



Robustness of Spiking Neural Networks under Noisy Event-based Inputs

MSc Project Report

Timur Khismatulin

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2024-25

Abstract

Spiking Neural Networks (SNNs) are biologically inspired models well-suited for event-based data from neuromorphic vision sensors. Their temporal coding and asynchronous computation offer energy-efficient processing, yet robustness to noise remains a major challenge in real-world scenarios.

This work investigates SNN robustness in classification under noisy conditions using the MNIST dataset encoded into spike trains via Poisson coding. Three configurations were tested: (i) SNN with Spike-Timing Dependent Plasticity (STDP) and Random Forest classification, (ii) SNN as a fixed feature extractor with Random Forest, and (iii) a baseline Random Forest on raw inputs. Two scenarios were considered: both training and testing on noisy data, and training on noisy data while testing on clean data.

Results show that SNNs are more sensitive to noise compared to the Random Forest baseline, with performance degradation evident in accuracy–SNR curves. The findings highlight the need for noise-robust encoding and learning rules, contributing to the development of reliable neuromorphic systems for event-driven applications.

Declarations

I certify that this project is my own work. Development of code and experiments was conducted by me, with occasional assistance from ChatGPT 5.0 for debugging, code optimisation, and clarification of errors. Drafting and structuring of the report were completed by me; ChatGPT was used for language refinement, summarisation of sections, and suggestions on clarity. All outputs generated with assistance have been critically reviewed, verified, and edited by me to ensure accuracy and consistency.

Notations

Symbol / Abbreviation	Meaning
$S_{ij}(t)$	Spike train of pixel (i, j) at time t
λ	Rate parameter for Poisson encoding
T	Simulation window length (ms)
Δt	Discrete time step (ms)
v_θ	Membrane threshold potential
v_{reset}	Reset potential after spike
τ_m	Membrane time constant
A_+, A_-	STDP learning rates for potentiation/depression
τ_+, τ_-	STDP time constants
Acc	Classification accuracy
SNN	Spiking Neural Network
STDP	Spike-Timing Dependent Plasticity
RF	Random Forest
DVS	Dynamic Vision Sensor
N+N	Training and testing on noisy data (noisy–noisy)
N+C	Training on noisy data, testing on clean data (noisy–clean)

Contents

Abstract	ii
Declarations	iii
List of Figures	vii
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Aims and Objectives	3
1.4 Contributions of this Work	4
1.5 Summary	5
2 Literature Review	6
2.1 Introduction	6
2.2 Spike Encoding and Noise Sensitivity	6
2.3 Training Approaches and Robustness	7
2.4 Neuromorphic Datasets and Noise	7
2.5 Neuromorphic Hardware	8
2.6 Applications and Robustness Considerations	8
2.7 Summary and Research Gaps	8
2.8 Discussion of Literature Review Table	12
3 Methodology	14
3.1 Overview of Methodology	14
3.1.1 Random Forest classifier	15
3.2 Dataset	15
3.2.1 Poisson rate coding	15
3.2.2 Preprocessing and rate scaling	16

3.2.3	Filtered dataset size	16
3.2.4	Implementation note	17
3.3	Model Configuration	17
3.3.1	Populations	17
3.3.2	Neuron dynamics	17
3.3.3	Connectivity	17
3.3.4	Spike-Timing Dependent Plasticity (STDP)	18
3.3.5	Spike raster under STDP	18
3.3.6	Simulation protocol and hyperparameters	18
3.3.7	Gaussian noise model (SNR)	19
3.3.8	Experiment Types and Scenarios	19
3.4	Evaluation metric	21
3.5	Summary	21
4	Results and Analysis	22
4.1	Gaussian noise: noisy training and noisy testing	23
4.1.1	Results for Experiment Type A (SNN + STDP + RF)	23
4.1.2	Results for Experiment Types B and C	24
4.2	Gaussian noise: noisy training and clean testing (N+C)	25
4.2.1	Results for Experiment Type A (SNN + STDP + RF)	25
4.2.2	Results for Experiment Types B and C (SNN frozen + RF, RF baseline)	26
4.2.3	Comparative analysis: N+N vs N+C	27
5	Conclusion and Future Work	28
5.1	Conclusion	28
5.2	Limitations	29
5.3	Future Work	29
5.4	Final Remarks	30
Appendix		31

List of Figures

1.1	Comparison of conventional video frames (bottom) and event-based DVS representation (top) [11].	2
1.2	Pedestrian detection: image-based (left) vs event-based (right) [20].	3
1.3	Spiking Neural Network architecture with excitatory and inhibitory populations connected via STDP synapses [12].	4
7figure.caption.10		
3.1	Workflow: MNIST → noise (SNR) → Poisson encoding → Random Forest (direct) or via SNN features.	15
3.2	Random Forest as an ensemble of decision trees with aggregation by majority voting (classification) or averaging (regression).	16
3.3	Spike raster during STDP training, illustrating excitatory and inhibitory activity under the plasticity rule [12].	18
3.4	Examples of noisy MNIST images at different SNR levels (10, 5, and 0 dB). .	20
4.1	Classification accuracy of Experiment Type A (SNN + STDP + RF) across SNR levels under Gaussian noise.	23
4.2	Experiment Types B (SNN frozen + RF) and C (RF baseline): classification accuracy across SNR levels under Gaussian noise (N+N scenario). .	24
4.3	Experiment Type A (SNN + STDP + RF): classification accuracy across SNR levels under Gaussian noise (N+C scenario).	26
4.4	Experiment Types B (SNN frozen + RF) and C (RF baseline): classification accuracy across SNR levels under Gaussian noise (N+C scenario). .	27

CHAPTER 1

Introduction

1.1 Introduction

Spiking Neural Networks (SNNs)¹ are often described as the third generation of neural network models, distinguished by their ability to exploit temporal coding and event-driven communication [15]. Unlike conventional artificial neural networks (ANNs), which rely on continuous activations and synchronous updates, SNNs communicate via discrete spikes. This feature makes them particularly suitable for neuromorphic hardware and energy-efficient computation [19]. In contrast to frame-based inputs, where redundant information is repeatedly processed, spike-based computation inherently reduces redundancy by focusing on meaningful temporal changes.

With the development of neuromorphic vision sensors, such as the Dynamic Vision Sensor (DVS)² [14], there has been a growing interest in applying SNNs to real-time event-based data. Event-based sensors differ fundamentally from traditional cameras: instead of capturing entire frames at fixed intervals, they asynchronously record only pixel-level intensity changes. This enables microsecond temporal resolution and significantly reduces data rates, which is particularly advantageous for robotics and embedded systems [9]. Figure 1.1 illustrates this contrast: conventional cameras record redundant frame sequences, while DVS sensors capture only pixel-level events corresponding to changes in the scene.

However, the shift towards event-driven perception also introduces challenges, such as robustness to noise and the development of algorithms that can fully exploit temporal coding.

While SNNs have demonstrated promising results in image and pattern recognition tasks [6, 21], their robustness to noisy inputs remains insufficiently explored. In real-

¹Neural networks where information is transmitted via discrete spikes.

²A camera that records only pixel-level intensity changes asynchronously.

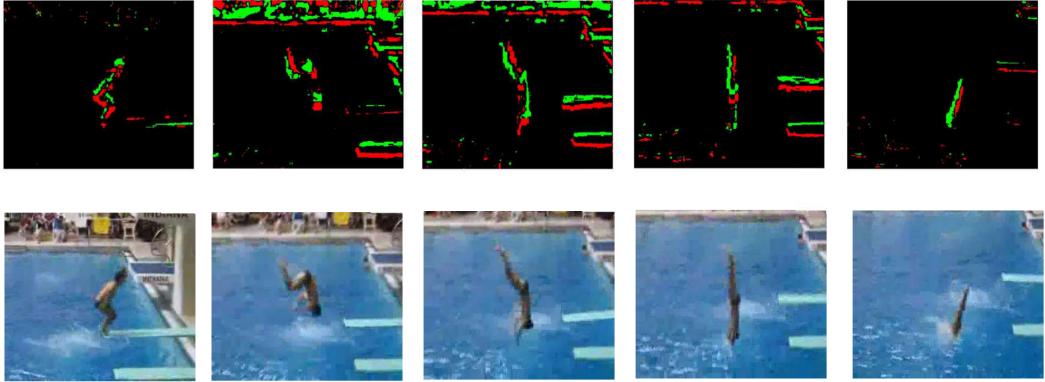


Figure 1.1: Comparison of conventional video frames (bottom) and event-based DVS representation (top) [11].

world scenarios, event-based data is affected by background noise, sensor variability, and transmission errors [27, 18]. Robustness is a critical property for practical deployment, especially in safety-critical domains, but much of the existing research has focused on demonstrating baseline performance rather than systematically analysing noise sensitivity. Recent surveys confirm that although surrogate gradient methods³ and ANN-to-SNN conversion techniques have advanced performance, noise handling is often left as an open problem [16].

1.2 Motivation

The motivation for this project stems from the increasing importance of neuromorphic vision in real-world applications such as robotics, autonomous driving, and mobile sensing. Unlike conventional frame-based cameras, event-based sensors capture only changes in the visual field, providing high temporal resolution and reduced redundancy [9]. This makes them highly efficient, but also inherently sensitive to background noise, hardware imperfections, and variable illumination. For example, DVS sensors are prone to spurious events caused by flickering light sources or rapid camera motion, which can degrade the quality of downstream classification.

