Supplementary Material
N-ImageNet: Towards Robust, Fine-Grained Object Recognition with Event Cameras

Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim
Dept. of Electrical and Computer Engineering, Seoul National University, Korea
{82magnolia, wogur110, ssonpull519, 96lives, youngmin.kim}@snu.ac.kr

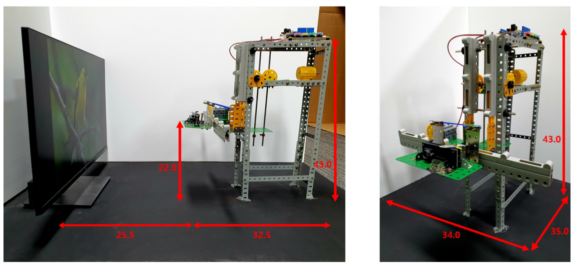

Figure A.1: Hardware setup for acquiring N-ImageNet. Note that the units are provided in centimeters.

A. Experimental Details

A.1. Hardware Setup

We report the dimensions of the hardware used for recording N-ImageNet data in Figure A.1. The custom hardware is utilized for generating N-ImageNet along with its variants, where the detailed generation process is described in Section 3.1. In addition, we provide a sample Arduino [2] script for generating arbitrary event camera motion in Figure A.2. Using the aforementioned setup, we make $50\mu s$ event camera recordings of each ImageNet [14] image.

A.2. Representation Implementation

In this section, we report the detailed implementations of event representations used in the paper.

A.2.1 Learning-based Representations

We first describe the details about learned representations, namely EST [7] and MatrixLSTM [4]. EST [7] was used in Section 4.1 and 4.2 for evaluating its performance on N-ImageNet and its variants. We adapt the implementation of Gehrig *et al.* [7] for implementing Event Spike Tensor (EST). We train EST with a batch size of 16.

MatrixLSTM [4] was tested in N-ImageNet and its variants, where the results are shown in Section 4.1 and 4.2. We use the implementation of the original paper [4] for MatrixLSTM. We train MatrixLSTM with a batch size of 16.

```
int MT_vertical_direction = 4;
int MT_vertical_magnitude = 5; //PWM
int MT_horizonal_direction = 6; //PWM
int MT_horizonal_magnitude = 7;
int BT_start = 15; //Blue Button
int BT_stop = 12; //Red Button

void setup() {
  pinMode(MT_vertical_direction, OUTPUT);
  pinMode(MT_vertical_magnitude, OUTPUT);
  pinMode(MT_horizonal_direction, OUTPUT);
  pinMode(MT_horizonal_magnitude, OUTPUT);
  pinMode(BT_start, INPUT);
  pinMode(BT_stop, INPUT);
  Serial.begin(9600);
}
void loop() {
  //initialize
  digitalWrite(MT_vertical_direction, LOW);
  analogWrite(MT_vertical_magnitude, 0);
  digitalWrite(MT_horizonal_direction, LOW);
  analogWrite(MT_horizonal_magnitude, 0);
  //start perpetual camera motion
  if (digitalRead(BT_start) == LOW) {
    while(digitalRead(BT_stop) == HIGH) {
      digitalWrite(MT_horizonal_direction, HIGH); //Right direction
      analogWrite(MT_horizonal_magnitude, 100);
      delay(50);
      digitalWrite(MT_vertical_direction, LOW); //Upward direction
      analogWrite(MT_vertical_magnitude, 200);
      delay(50);
      digitalWrite(MT_horizonal_direction, LOW);  //Left direction
      analogWrite(MT_horizonal_magnitude, 100);
      delay(50);
      digitalWrite(MT_vertical_direction,HIGH);  //Downward direction
      analogWrite(MT_vertical_magnitude, 0);
      delay(50);
    }
  }
}
```

Figure A.2: Arduino code for acquiring N-ImageNet.

A.2.2 Non Learning-based Representations

We report the details about representations that do not incorporate learning. For all our experiments conducted on N-ImageNet, we use a batch size of 256. We make further specifications on time surface [8], HATS [15], and DiST.

