Constructing and Evaluating Complex Event-based Datasets for Increasing Performance of Instance Segmentation Models

Alexandru-Dragoş Manolache
Supervisor(s): Nergis Tömen, Ombretta Strafforello, Xin Liu
EEMCS, Delft University of Technology, The Netherlands

22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Event-based cameras represent a new alternative to traditional frame based sensors, with advantages in lower output bandwidth, lower latency and higher dynamic range, thanks to their independent, asynchronous pixels. These advantages prompted the development of computer vision methods on event data in the last decade, however event-based datasets are still in early stages in terms of size and complexity compared to normal datasets (e.g. ImageNet). This paper explores event data augmentation by superimposing two existing event datasets (N-MNIST and N-Caltech101) and by adding uniform noise. It shows that training an instance segmentation model on noisy datasets does not improve its performance, but the amount and type of noise added in the background decreases the performance of such model. Code is available at: https://github.com/alexmanoo/dvs_datasets_transforms.

Keywords: Event Datasets, Data Augmentation, Event-based Vision, Dynamic Vision Sensor

1 Introduction

The retina, as well as other neurological and biological processes, serve as inspiration for event cameras. These cameras use a new type of sensor, different from regular Active-Pixel sensor (APS) cameras. A shutter on a typical APS captures light by exposing the entire light-sensitive surface to a specific amount of light for a specific amount of time. In the case of event-based sensors (also called Dynamic Vision Sensors - DVS), each time a light intensity change exceeds a predetermined threshold, the light-sensitive pixels individually turn on or off, outputting a stream of events (Figure 1). Thus, in comparison to APS, DVS pixels records these intensity events asynchronously and independently over time, resulting in reduced motion blur, increased temporal resolution, and a high dynamic range [1].

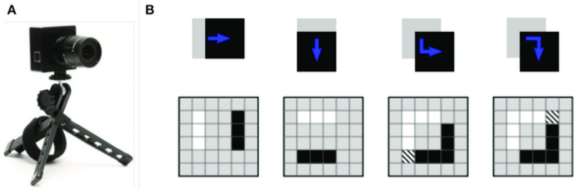


Figure 1: (A) A DVS camera, (B) The generated events by a moving simple square. Image taken from [2].

The data output from the new, dynamic vision sensors, differs in structure and content from APS, necessitating the development of new Machine Learning processing methods. There are a number of state-of-the-art deep learning methods (shown in Figure 2) that can achieve above 90% accuracy on image (that is, frame-based) datasets. However their input format is incompatible with event-based data. In recent years, work in the event-based field has progressed, and event (or neuromorphic) datasets have been created or converted from frame-based datasets [3].

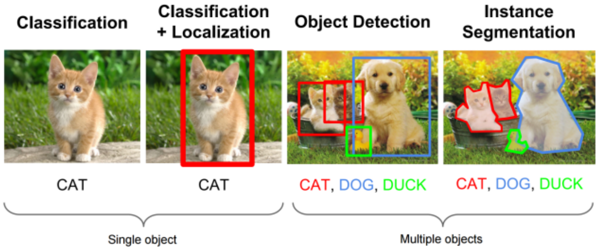


Figure 2: Classification, object detection and instance segmentation examples on frame-based images.[1]

Naturally, event-based data has prompted the development of computer vision methods such as object detection, tracking, and instance and semantic segmentation. The datasets used, however, are either proprietary or overly complex. As such, it must be determined how much information can be extracted from event-based video datasets in order to reliably perform object detection and, the focus of this paper, segmentation.

This paper proposes the following question to clarify the data complexity problem: **Consider constructing event-based segmentation datasets with noisy background, by a superimposition of two event datasets. Would such datasets bring improvements over the original datasets for simulating noisy, real-world environments thereby increasing the performance of segmentation Machine Learning models?**

Two related subquestions follow from the main question:

1. What is the optimal amount of noise that can be superimposed from an existing event-based dataset over a different event-based dataset to gain an improvement in segmentation tasks?
2. How does applying random noise over an existing event-based dataset affect the performance of instance segmentation models?

In summary, the main contributions of this paper are:

- Propose a new approach to creating noisy event-based datasets.
- Create and evaluate event-based datasets with varying amounts of noise on an instance segmentation model.

This paper presents a literature study on image and neuromorphic datasets, data augmentation, and image and event segmentation in Section 2. The methodology is explained in Section 3, with details about generating datasets and segmentation models. Section 4 presents the experiments and results. Section 5 gives a further explanation of results and a reflection on the outcomes. In Section 6, the ethical aspects of this research are discussed, as well as the reproducibility of the methods. Finally, Section 7 summarizes the research questions and provides a conclusion.

[1] https://medium.com/swlh/94ca109274f2

2 Related Work

This section provides a literature review of relevant event-based computer vision research. Existing image and neuromorphic datasets, data augmentation methods, and image and event segmentation literature are all discussed.

2.1 Neuromorphic Datasets

During the 1990s when frame-based computer vision was starting to develop, datasets available were small at first, had little variability in their representations and needed improvements. MNIST [4] had uniform backgrounds, while Caltech-5 [5] had a low number of classes with objects positioned in roughly the same parts of the image. Over time, the importance of how well these datasets represent real-life was recognized and more focus was put towards creating bigger and more complex datasets [6]. This allowed researchers to create more complex computer vision models and, more importantly, perform comparisons with other work using bench-marking datasets.

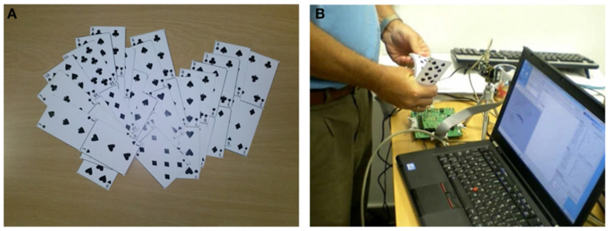


Figure 3: (A) Custom made poker card deck with all pips in black. (B) Browsing the custom poker card deck in front of the DVS camera. Image taken from [7].

Similarly, datasets have a noticeable influence in the field of event-based vision. Most importantly, well-constructed event-based (or neuromorphic) datasets allow for cost reductions of research – since current prices of event-based cameras are very high – and they enable quantitative bench-marking [8]. Small event-based datasets have been created by researchers who want to evaluate their event-based computer vision models: Poker-DVS [7] (cards of a deck, 4 classes, example in Figure 3), faces [9] (7 classes), MNIST-DVS [7] (handwritten digits, 36 classes), DVS-128 [10] (gestures in dynamic scenes). Perhaps an important achievement is the conversion of the well-known frame-based MNIST [4] and Caltech101 [11] datasets into neuromorphic type [3]. Their event representations are shown in Figure 10.

