Repetitive Activity Counting by Sight and Sound

Yunhua Zhang[1] Ling Shao[2] Cees G. M. Snoek[1]
[1]University of Amsterdam [2]Inception Institute of Artificial Intelligence

Overview of Appendix

The inference procedures of our model are first illustrated in Appendix A. Then we present additional details of our Countix-AV and Extreme Countix-AV datasets in Appendix B and Appendix C. The effect of hyperparameter θ_s and margin m in our temporal stride decision module is evaluated in Appendix E, as well as the comparison with fixed temporal strides. For the reliability estimation module, we also study the effect of two hyperparameters θ_r^v and θ_r^a and compare with two alternatives in Appendix F. In Appendix G, we analyze the performance of training our model with action class supervision and we demonstrate the implementation details of our sight model on the UCFRep [42] dataset in Appendix H. We finally explain the videos provided in the supplementary material in Appendix I and Appendix J.

A. Inference Procedures

For each video, we first divide it into video clips with temporal strides of $\{1, 2, ..., S_K\}$ and their corresponding audio signals, which are sent into the networks simultaneously. In experiments, we find S_K=5 is enough for the used datasets, and it can be enlarged for situations where the action takes place slowly. Then, we choose the stride with the maximum score outputted by the temporal stride decision module to resample the video and preserve the estimated reliability score of the selected stride for later fusion. In the end, after obtaining the counting results from both streams, the final prediction of our model is computed by Eq. 10.

B. Countix-AV Dataset Statistics

The Countix-AV dataset is a subset of Countix [11] and the videos come from YouTube. It includes a total number of 19 classes for which the repetitive actions have a clear sound. The statistics per class are summarized in Table 6, including the number of videos per train, val and test fold, as well as the average count ground truth per class and fold. Some example videos are included in the supplementary material ("Learned repetition classes/Sound").

C. Extreme Countix-AV Dataset Details

The Extreme Countix-AV contains 214 videos in total, with 156 from Countix-AV and 58 from the VGGSound dataset [8]. We define 7 vision challenges to collect videos. First, we manually check every video and choose those that have camera viewpoint changes, disappearing activity and scale variation based on our visual observation. Then, for the cluttered background challenge, we also manually select the videos in which there are multiple persons appearing simultaneously while only one person is doing the repetitive actions or the object conducting repetitive activity is too small and hard to be distinguished (*e.g.*, some videos of bouncing ball). To collect videos captured in low illumination, we compute the average pixel intensity of each video, and add those with values below 100 (in the range of 0 and 255) into our dataset. For the fast motion challenge, we compute the average period length of each video according to the counting annotations, and find the videos with the average period length shorter than 3 frames. Finally, together the videos of those 7 challenges form our Extreme Countix-AV dataset.

D. Effect of L1 and L2 Loss Terms

In Eq. 6 and Eq. 7, both L1 and L2 terms are used in the loss functions of the sight and the sound streams to balance accuracy on small and large counts. To illustrate their effectiveness, we perform an ablation on different term combinations and report the sight-only model performance on

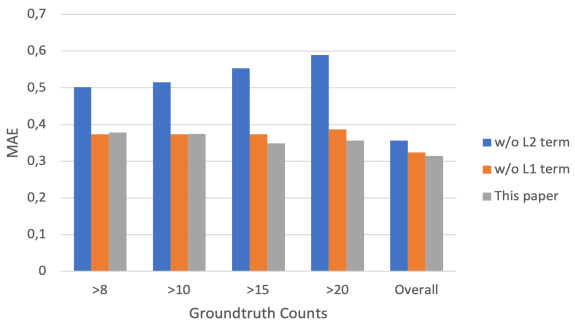

Figure 5: Performance on videos with more repetitions.