In practice, noisy inputs can significantly reduce system performance, particularly in safety-critical applications such as collision avoidance in robotics or pedestrian detection in autonomous vehicles [24, 22]. Figure 1.2 illustrates this challenge: while a conventional camera (left) struggles to detect a pedestrian emerging from behind a vehicle, the event-based sensor (right) captures rapid pixel-level changes that allow for earlier and more reliable detection. Such capabilities highlight the potential safety advantages of event-driven vision, but also emphasise the risks if robustness to noise and artefacts is not

³Surrogate gradients are approximate derivatives that enable gradient-based optimisation in SNNs despite the non-differentiability of spike generation.

properly addressed.

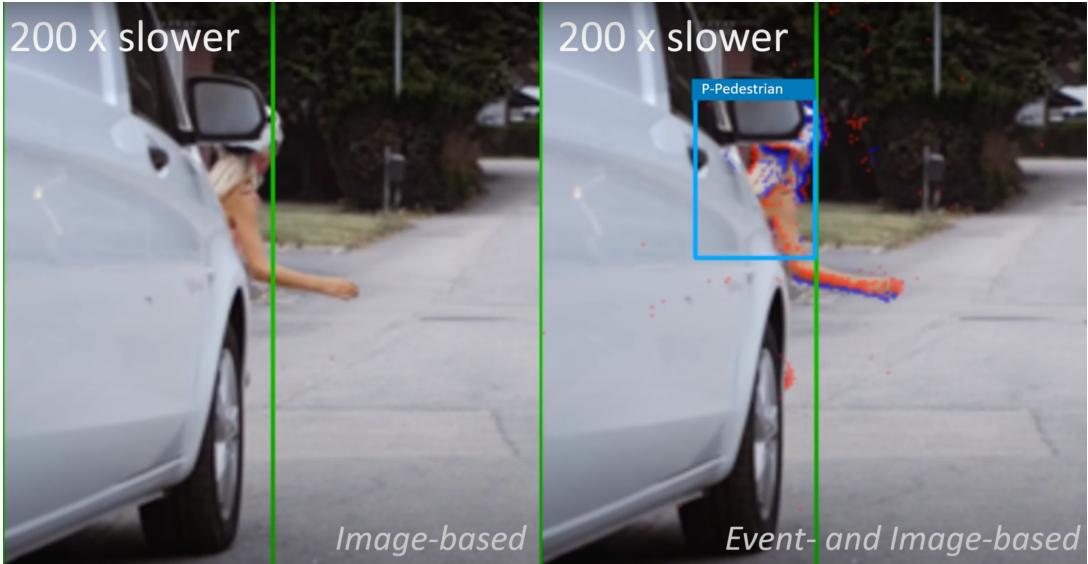


Figure 1.2: Pedestrian detection: image-based (left) vs event-based (right) [20].

Despite recent advances in spiking neural networks and neuromorphic hardware, relatively little systematic research has been conducted on the interplay between noise, encoding strategies, and classification accuracy. Most prior work has emphasised achieving state-of-the-art benchmarks under clean conditions rather than evaluating robustness to imperfections [30, 31].

The practical importance of robustness cannot be overstated. In autonomous systems, models must function reliably under imperfect sensory conditions, such as rain, motion blur, or sensor degradation. Similarly, in embedded IoT and mobile devices, constraints on bandwidth and power often lead to lossy transmission of data streams, further introducing artefacts that may affect SNN performance. By addressing this gap, the project aims to provide a more realistic understanding of how SNNs can be deployed in environments where imperfect data is the norm rather than the exception.

1.3 Aims and Objectives

The overall aim of this project is to investigate the robustness of Spiking Neural Networks (SNNs) when subjected to noisy event-based data generated from frame-based datasets, and to analyse how different model configurations respond to varying noise levels. The focus is on controlled experiments with the MNIST dataset, where noise is directly added to the images prior to spike encoding.

To achieve this aim, the following objectives are defined:

- Review existing approaches to training and applying SNNs with neuromorphic datasets [19, 21].

- Identify limitations in current methods, particularly the insufficient treatment of noisy inputs in robustness analysis [27, 16].
- Implement an experimental framework based on an existing SNN model architecture with STDP learning rules [12]. Figure 1.3 illustrates the architecture of the model, used for the robustness experiments in this report.

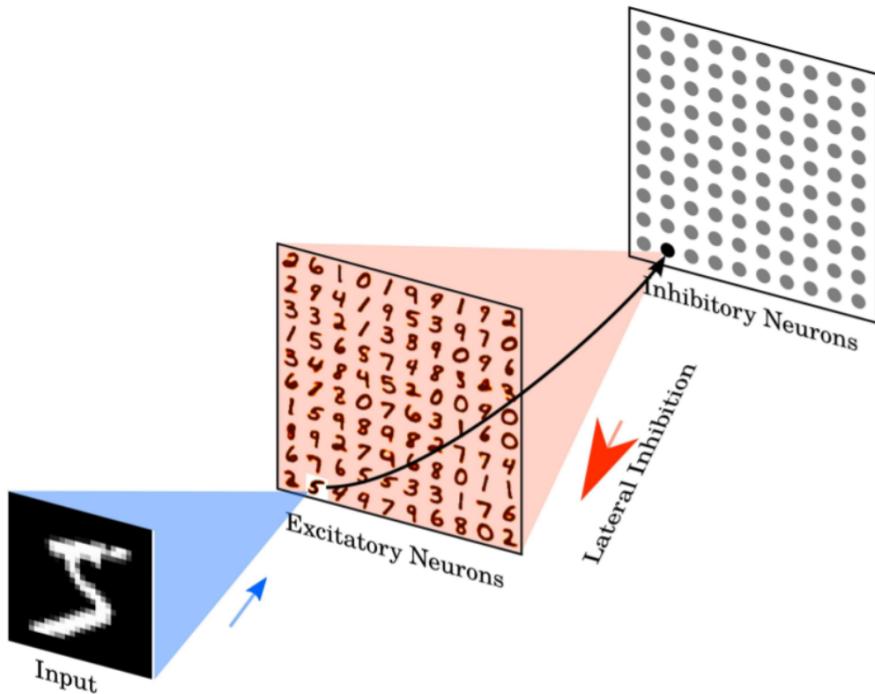


Figure 1.3: Spiking Neural Network architecture with excitatory and inhibitory populations connected via STDP synapses [12].

- Conduct controlled experiments by adding Gaussian noise at different signal-to-noise ratio (SNR) levels to the MNIST dataset prior to spike encoding.
- Evaluate three model configurations (STDP-based SNN with Random Forest, SNN feature extractor with Random Forest, and baseline Random Forest) under varying noise levels.
- Demonstrate and analyse the dependency of classification accuracy on noise levels across all models, providing quantitative evidence of robustness and sensitivity.

1.4 Contributions of this Work

The contributions of this dissertation are as follows:

- Use of an existing open-source SNN implementation (STDP-based MNIST classifier) as the experimental basis for robustness analysis.
- Addition of controlled Gaussian noise to MNIST images to simulate degraded input conditions at different signal-to-noise ratio (SNR) levels.
- Design of two experimental protocols: (i) training and testing on noisy data, and (ii) training on noisy data while testing on clean data, to investigate the effect of noise on generalisation.
- Evaluation of three model settings: (i) SNN with STDP and Random Forest, (ii) SNN as a fixed feature extractor with Random Forest, and (iii) a baseline Random Forest directly on image pixels.
- Empirical analysis of classification accuracy across noise levels, including accuracy–SNR curves.
- Analytical discussion of the observed results, highlighting the relative robustness of SNN-based methods compared to the baseline under increasing noise.

1.5 Summary

This chapter introduced the context of spiking neural networks and their relevance to event-based vision, with emphasis on the challenge of robustness to noise. The project aims to evaluate SNN performance under noisy MNIST inputs, focusing on experimental protocols, model comparison, and analysis. An existing SNN implementation was used to prioritise representative experiments and systematic robustness evaluation over low-level reimplementation. The next chapter presents a literature review, highlighting prior work and the gaps this study addresses.

CHAPTER 2

Literature Review

2.1 Introduction

Spiking Neural Networks (SNNs) are increasingly applied to event-based vision tasks due to their temporal precision and energy efficiency. However, robustness to noise remains a critical challenge. Noise arises naturally in neuromorphic sensors, including background events, illumination changes, and hardware variability. Existing research has addressed spike encoding, learning rules, hardware deployment, and dataset design, but rarely with systematic evaluation of robustness. This chapter surveys the field along five axes—encoding, training, datasets, hardware, and applications—and identifies how prior work leaves open questions about resilience to noise. Table 2.1 provides a structured overview.