Representations and contents of neuromorphic datasets vary depending on the source camera and the context of the recording. DDD17 [12] for example has recordings of roads from the driver's perspective at different times of the day. It provides the events recorded by a DVS camera alongside the equivalent gray-scale images, plus extra telemetry (vehicle speed, GPS position, driver steering, throttle, and brake) captured from the car's on-board diagnostics interface. Neuromorphic-MNIST [13] and Neuromorphic-Caltech101 [14], however, provide only the events. No telemetry data is relevant in this case, and the frame-based representations are available online, albeit they require some processing if they are to be used for any related task (as explained later in 3.3).

As MNIST [4] and Caltech101 [11] provided a building block for computer vision in its inception, it is worth exploring the usefulness of their neuromorphic representations in the quite new and evolving event-based computer vision field, more specifically using data processing and augmentation techniques in segmentation scenarios.

2.2 Data augmentation

In frame-based vision, data augmentation is a useful tool for enhancing the size and diversity of image datasets to ultimately help reduce overfitting of CNNs [15]. Image data augmentation encompasses two approaches: basic image manipulations and deep learning approaches. In the former, geometric transformations and color space augmentation are applied on the datasets (shown in Figure 4), as used by the revolutionary image classifier AlexNet CNN [16]. The latter approach focuses on improving the model's architecture [17; 18].

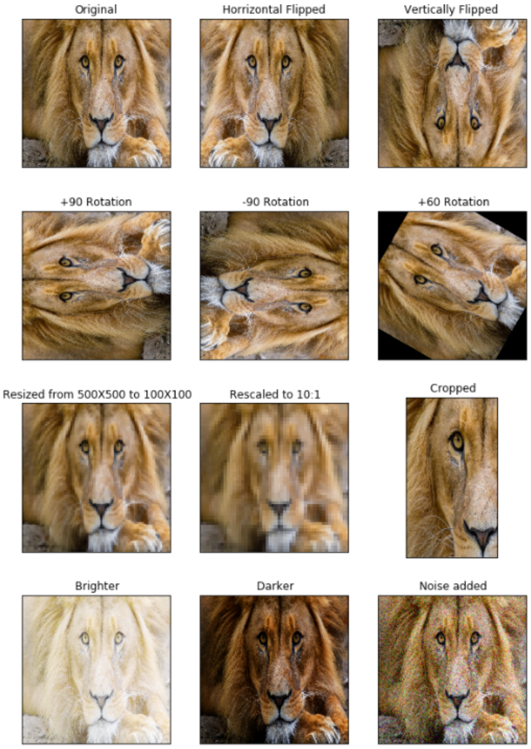


Figure 4: Examples of possible augmentations on one image.[2]

Current approaches to data augmentation for event datasets are scarce. In [19], authors benchmark state-of-the-art models over two neuromorphic datasets by applying data augmentation over frame-based encodings of the events. More recent work applies augmentation directly on event data. EventDrop [20] filters and drops events from the

[2] https://www.mygreatlearning.com/blog/understanding-data-augmentation/

dataset to simulate different levels of occlusion, as seen in Figure 5. In [21], authors employ a collection of geometric augmentations on events (e.g. mixup [22], flipping, rolling, rotation, etc.) by randomly sampling and applying subsets of these augmentations with different probabilities and intensities. Both papers achieve significant improvements over previous state-of-the-art results, but ignore relationships between events when processing them. In [23], authors have taken into consideration spatio-temporal features of events for applying augmentations and therefore achieved better results than previously mentioned papers.

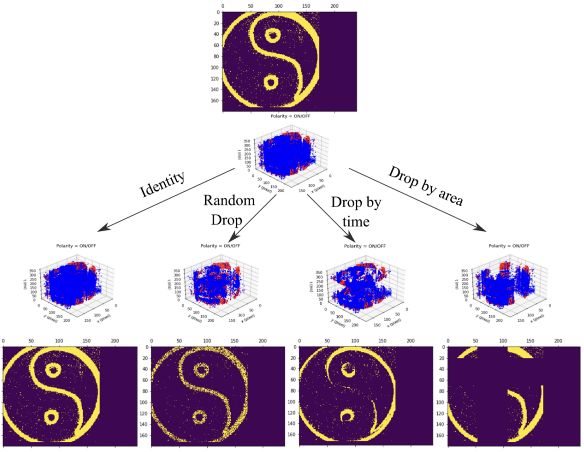


Figure 5: An example of augmented events with EventDrop. Image taken from [20].

The mixup augmentation first introduced in [22] and previously used in [21] serves as inspiration for this paper. In the context of event data, applying this augmentation picks two random samples of events from the same dataset and returns a new sample as a linear interpolation. Naturally, the possibility of combining two or more samples from different datasets (like N-MNIST [13] and N-Caltech101 [14]) arises, which is explored in more detail in subsequent chapters of this paper.

2.3 Image & Event Segmentation

Image segmentation can take two approaches, *semantic* segmentation where the aim is pixel-level labeling using a set of object categories, and *instance* segmentation, which further detects and labels each separate object in the image [24]. For example, using semantic segmentation, all goats in Figure 6 are segmented as one blue object, as is the dog. It is thus a harder task than image classification, where the output is one label per image. Instance segmentation extends this semantic task by further detecting distinct instances of an object in the image – in Figure 6, the goats are labeled individually as different objects of interest.

Segmentation for frame-based computer vision is a thoroughly researched topic. Mask R-CNN is a simple, flexible, and general framework for object *instance* segmentation [25]. It detects objects in the image while also generating masks for each of them, by extending Faster R-CNN's [26] bounding box detection functionality. DeepLabV3+ [27] is a state-of-the-art *semantic* segmentation model that employs an encode-decoder module, using the DeepLabV3 framework as the encoder for feature extraction, and a decoder that recovers sharper segmentations. Its backbone is based on an improved version of Xception [28]. It achieves a new state-of-the-art performance on two important datasets: PASCAL VOC 2012 [29] and Cityscapes [30]. These state-of-the-art models developed for image segmentation serve as an important baseline for working with event-based neural networks.

