You Can Run but not Hide: Improving Gait Recognition with Intrinsic Occlusion Type Awareness
Supplementary material

Ayush Gupta, Rama Chellappa
Johns Hopkins University
3400 North Charles Street, Baltimore, MD, 21218
{agupt120, rchella4}@jhu.edu

1. Introduction

In this supplementary material, we provide more details about the BRIAR dataset and its evaluation protocol. Next, we elaborate on the synthetic occlusions used in our experiments. We further train new models on dynamic occlusions and show that occlusion awareness can help even in dynamic occlusions. Next, we provide additional details and analysis on the Learnable 3D Conv method. Further, we perform experiments with different occlusion types, and restrict occlusion types in the gallery and probe set to analyse how difficult different occlusions types are. Lastly, we provide more details and experimental evaluation results regarding the occlusion detector.

2. BRIAR Dataset

The BRIAR [1] dataset is a recently collected dataset for gait recognition in outdoor, uncontrolled conditions. It has a lot of challenging outdoor scenes containing large variations in illumination, camera quality, distance of the subject from the camera, and extreme viewpoints. This makes it one of the most challenging gait recognition datasets.

The dataset contains videos captured systematically from distances of 100m, 200m, 400m and 500m. Additionally, some videos are captured from UAVs and some are captured at close range from an elevated viewpoint. Some video frames captured from UAVs are visualized in Figure 2. In BRIAR, the subjects move inside a fixed square boundary. The movement of the subjects may be 1) structured, where they walk along pre-defined straight lines inside the boundary, or 2) random, where subjects can move arbitrarily inside the boundary. While walking, the subjects are free to use their mobile phones and walk naturally, to represent a more practical scenario.

We use the BRS-1 and BRS-1.1 subsets of the BRIAR dataset for training, giving us a total of 212 training subjects. We use the BTS-1 subset for evaluation, containing 90 subjects. The BRS and BTS subsets are mutually exclusive, so no subjects used for training are used for evaluation and vice versa. The dataset defines the protocol for evaluation, containing the subject IDs, and start and end frame for each of the probe and gallery sequences.

Additionally, the videos captured from the 200m range are kept at a position which introduces jagged occlusions where the lower part of the subject is always occluded from view from tall grass of varying height. This makes recognizing the gait especially difficult, since the legs are partially hidden from view of the camera and the occlusion is also not consistent across the video. Some examples of these jagged occlusions have been shown in Figure 4 of the main paper.

2.1. Evaluation Protocol

The BRIAR dataset contains a variety of different conditions and distances. We take the non-occluded videos from the dataset and introduce synthetic occlusions in them for evaluation. The BRIAR protocol provides the probe and gallery split. A single video may have multiple probes within it, as specified by the start and end frames of each probe according to the protocol. It should be noted that none of the probes overlap with each other. The BRIAR dataset also contains single images as probes, but we filter them out because temporal information is required to run gait recognition models.

The controlled, indoor sequences in BRIAR are of higher quality and are treated as the gallery set. Meanwhile, the outdoor, more challenging conditions constitute the probe set. For evaluation, we use the Top-K rank retrieval metric. We compute the euclidean distance between each probe-gallery pair, and select the top K gallery matches for each probe. If the correct identity of the probe subject is within the top K predictions, the subject is regarded as being identified correctly. Since each subject also has multiple entries within the gallery, we select the top K gallery videos instead of the top K subjects. This list may have a subject being repeated, effectively reducing the number of possible candidates to choose from. Thus, this is a tougher evaluation metric than selecting the top K unique subjects while also being a more practical one to evaluate the model on.

3. Synthetic Occlusions

3.1. Consistent Occlusions

We use synthetic consistent occlusions to train the occlusion detector $\mathcal{D}$ as well as the gait recognition backbone $\mathcal{F}$ in most of our experiments. Consistent occlusions are one where all the frames have the occlusion patch at the same position, thereby blocking a body part from view for the entire length of the video. The consistent occlusion types we use in our experiments are described in Figure 1 and Section 3.2 of the main paper. The range R of these synthetic occlusions is set to be 20% - 50% of the frame size. The level of occlusion in a video is randomly chosen from this range.

3.2. Dynamic Occlusions

Dynamic occlusions are one where the position of the occlusion patch changes with time. We perform some additional experiments using such dynamic occlusions to check the generalizability of the occlusion aware model to unseen occlusion types, and to verify whether using the occlusion detector in the transient mode through the Learnable 3D Conv technique helps with dynamic occlusions.

