SAFE-DNN: A DEEP NEURAL NETWORK WITH SPIKE ASSISTED FEATURE EXTRACTION FOR NOISE ROBUST INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a Deep Neural Network with Spike Assisted Feature Extraction (SAFE-DNN) to improve robustness of classification under stochastic perturbation of inputs. The proposed network augments a DNN with unsupervised learning of low-level features using spiking neuron network (SNN) with Spike-Time-Dependent-Plasticity (STDP). The complete network learns to ignore local perturbation while performing global feature detection and classification. The experimental results on CIFAR-10 and ImageNet subset demonstrate improved noise robustness for multiple DNN architectures without sacrificing accuracy on clean images.

1 INTRODUCTION

Statistical machine learning using deep neural network (DNN) has demonstrated high classification accuracy on complex inputs in many application domains. Motivated by the tremendous success of DNNs in computer vision (Krizhevsky et al. (2012); He et al. (2015); Szegedy et al. (2015); Sandler et al. (2018)) there is a growing interest in deploying DNNs in autonomous systems interacting with physical world such as autonomous vehicles and robotics. However, an autonomous vehicle needs to make reliable classifications even with noisy sensor data. For deep convolutional neural networks that depend on statistical training methods, perturbation of pixel level information can cause kernels to generate incorrect feature maps. Such errors can propagate through network and degrade the classification accuracy (Nazaré et al. (2017)). The impact of noise on image classification has received significant interest in recent years. Nazar (Nazaré et al. (2017)) and Luo (Luo & Yang (2014)) shows that noise in inference images causes degradation of image classification performance of DNN. Solutions that have been proposed include training with dataset containing noise (Milyaev & Laptev (2017), Nazaré et al. (2017)) and manually introducing noise to network parameters (Luo & Yang (2014)). Other approaches consider pre-processing images with trained de-noising network or using pixel level regularization while training with noisy images (Ronneberger et al. (2015), Na et al. (2019), Na et al. (2018)). The prior works show improved accuracy when noise pattern used in training is similar to that experienced in the inference. However, it is improbable to pre-estimate all sources and structures of noise that an autonomous system may experience during operation. Moreover, training a network with noisy data or use of de-noising network can degrade performance on clean data. Therefore, a new class of DNN architecture is necessary for autonomous applications that is inherently resilient to input perturbations and does not require extensive training on noisy data.

The neuro-inspired learning, in particular, Spiking Neural Network (SNN) with Spike Time Dependent Plasticity (STDP) is an alternative and unsupervised approach to learning features in input data (Hebb et al. (1950); Bi & Poo (2001); Diehl & Cook (2015); She et al. (2019a); Querlioz et al. (2013); Srinivasan et al. (2016)). STDP based SNN optimizes network parameters according to causality information (Moreno-Bote & Drugowitsch (2015); Lansdell & Kording (2019)) with no labels required. However, the classification accuracy of a STDP-learned SNN for complex datasets is still much lower than what is achievable with a traditional DNN.

This paper presents a hybrid network architecture where the feature space of a DNN is augmented with features extracted via an SNN with STDP-based learning (Figure 1). The proposed network is referred to as Spike Assisted Feature Extraction based Deep Neural Network (SAFE-DNN). We argue that supervise training in DNN enables global learning between low-level pixel-to-pixel interactions

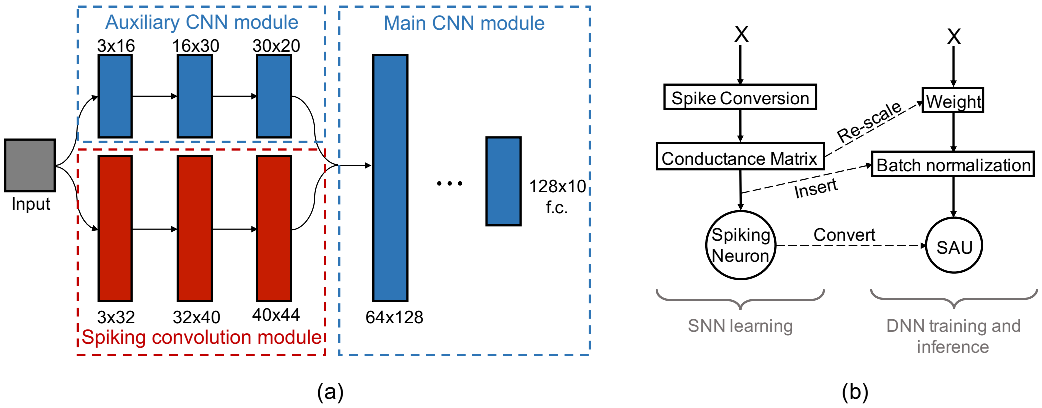

Figure 1: (a) An example architecture of SAFE-DNN. (b) Transition of building blocks from SNN to spiking convolution module of SAFE-DNN, with a special activation unit (SAU)