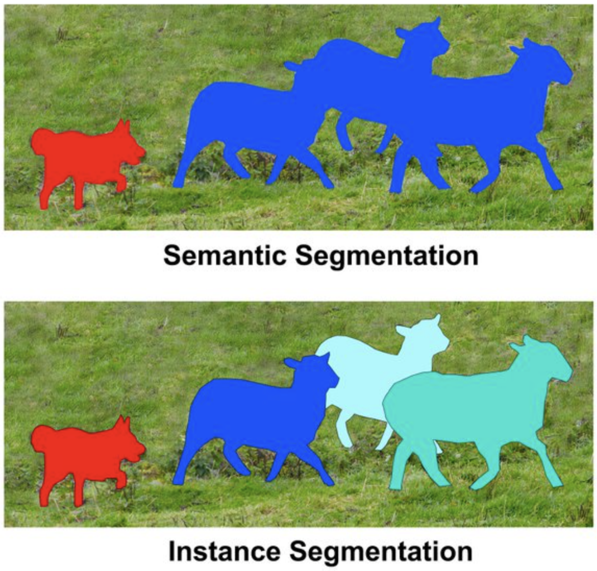


Figure 6: Semantic and instance segmentation on 2 different types of animals.[3]

Segmentation is still rather unexplored in event-based vision [8]. More works addressing this difficult problem begin to appear, as more advanced event-based vision techniques are developed. EV-SegNet [31] introduces a first baseline for semantic segmentation with event data. The authors draw inspiration from the state-of-the-art DeepLabV3+ model and Xception framework to build a semantic segmentation CNN that takes event data as input. Additionally, they propose a superior, new 6-channel representation of the events that surpasses previous representations for related tasks. Their evaluations demonstrate how events and their corresponding gray-scale images are complementary and can yield better results than using only events. Another interesting solution is EvDistill [32], which proposes a new approach to segmentation of unlabeled event data, where a teacher network is trained on labeled, frame-based data and transfers knowledge to learn a student network on un-labeled, event data.

EV-Mask-RCNN [33] is an instance segmentation model based on the state-of-the-art Mask R-CNN [25] framework. It transforms the input event data into RGB-Depth images, creates the ground truth masks and segments each object

[3] https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Mask_RCNN

instance in the frame. Since it is compatible with the dataset generation methods presented in the paper and can generate the segmentation masks, the model used for evaluations is EV-Mask-RCNN [33].

3 Methodology

The main research question proposes to analyze the effect of applying noise to an existing event-based dataset on segmentation models. For this reason, three related steps are considered for constructing noisy event-based datasets and evaluating them on segmentation tasks. They provide insights for answering the two related subquestions stated in Section 1.

3.1 Neuromorphic Datasets

The datasets used in this paper are two simple and accessible datasets: **Neuromorphic-MNIST** [13] and **Neuromorphic-Caltech101** [14]. They are the neuromorphic representations of the well-known MNIST [4] and Caltech101 [11] frame-based datasets, which have been converted to event representations by Orchard et. al [3].

The two datasets are saved as a set of separate binary files, each consisting of a list of events. Each event is represented by 40 bits as follows:

- bit 39-32: x coordinate in the sensor
- bit 31-24: y coordinate in the sensor
- bit 23: polarity – 0 for OFF, 1 for ON
- bit 22-0: timestamp – in microseconds

Tonic[4] library is used for loading and applying transformations on event datasets like N-MNIST and N-Caltech101. The library loads and processes the binary files for N-MNIST and N-Caltech101 into a representation easy to work with in Python. It returns a list of events represented by 4-tuples in the form of *(x, y, timestamp, polarity)* and the corresponding digit label.

3.2 Superimposing strategies

The main objective is to create noisy datasets containing N-MNIST digits overlapping background noise, with the evaluation goal of identifying the digits via segmentation methods. Three such datasets are created with different degrees of background noise, as explained in the next paragraphs. Appendix B shows a sample of each digit from each noisy dataset generated and from N-MNIST. The average amount of events per digit, in each dataset, is plotted in Figure 7.

Superimposed-Noisy dataset

This dataset contains the most amount of background noise. One entry contains a merged background with a foreground, with the former as a random entry from N-Caltech101 and the latter being a digit from N-MNIST, as seen in Figure 12.

Each entry of the Superimposed-Noisy dataset is constructed by applying a series of manipulations to the background and the foreground. A denoise filter is applied to the background and foreground events to filter out any random sensor noise. The background is cropped to the size of the foreground (34 x 34 pixels) to make it compatible with the evaluation model (later explained in 3.3) – that is, events occurring outside the selected area are removed. The background events are filtered once again to remove those with a timestamp before or after the foreground events.

The final step is superimposing the foreground over the background. Their array representations are concatenated and a remove-overlaps filter is applied. Note that now some events of the background may occur at the same place and time with the foreground ones. They may have the same (x, y) coordinates and timestamp. These ’conflicts’ are found by sorting the concatenated events array into buckets for coordinates and timestamps to find the overlapping ones, then only one ON event of any overlap is kept.

Centered-Filtered dataset

This dataset is a less noisy variant of the aforementioned Superimposed-Noisy dataset. It is therefore constructed similarly, with the addition of two manipulations.

Analyzing the contents of N-Caltech101 entries, the center of the frame almost always contains lots of events. Therefore it is worth constructing a dataset with a crop of the background in the center instead of a random crop. This yields quite a noisy frame, so the background events are downsampled to a predefined maximum number, which is picked to be 3000 after experimental testing. Figure 13 shows some samples from this dataset.

Uniform-Noisy dataset

The previous datasets effectively use (partial) shapes or objects as background noise. When cropping into the N-Caltech101 frame, a section of an object might be used as noise. Or it can be the case a bigger object is captured in the cropped frame and fills it almost entirely.

The Uniform-Noisy dataset is inspired by the case where the entire frame is full of noise. Instead of N-Caltech101 events, a fixed number of events are introduced in the background, drawn from a uniform distribution across event dimensions *(x, y, timestamp, polarity)*. A sample from this dataset is shown in Figure 14.

Implementation

To generate superimposed entries for all of N-MNIST, there is a lot of complexity for processing. The size of the dataset is around 60000 entries. Doing complex computations like the previously mentioned sorting into buckets for each entry does not scale well, taking a minimum of 9 hours to complete for the whole dataset.

Taking advantage of multiple cores of the CPU, the work can be optimized by splitting ranges of entries on different cores. If there are 10 cores available, the 60000 entries would be split into 6000 per core. Using the 10 cores, 10000 events are done processing in about 7 minutes, while 60000 events are done in about 50 minutes.