To simulate dynamic occlusions, we place black patches of different shapes on the image frames, and the position of these patches can change with time. Specifically, we place either a small rectangular moving patch which occludes a portion of the subject, or a tall rectangular moving patch which covers the entire height of the frame. Some examples of these dynamic occlusions are shown in Figure 1. The height and width of the small patch are chosen randomly within the range $R_{ds} = (0.3, 0.5)$ which corresponds to 30% - 50% of the frame dimension. The height of the tall rectangular patch of occlusion is fixed to the height of the frame, and its width is chosen randomly within the range $R_{dt} = (0.2, 0.4)$ for each video. We decide these ranges of the occlusion patch size by manually visualizing the occluded video for different ranges, and choosing one which looks most similar to occlusion patterns which might be caused by objects like trees or poles covering the height of the frame, or small stationary objects like cones or boxes blocking a part of the moving subject.

To make the occlusions dynamic and realistic, we decide to give a velocity to the occlusion patches as opposed to randomly deciding the position across each frame. The direction of movement of the patch is decided randomly from left to right or right to left, and the velocity of these patches is chosen randomly from the range $R_v = (0.5, 1.0)$ pixels per frame. This range has also been chosen by manually inspecting the synthetically occluded videos with different velocities for the occlusion patch.

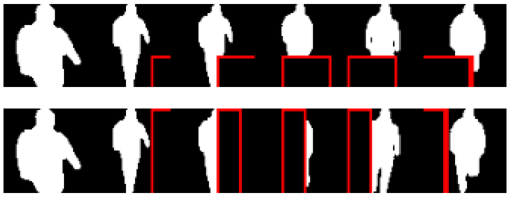

Figure 1. Examples of the synthetic dynamic occlusions we use, applied on video frames taken from the GREW dataset. The top row shows a small moving rectangular patch, and the bottom row shows a tall patch which covers the height of the frame applied on the same video. The occlusion patches are shown with a red boundary for visualization purposes only.

Method	Rank-1	Rank-5	Rank-10	Rank-20
Learnable 3D Conv	22.45	37.45	44.42	51.93
Deferred Concat	30.32	47.08	54.45	62.62

Table 1. Different occlusion awareness methods used for dynamic occlusions. Even for occlusions which change with time, inserting occlusion information in the deeper layers of the network performs better.

4. Occlusion Awareness in dynamic occlusions

Here, we train occlusion aware networks on dynamic occlusions. We experiment with Learnable 3D Conv and the Deferred Concat method to insert occlusion awareness in the GaitGL backbone. The results are summarized in Table 1. We observe that even in dynamic occlusions, the Deferred Concat method, where $\mathcal{D}$ operates in cumulative mode and outputs β_c, performs better than Learnable 3D Conv where $\mathcal{D}$ operates in transient mode and outputs β_t. This further demonstrates that the position where Learnable 3D Conv inserts occlusion aware features is not optimal for the gait recognition backbone, and occlusion awareness helps in the deeper layers of the network. It remains to be seen whether inserting transient occlusion features β_t into these deeper layers further improves performance on dynamic occlusions, and we leave that to future work.

5. Learnable 3D Conv

5.1. Additional details

The occlusion awareness module $\mathcal{M}$ takes as input the occlusion feature β and the intermediate feature X. It outputs a new occlusion aware intermediate feature $X^{'}$ which is replaced by X in the backbone. In most of the experiments, the size of $X^{'}$ is same as X, so that the architecture of the backbone remains unchanged. However, in Section

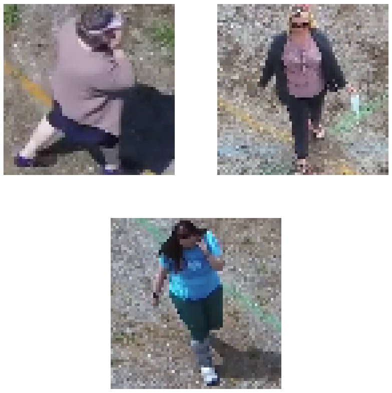

Figure 2. Some sample frames taken from the videos captured from UAVs in the BRIAR [1] dataset. We can see the extreme viewpoint angle in these videos, making recognition a more challenging problem from these.

5.2, we experiment with a larger size of the intermediate feature $X^{'}$.

The size of the occlusion feature β is $64 \times f$ in transient mode (f being the number of frames in the video), and 64×1 in cumulative mode. In the Learnable 3D Conv method, the transient occlusion feature β_t is repeated along height and width dimensions and concatenated with the intermediate feature X (of size $32 \times f \times h \times w$) along the channel dimension to give a feature size of $96 \times f \times h \times w$. The learnable 3D Conv layer reduces this again to $32 \times f \times h \times w$. However, the 3D Conv described in Section 5.2 transforms it into another block of $96 \times f \times h \times w$.