2.2 Spike Encoding and Noise Sensitivity

Encoding strategies define how information is represented in spikes and directly affect noise robustness. Rate coding [15, 6, 21] demonstrated competitive accuracy but showed sensitivity to perturbations, as all spikes contribute equally to rate estimates. Latency and temporal encodings [28, 12] preserved timing information and improved efficiency, though their behaviour under sensor noise remains underexplored. Poisson-based encoding introduced stochasticity that can act as implicit regularisation, potentially improving robustness, but at the expense of reproducibility and determinism. *Gap: Existing evaluations rarely compare encodings explicitly under controlled noise, motivating the experiments in this dissertation.*

2.3 Training Approaches and Robustness

Learning rules strongly influence robustness. STDP [6] provided biologically inspired adaptation yet struggled with noisy or inconsistent patterns. Conversion-based methods [21, 22] achieved high accuracy under clean conditions but often degraded under variability in event streams. Surrogate gradient training [30, 31, 2] enabled deeper SNNs and better generalisation, but robustness to noisy DVS data has not been systematically tested. *Gap:* *No common training framework exists for directly contrasting these approaches in the presence of noise. This project addresses this gap by implementing three comparable pipelines.*

2.4 Neuromorphic Datasets and Noise

The Dynamic Vision Sensor (DVS) [14] introduced event-driven data by emitting asynchronous events that encode pixel location, timestamp, and polarity of luminance changes. Unlike conventional frame-based cameras, this architecture provides sparse and low-latency input streams, but it is inherently prone to noise from illumination flicker, background motion, and sensor artefacts. An illustration of a DVS device and its event-integration pipeline is shown in Figure 2.1, adapted from [29]. Benchmarks such as N-MNIST, CIFAR10-DVS, and DVS Gesture provided standardised evaluation, though most prior work reported results on clean subsets. Few studies explicitly analysed noise resilience in these datasets.

Gap: *Robustness is rarely treated as a first-class metric in benchmark reporting. The present study uses MNIST as a controlled setting to inject and quantify noise explicitly.*

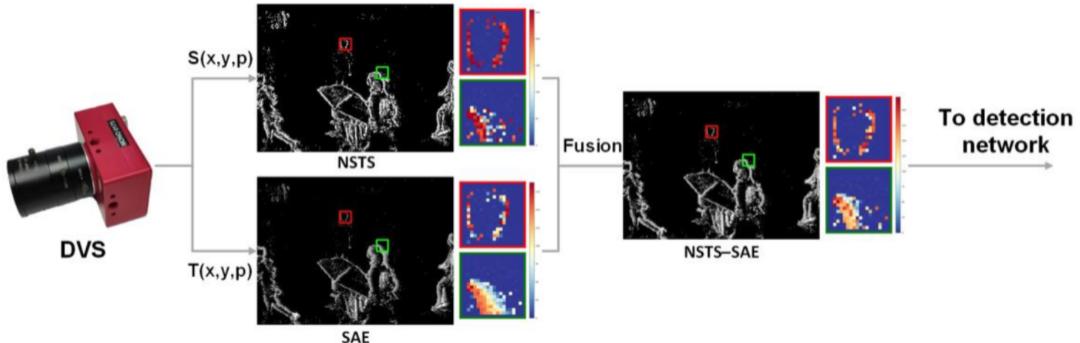


Figure 2.1: Illustration of a Dynamic Vision Sensor (DVS) and event integration pipeline. The DVS produces asynchronous events $S(x, y, p)$ and $T(x, y, p)$, which can be fused through methods such as NSTS and SAE¹[29].

¹NSTS (Neighborhood Suppression Time Surface) suppresses redundant events to emphasise high-contrast local changes, while SAE (Surface of Active Events) integrates recent events into a time-decaying map.

2.5 Neuromorphic Hardware

Neuromorphic processors such as IBM TrueNorth [1, 7], Intel Loihi [5], and SpiNNaker [8] demonstrated scalable SNN deployment. While these platforms offer architectural robustness through redundancy and low-precision computation, systematic evaluation of robustness is limited. Most reported results emphasised throughput or energy efficiency. *Gap: Hardware-level noise tolerance is not systematically characterised. By remaining hardware-agnostic, this dissertation focuses on algorithmic robustness independent of specific platforms.*

2.6 Applications and Robustness Considerations

Event-based recognition systems [18, 24] highlighted the feasibility of real-time processing. Handcrafted descriptors like HATS filtered some noise but lacked adaptability. Alternative asynchronous models such as Phased LSTM [17] addressed temporal irregularities outside the spiking paradigm. Overall, applications validated SNN potential yet rarely included controlled robustness studies. *Gap: Application-oriented papers emphasise performance in clean scenarios; robustness evaluation remains ad hoc.*

2.7 Summary and Research Gaps

Table 2.1 shows that research on SNNs has advanced across theory, learning rules, datasets, and hardware platforms. Strong results have been achieved in areas such as STDP, surrogate gradients, ANN→SNN conversion, and large-scale neuromorphic systems like DVS, SpiNNaker, TrueNorth, and Loihi. However, most studies focus on clean input data, with robustness to noisy event-based signals rarely evaluated in a systematic way.

Three primary gaps are evident:

1. Lack of systematic evaluation of SNN performance under varying noise conditions.
2. Limited integration of robust training strategies with neuromorphic datasets.
3. Minimal hardware-level or application-level studies explicitly targeting noise resilience.

The present work addresses these gaps by analysing the robustness of SNNs under noisy event-based inputs, providing empirical insights into the resilience of spike-based computation.

Table 2.1: Summary of Literature Review on Spiking Neural Networks and Neuromorphic Vision

Author(s)	Year	Method / Model	Key Findings	Gap / How Addressed
Stromatias et al. [28]	2017	Event-driven SNN classifier with histogram-based learning	Achieved 97.7% on N-MNIST; robust supervised SNN learning	Did not analyse robustness to sensor noise. <i>This work investigates noise effects explicitly</i>
Larionov [?]	2021	MNIST SNN with latency/Poisson encoding	Shows feasibility of SNN training on simple dataset	Prototype only, no DVS and no noise. <i>This work extends to noisy neuromorphic vision</i>
Maass [15]	1997	Theory of SNNs (third generation)	Higher computational power vs ANN/RNN	Purely theoretical, no robustness analysis. <i>This work applies theory to noisy vision data</i>
Pfeiffer & Pfeil [19]	2018	Survey of deep SNNs	Reviews STDP, surrogate gradients, ANN→SNN	Focus on learning rules, not robustness. <i>This work studies robustness empirically</i>
Lichtsteiner et al. [14]	2008	DVS hardware	First DVS: 120 dB, 15μs latency	Introduced sensor but not robustness tests. <i>This work evaluates SNNs under noisy DVS input</i>
Diehl & Cook [6]	2015	STDP unsupervised SNN	Digit recognition with STDP	Limited accuracy; no analysis under noise. <i>This work tests STDP-trained models on noisy data</i>

Continued on next page

Table 2.1 (continued)

Author(s)	Year	Method / Model	Key Findings	Gap / How Addressed
Rueckauer et al. [21]	2017	ANN→SNN conversion	Near-ANN accuracy with rate coding	Conversion ignores noise robustness. <i>This work examines noise sensitivity in converted models</i>
Stromatias et al. [27]	2017	Spiking DBN on neuromorphic HW	Robust to quantisation and low precision	Did not address input noise. <i>This work focuses on robustness to event noise</i>
Wu et al. [30]	2019	Surrogate gradients	Direct SNN training, high accuracy	Evaluated on clean data only. <i>This work applies surrogate training to noisy DVS</i>
Zenke & Ganguli [31]	2018	SuperSpike surrogate rule	Enables deep SNN training	No robustness tests. <i>This work extends evaluation to noisy inputs</i>
Bellec et al. [2]	2018	e-prop (online)	Alternative to BPTT	Did not study noise. <i>This work adapts e-prop to noisy DVS streams</i>
Neftci et al. [16]	2019	Surrogate gradient review	Theoretical foundation	No empirical robustness analysis. <i>This work provides empirical noise evaluation</i>
Amir et al. [1]	2017	IBM TrueNorth system	Ultra-low-power event pipeline	Hardware focus, robustness not tested. <i>This work evaluates noise effects independent of hardware</i>

Continued on next page

Table 2.1 (continued)

Author(s)	Year	Method / Model	Key Findings	Gap / How Addressed
Davies et al. [5]	2018	Intel Loihi	On-chip learning, scalable	HW-specific, limited robustness studies. <i>This work analyses robustness across platforms</i>
Orchard et al. [18]	2015	Event-based recognition	Efficient event-based vision	Datasets limited; no noise analysis. <i>This work applies modern SNNs under noise</i>
Neil & Liu [17]	2016	Phased LSTM	Efficient modeling	Not spiking; no robustness comparison. <i>This work evaluates SNNs as robust temporal models</i>
Sironi et al. [24]	2018	HATS descriptor	SOTA with time surfaces	Handcrafted, limited robustness tests. <i>This work compares end-to-end SNN robustness</i>
Shrestha & Orchard [23]	2018	SpikeNorm (ANN→SNN)	Better converted SNNs	Conversion only; no noise analysis. <i>This work trains SNNs for robustness</i>
Sengupta et al. [22]	2019	Deep converted SNNs	Near-SOTA large-scale	Static images; no noise robustness. <i>This work targets noisy DVS data</i>
Esser et al. [7]	2016	CNNs on TrueNorth	Scalable low-power	HW-bound; robustness not central. <i>This work evaluates noise-robust methods independently</i>

Continued on next page

Table 2.1 (continued)

Author(s)	Year	Method / Model	Key Findings	Gap / How Addressed
Furber et al. [8]	2014	SpiNNaker system	Real-time large SNN sim	HW design focus; no robustness study. <i>This work adds robustness analysis on noisy inputs</i>
Indiveri & Liu [10]	2015	Neuromorphic circuits (survey)	Energy-efficient circuits	Survey only; no noise evaluation. <i>This work empirically evaluates on noisy DVS</i>

2.8 Discussion of Literature Review Table

Table 2.1 provides more than a catalogue of prior work: it reveals patterns and trajectories across the SNN field. Looking horizontally across decades of research, one can trace a chronological shift from early theoretical claims of computational power [15] towards hardware prototypes [14, 8], conversion pipelines [21], and, most recently, surrogate gradient methods [30, 31]. Each wave of research brought genuine progress, yet the “Gap” column makes it clear that robustness has remained peripheral throughout.