With the help of multi-core processing, faster merging is achieved, which allows for more evaluations. The source code for generating these datasets as well as instructions

[4] https://tonic.readthedocs.io/en/latest/

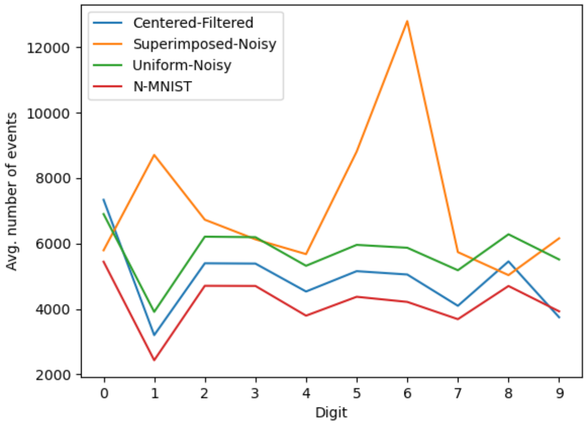


Figure 7: Average amount of noise per digit for each generated dataset.

for the instance segmentation model are available at https://github.com/alexmanoo/dvs_datasets_transforms.

3.3 Evaluations

The three generated datasets are evaluated using an instance segmentation model, to analyze the effect of different amounts of noise on the model's performance and generalizability. It is expected that a model trained on a dataset with background noise should perform worse on the same dataset without noise, because the data itself (i.e. the digits) is not necessarily modified, only the backgrounds. The model used for evaluations is EV-Mask-RCNN [33], which generates the ground truth of the input data, and performs instance segmentation.

Ground truth of N-MNIST

To perform segmentation on N-MNIST, the ground truth masks of each digit are required. They are not provided in the original dataset, but they can be generated. In [33], the author generates a proprietary representation of the masks for N-MNIST. The digit shapes are approximated by making an average frame of events from a time window (e.g. 10ms, 20ms, 50ms, etc.), and each digit is matched with the frame-based MNIST to calculate exact positions of the mask. The same procedure was used in this paper, with a time-window of 20ms. Examples of such masks are shown in Figure 8.

4 Experiments

This section describes the experimental setup and results for evaluating noisy datasets created according to the methodology in Section 3.

4.1 Setup

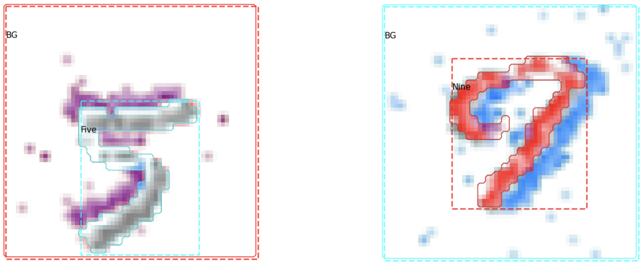


Figure 8: Ground truth of digits 5 and 9, left and right, respectively, generated using [33].

The datasets are generated on a device with 10-core M1 CPU and 16GB RAM, following the dataset construction strategies and implementation optimizations explained in section 3.2. The following three datasets are created: Superimposed-Noisy, Centered-Filtered, and Uniform-Noisy. Additionally, the original N-MNIST dataset which does not contain any noise is used together with the noisy datasets to provide a baseline for the experiments. Then each dataset is converted to the representation accepted by EV-Mask-RCNN [33]. A time-window of 20ms is used.

For TensorFlow compatibility reasons, the training and testing using EV-Mask-RCNN [33] are executed on a laptop with *Intel Core i7 8th Gen 8750H* CPU, *Nvidia GeForce GTX 1050 Ti* GPU, and 32GB RAM. The model is set up with a starting leaning rate of 0.001 which is then adjusted after a number of epochs for each dataset. The loss is calculated by a sum of different losses for anchor boxes, localization accuracy, object classification in the region proposal, localization of the bounding box, and masks of identified objects. The loss weights tell the model which aspects are most important. Since this is a segmentation task, and identification at pixel level is of importance, the mask importance loss weight is increased the most.

Three instances of the model are trained for each of the four datasets, resulting a total of 12 trained models. Instance 1 is trained for 12 epochs, instance 2 for 20 epochs, and instance 3 for 30 epochs. For each instance, the learning rate starts at 0.001 and drops at 0.0001 after 5, 8, and 15 epochs respectively.

Three metrics used for scoring the experiments are recorded by EV-Mask-RCNN [33], all stated in percentages – mean Accuracy (*mAcc*), mean Average Precision (*mAP*), and mean Intersection over Union (*mIoU*). They have been used as main metrics to evaluate performance in past event segmentation models, e.g. Ev-SegNet [31] and EvDistill [32].

4.2 Results

Tables 1, 2, 3, 4 contain the evaluation metrics calculated after testing each model on its respective test dataset. All results of noisy models (Tables 2, 3, 4) have comparable accuracy but smaller *mAP* and *mIoU* than N-MNIST model (Table 1). The *mAcc* is similar because it measures pixel-wise correct labeling in a frame where the background is always the majority of pixels. It is worth noting that the model trained and evaluated on Uniform-Noisy dataset, the second noisiest out of all, is closest to N-MNIST trained model.

The best model for all noisy datasets is the one trained on 30 epochs. For the non-noisy one, 20 epochs is the optimal amount. It is clear the noise introduced in the datasets requires additional training time for the model to extract

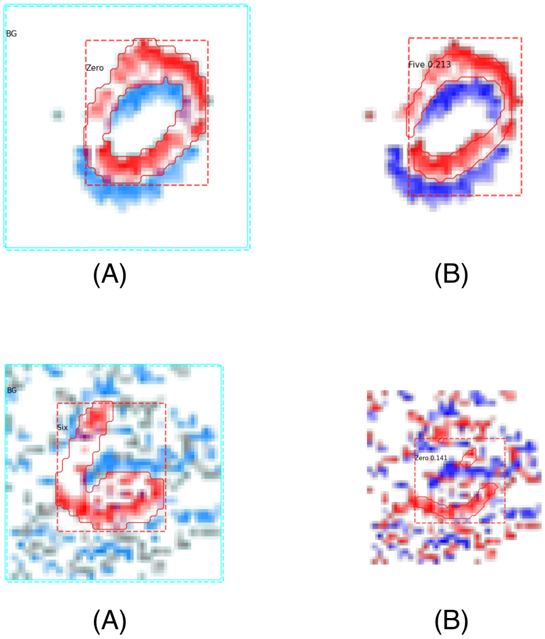


Figure 9: (A) Ground truth of digits zero and six. (B) Segmentation mask and class predictions.

relevant features. Segmentation results examples are shown in Figure 9.

Cross evaluations. To further understand if training models on noisy datasets results in improvements, each model presented in this section is evaluated on the original N-MNIST test set for 10 times. The best models for each dataset (highlighted in gray in Tables 1 to 4) were used, and the averaged results are presented in Table 5.

Table 5: Results of training on noisy datasets and testing on N-MNIST test dataset.