Dataset	Input Shape ($H \times W$)	# of Epochs	# of Train Data	# of Test Data
N-Cars [15]	128×128	12	15422	8607
CIFAR10-DVS [9]	128×128	23	8000	2000
ASL-DVS [3]	180×240	5	800	100000
N-Caltech101 [12]	240×304	30	7000	1709

Table A.1: Dataset statistics for fixed epoch training evaluation.

Dataset	Minimum Train Data	Maximum Train Data	Increment	# of Test Data
N-Cars [15]	1000	14000	1000	8607
CIFAR10-DVS [9]	500	8000	500	2000
ASL-DVS [3]	200	1000	200	2000
N-Caltech101 [12]	1000	7000	1000	1709

Table A.2: Dataset statistics for resource-constrained training evaluation.

For the time surface, we set the time constant τ of the exponential smoothing kernel to 0.3.

For HATS [15], we make a slight modification from the original paper to facilitate batch-wise parallelizable implementation. The original version of HATS utilizes memory cells that keep track of the recent events in a fixed time window. While this is suitable for asynchronous inference, it hinders large batch training, as sequential operations are present. Thus we opt to keep track of top k events for each pixel, which could be efficiently implemented using PyTorch Scatter [11]. In our experiments we set $k = 5$. Also, while HATS originally produces a low-resolution representation from neighborhood aggregation, we apply padding before aggregation to keep the resulting representation at high resolution. This lead to the enhanced performance on N-ImageNet shown in Table 4, 6.

For DiST, we set the discount factor from Equation 1 to $\alpha = 3$ and the neighborhood size from Equation 2 to $\rho = 5$. To efficiently implement the sorting operation, we utilize the `scatter_max` operation from Pytorch Scatter [11].

A.3. Pre-training Experiment Setup

We further provide details about the experiments for validating N-ImageNet pretraining, where the results are displayed in Section 4.1. Unlike the N-ImageNet validation experiment in Table 4, input representations are reshaped to fit the spatial resolution of the tested datasets. The input resolution for each dataset is shown in Table A.1.

Fixed Epoch Training We first train event-based object recognition algorithms with various initialization schemes (N-ImageNet pretraining, ImageNet pretraining, random initialization) on existing benchmarks. Details about the experimental setup are provided in Table A.1. Under the fixed number of train/test data, seven models from Table 4 are trained with a learning rate of 0.0003.

Factor	Trajectory		Brightness	
Change Amount	Small	Big	Small	Big
Validation Dataset Number	1, 2	3, 4, 5	7, 8	6, 9
Timestamp Image [13]	38.31	33.70	33.27	28.04
Timestamp Image$^{\mathbf{D}}$ [13]	35.86	32.37	30.67	26.41
Sorted Time Surface [1]	38.34	31.95	33.47	28.38
Sorted Time Surface$^{\mathbf{D}}$ [1]	35.97	32.28	30.73	26.27
DiST	**40.15**	**34.42**	**35.87**	**30.88**

Table B.1: Mean accuracy of models with explicit event denoising (superscripted $^{\mathbf{D}}$) measured on N-ImageNet variants.

Resource-constrained Training We further evaluate the different initialization schemes under resource-constrained settings. All networks are trained for 5 epochs on the datasets shown in Table A.2. We train each model with an increasing number of train data where the amount of increment is provided in Table A.2, starting from the minimum value until it reached the maximum. For example, in N-Caltech101, we trained models with training data sizes of 1000, 2000, . . . , 7000.

A.4. Clarification on Structural Similarity Index Measure (SSIM)

We used SSIM in Section 4.2 to evaluate the visual consistency of event representations amidst camera trajectory and brightness changes. SSIM between two windows x and y of size $N \times N$ is defined as follows,

$$\textbf{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (1)$$

where μ_x, μ_y are the mean value of the windows, σ_x, σ_y are the standard deviation of the windows, σ_{xy} is the covariance between the windows, and $c_1 = 6.5025, c_2 = 58.5225$. In all our experiments, we used SSIM with a window size of $N = 11$.

