the class of the input sequence is predicted using a Linear Discriminant Analysis (LDA) classifier.

We carry out classification of the pose-based *BEI* feature using an LDA classifier corresponding to each pose in the set of maximal unique walking poses, and next aggregate the responses given by the different classifiers to predict the final class of a test subject. Let $\mathcal{P}^m_j$ denote the probability that the feature of the test subject for the j^{th} key pose is mapped to the m^{th} subject. Similarly, we compute the probabilities for all the poses into which the frames of the test sequence got mapped, and next average these probabilities to obtain a metric $\mathcal{P}^m$ representing

the similarity of the test subject with respect to the m^{th} gallery subject. The test subject is assigned to class $\mathcal{C}$ if

$$P^{\mathcal{C}} > P^{m} \; \forall m = 1, 2, ..., M, m \neq \mathcal{C}. \tag{3.3}$$

3.3 Experiments and Results

Experimental evaluation of the proposed pose-based *BEI* feature has been carried out on the same platform, as mentioned in Section 3.1.3. Tables 3.4 and 3.5 show the accuracy obtained using the proposed pose-based *BEI* feature for the CASIA B and the TUM-GAID data, respectively. In the same tables, we also make a comparative study with several existing non-Deep Learning and Deep Learning-based gait recognition techniques. Among the non-Deep Learning-based approaches, we consider the popular *GEI* [27], *AEI* [20], *DTW* [69], *PEI* [1], and the contour-based feature discussed in Section 3.1, i.e., *BEI*, whereas among the Deep Learning-based techniques we consider that using *Deep CNN* [88], *GEINet* [91], *MGANs* [50], *GaitSet* [94], and *GaitGAN* [95].

The training-test scenarios C_1, C_4, C_5 for the CASIA B data and T_1, T_4, T_5 for the TUM-GAID data, as discussed in Table 2.1, correspond to the second, third, and fourth columns in both the tables. The first column represents the gait feature, while the fifth and sixth columns show the mean accuracy obtained from the various scenarios and the classification time (i.e., the time required to compare a test feature with the entire gallery set) for each gait feature. In Tables 3.4 and 3.5, the best accuracy values obtained for each of the scenarios C_1, C_4, C_5 in case of the CASIA B data, and T_1, T_4, T_5 in case of the TUM-GAID data (as shown in Table 2.1) have been shown in bold fonts. It is seen that generally each of the existing techniques performs quite well if the test sequence consists of normal walking and are devoid of co-variate factors such as carrying bag, wearing coat, etc. (refer to the first column of both the tables). If the training and test conditions are made different by introducing varying co-variate conditions, the accuracy of each of these methods usually suffers (refer to the third and fourth columns of the two tables). A general observation from both the tables is that contour-based features like *AEI* [20] and our proposed *BEI* are

Table 3.4: Comparative performance analysis of the different gait recognition methods on CASIA B data [2] in terms of Rank 1 accuracy

Approach Name	C_1(%)	C_4(%)	C_5(%)	Mean (%)	Time (sec)
GEI [27]	**99.59**	65.44	37.39	67.47	0.42
AEI [20]	**99.59**	88.61	45.93	78.04	0.42
Deep-CNN [88] batch size 16	97.15	35.77	19.51	50.81	18.12
GEINet [91]	95.12	57.72	19.91	57.58	14.76
DTW [69]	94.71	56.50	27.64	59.61	3.21
MGANs [50]	**99.59**	91.00	48.00	79.53	1.52
GaitSet [94] LT	91.70	81.00	**70.10**	80.93	21.21
GaitGAN [95]	98.39	64.52	48.39	70.43	0.42
PEI with 6 poses [1]	**99.59**	78.04	43.90	73.84	1.83
PEI with 13 poses [1]	98.79	81.04	53.62	77.81	3.92
BEI	**99.59**	83.87	46.77	76.74	0.42
Pose-based BEI with 6 poses	**99.59**	**92.33**	57.25	**83.05**	1.72

capable of handling the differences in the co-variate conditions more effectively than those derived from complete silhouette structures like *GEI* [27], *DTW* [69], etc. Further, recent sophisticated Deep Learning-based approaches using *MGANs* [50], *GaitSet* [94], and *GaitGAN* [95] also have a good overall average accuracy. The proposed pose-based *BEI* feature performs with the best average accuracy of 83.05% for the CASIA B data (refer to Table 3.4) and with the second-best average accuracy of 81.23% for the TUM-GAID data (refer to Table 3.5), which is less than the best one given by *MGANs* by only 1.11%. However, the response time of MGANs is also higher than that of Pose-based *BEI* by about 2 seconds. From the tables, we observe that our proposed pose-based *BEI* feature performs consistently well for the varying training-test scenarios corresponding to both the data sets and also with a reasonably low response time.