Model	mAcc	mAP	mIoU
Superimposed-Noisy	94.75	21.97	24.36
Centered-Filtered	95.12	28.44	34.24
Uniform-Noisy	95.31	29.3	36.58
No-noise	95.53	33.71	42.3

Table 6: Standard deviations of metrics from Table 5.

Standard deviations			
Model	mAcc	mAP	mIoU
Superimposed-Noisy	0.06	1.01	1.23
Centered-Filtered	0.08	1.04	1.32
Uniform-Noisy	0.06	1.25	1.52
No-noise	0.03	0.88	1.18

Table 1: Results of training and testing on N-MNIST dataset. Best model highlighted in light gray.

Model	mAcc	mAP	mIoU
5-12eps	95.07	23.1	26.63
8-20eps	95.57	34.4	42.99
15-30eps	95.22	33.3	40.14

Table 2: Results of training and testing on Superimposed-Noisy dataset. Best model highlighted in light gray.

Model	mAcc	mAP	mIoU
5-12eps	84.85	5.1	4.0
8-20eps	92.99	10.3	8.58
15-30eps	93.55	20.4	20.13

Table 3: Results of training and testing on Centered-Filtered dataset. Best model highlighted in light gray.

Model	mAcc	mAP	mIoU
5-12eps	94.11	18.5	18.24
8-20eps	93.4	23.4	25.28
15-30eps	94.54	27.2	30.31

Table 4: Results of training and testing on Uniform-Noisy dataset. Best model highlighted in light gray.

Model	mAcc	mAP	mIoU
5-12eps	90.32	11.6	10.67
8-20eps	94.13	20.2	20.57
15-30eps	94.66	28.9	32.43

4.3 Statistical significance of cross evaluations

Looking at cross evaluations averaged results in Table 5 (and standard deviations in Table 6), there is significant difference between *mAP* and *mIoU* metrics of each model. One can conclude they are not drawn from the same distribution, that the noisy models perform worse than the no-noise one, and, therefore, do not provide an improvement for segmentation tasks.

To test this hypothesis, a statistical significance test (two-tailed, independent T-Test [34]) is performed for all three metrics, to determine if the means of two sets of data are significantly different from each other. The chosen null hypothesis is ”The means of the two sets of data are equal”. Each metric (*mAcc*, *mAP*, *mIoU*) of noisy models is tested against the corresponding metric of No-noise model, and the resulting p-value is analyzed. To accept the null hypothesis, the p-value has to be greater than 0.05. Performing the t-test yields all *p-values* < 0.001 , which signifies no underlying features exist in the sets of data that would make them belong to the same distribution. The outcome of this test verifies there is no improvement for the presented segmentation task when adding noise to an event-based dataset. All p-values and t-statistics are shown in Table 7.

5 Discussion

The noisy datasets created in this paper are intended to simulate noisy, real-world environments. In addition, the main research question is whether these datasets can make machine learning model training more generalizable.

The significance test in section 4.3 concluded that the models trained on noisy datasets perform worse on a dataset with no noise, than a model trained directly on the No-noise dataset. Intuitively, this result is expected, but the results on Uniform-Noisy dataset are worth mentioning. Although it was the second noisiest dataset (see Figure 7), the model performed the best when trained on this dataset, surpassing Centered-Filtered which had 1000 less events on average per digit. The uniformity of the noise did not impact the model's learning ability as much as the other types of noise and allowed it to perform best in the testing scenario.

Despite the fact that the evaluation metrics in Tables 1, 2, 3, and 4 reflect the models' lower ability on noisy data, the segmentation masks placement is comparable across all models. The mask losses for all trained models are shown in Appendix D. They are all between 1 and 1.5, implying that the noisy models are almost as good as the no-noise model at applying (partial) masks to the objects. However, because the test datasets differ in noise, this comparison should be viewed with caution.

The fact that the dataset backgrounds have been altered with noise is a reason for the negative results of this paper. Random data is essentially added to the digits, which can only negatively impact a model. Data augmentations are usually done on the data itself, i.e. the digit, like in EventDrop [20]. Because they alter the data itself rather than the background, the generalizability of predictions improves.

Furthermore, superimposing two datasets as presented in the paper is not natural. Normally, when an object is placed in front of another, there is some depth between them and their movements are not exactly matched. In other words, there is a moving object and a more static background, or a static object and moving background. The dataset merging results of this paper match the movements of objects exactly. By applying rotations or time shifting on the digits, some variability in the movement directions could be achieved.

A methodology improvement for this paper would be using different time-window intervals for converting to events to frames. The experiments in Section 4 have been carried out on a 20 milliseconds average frame of events on all four event-based datasets, however in [33] best results follow on 50 milliseconds time-windows. With noisier datasets, smaller time-windows of 15, 10, or 5 milliseconds might improve model metrics. Additionally, cross evaluations have only been done on the no-noise MNIST dataset. Using the other noisy datasets to perform cross evaluations can also give some more insights into relationships between the noise and model.

This paper demonstrated superimposing of two toy neuromorphic datasets (N-MNIST [13] and N-Caltech101 [14]) using the computational power of consumer hardware. For each entry of N-MNIST, some events from the other dataset were added, with some filtering and frame cropping being done. On the N-MNIST dataset with 60000 entries, merging duration is 9 hours for single core operations, and 50 minutes when scaling to 10 cores. This type of operation is manageable on a consumer system because the chosen dataset is small both in terms of size and resolution. For much larger datasets, similar computations might take days. It is therefore recommended to utilize systems with superior CPUs and RAM, as well as keeping the complexity of operations on each dataset entry as low as possible.

6 Responsible Research

This paper's research is conducted using already available datasets and Machine Learning models. Therefore, data integrity and the ethicality of models could impact the credibility and reproducibility of the results.

The two datasets of this research are neuromorphic adaptations of the popular MNIST [4] and Caltech-101 [11] computer vision datasets. They are public and easy to load with the help of open-source libraries like Tonic[5]. MNIST contains digits written by humans, however they are not identifiable or traceable back to the original persons. Caltech-101 consists of pictures of objects and some of faces. On the Caltech website[6], they state the images are for non-profit scientific experiments, they are not Caltech property, and any use other than *fair use* should be negotiated with the pictures' owners.

Machine Learning models are well-known for their ability to extract features and information from all types of data not visible by humans. A human expert in Artificial Intelligence may not be able to read or explain the results of a model, and it may be difficult to understand why the model came to certain conclusions. Similarly, the human eye cannot identify objects or features in some representations of neuromorphic datasets because the events might be sparse. That is why care is taken when using the machine learning model of this paper. The only features it is trained to identify are the digits in the frame, and nothing more.

Results achieved in this paper contradict the research question and invalidate this research goal. However, negative results are equally important as new findings and contribute to the understanding of the topic.