Action class	Number of videos				Average count groundtruth			
	Train	*Validation*	*Test*	*Total*	*Train*	*Validation*	*Test*	*Average*
battle rope training	57	17	41	115	14	10	6	11
bouncing ball (not juggling)	63	25	41	129	7	9	7	7
bouncing on trampoline	22	7	15	44	6	6	5	6
clapping	16	14	37	67	7	7	9	8
gymnastics tumbling	27	6	15	48	4	3	4	4
juggling soccer ball	65	23	9	97	11	11	9	11
jumping jacks	31	16	24	71	7	5	5	6
mountain climber (exercise)	37	12	22	71	10	9	10	10
planing wood	37	16	25	78	5	6	5	5
playing ping pong	79	25	34	138	3	3	3	3
playing tennis	42	11	24	77	3	3	3	3
running on treadmill	51	13	24	88	13	13	10	12
sawing wood	55	12	41	108	9	7	7	8
skipping rope	62	24	36	122	12	11	9	11
slicing onion	110	40	66	216	12	13	11	12
swimming	80	13	32	125	5	5	6	5
tapping pen	38	12	24	74	19	25	24	22
using a wrench	22	3	9	34	5	3	5	5
using a sledge hammer	93	22	43	158	5	5	5	5
Total	987	311	562	1860	-	-	-	-

Table 6: Countix-AV dataset statistics. Note our model does not use the action class labels.

Countix [11] towards videos of various groundtruth counts. As shown in Figure 5, for videos with many repetitions, results do not degrade due to the L1 loss, with 0.356 MAE for videos with more than 20 cycles compared to 0.387 (w/o L1 loss) and 0.553 (w/o L2 loss). However, for videos with few repetitions, results get worse without L1 loss as shown in Figure 6. The MAE for the sight stream increases from 0.217 to 0.348 on videos having only 2 repetitions. Therefore, the combination of L1 and L2 terms results in the best overall performance.

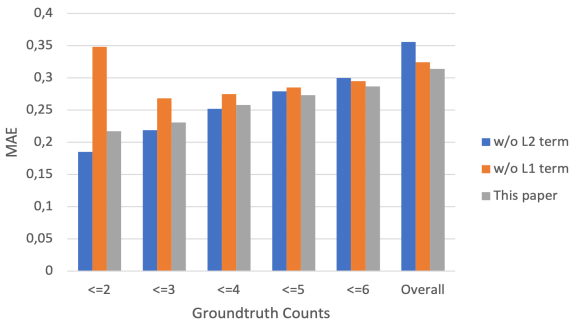

Figure 6: Performance on videos with a few repetitions.

E. Sight Stream Results

Here, we study the effect of hyperparameter θ_s and margin m in our temporal stride decision module. All the experiments are based on the sight stream with visual modality only and the original Countix [11] dataset. Note that despite the performance varies under different settings, all the results by our sight stream outperform the state-of-the-art by Dwibedi *et al.* [11] considerably.

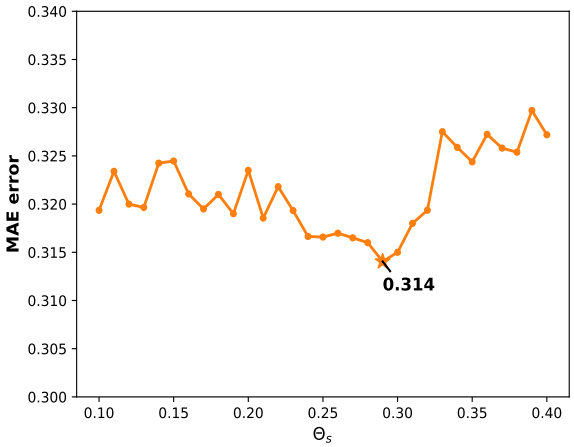

Figure 7: Effect of θ_s. The performance of the sight stream is not very sensitive to θ_s and the best result is achieved at θ_s=0.29.

Temporal stride	1	2	3	4	5	6
MAE ↓	0.607	0.378	0.387	0.427	0.467	0.475

Table 7: Effectiveness of temporal stride module. Using a fixed temporal stride results in inferior performance compared to 0.314 MAE by our temporal stride module.

Effect of θ_s. As illustrated in Section 3.3, θ_s is used to select the negative strides for training. With a higher θ_s, the chosen negative strides are larger and lead the sight stream to have more omissions. In contrast, a small θ_s makes the selection rule strict and thus results in over-fit issues. We study the effect of θ_s by setting it in the range of [0.1, 0.33), and the results are shown in Figure 7. We can conclude that the performance is not very sensitive to θ_s, and empirically θ_s=0.29 represents the best trade-off. We also observe that the average MAE error increases when $\theta_s \geq 0.33$, since the trained sight stream tends to select larger temporal strides and omit certain repetitions.