5.2. Increasing number of channels in 3D Conv

In this experiment, we try a larger size of the intermediate feature $X^{'}$ to see if a larger size of the occlusion feature benefits the model. Specifically, we use the Learnable 3D Conv method and increase the number of output channels in the 3D Conv to 96 from the earlier 32 channels. We use the GaitGL [3] backbone and train the model on the BRIAR [1] dataset. The results are mentioned in Table 2. We compare the model to the earlier Learnable 3D Conv and the Deferred Concat method, and observe that increasing the number of channels actually hurts the model. Thus, the occlusion information is able to fit better in the original number of channels of $X^{'}$ and introducing more channel confuses the model.

6. Occlusion Type Analysis

Evaluation by occlusion type: During evaluation, we randomly apply occlusions of different types on the input. In this section, we evaluate our model on these occlusion types separately to get an idea about which occlusion types are easier and which are difficult for the model. Our results are summarized in Table 3. As expected, the model is able to perform better when the size of the occlusion patch is small (corresponding to occlusions #1-#4). However, the task becomes much more difficult when half of the body is missing in occlusions #5-#8.

Different occlusions in gallery and probe: In our experiments, the occlusion type for each video is chosen at random, independent of other videos. As a consequence, the gallery and probe videos of a subject may have different or the same type of occlusion within them. In this section, we analyse the effect of enforcing the gallery and probe set to have different occlusions. As such, we apply occlusion types #1-#4 on the gallery set, and restrict occlusions on the probe set to #5-#6 and #7-#8 in separate experiments. Our results are shown in Table 4. Here as well, we observe that the model with occlusion awareness is able to perform better than the baselines.

7. Occlusion Detector

The occlusion detector $\mathcal{D}$ takes a video of silhouette masks as input and outputs the occlusion feature β. It is trained on silhouette images to classify the image into nine classes - eight types of occlusions or no occlusion. When working on videos, it outputs an occlusion feature for every frame and depending on its mode of operation, it can either output the entire block β_t or the mean-pooled feature β_c.

Optimal architecture: In our experiments, we use a three-layer convolutional neural network as the occlusion detector. In this section, we try out different depths of the CNN architecture to see which one would be the best for introducing occlusion awareness. We try out networks with 1, 3 and 5 convolutional layers, and they are able to achieve classification accuracies of 89.1%, 98.8% and 99.1% respectively. Even though the 5 layer network performs the best, the difference in performance is not much between the latter two variants. Thus, we choose the 3 layer network to introduce occlusion awareness considering the trade-off between computational cost and performance.

Implementation Details: The occlusion detector $\mathcal{D}$ we use is a three layer CNN with two additional linear layers. The ReLU activation function is used after each layer, except the last one where we use the softmax activation function while training. The occlusion detector is trained on the occlusion classification task using Cross Entropy Loss [4] and the Adam optimizer [2] with a learning rate of 0.001. We use a batch size of 32 for training the occlusion detector. The occlusion detector is trained on the BRIAR dataset, from which silhouette masks are extracted using Detectron2 [5].

Occlusion Awareness Method	100m		400m		500m		Extreme Angle		Aerial	
	Rank1	Rank20	Rank1	Rank20	Rank1	Rank20	Rank1	Rank20	Rank1	Rank20
Learnable 3D Conv more channels	22.27	77.94	9.48	62.36	9.77	57.96	14.25	66.79	12.7	71.43
Learnable 3D Conv	27.3	81.05	14.15	72.39	13.38	74.31	21.37	73.55	19.05	82.54
Deferred Concat	**34.58**	82.12	**21.15**	70.19	**18.47**	70.91	**25.73**	78.27	**28.57**	82.54

Table 2. Comparison of the performance of the occlusion aware network when the intermediate feature has more channels, compared to the regular Learnable 3D Conv method. The best deferred concat method is also shown for reference. We observe that increasing the number of channels actually hurts the performance, and the occlusion information better fits in the original number of channels inside the backbone.

Occlusion types	Rank-1	Rank-5	Rank-10	Rank-20
Corner patch (#1-#4)	20.92	35.12	42.13	48.82
Half Horizontal (#5-#6)	7.97	14.35	17.87	21.8
Half vertical (#7-#8)	11.67	21.37	27.9	36.35

Table 3. Evaluation of the occlusion aware GaitGL model on the GREW dataset, where synthetic occlusions are restricted to particular types during evaluation. #1-#4 correspond to an occlusion patch placed in any of the four corners of the frame, #5-#6 correspond to half horizontal occlusions where the top or bottom half of the body may be missing, and #7-#8 correspond to occlusions where either the left or the right half of the body may be missing. Half horizontal is the toughest occlusion type and corner patches are relatively the easiet occlusion types for the model.