All the code is publicly available at this link[7] to allow anyone to inspect the code of generating the results of this paper. Furthermore, comments and instructions in the README file provide support and guides to enable easy code running and reproducibility.

[5] https://tonic.readthedocs.io/en/latest/

[6] http://www.vision.caltech.edu/datasets/caltech_10k_webfaces

[7] https://github.com/alexmanoo/dvs_datasets_transforms

7 Conclusions and Future Work

This paper presents a method to add noisy backgrounds to event-based datasets, by superimposing two toy datasets and by adding uniform noise. It has demonstrated that training an instance segmentation model on such noisy datasets does not increase its performance, but the type of added noise decreases the performance of such model. For future work, additional methods of constructing superimposed datasets can be implemented. An idea is to increase the variability of noise between the generated datasets. Additionally, larger or smaller time windows than the one used in this paper's experiments might improve outcomes.

References

[1] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 microsecond latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, 2011.

[2] S. Seifozzakerini, W.-Y. Yau, K. Mao, and H. Nejati, "Hough transform implementation for event-based systems: Concepts and challenges," *Frontiers in computational neuroscience*, vol. 12, p. 103, 2018.

[3] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, p. 437, 2015.

[4] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[5] Caltech Vision Lab, "Caltech-5." http://www.vision.caltech.edu/datasets/, 2022.

[6] C. Tan, S. Lallee, and G. Orchard, "Benchmarking neuromorphic vision: lessons learnt from computer vision," *Frontiers in neuroscience*, vol. 9, p. 374, 2015.

[7] T. Serrano-Gotarredona and B. Linares-Barranco, "Poker-dvs and mnist-dvs. their history, how they were made, and other details," *Frontiers in neuroscience*, vol. 9, p. 481, 2015.

[8] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[9] G. Lenz, S.-H. Ieng, and R. Benosman, "Event-based face detection and tracking using the dynamics of eye blinks," *Frontiers in Neuroscience*, p. 587, 2020.

[10] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, *et al.*, "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178, IEEE, 2004.

[12] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "Ddd17: End-to-end davis driving dataset," *arXiv preprint arXiv:1711.01458*, 2017.

[13] Garrick Orchard, "Neuromorphic mnist." https://www.garrickorchard.com/datasets/n-mnist, 2022.

[14] Garrick Orchard, "Neuromorphic caltech-101." https://www.garrickorchard.com/datasets/n-caltech101, 2022.

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[19] K. S. Krishnan and K. S. Krishnan, "Benchmarking conventional vision models on neuromorphic fall detection and action recognition dataset," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0518–0523, IEEE, 2022.

[20] F. Gu, W. Sng, X. Hu, and F. Yu, "Eventdrop: data augmentation for event-based learning," *arXiv preprint arXiv:2106.05836*, 2021.

[21] Y. Li, Y. Kim, H. Park, T. Geller, and P. Panda, "Neuromorphic data augmentation for training spiking neural networks," *arXiv preprint arXiv:2203.06145*, 2022.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[23] G. Shen, D. Zhao, and Y. Zeng, "Eventmix: An efficient augmentation strategy for event-based data," *arXiv preprint arXiv:2205.12054*, 2022.

[24] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[29] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[31] I. Alonso and A. C. Murillo, "Ev-segnet: Semantic segmentation for event-based cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[32] L. Wang, Y. Chae, S.-H. Yoon, T.-K. Kim, and K.-J. Yoon, "Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 608–619, 2021.

[33] A. Băltărețu, "Ev-mask-rcnn: Instance segmentation in event-based videos," 2022.

[34] T. K. Kim, "T test as a parametric statistic," *Korean journal of anesthesiology*, vol. 68, no. 6, pp. 540–546, 2015.