B. Additional Ablation on DiST

We perform an additional ablation study with the discounting operation of DiST by establishing comparisons against explicit denoising. Recall that the DiST serves as a robust event representation thanks to its noise suppression from discounting and speed invariance from sorting (Section 3.2).

We apply the density-based denoising scheme of Feng *et al.* [6] on the events from the N-ImageNet variants. Density-based denoising first voxelizes input events and further removes background activities by thresholding on the voxel-wise event densities. Hot pixels are eradicated by convolving the voxel representation with a hand-crafted filter.

The denoised events are given as input to the timestamp image [13] and sorted time surface [1], which are discount-ablated versions of DiST. The validation accuracies of these models are shown in Table B.1, where the denoised inputs do not lead to enhanced robustness. Such hand-crafted denoising algorithms have been effective for robust classification in existing datasets with a small number of classes [16, 17]. However, these methods often remove subtle visual details (e.g. texture), which are crucial cues for fine-grained object recognition.

The discount operation of DiST adaptively aggregates neighborhood evidence to suppress noise. DiST is capable of preserving visual subtleties, which is observable from the high SSIM values reported in Figure 7. Thus DiST is more suitable for robust, fine-grained object recognition than explicit denoising.

C. Full Robustness Evaluation Results

We report the full results on the N-ImageNet variants, as shown in Table C.1. Recall we have generated nine validation datasets for quantifiable robustness evaluation. DiST outperforms existing event representations in most N-ImageNet variants. Notably, DiST shows superior performance in all N-ImageNet variants with brightness change. This indicates that the discounting operation effectively suppresses noise frequently triggered from such environments.

We make further analysis on the effect of camera trajectory and scene illumination changes in object recognition accuracy. Table C.2, C.3 display the average accuracy of models from Table C.1 for each validation dataset. Both tables demonstrate that performance drop increases as the amount of change intensifies. For trajectory changes, a stark accuracy drop occurs from original to validation 5, validation 1 to validation 3, and validation 2 to validation 4. This indicates that given a fixed trajectory shape, performance deteriorates as more dynamic camera motion takes place. For brightness changes, a similar phenomenon is observable by comparing validation 6 with 7, and validation 8 with 9: accuracy drops rapidly as brightness change increases.

D. Visualization of DiST

We visualize DiST compared with the timestamp image [13] and sorted time surface [1] in Figure D.1. DiST is created by first sorting the timestamp image [13] and applying the discount mechanism stated in Equation 2. Notice that, unlike the other two representations, DiST not only suppresses background activities and hot pixels, but also demonstrates consistent representation in various conditions with high SSIM.

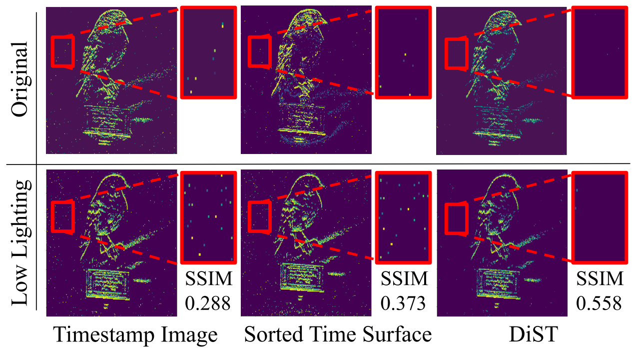

Figure D.1: Comparison of DiST against other representations in the original N-ImageNet dataset and Validation 6 variant (low lightning condition). DiST is able to suppress noise in both conditions.

References

[1] I. Alzugaray and M. Chli. Ace: An efficient asynchronous corner tracker for event cameras. In *2018 International Conference on 3D Vision (3DV)*, pages 653–661, 2018. 2, 3, 4

[2] M. Banzi. *Getting Started with Arduino*. Make Books - Imprint of: O'Reilly Media, Sebastopol, CA, ill edition, 2008. 1

[3] Y. Bi, A. Chadha, A. Abbas, , E. Bourtsoulatze, and Y. Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019. 2

[4] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision (ECCV)*, August 2020. 1, 4

[5] G. Cohen, S. Afshar, G. Orchard, J. Tapson, R. Benosman, and A. van Schaik. Spatial and temporal downsampling in event-based visual classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):5030–5044, 2018. 4