A historical view highlights how different research communities approached the same challenge. In the late 1990s, theoretical studies positioned SNNs as a more expressive model class, but did not attempt to quantify robustness. In the 2000s, the emergence of neuromorphic sensors and early plasticity rules gave the first demonstrations of event-driven processing, yet noise was largely treated as an inconvenience rather than a scientific variable. The 2010s saw the rise of deep SNNs through conversion from ANNs and handcrafted event descriptors. Here, accuracy metrics became dominant, reflecting the influence of mainstream computer vision benchmarks, while robustness metrics were absent. By the end of the decade, surrogate gradient learning and new hardware platforms such as Loihi and SpiNNaker broadened the scope and scale of SNNs, but still without systematic evaluation under noisy inputs.

Another insight from Table 2.1 is the convergence of different subfields on the same blind spot. Hardware-oriented papers emphasised throughput and energy savings; algorithmic work emphasised training efficiency or biological plausibility; application-driven papers emphasised feasibility. Yet in every case, the gap column records the same omission: robustness was rarely the central research question. The consistency of this pattern across categories suggests that the field has normalised clean-data benchmarks as sufficient, even though real-world neuromorphic sensors are unavoidably noisy.

Taken together, the entries in Table 2.1 highlight both the diversity of approaches

and the uniformity of what is missing. Progress has been incremental and multi-dimensional—new rules, new architectures, new chips, new datasets—but resilience to degraded input has not been systematically interrogated. This observation motivates the present dissertation: by treating robustness not as a side note but as a primary evaluation axis, the work contributes a dimension of empirical evidence that is absent from prior literature.

Chapter Summary

This chapter provided the theoretical background and reviewed related work on spiking neural networks, neural coding schemes, synaptic plasticity rules, and robustness under noise. The discussion identified current research gaps and positioned the contribution of this study within the broader literature.

The remainder of the dissertation is organised as follows. Chapter 3 presents the adopted methodology, including dataset preparation, spike encoding strategies, network configuration, noise models, and the experimental protocols. Chapter 4 reports and analyses the experimental results, with a particular focus on how classification accuracy changes under varying noise conditions across different model configurations. Finally, Chapter 5 concludes the dissertation, summarising the main findings, discussing limitations, and outlining potential directions for future research.

CHAPTER 3

Methodology

This chapter describes the methodology adopted to investigate the robustness of Spiking Neural Networks (SNNs) under noisy event-based data. The focus is on controlled experiments with the MNIST dataset, where Gaussian noise was added directly to the images before spike encoding. The methodology outlines dataset preparation, preprocessing, spike encoding, model configurations, experimental protocols, and evaluation procedure. Illustrations are included to clarify the workflow.

3.1 Overview of Methodology

The overall workflow of the experimental study is shown in Figure 3.1. It consists of five main stages:

1. Preparation of the MNIST dataset, including selection of three classes (digits 0, 1, and 8) and scaling of pixel intensities.
2. Encoding of images into Poisson-distributed spike trains to serve as input to the SNN.
3. Configuration of three model setups: (i) an SNN with Spike-Timing Dependent Plasticity (STDP) and a Random Forest classifier, (ii) an SNN as a fixed feature extractor combined with Random Forest, and (iii) a baseline Random Forest applied directly to image pixels.
4. Injection of Gaussian noise at different signal-to-noise ratio (SNR)¹ levels to simulate degraded inputs.

¹SNR is defined as $10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right)$, where P_{signal} and P_{noise} are the average power of the signal and the noise, respectively.

5. Evaluation of all models under two experimental protocols: training and testing on noisy data, and training on noisy data while testing on clean data.

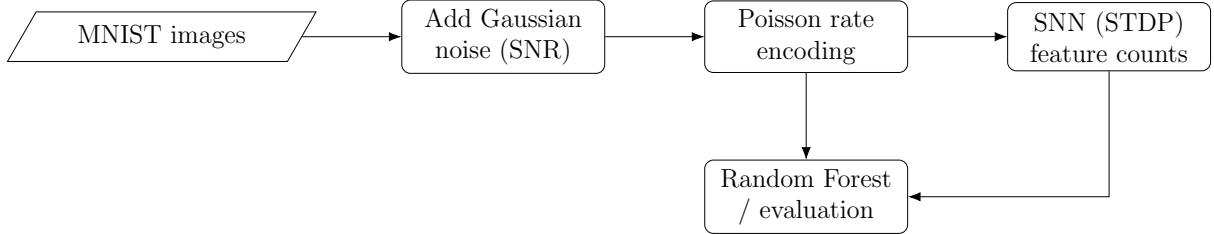


Figure 3.1: Workflow: MNIST \rightarrow noise (SNR) \rightarrow Poisson encoding \rightarrow Random Forest (direct) or via SNN features.

3.1.1 Random Forest classifier

Random Forest (RF) is a widely used ensemble learning method introduced by Breiman [4]. It combines the predictions of multiple decision trees, each trained on a bootstrap sample of the dataset and using a random subset of input features at each split. This randomness reduces correlation between individual trees and mitigates overfitting, while majority voting (for classification) or averaging (for regression) aggregates their outputs into a final prediction (Figure 3.2).

In this work, the RF serves as a downstream classifier on either raw image pixels or features extracted from the spiking neural network (SNN). The RF is configured with a maximum tree depth of 4, balancing computational efficiency with predictive performance.

3.2 Dataset

The *MNIST* handwritten digits dataset was used [13, 12]. The task was restricted to a ternary subset with labels $\{0, 1, 8\}$, selecting only those images from both the original training and test partitions. Image size is 28×28 grayscale.

3.2.1 Poisson rate coding

Let $I_{ij} \in [0, 1]$ denote the normalised pixel intensity at location (i, j) and T the simulation window. Under Poisson rate coding, each pixel drives one input neuron that emits spikes according to an independent homogeneous Poisson process with rate

$$r_{ij} = \lambda I_{ij} \quad [\text{Hz}]. \quad (3.1)$$

For a time bin of width Δt , the spike count in that bin satisfies $k \sim \text{Poisson}(r_{ij}\Delta t)$. In discrete-time simulators this is implemented equivalently as a Bernoulli trial per bin,

$$S_{ij}(t) \sim \text{Bernoulli}(1 - e^{-r_{ij}\Delta t}), \quad t = 1, \dots, T/\Delta t, \quad (3.2)$$

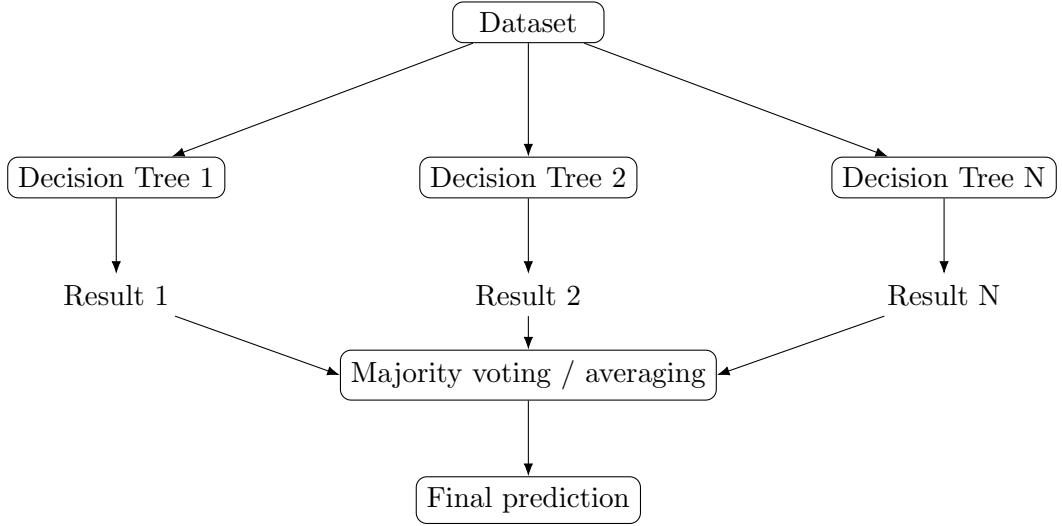


Figure 3.2: Random Forest as an ensemble of decision trees with aggregation by majority voting (classification) or averaging (regression).

with at most one spike per bin. The simulation step was set to $\Delta t = 1$ ms. During the *active* presentation window the rate r_{ij} is held constant for each input pixel; during the *passive* window the input is silenced ($r_{ij} = 0$). In practice, spike trains were generated using a `PoissonGroup` in Brian2, which realises independent Poisson sources per input neuron consistent with the Bernoulli formulation above.

3.2.2 Preprocessing and rate scaling

Raw MNIST intensities $X_{ij} \in [0, 255]$ were linearly mapped to firing rates by a global scaling factor of 1/4, i.e.

$$r_{ij} = \frac{X_{ij}}{4} \text{ [Hz]},$$

so that a saturated pixel ($X_{ij} = 255$) corresponds to approximately 63 Hz. Equivalently, if $I_{ij} = X_{ij}/255$ denotes the normalised intensity, this mapping corresponds to choosing $\lambda \approx 63.75$ Hz in $r_{ij} = \lambda I_{ij}$. The resulting rate map was then used as the Poisson parameter during the *active* window of each image; typical digits yield average input firing rates within the 20–50 Hz range.

3.2.3 Filtered dataset size

After restricting the MNIST dataset to digits {0, 1, 8}, the resulting dataset contained 18,516 training images and 3,089 test images, each of size 28×28 pixels.