Effect of margin m. As detailed in Section 3.3, the max-margin ranking loss is adopted for training. In Figure 8, we show the performance of the sight stream when m varies from 1.0 to 4.0. We can see that the MAE error fluctuates between 0.314 and 0.330, so the value of margin m does not affect the results much.

Effectiveness of temporal stride module. We report sight-stream results for fixed temporal strides on Countix in Table 7. Our temporal stride decision module obtains a much better 0.314 MAE.

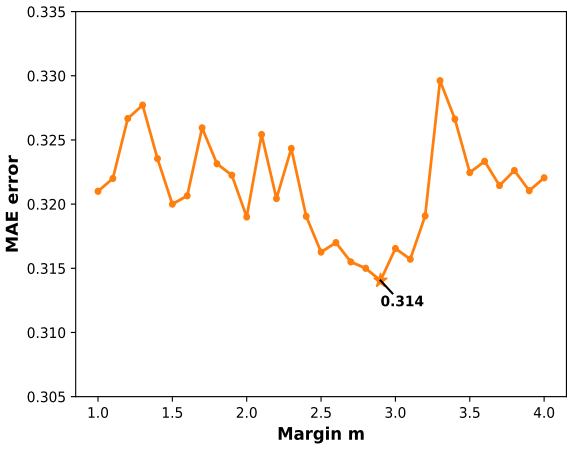

Figure 8: Effect of margin m for the max-margin ranking loss. The value of m does not influence the performance much and m=2.9 results in the lowest MAE error.

θ_r^v	θ_r^a	*MAE* ↓
0.365	0.420	0.294
0.365	0.400	0.291
0.365	0.390	0.292
0.365	0.380	0.294
0.360	0.420	0.292
0.360	**0.400**	**0.291**
0.360	0.390	0.292
0.360	0.380	0.295
0.355	0.420	0.294
0.355	0.400	0.293
0.355	0.390	0.293
0.355	0.380	0.296
0.350	0.420	0.292
0.350	0.400	0.293
0.350	0.390	0.292
0.350	0.380	0.295

Table 8: Effect of thresholds θ_r^v and θ_r^a in the reliability estimation module. The performance of the full sight and sound model varies slightly with different thresholds, but is always better than simply averaging the predictions from both streams.

Model components	*MAE* ↓	*OBO* ↑
Predictions from the final models	0.297	0.436
Fully connected layer	0.301	0.421
Full sight and sound model	0.291	0.479

Table 9: Comparison with other fusion schemes on Countix-AV. Using empirical predictions is better than the predictions from final models and our fusion scheme by predicting reliability score performs superior to two additional fully connected layers for feature integration and counting prediction.

F. Reliability Estimation Module Results

Effect of θ_r^v and θ_r^a. In Section 3.4, we use two thresholds θ_r^v and θ_r^a to collect predictions from both modalities for training the reliability estimation module. To study the effect of these two thresholds, we set them to different values and the results are shown in Table 8. It is clear that the MAE fluctuates slightly between 0.291 and 0.296 under various settings, and the best performance is achieved when θ_r^v=0.360 and θ_r^a=0.400. In particular, the reliability estimation module is always superior to simply averaging predictions.

Comparison with other fusion methods. To illustrate the superiority of our proposed scheme, which uses empirical predictions for training, we compare our approach with two alternatives. One is to directly use the predictions from the final trained models over the training videos for learning.

The other is similar to the fusion method described in [7] that trains two additional fully connected layers working in tandem built upon the penultimate layers of both sight and sound streams, which take the concatenated features from both modalities as inputs. As our original counting model, one fully connected layer outputs the repetition classification results and the other predicts the counting result of each class. The loss function is the same as Eq. 6 and Eq. 7 with the same hyperparameters but P is set to 41 for the best result.