7.1. Training and evaluation

During training, we sample one frame from every 50 frames in the video. During evaluation of the occlusion detector, we randomly pick one frame from each video. During both training and evaluation, synthetic occlusions of the previously discussed eight types are randomly introduced during the data loading step of the input frame and the classification accuracy is measured.

7.2. Architecture

The architecture of the occlusion detector is described in Table 5. It is a three-layer convolutional neural network followed by two linear heads. During training, the output of the FC2 layer is used to calculate the cross-entropy loss. However, during inference, and when it is being used along with the backbone $\mathcal{F}$, the FC2 layer is removed and the output of FC1 is used as the occlusion feature β.

8. Cross Domain Evaluation of Occlusion Detector

For our experiments with the gait recognition backbone $\mathcal{F}$, we use the weights of the occlusion detector obtained after training it on the relatively smaller BRIAR dataset. We use it directly on GREW without additional training to demonstrate its robustness across different domains. In this section, we further demonstrate its cross-domain generalization capability. We train and evaluate it on both BRIAR and GREW datasets, and also perform cross domain evaluation on the occlusion classifying task.

The results obtained are presented in Table 6. We observe that while the model performs best in-domain, the performance does not drop significantly across domains, thus demonstrating the robustness of the occlusion detector $\mathcal{D}$.

References

[1] David Cornett, Joel Brogan, Nell Barber, Deniz Aykac, Seth Baird, Nicholas Burchfield, Carl Dukes, Andrew Duncan, Regina Ferrell, Jim Goddard, Gavin Jager, Matthew Larson, Bart Murphy, Christi Johnson, Ian Shelley, Nisha Srinivas, Brandon Stockwell, Leanne Thompson, Matthew Yohe, Robert Zhang, Scott Dolvin, Hector J. Santos-Villalobos, and David S. Bolme. Expanding accurate person recognition to new altitudes and ranges: The briar dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 593–602, January 2023. 1, 3

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3

[3] Beibei Lin, Shunli Zhang, and Xin Yu. Gait Recognition via Effective Global-Local Feature Representation and Local Temporal Aggregation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14628–14636, Montreal, QC, Canada, Oct. 2021. IEEE. 3

[4] Shie Mannor, Dori Peleg, and Reuven Rubinstein. The cross entropy method for classification. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 561–568, New York, NY, USA, 2005. Association for Computing Machinery. 3

[5] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 3

Method	Rank-1		Rank-5		Rank-10		Rank-20	
	P #5-#6	P #7-#8	P #5-#6	P #7-#8	P #5-#6	P #7-#8	P #5-#6	P #7-#8
Baseline-1	0.23	1.27	0.6	3.23	1.18	4.8	2.33	7.03
Baseline-2	9.8	15.3	19.67	28.95	25.2	35.78	31.5	43.85
Occlusion Aware	11.4	18.22	21.68	33.62	27.88	40.7	35.05	47.8

Table 4. Performance of the baselines and occlusion aware model when gallery and probes have different occlusion types. Here, gallery occlusion is chosen between #1-#4 and probe occlusion is chosen from either #5-#6 or #7-#8 as specified.

Layer Name	**Input shape**	**Output Shape**
Conv1	64 * 64 * 1	64 * 64 * 32
ReLU, MaxPool1	64 * 64 * 32	32 * 32 * 32
Conv2	32 * 32 * 32	32 * 32 * 64
ReLU, MaxPool2	32 * 32 * 64	16 * 16 * 64
Conv3	16 * 16 * 64	16 * 16 * 128
ReLU, MaxPool3	16 * 16 * 128	8 * 8 * 128
AdaptiveAvgPool	8 * 8 * 128	128
FC1	128	64
FC2	64	9

Table 5. The architecture of the occlusion detector. It is a three layer convolutional neural network followed by two fully connected layers.

	Test on BRIAR	Test on GREW
Train on BRIAR	**98.0**	94.9
Train on GREW	97.9	**98.8**

Table 6. In-domain and cross-domain evaluation of the occlusion detector on the BRIAR and GREW datasets. As expected, the performance is highest in the in-domain evaluation, but it does not drop significantly across domains. This demonstrates the robustness of the occlusion detector across domains.