3.2.4 Implementation note

All spiking simulations were performed in Brian2 [26]; for computational efficiency, Cython code generation was enabled (`prefs.codegen.target = cython`).

3.3 Model Configuration

The spiking architecture was implemented in Brian2 [26, 12] with `prefs.codegen.target=cython`. The network consists of a Poisson input layer, one excitatory population (EG), and one inhibitory population (IG), coupled via feedforward plastic synapses and lateral inhibition.

3.3.1 Populations

- **Input (PG):** PoissonGroup with $n_{\text{input}} = 28 \times 28$ independent generators. Pixel intensities (rescaled to Hz) provide the rate for each generator.
- **Excitatory (EG):** $n_e = 100$ leaky integrate-and-fire neurons with conductance variables v , g_e , g_i . Dynamics were simulated using Euler integration with specified threshold, reset, and refractory mechanisms.
- **Inhibitory (IG):** $n_i = 100$ leaky integrate-and-fire neurons receiving excitation from EG and providing global (all-except-self) inhibition back to EG.

3.3.2 Neuron dynamics

Excitatory neurons follow:

$$\frac{dv}{dt} = \frac{g_e(0 \text{ mV} - v) + g_i(-100 \text{ mV} - v) + (v_{\text{rest}}^{(e)} - v)}{100 \text{ ms}}, \quad \frac{dg_e}{dt} = -\frac{g_e}{5 \text{ ms}}, \quad \frac{dg_i}{dt} = -\frac{g_i}{10 \text{ ms}}, \quad (3.3)$$

with threshold $v > v_{\text{th}}^{(e)}$, reset $v \leftarrow v_{\text{reset}}^{(e)}$, and refractory 5 ms. Inhibitory neurons:

$$\frac{dv}{dt} = \frac{g_e(0 \text{ mV} - v) + (v_{\text{rest}}^{(i)} - v)}{10 \text{ ms}}, \quad \frac{dg_e}{dt} = -\frac{g_e}{5 \text{ ms}}, \quad (3.4)$$

with threshold $v > v_{\text{th}}^{(i)}$, reset $v \leftarrow v_{\text{reset}}^{(i)}$, refractory 2 ms.

3.3.3 Connectivity

- **PG→EG (Synapses S1):** all-to-all plastic synapses with pair-based STDP (Sec. 3.3.4), initialised $w \sim \mathcal{U}(0, g_{\text{max}})$.
- **EG→IG (S2):** one-to-one, fixed excitatory weights $w = 3$, delays $\sim \mathcal{U}(0, 10 \text{ ms})$.
- **IG→EG (S3):** all-except-self inhibition, fixed $w = 0.03$, delays $\sim \mathcal{U}(0, 5 \text{ ms})$.

3.3.4 Spike-Timing Dependent Plasticity (STDP)

Spike-Timing Dependent Plasticity is a biologically inspired Hebbian learning rule that modifies synaptic weights according to the relative timing of pre- and post-synaptic spikes [3, 25]. If a pre-synaptic spike occurs shortly before a post-synaptic spike, the synapse is potentiated (long-term potentiation, LTP). Conversely, if the post-synaptic spike occurs first, the synapse is weakened (long-term depression, LTD). This mechanism enables unsupervised feature learning in spiking neural networks.

In the present model, each PG→EG synapse maintains pre- and post-synaptic traces A_{pre} , A_{post} :

$$\frac{dA_{\text{pre}}}{dt} = -\frac{A_{\text{pre}}}{\tau_{\text{pre}}}, \quad \frac{dA_{\text{post}}}{dt} = -\frac{A_{\text{post}}}{\tau_{\text{post}}}, \quad (3.5)$$

with $\tau_{\text{pre}} = \tau_{\text{post}} = 20$ ms. On a pre-spike: $g_e += w$, $A_{\text{pre}} += dA_{\text{pre}}$, and $w \leftarrow \text{clip}(w + \text{lr} \cdot A_{\text{post}}, 0, g_{\max})$. On a post-spike: $A_{\text{post}} += dA_{\text{post}}$ and $w \leftarrow \text{clip}(w + \text{lr} \cdot A_{\text{pre}}, 0, g_{\max})$. Here $g_{\max} = 0.05$, $dA_{\text{pre}} = 0.01 \cdot g_{\max}$, $dA_{\text{post}} = -dA_{\text{pre}} \cdot \frac{\tau_{\text{pre}}}{\tau_{\text{post}}} \cdot 1.05$. Plasticity is controlled by a shared learning-rate parameter `lr` (enabled during training, disabled at evaluation).

3.3.5 Spike raster under STDP

Figure 3.3 shows the spiking activity of the excitatory (red) and inhibitory (blue) populations during training. The horizontal axis represents simulation time in milliseconds, while the vertical axis indicates neuron indices. Each dot corresponds to a single spike generated by a neuron at the given time. Vertical bands correspond to bursts of spikes in subsets of neurons, reflecting the dynamics of excitation and inhibition as synaptic weights are modified.

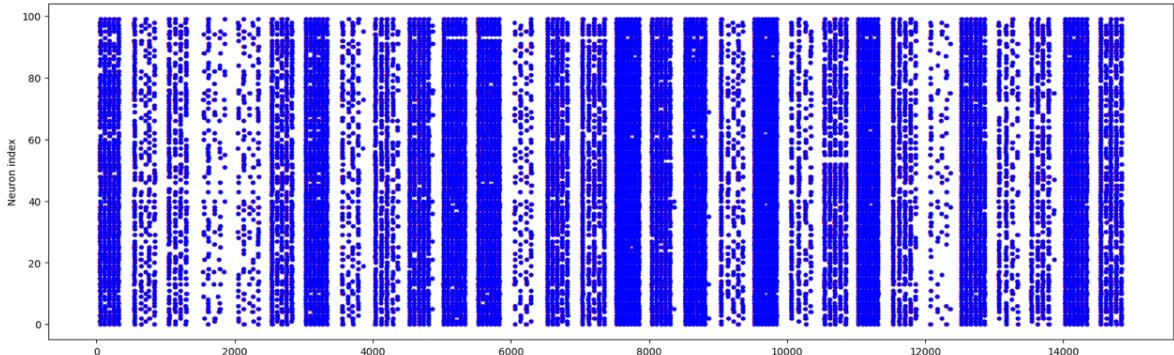


Figure 3.3: Spike raster during STDP training, illustrating excitatory and inhibitory activity under the plasticity rule [12].

3.3.6 Simulation protocol and hyperparameters

Each input image was presented for an *active* period of 0.35 s, followed by a *passive* period of 0.15 s with no input. During training, plasticity was enabled (`lr=1`), whereas during

evaluation it was disabled ($l_r=0$). A summary of all model parameters is provided in Table 3.1.

Table 3.1: Key model and simulation hyperparameters.

Input neurons	$n_{\text{input}} = 784$ (Poisson)
Excitatory / Inhibitory	$n_e = n_i = 100$
$v_{\text{rest}}^{(e)}, v_{\text{reset}}^{(e)}, v_{\text{th}}^{(e)}$	-60, -65, -52 mV
$v_{\text{rest}}^{(i)}, v_{\text{reset}}^{(i)}, v_{\text{th}}^{(i)}$	-60, -45, -40 mV
Time constants (EG)	$\tau_v = 100$ ms, $\tau_{g_e} = 5$ ms, $\tau_{g_i} = 10$ ms
Time constants (IG)	$\tau_v = 10$ ms, $\tau_{g_e} = 5$ ms
STDP	$\tau_{\text{pre}} = \tau_{\text{post}} = 20$ ms; $g_{\max} = 0.05$
Delays	EG→IG: [0, 10] ms; IG→EG: [0, 5] ms
Presentation	Active 0.35 s + Passive 0.15 s per image
Features	EG spike counts (100-dim)

3.3.7 Gaussian noise model (SNR)

To emulate sensor corruption, we inject additive white Gaussian noise with a prescribed signal-to-noise ratio (SNR). For an input image \mathbf{x} , the noisy sample is obtained as

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}, \quad \mathbf{n} \sim \mathcal{N}(0, \sigma^2),$$

where the noise variance σ^2 is derived from the average signal power and the target SNR (dB).

Figure 3.4 illustrates typical MNIST digits under increasing noise levels (SNR = 10, 5, 0 dB).

3.3.8 Experiment Types and Scenarios

To evaluate robustness, three complementary experiment types are defined:

- **Type A (SNN + STDP + RF).** The spiking neural network (SNN) is trained for one epoch with spike-timing dependent plasticity (STDP) enabled, producing feature representations. A Random Forest classifier is then trained on these features.
- **Type B (SNN frozen + RF).** The SNN is used as a fixed feature extractor without further STDP training. A Random Forest classifier is trained on the extracted features.
- **Type C (RF baseline).** A Random Forest classifier is trained directly on flattened image pixels, without an SNN feature extraction stage. This provides a direct baseline for comparison with SNN-based methods.