Finally, we study the rank-based improvement in the average classification accuracy of the different methods used in the previous comparative study using the test sets of the CASIA B and the TUM-GAID data. Results through cumulative match characteristic (CMC) curves are shown in Figures 3.12 and 3.13 respectively for the two data sets. As expected, it is observed from both the plots that the classification accuracy of each of the gait recognition approaches increases with increment in the rank. Our method achieves above 85% accuracy mark within a

Table 3.5: Comparative performance analysis of the different gait recognition methods on TUMGAID data [3] in terms of Rank 1 accuracy

Approach Name	T_1(%)	T_4(%)	T_5(%)	Mean (%)	Time (sec)
GEI [27]	**97.03**	20.72	**87.82**	68.52	0.58
AEI [20]	91.14	**72.78**	79.01	80.97	0.58
Deep-CNN [88] batch size 16	90.78	29.60	60.36	60.24	22.10
GEINet [91]	90.29	25.49	78.61	64.79	11.33
DTW [69]	65.90	12.29	59.67	45.95	4.12
MGANs [50]	95.23	66.61	85.19	**82.34**	5.42
GaitSet [94] LT	86.18	58.55	76.15	73.62	32.43
GaitGAN [95]	94.57	54.27	83.22	77.35	0.58
PEI with 6 poses [1]	93.09	17.76	84.04	64.96	6.20
PEI with 13 poses [1]	65.08	47.04	50.65	40.92	12.13
BEI	91.31	66.39	76.22	77.97	0.58
Pose-based BEI with 6 poses	93.93	68.29	81.47	81.23	3.01

rank of 4 for the CASIA B data, and within a rank 3 for the TUM-GAID data, which is quite good. At higher ranks, the CMC curve for our proposed Pose-based *BEI* feature shows little improvement for the CASIA B data, whereas that by *BEI* and few other Deep Learning approaches like MGANs [50] and GaitGAN [95] show a continuous rise in the accuracy. This is mostly due to predetermining a fixed set of key poses before gait feature extraction and constraining consecutive frames of a gait sequence to get mapped to either the same key pose or successive key poses (refer to Section 3.2.1) which might not work effectively for gait sequences of all persons. This also results in deriving mismatched gait signatures for the same identity if the training and test sequences are captured at even slightly different frame rates or walking speeds.

From the above experiments, we conclude that deriving pose-based features using a pre-determined set of key poses will be more effective compared to the the corresponding aggregated features as long as the training and test sequences are captured under similar frame rate or waking speed. This is because in case of pose-based features, intrinsic characteristics of gait at different fractions of a gait cycle are obtained which is ignored while extracting aggregated features from a complete cycle. However, this approach is likely to suffer if the training and test conditions are made slightly different. In such cases, the frames from one of the sequences

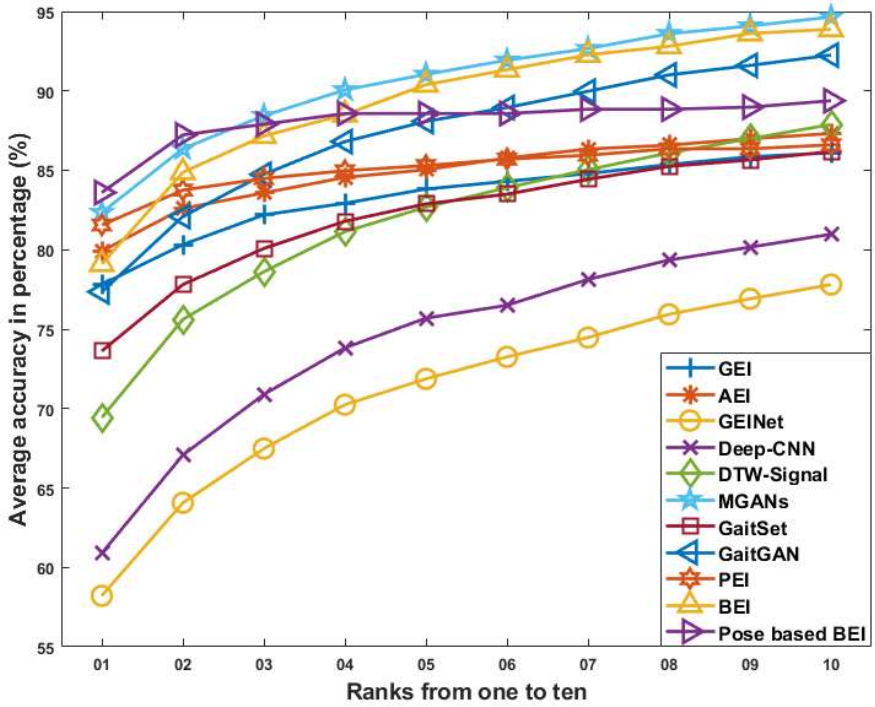

Figure 3.12: Rank based performance comparison between average accuracy of different gait recognition approaches on the CASIA B data-set