The results are shown in Table 9. We observe empirical predictions perform better than directly adopting predictions from the final models, and our reliability estimation module outputting the reliability score outperforms the counterpart that uses fully connected layers for feature fusion as well as counting prediction. Therefore, our proposed reliability estimation scheme effectively integrates information from both modalities for more accurate counting prediction.

G. Counting with Action Class Supervision

To verify whether the action class labels can improve the counting accuracy further, we replace the L_{div} in Eq. 2 and Eq. 3 with a cross-entropy loss using action class labels for supervision to train the repetition classification branch. The results are shown in Table 10, while the performances of our original models can be found in Table 5. We observe that action class supervision can only improve the counting accuracy of the sight stream by a small margin, while degrade the performance of the sound stream and the full sight and sound model. The results demonstrate that repetition classes cannot be simply regarded as action classes. Although action class supervision can guide the network to count the correct repetitive movements inside each video, each action class may contain various repetition classes (*i.e*. repetitive motions) in different videos, which should not be counted in the same way. For instance, for the sight stream, videos of doing aerobics contain different movements that needed to be counted. In contrast, the arm shows similar motions in some videos belong to action classes of playing table tennis and playing tennis. Similar phenomenon can also be found in the sound stream. On one hand, some videos of "Slicing onion" and "Tapping pen" contain similar sound patterns and thus can be counted in the same way. On the other hand, the sound stream needs to focus on various tones in different videos that belong to the action class "skipping rope". In some videos, it is easy and reliable to count the repetitions by hearing how many times the feet of the person touch the ground. However, in some other videos, only the sound of rope is clear and usable. We provide example videos of learned repetition classes in the folder "Learned Repetition Classes" of the supplement for both sight and sound streams with illustration in Appendix I. Therefore, we can conclude that for temporal repetition counting, our automatically learned repetition classification layer is superior to its counterpart that uses action class supervision.

	Countix		Countix-AV	
	MAE ↓	*OBO* ↑	*MAE* ↑	*OBO* ↑
Sight	0.309	0.490	0.330	0.407
Sound	-	-	0.400	0.301
Sight & Sound	-	-	0.316	0.424

Table 10: Counting with action class supervision. Only the sight stream can benefit from the action class supervision marginally, while the performances of the sound stream and the full sight and sound model degrade. Therefore, action class supervision cannot effectively guide the learning of repetition counting models.

H. Implementation Details for UCFRep

Here, we illustrate the training details of our sight-only model on the UCFRep [42] dataset. Similar to the training on the Countix [11] dataset, we also initialize the weights of the model from a Kinetics pretrained checkpoint. Hyperparameters, like λ_v^1, λ_v^2, margin m, batch size, learning rate, etc, remain the same, as described in Section 4.3. The only difference is the number of repetition classes P, which is adjusted by greedy search, and we find P=24 works best.

I. Learned Repetition Classes

As our model learns to classify the input videos into different repetition classes automatically during training, here we visualize these learned classes by example videos in the supplementary material. The sight and sound streams are illustrated separately.

Learned repetition classes of the sight stream. In the folder "Learned Repetition Classes/Sight" of the supplementary material, we prepare 4 groups, named from "1.mp4" to "4.mp4", for the illustration of videos which have similar repetition class distributions from the repetition classification layer but belong to different action classes. Similar movements can be discovered in videos of each group despite there are significant variations in appearance. In "Doing aerobics.mp4", we can see that the video segments contain different repetitive motion patterns and thus they are classified into different repetition classes for counting by our sight model. However, in the field of action recognition, these segments belong to the same action class "doing aerobics".

Learned repetition classes of the sound stream. Similar to the sight stream, in the folder "Learned Repetition Classes/Sound" of the supplementary material, we prepare 4 groups named from "1.mp4" to "4.mp4", in which videos

of each group have similar class distributions by the repetition classification layer. It is clear that the audio tracks in each group have similar sound patterns but belong to various action classes. We also present some videos belong to the the same action class (*i.e*. skipping rope) but are treated as different repetition classes due to various types of sound in "skipping_rope.mp4".

J. Example Videos

In "demo_video.mp4" of the supplementary material, we show some example videos of our Extreme Countix-AV dataset with corresponding challenges as well as the predictions from the sight and the sound stream, our full sight and sound model and groundtruth.