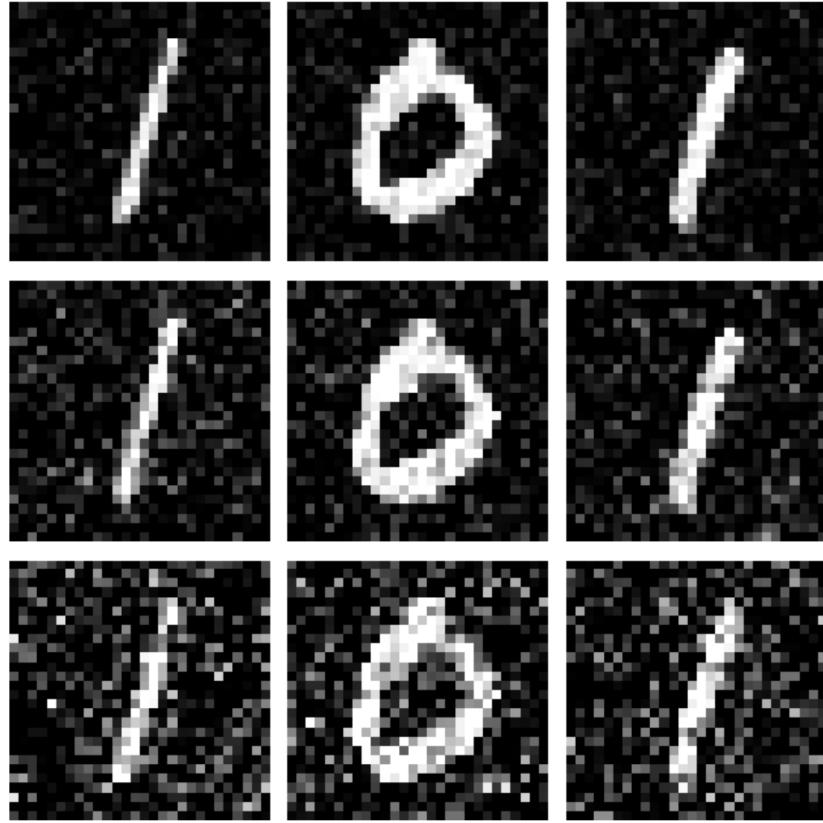


Figure 3.4: Examples of noisy MNIST images at different SNR levels (10, 5, and 0 dB).

Each experiment is conducted using 1000 training samples, 1000 samples for feature assignment, and 500 evaluation samples from the MNIST dataset. The SNN is trained for one epoch (when enabled), and the downstream classifier is a Random Forest with maximum depth 4.

These experiment types are applied under two scenarios:

1. **Noisy training and noisy testing:** both training and testing data are augmented with noise.
2. **Noisy training and clean testing:** the classifier is trained on noisy data but evaluated on clean test data, in order to assess robustness under mismatched conditions.

For both scenarios, experiments are repeated across signal-to-noise ratio values $\{40, 30, 20, 10, 5, 0\}$ dB. Classification accuracy is reported separately for each setting, and the results are visualised as accuracy–SNR curves.

3.4 Evaluation metric

Model performance is measured in terms of classification accuracy, defined as the proportion of correctly predicted labels on the test set:

$$\text{Acc} = \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{\hat{y}_n = y_n\}.$$

All reported results are based on this accuracy metric under different noise conditions.

3.5 Summary

This chapter described the methodology used to evaluate the robustness of spiking neural networks under noisy conditions. The input encoding relied on Poisson rate coding with linear preprocessing of pixel intensities. Excitatory and inhibitory neurons were modelled as leaky integrate-and-fire units with conductance-based synapses and STDP plasticity. The network configuration, simulation protocol, and hyperparameters were specified in detail. Three experiment types were defined: (A) SNN with STDP and Random Forest, (B) frozen SNN features with Random Forest, and (C) Random Forest baseline on raw pixels. Each type was applied under two scenarios, with noise injected in both training and testing, or in training only. Evaluation was performed using classification accuracy on MNIST across multiple noise levels. The next chapter presents experimental results and their analysis.

CHAPTER 4

Results and Analysis

This chapter presents the experimental results obtained using the methodology from Chapter 3. The focus is on classification accuracy under varying noise conditions for three experiment types (A–C) and two training–testing settings. Before presenting accuracy–SNR curves, Table 4.1 summarises the experimental configurations (A–C) and the two conditions N+N¹ and N+C², providing a concise reference for the model pipelines and data setups.

Table 4.1: Experimental scenarios used in Chapter 4. For each experiment type (A–C), results are reported under two settings (N+N, N+C).

Type	Model pipeline	Training / Testing condition
A	SNN with STDP feature extraction + RF	Noisy spike trains (Poisson-encoded MNIST with Gaussian noise). N+N: noisy test spike trains N+C: clean test spike trains
B	SNN (weights frozen; no STDP) + RF, features from spiking activity	Noisy spike trains (same noise protocol as A). N+N: noisy test spike trains N+C: clean test spike trains
C	RF only (no SNN front-end), features from non-spiking inputs	Noisy features derived from MNIST. N+N: noisy test features N+C: clean test features

¹Training and testing on noisy data.

²Training on noisy data, testing on clean data.

4.1 Gaussian noise: noisy training and noisy testing

The first group of results considers the case where Gaussian noise is applied consistently to both training and testing data (*noisy-noisy* scenario). This setting provides a direct assessment of how well the models can adapt when exposed to noisy inputs throughout both phases. Classification accuracy across different SNR levels is reported below.

4.1.1 Results for Experiment Type A (SNN + STDP + RF)

The first set of results corresponds to Experiment Type A, where the spiking neural network is trained for one epoch with STDP enabled and a Random Forest classifier is trained on the extracted features. Figure 4.1 presents classification accuracy for different signal-to-noise ratios in the range 40–0 dB. Accuracy is high at weak noise levels (above 0.92 for 40 dB and 20 dB), with a slight non-monotonic variation between 30 and 20 dB. A clear degradation is observed as noise increases, with accuracy dropping to 0.748 at 0 dB.

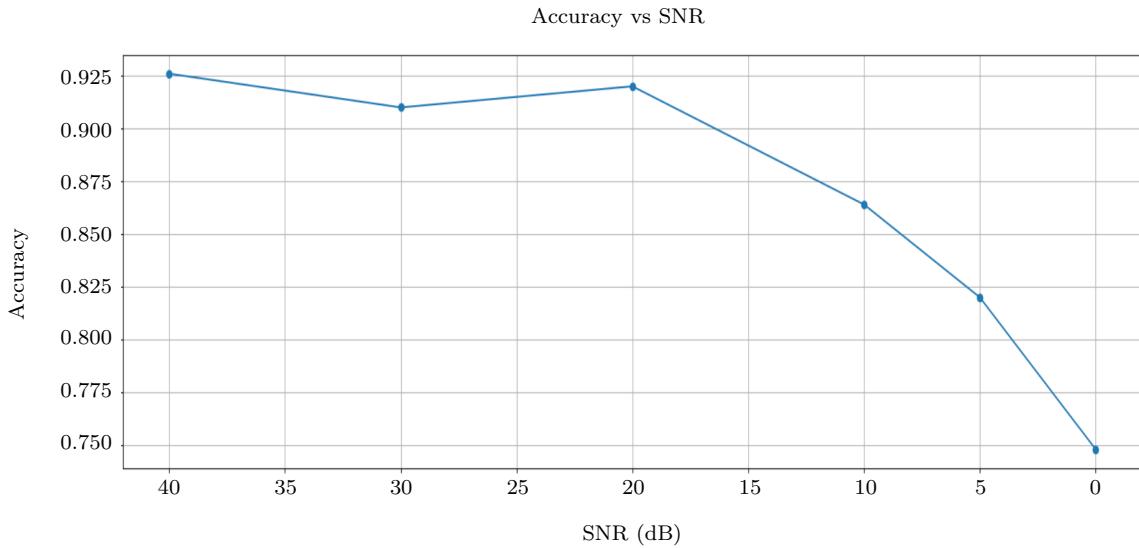


Figure 4.1: Classification accuracy of Experiment Type A (SNN + STDP + RF) across SNR levels under Gaussian noise.

Analysis. Accuracy remains high at 40–20 dB (around 0.92), but declines notably at stronger noise levels, reaching 0.748 at 0 dB. The curve is not strictly monotonic, with performance at 20 dB slightly higher than at 30 dB. A plausible explanation is that moderate noise acts as a regulariser on the limited training set (1000 samples), helping STDP avoid overfitting and thus improving generalisation at 20 dB. At stronger noise (10–0 dB), however, class-discriminative details in the images are largely destroyed. Because Poisson encoding adds additional stochasticity and spike counts compress temporal information,

the features lose reliability, causing the downstream Random Forest to misclassify more frequently.

4.1.2 Results for Experiment Types B and C

Figure 4.2 compares the performance of Experiment Type B (SNN frozen + RF) and Type C (RF baseline) under Gaussian noise applied during both training and testing. Accuracy is reported for SNR values from 40 dB to 0 dB.

The Random Forest baseline (Type C) achieves consistently high performance, remaining above 0.94 even at 0 dB. By contrast, the frozen SNN feature extractor (Type B) exhibits substantially lower accuracy, with performance degrading from 0.85 at 40 dB to 0.66 at 0 dB. This indicates that, without synaptic plasticity, the learned SNN features are less robust to additive noise than raw-pixel features used directly by the Random Forest classifier.