may not match exactly with any of the derived key poses, but the constraints imposed on the state transition model would forcefully match the input frames to the already derived key poses. The use of multiple key pose sets or relaxing of the constraints in the state transition model may help in mitigating the above problems to a certain extent which we will study in the following chapters.

3.4 Summary

The chapter introduces a pose-based gait feature derived from silhouette contours termed as the Pose-based *Boundary Energy Image*. Most traditional gait features like Gait Energy Image, Pose Energy Image and also recent Deep Learning models for gait recognition, namely, MGANs, GaitGAN, etc., use the complete silhouette

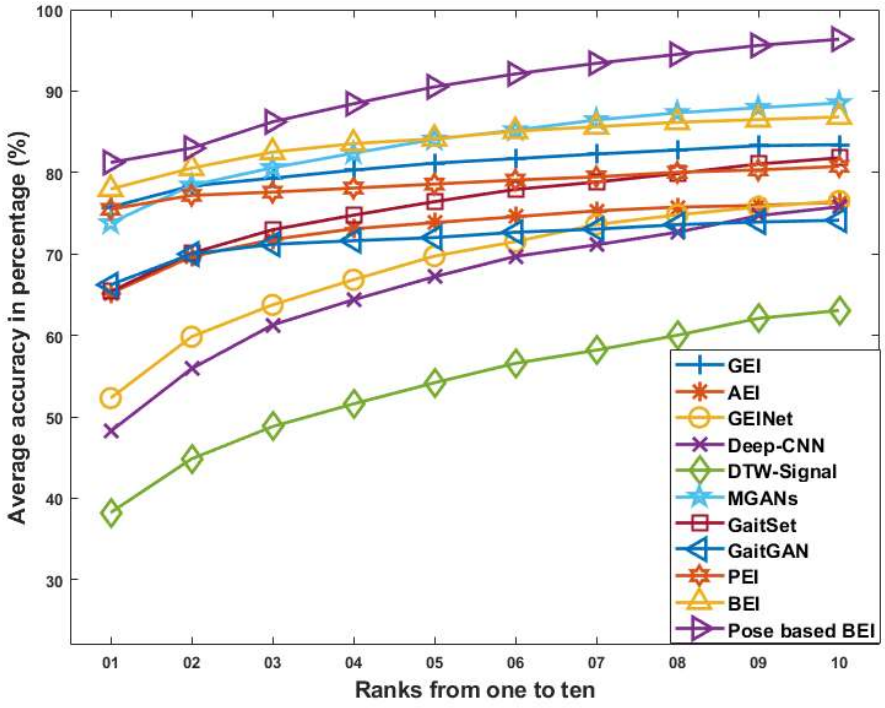

Figure 3.13: Rank based performance comparison between average accuracy of different gait recognition methods on the TUMGAID data set

structures over the frames of a gait cycle to derive the relevant features. Since binary silhouettes contain significant amount of redundant pixel information that is not useful for gait recognition, we propose the use of contour-based features that can capture the kinematics of gait more effectively. Existing contour-based features like Active Energy Image, Frame Difference Energy Image are constructed by computing the absolute differences between each pair of consecutive silhouettes in a sequence and aggregating this difference information. However, for slow walking speed or high frame rate, two consecutive frames may be closely similar to each other due to which useful kinematic information may not always be obtained from the difference images. Hence, we propose to extract the silhouette boundary pixels corresponding to each frame and generate the boundary image by applying standard morphological operations and average these boundary images to form a gait

template termed *BEI*. Next, we extend this work to form the Pose-based *BEI* by computing a set of key poses and constructing an aggregated feature corresponding to each key pose. Experimental results show that our proposed Pose-based *BEI* is time-efficient and performs robustly against varying co-variate conditions during training and testing. It also outperforms several other non-Deep Learning and Deep Learning-based gait recognition methods in terms of recognition accuracy. However, a limitation of this method is that it requires predetermining a set of key poses prior to gait feature construction due to which the method is ineffective in the presence of varying frame rates or walking speeds. A plausible solution to this problem has been addressed in the following chapter, i.e., Chapter 4 using a dictionary of key pose sets instead of a single key pose set.