A Supplementary

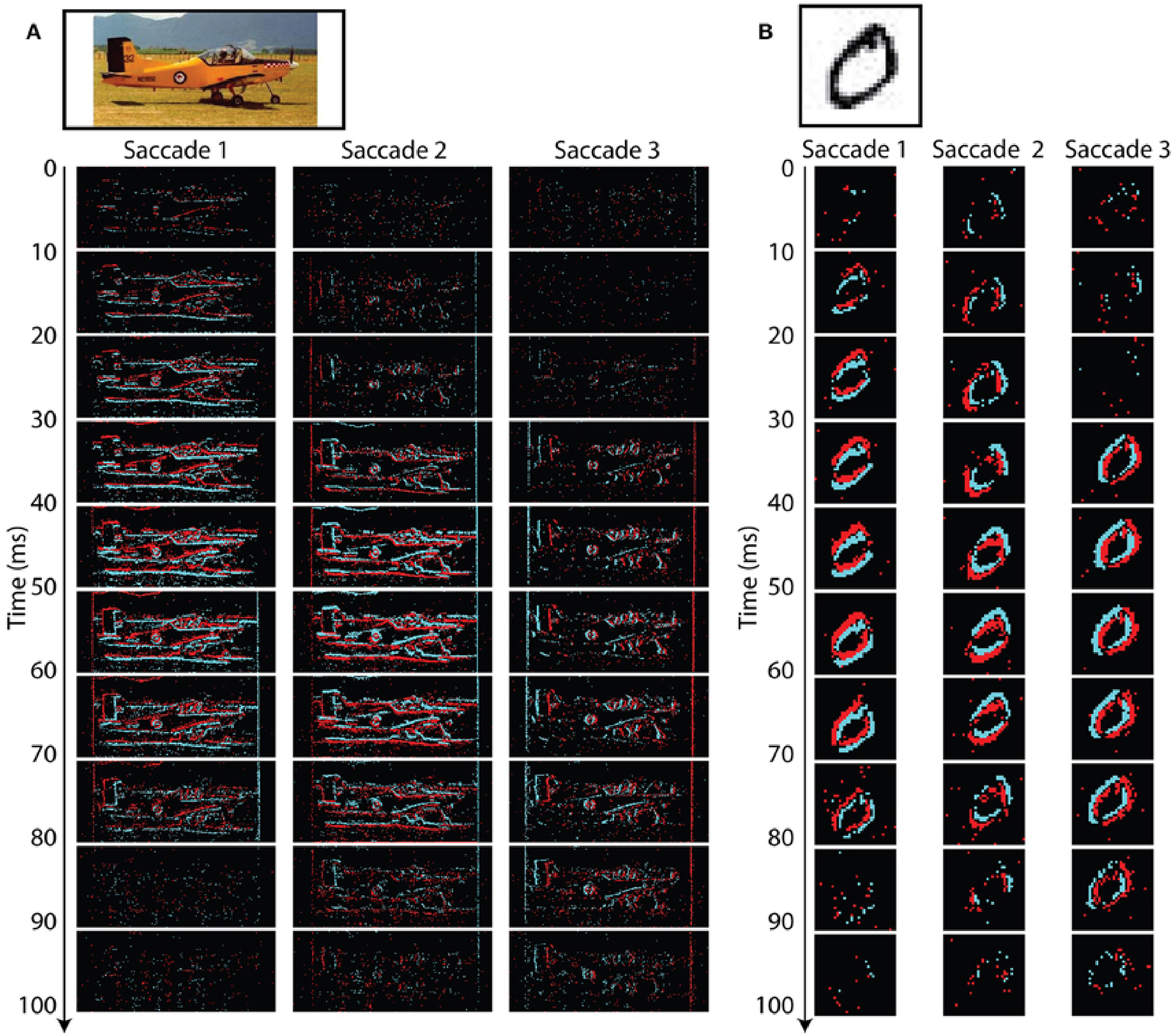


Figure 10: Recordings with a DVS camera for Caltech101 (A) and MNIST (B). Image taken from [3].

B Noisy datasets samples

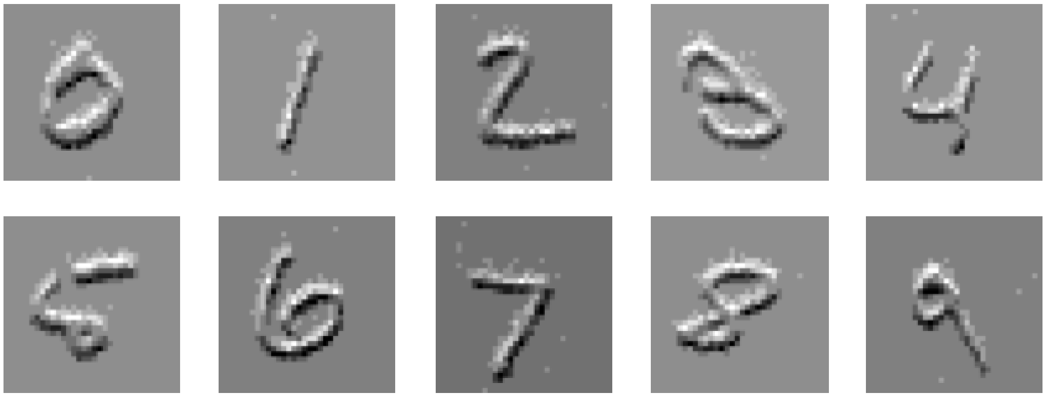


Figure 11: Sample of each digit from N-MNIST dataset.

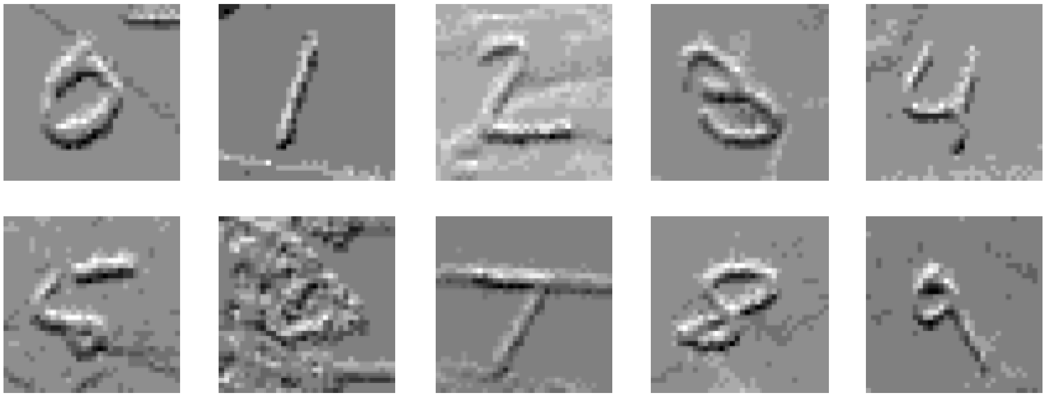


Figure 12: Sample of each digit from Superimposed-Noisy dataset.

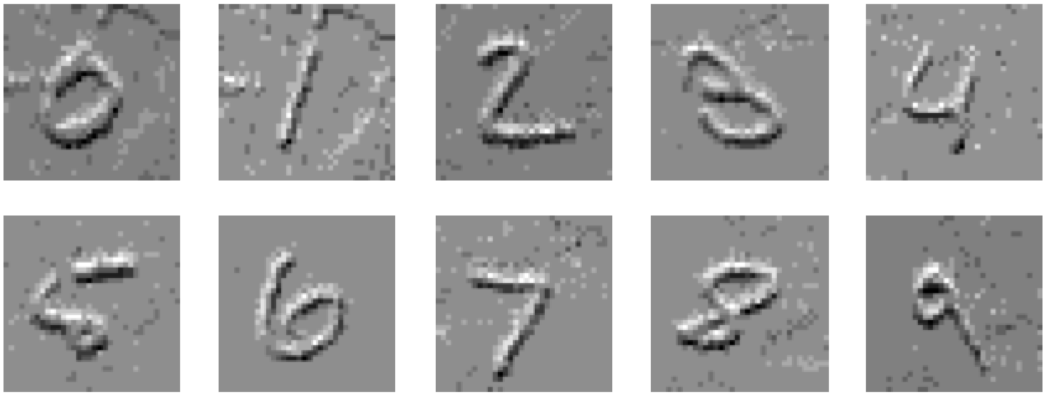


Figure 13: Sample of each digit from Centered-Filtered dataset.