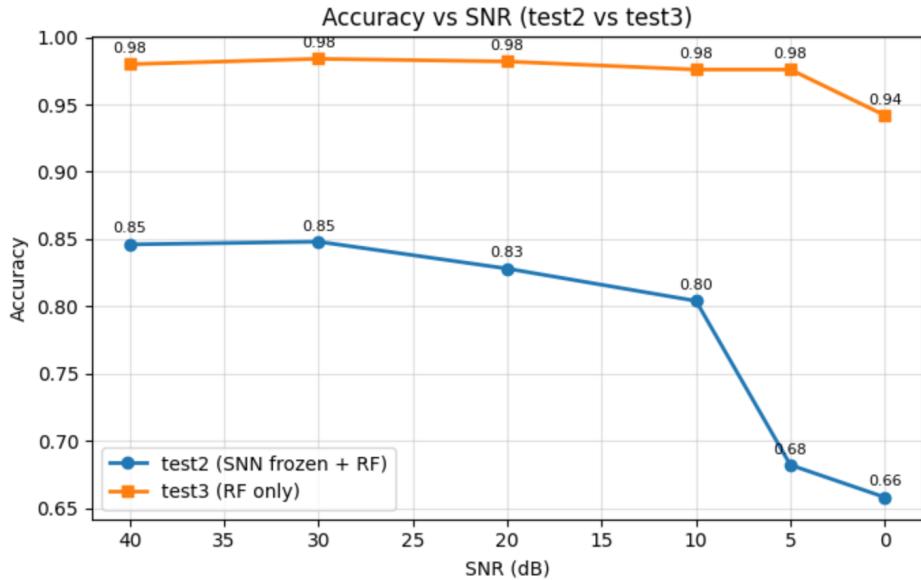


Figure 4.2: Experiment Types B (SNN frozen + RF) and C (RF baseline): classification accuracy across SNR levels under Gaussian noise (N+N scenario).

Analysis. Figure 4.2 compares Experiment Types B and C under the N+N scenario. The Random Forest baseline (Type C) shows almost noise-invariant performance, maintaining accuracy close to 0.98 for 40–10 dB and still above 0.94 at 0 dB. This robustness arises because the classifier directly exploits pixel-level features, and with noise present in both training and testing, the model adapts its decision boundaries to the same distribution. The ensemble nature of Random Forest further mitigates local perturbations by averaging across trees. In contrast, the frozen SNN feature extractor (Type B) suffers a gradual but substantial decline, from 0.85 at 40–30 dB to 0.66 at 0 dB. Without synaptic plasticity, its spike-based features cannot adapt to the statistics of noisy inputs.

Moreover, Poisson encoding transforms pixel-level perturbations into variability in spike counts, amplifying noise and weakening class separability. Thus, while Type C benefits from direct access to raw inputs and ensemble averaging, Type B illustrates how fixed spike-based representations can become fragile under additive Gaussian noise.

Comparative summary. In the N+N scenario, the Random Forest baseline (Type C) achieves the best robustness because it can directly learn from noisy pixels and stabilise predictions through ensemble averaging. The SNN with STDP (Type A) shows reasonable resilience at moderate noise, but strong noise overwhelms the spike-based representation, leading to accuracy loss. The frozen SNN (Type B) is most vulnerable: without adaptive plasticity, its spike features simply mirror noise distortions from the input, producing weak and unstable representations. Overall, these results highlight the importance of both adaptation mechanisms (STDP) and direct feature availability (pixels) for robust performance under noise.

4.2 Gaussian noise: noisy training and clean testing (N+C)

The second group of results corresponds to the scenario where Gaussian noise is applied during training but evaluation is performed on clean test data. This setting highlights the ability of the models to generalise from noisy inputs to noise-free conditions, providing a complementary view of robustness.

4.2.1 Results for Experiment Type A (SNN + STDP + RF)

Figure 4.3 shows classification accuracy across SNR levels for Experiment Type A under the noisy-clean (N+C) scenario. Accuracy remains above 0.92 at high SNR values (40–30 dB) but degrades substantially as the training noise increases, dropping to 0.396 at 0 dB.

Analysis. In the N+C scenario, accuracy for Experiment Type A declines sharply as the noise level in training increases: from above 0.92 at 40–30 dB to only 0.40 at 0 dB. Unlike the N+N case, here the model must generalise from noisy training data to clean evaluation samples. This domain mismatch causes a severe drop in performance once the training noise becomes strong. At high SNR, the features extracted by the SNN still capture relevant structures, and the Random Forest can map them to clean test labels. However, when training inputs are heavily corrupted, STDP adapts synaptic weights to noise-specific patterns rather than class-defining features. As a result, the downstream classifier receives distorted representations that do not align with the clean test distribution. This explains the steep loss of accuracy and highlights the sensitivity of spike-based representations to mismatched training and testing domains.

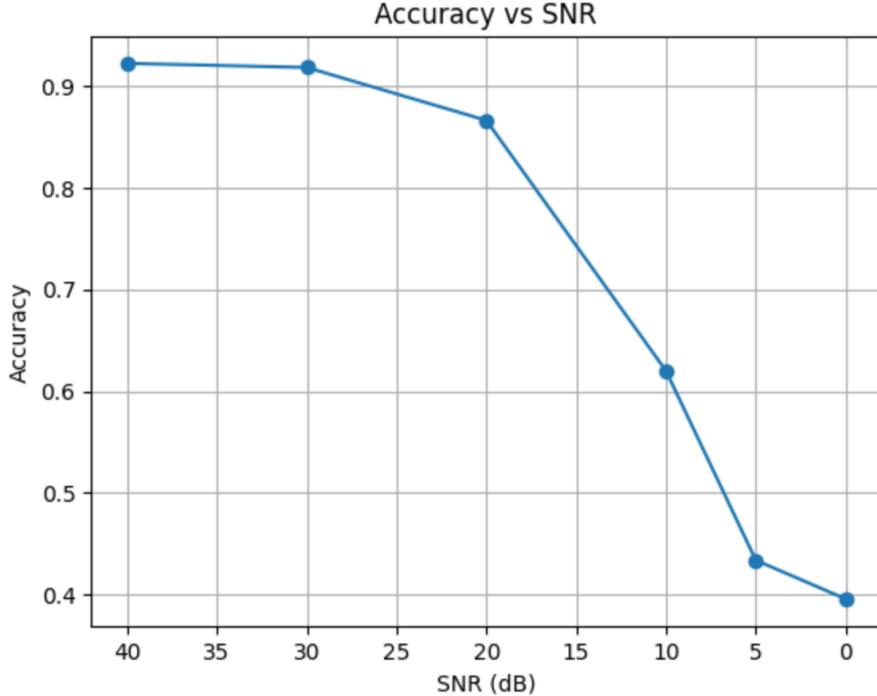


Figure 4.3: Experiment Type A (SNN + STDP + RF): classification accuracy across SNR levels under Gaussian noise (N+C scenario).

4.2.2 Results for Experiment Types B and C (SNN frozen + RF, RF baseline)

Figure 4.4 compares accuracy for Experiment Types B and C under the N+C scenario. The Random Forest baseline (Type C) maintains consistently high performance, remaining above 0.94 across all noise levels. In contrast, Type B shows strong degradation as training noise increases, with accuracy falling from 0.880 at 40 dB to 0.408 at 0 dB. This highlights the limited robustness of frozen SNN features when transferred to clean evaluation data.

Analysis. Figure 4.4 compares Experiment Types B and C under the N+C scenario. The Random Forest baseline (Type C) again maintains almost flat performance, with accuracy close to 0.98 across most SNR levels and only a small decline to 0.946 at 0 dB. This indicates that training on noisy pixels does not hinder generalisation to clean test inputs, since the decision boundaries learned by the ensemble remain well aligned with class-relevant structures. In contrast, the frozen SNN (Type B) shows a pronounced degradation: accuracy drops from 0.88 at 40 dB to 0.41 at 0 dB. Because its spike-based features cannot adapt, the synaptic representation learned under strong training noise encodes distortions rather than useful signal. When applied to clean evaluation data, these distorted features no longer match class structure, producing a sharp mismatch

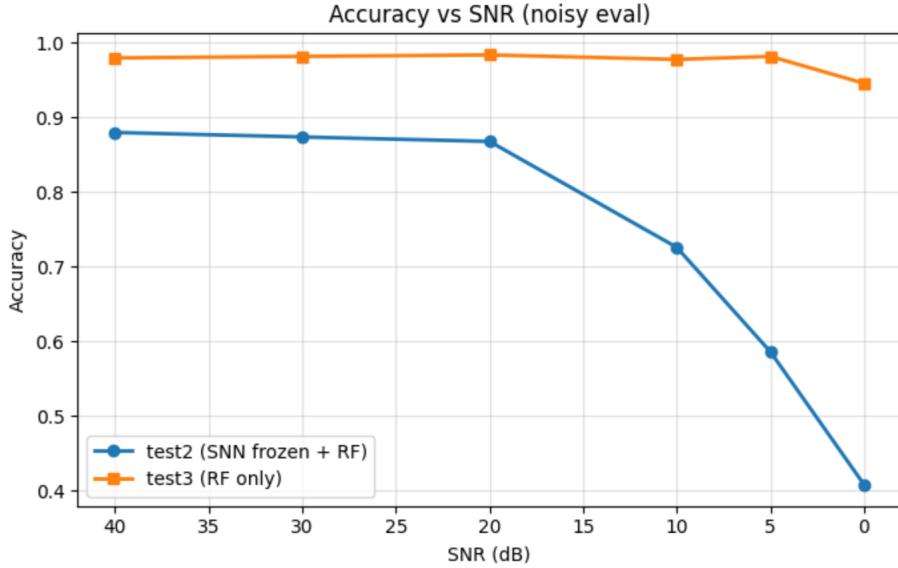


Figure 4.4: Experiment Types B (SNN frozen + RF) and C (RF baseline): classification accuracy across SNR levels under Gaussian noise (N+C scenario).

and significant loss in accuracy.