[6] Y. Feng, H. Lv, H. Liu, Y. Zhang, Y. Xiao, and C. Han. Event density based denoising method for dynamic vision sensor. *Applied Sciences*, 10(6), 2020. 2

[7] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5632–5642, 2019. 1, 4

[8] X. Lagorce, G. Orchard, F. Galluppi, B. Shi, and R. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions*

Change	None	Trajectory					Brightness				Average
Validation Dataset	Orig.	1	2	3	4	5	6	7	8	9	All
MatrixLSTM [4]	32.21	32.87	33.13	26.84	24.00	26.02	17.72	25.57	32.24	29.47	27.54
Event Spike Tensor [7]	**48.93**	**45.82**	28.12	32.78	21.77	**42.51**	16.56	21.79	28.00	28.16	29.50
Binary Event Image [5]	46.36	42.68	30.68	37.74	22.99	34.74	19.00	27.85	34.03	32.08	31.31
Event Histogram [10]	47.73	43.73	33.72	37.69	24.56	35.24	20.89	29.68	36.33	34.56	32.93
Event Image [16]	45.77	40.93	32.10	38.13	21.93	32.82	21.21	29.73	34.78	32.86	31.61
Time Surface [13]	44.32	41.01	34.63	40.00	25.48	34.89	22.12	31.27	37.12	35.36	33.54
HATS [15]	47.14	43.51	34.38	38.53	24.54	36.78	21.98	30.53	35.99	34.47	33.41
Timestamp Image [13]	45.86	43.01	33.62	39.37	25.39	36.23	21.16	30.02	36.52	34.92	33.37
Sorted Time Surface [1]	47.90	44.33	33.50	40.17	23.72	37.19	21.57	30.31	36.63	35.18	33.62
DiT	46.10	42.96	33.46	39.62	23.95	37.25	22.21	29.64	35.68	34.63	33.27
DiST	48.43	45.17	**36.58**	**42.28**	**26.57**	38.70	**24.39**	**32.76**	**38.99**	**37.37**	**35.89**

Table C.1: Full robustness evaluation results on N-ImageNet and its variants.

Dataset	Change Amount	Shape	Average Accuracy
Original	None	Square ↺	45.52
Validation 1	Small	Vertical	42.37
Validation 2	Small	Horizontal	33.08
Validation 3	Big	Vertical	37.57
Validation 4	Big	Horizontal	24.08
Validation 5	Big	Square ↺	35.67

Table C.2: Average accuracy of models evaluated on N-ImageNet and its trajectory variants. ↺ indicates counter-clockwise rotation.

Dataset	Change Amount	Relative Brightness	Average Accuracy
Original	None	Normal	45.52
Validation 6	Big	Dark	20.80
Validation 7	Small	Dark	29.01
Validation 8	Small	Bright	35.12
Validation 9	Big	Bright	33.55

Table C.3: Average accuracy of models evaluated on N-ImageNet and its brightness variants.

on pattern analysis and machine intelligence, 39, 07 2016. 1

[9] H. Li, H. Liu, X. Ji, G. Li, and L. Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017. 2

[10] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018. 4

[11] F. Matthias. Pytorch scatter, 2021. 2

[12] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015. 2

[13] P. K. J. Park, B. H. Cho, J. M. Park, K. Lee, H. Y. Kim, H. A. Kang, H. G. Lee, J. Woo, Y. Roh, W. J. Lee, C. Shin, Q. Wang, and H. Ryu. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1624–1628, 2016. 2, 3, 4

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[15] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman. HATS: Histograms of Averaged Time Surfaces for Robust Event-based Object Classification. *arXiv preprint arXiv:2018.00186*, June 2018. 1, 2, 4

[16] Y. Wang, B. Du, Y. Shen, K. Wu, G. Zhao, J. Sun, and H. Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6351–6360, 2019. 3, 4

[17] J. Wu, C. Ma, X. Yu, and G. Shi. Denoising of event-based sensors with spatial-temporal correlation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4437–4441, 2020. 3