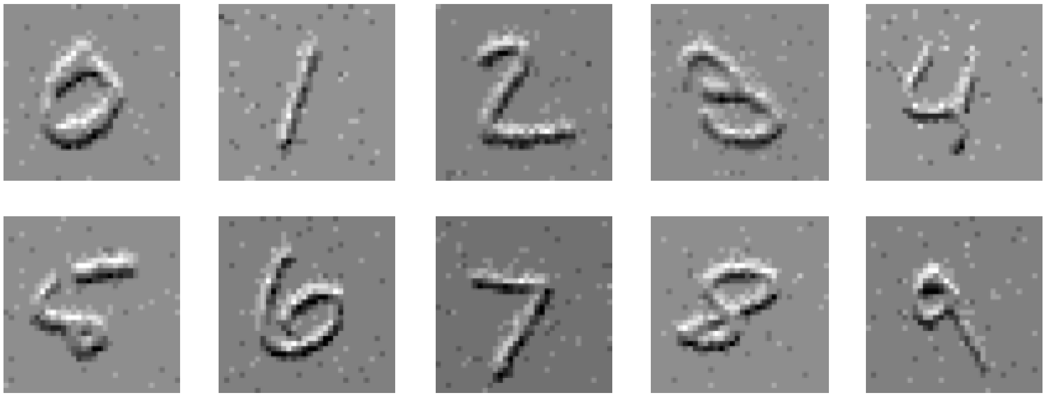


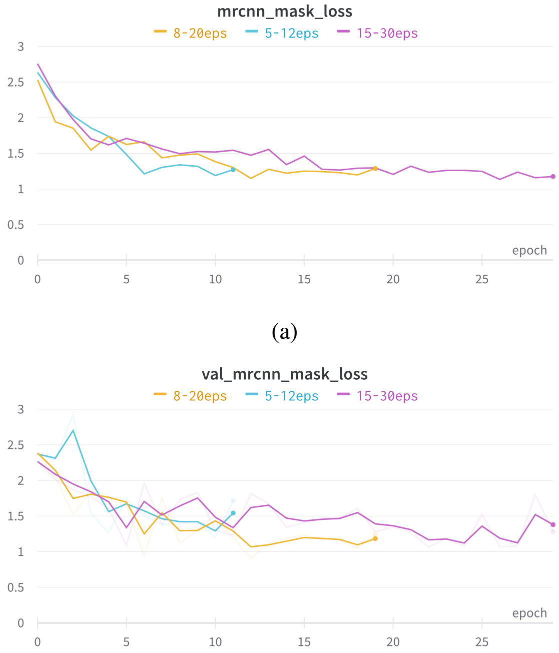
Figure 14: Sample of each digit from Uniform-Noisy dataset.

C T-test results

Table 7: P-value and t-statistic results for t-tests on three noisy datasets. Null hypothesis is ”The means of the two sets of data (noisy and no-noise) are equal”. P-values are < 0.001, suggesting the rejection of the null hypothesis.

	Superimposed-Noisy			**Centered-Filtered**			**Uniform-Noisy**		
Metric	mAcc	mAP	mIoU	mAcc	mAP	mIoU	mAcc	mAP	mIoU
p-value	1.99e-17	8.29e-16	3.31e-17	5.55e-11	8.60e-10	6.01e-11	3.89e-08	7.57e-08	5.05e-08
t-statistic	32.45	26.27	31.53	13.73	11.60	13.67	9.07	8.67	8.91

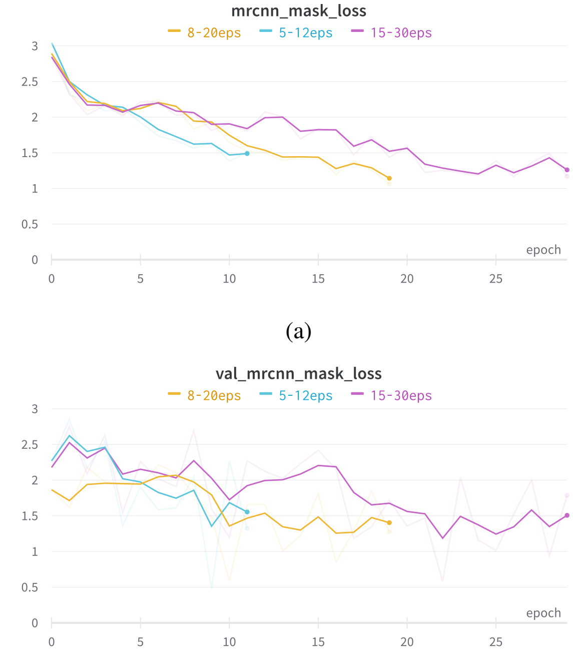
D Validation losses



(a)

(b)

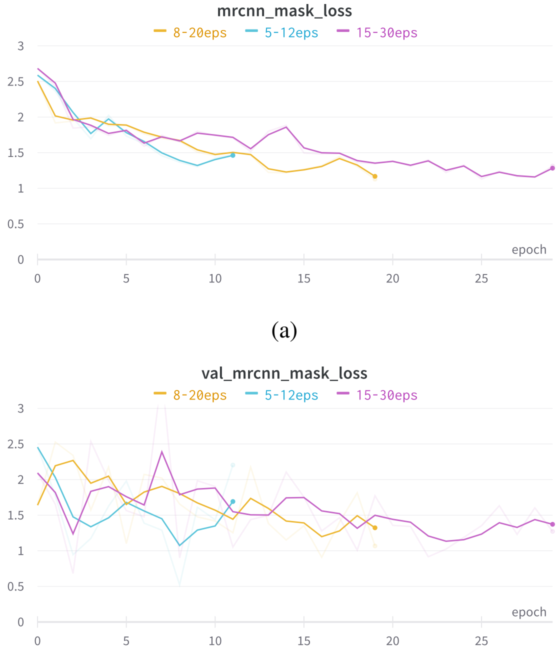
Figure 15: (a) Mask loss of models trained on N-MNIST dataset. (b) Respective validation mask loss.



(a)

(b)

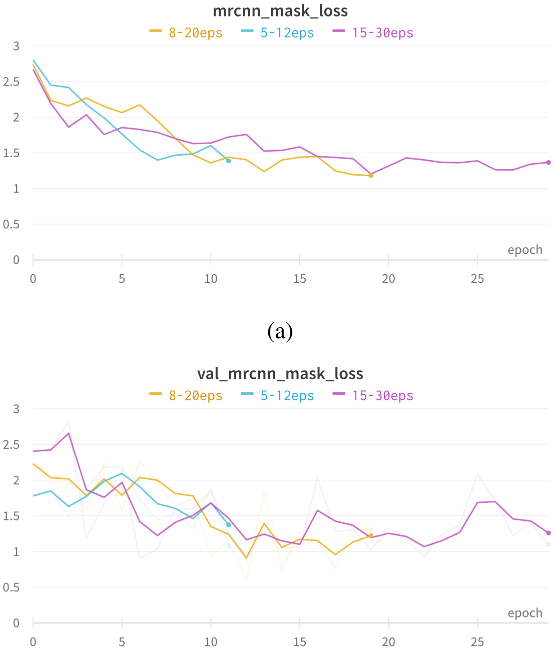
Figure 16: (a) Mask loss of models trained on Superimposed-Noisy dataset. (b) Respective validation mask loss.



(a)

(b)

Figure 17: (a) Mask loss of models trained on Centered-Filtered dataset. (b) Respective validation mask loss.



(a)

(b)

Figure 18: (a) Mask loss of models trained on Uniform-Noisy dataset. (b) Respective validation mask loss.