4.2.3 Comparative analysis: N+N vs N+C

A key difference emerges when comparing the two scenarios. For the Random Forest baseline (Type C), performance remains almost unchanged between N+N and N+C. In both cases, accuracy stays very high across all SNRs, indicating that decision trees trained on noisy pixels can generalise equally well to noisy or clean test sets. This robustness stems from two properties: (i) the raw pixel space preserves discriminative information even under noise, and (ii) ensemble averaging across trees smooths over noise-specific variations, leading to noise-invariant decision boundaries.

For the SNN-based approaches (Types A and B), the difference between N+N and N+C is substantial. In the N+N case, performance declines with noise but remains moderate, because the same corrupted distribution is seen in both training and testing, and the downstream classifier can still exploit consistent patterns. In the N+C case, however, a domain mismatch arises: the SNN learns weights or produces features shaped by noisy data, but these features no longer align with the structure of clean test inputs. As a result, the extracted spike patterns become misleading, and classification accuracy drops sharply.

Overall, the results suggest that Random Forest gains from direct pixel-level access and noise-robust averaging, whereas spike-based representations remain sensitive to mismatched training and testing domains. STDP aids under matched conditions (N+N), but without domain generalisation, SNN features remain fragile when noise differs between training and testing.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

This work investigated the robustness of spiking neural networks (SNNs) under noisy event-based data, focusing on synthetic perturbations applied to the MNIST dataset. The motivation arose from the increasing role of neuromorphic vision in real-world applications, where imperfect sensory data is the norm rather than the exception.

A systematic methodology was developed comprising dataset preparation, Poisson encoding strategies, three model configurations (STDP-based SNN with Random Forest, SNN as feature extractor with Random Forest, and a baseline Random Forest), and controlled Gaussian noise injection. Robustness was evaluated by adding noise directly to MNIST images prior to spike encoding.

The experimental results led to several key findings:

- **SNNs with STDP** showed improved robustness in some settings, particularly when noise was moderate, compared to frozen-feature SNNs. However, their performance under strong Gaussian noise was less stable than the Random Forest baseline.
- **Poisson encoding** provided stable performance by exploiting temporal redundancy, though stochasticity also amplified variability under strong perturbations.
- **Gaussian background noise** proved to be highly disruptive, strongly affecting classification accuracy across all configurations. Unlike SNNs, the Random Forest baseline maintained consistently high accuracy across all noise levels.

Overall, the report contributes to understanding the interplay between noise and classification performance in SNNs. The study emphasises that robustness, rather than

clean accuracy alone, should serve as a primary benchmark for evaluating neuromorphic systems intended for deployment in safety-critical or resource-constrained environments.

5.2 Limitations

Several limitations must be acknowledged:

- The experiments were restricted to MNIST with synthetic noise added directly to image pixels. Other types of perturbations such as event drop or temporal jitter, while discussed conceptually, were not fully implemented in the final evaluation.
- Only pair-based STDP and basic feature-extraction pipelines were explored. More advanced learning rules (e.g., e-prop or multi-compartment neuron models) could improve feature quality and adaptability.
- The evaluation focused on Random Forests as classifiers. Other classifiers (e.g., spiking CNNs, graph neural networks) may further enhance performance and leverage spike-based features more effectively.
- Computational efficiency metrics were limited. Real-time deployment on neuromorphic hardware (Loihi, SpiNNaker, TrueNorth) was not directly tested.

5.3 Future Work

Based on the above limitations, several directions for future research are identified:

- **Evaluation on real neuromorphic datasets:** Extending the study to datasets such as N-MNIST [18], N-CARS [24], and DVS Gesture to provide more realistic validation.
- **Extended noise models:** Implementing event drop and temporal jitter perturbations in addition to Gaussian noise, in order to approximate real sensor variability more closely.
- **Advanced encoding schemes:** Investigating alternative temporal codes (e.g., burst coding, phase coding, population coding) and compression methods tailored to neuromorphic sensors.
- **Improved learning rules:** Incorporating biologically plausible yet scalable algorithms such as e-prop [2] or surrogate gradient descent [30] for end-to-end training of deep SNNs.
- **Hardware deployment:** Benchmarking robustness on neuromorphic hardware platforms (Intel Loihi [5], SpiNNaker [8]) to assess energy efficiency and real-time performance.

5.4 Final Remarks

This work has shown that spiking neural networks offer valuable insights into the design of robust neuromorphic systems when subjected to noisy event-based data. While Random Forests on raw pixels demonstrated superior stability under Gaussian perturbations, SNNs with STDP revealed both strengths and vulnerabilities, illustrating the nuanced trade-offs between biological realism and engineering performance.

The integration of biologically inspired temporal coding and plasticity mechanisms can improve resilience in certain conditions, but further work is required to achieve noise-invariant encoding and domain generalisation. By bridging theoretical advances in neural coding with practical evaluation, this report contributes to the broader goal of developing energy-efficient, noise-tolerant intelligent systems suitable for deployment in real-world, safety-critical environments.

Appendix

GitLab Repository Information

This appendix provides information about the GitLab repository associated with this report. The repository is hosted at: <https://git.cs.bham.ac.uk/projects-2024-25/txk402.git>

- `C1_noisy_train+_clean_test.ipynb` – Training with noisy data and evaluation on clean data.
- `C23_noisy_train+clean_test.ipynb` – Extended noisy-to-clean training experiments.
- `N1_noisy_train+_noisy_test.ipynb` – Training and evaluation under noisy conditions.
- `N23_noisy_train+noisy_test.ipynb` – Extended noisy-to-noisy training experiments.
- `README.md` – Project description, background, and experimental design.

Execution Instructions

1. Clone the repository:
2. Install required dependencies.
3. Launch Jupyter Lab/Notebook and open the desired notebook:
4. Run the notebook cells in order to reproduce the experiments.

Bibliography

- [1] Arnon Amir, Brian Taba, David Berg, Thomas Melano, Jeffrey McKinstry, Christian Di Nolfo, Tapas Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcelo Mendoza, and et al. A low power, fully event-based gesture recognition system. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7243–7252, 2017.
- [2] Guillaume Bellec, Franz Scherr, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.
- [3] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998.
- [4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [5] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sankar Choday, George Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, and et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.
- [6] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99, 2015.
- [7] Steven K Esser, Paul A Merolla, John V Arthur, Andrew S Cassidy, Raja Appuswamy, Alexander Andreopoulos, David R Berg, Jeffrey L McKinstry, Thomas J Melano, David R Barch, and et al. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, 113(41):11441–11446, 2016.

- [8] Steve B Furber, Francesco Galluppi, Steve Temple, and Luis A Plana. Spinnaker: A spiking neural network architecture. *IEEE Proceedings*, 102(5):652–665, 2014.
- [9] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.
- [10] Giacomo Indiveri and Shih-Chii Liu. Memory and information processing in neuromorphic systems. *Proceedings of the IEEE*, 103(8):1379–1397, 2015.
- [11] Nabeel Khan, Khurram Iqbal, and Maria G. Martini. Lossless compression of data from static and mobile dynamic vision sensors—performance and trade-offs. *IEEE Access*, 8:101159–101172, 2020. [doi:10.1109/ACCESS.2020.2996661](https://doi.org/10.1109/ACCESS.2020.2996661).
- [12] D. Larionov. Mnist spiking neural network. Kaggle Notebook, 2021. Accessed: 2025-08-30. URL: <https://www.kaggle.com/code/dlarionov/mnist-spiking-neural-network>.
- [13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- [14] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbrück. A 128×128 120 db 15 s latency asynchronous temporal contrast vision sensor. In *IEEE Journal of Solid-State Circuits*, volume 43, pages 566–576. IEEE, 2008.
- [15] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.
- [16] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.
- [17] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- [18] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. In *Frontiers in Neuroscience*, volume 9, page 437. Frontiers, 2015.
- [19] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, 12:774, 2018.
- [20] Robotics and University of Zurich Perception Group. Event-based vision, event cameras, event camera slam. https://rpg.ifi.uzh.ch/research_dvs.html, 2020. Accessed: 2025-09-04.

- [21] Bodo Rueckauer, Ioana-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.
- [22] Abhronil Sengupta, Yuting Ye, Yanan Wang, Chen Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.
- [23] Sumit Bam Shrestha and Garrick Orchard. Spikenorm: Normalization method for converting artificial neural networks to spiking neural networks. *2018 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.
- [24] Amos Sironi, Marco Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.
- [25] Sen Song, Kenneth D Miller, and L F Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3:919–926, 2000.
- [26] Marcel Stimberg, Dan FM Goodman, and Romain Brette. Brian 2, an intuitive and efficient neural simulator. *eLife*, 2019.
- [27] Evangelos Stamatias, Daniel Neil, Michael Pfeiffer, Shih-Chii Liu, and Steve B Furber. Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Frontiers in Neuroscience*, 11:240, 2017.
- [28] Evangelos Stamatias, Miguel Soto, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Frontiers in neuroscience*, 11:350, 2017.
- [29] Jixiang Wan, Ming Xia, Zunkai Huang, Li Tian, Xiaoying Zheng, Victor Chang, Yongxin Zhu, and Hui Wang. Event-based pedestrian detection using dynamic vision sensors. *Electronics*, 10(8):888, 2021. [doi:10.3390/electronics10080888](https://doi.org/10.3390/electronics10080888).
- [30] Yujie Wu, Liang Deng, and Guoqi Li. Direct training for spiking neural networks: Faster, larger, better. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1311–1318, 2019.
- [31] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.