and high-level detection and classification. On the other hand, STDP performs unsupervised local learning and extracts low-level features under spatial correlation. By integrating features from global (supervised training) and local (STDP) learning, the hybrid network "learns to ignore" locally uncorrelated perturbations (noise) pixels while extracting the correct feature representation from the overall image. Consequently, SAFE-DNN achieves more robust image classification under stochastic perturbation of the input. This paper makes the following key contributions:

- We present SAFE-DNN, a hybrid network architecture that can integrate features extracted via supervised training and unsupervised neuro-inspired learning. The proposed design is versatile and can be easily implemented with different deep learning architectures to improve their robustness.
- We develop an SNN architecture with spiking convolution layers and optimized neuron activation functions that facilitate integration of SNN generated features within the conventional DNN pipeline creating a single network.
- We present a cohesive learning methodology for SAFE-DNN that couples STDP-based robust learning of local features with stochastic gradient descent (SGD) based supervised training of the network.
- We demonstrate that a SAFE-DNN is more resilient to input perturbation than a conventional DNN while requiring no prior knowledge of the perturbation during training and inference. Moreover, we show that, unlike pre-processing based noise removal of images or training the DNN with noisy data, SAFE-DNN does not impact the accuracy for clean images.

The rest of the paper is organized as follows: Section 2 presents the background on SNN; Section 3 discusses motivation behind SAFE-DNN; Section 4 presents architecture and learning method for SAFE-DNN; Section 5 presents the experimental results; and Section 6 concludes the paper.

2 Background on SNN

Spiking neural network uses biologically plausible neuron and synapse models that can exploit temporal relationship between spiking events (Moreno-Bote & Drugowitsch (2015); Lansdell & Kording (2019)). There are different models that are developed to capture the firing pattern of real biological neurons. We choose to use Leaky Integrate Fire (LIF) model in this work described by:

$$dv/dt = a + bv + cI; \text{ and } v = v_{reset}, \text{ if } v > v_{threshold} \tag{1}$$

where, a, b and c are parameters that control neuron dynamics, and I is the sum of current signal from all synapses that connects to the neuron.

In SNN, two neurons connected by one synapse are referred to as pre-synaptic neuron and post-synaptic neuron. Conductance of the synapse determines how strongly two neurons are connected and learning is achieved through modulating the conductance following an algorithm named spike-timing-dependent-plasticity (STDP) (Hebb et al. (1950); Bliss & GardnerMedwin (1973); Gerstner et al. (1993)). With two operations of STDP: long-term potentiation (LTP) and long-term depression (LTD), SNN is able to extract the causality between spikes of two connected neurons from their temporal relationship. More specifically, LTP is triggered when post-synaptic neuron spikes closely after a pre-synaptic neuron spike, indicating a causal relationship between the two events. On the other hand, when a post-synaptic neuron spikes before pre-synaptic spike arrives or without receiving a pre-synaptic spike at all, the synapse goes through LTD. We choose to use a frequency-dependent (FD) stochastic STDP model that has been tested in machine vision applications (She et al. (2019b)). For this model the magnitude of modulation is determined by (Querlioz et al. (2013)):

$$\Delta G_p = \alpha_p e^{-\beta_p (G-G_{min})/(G_{max}-G_{min})} \text{ and } \Delta G_d = \alpha_d e^{-\beta_d (G_{max}-G)/(G_{max}-G_{min})} \tag{2}$$

$$P_{pot} = \gamma_{pot} e^{(-\Delta t/(\tau_{pot}(1+\phi_{pot}\frac{f-f_{min}}{f_{max}-f_{min}})))} \text{ and } P_{dep} = \gamma_{dep} e^{(\Delta t/(\tau_{dep}(1+\phi_{dep}\frac{f-f_{min}}{f_{max}-f_{min}})))} \tag{3}$$

In the functions above, ΔG_p is the magnitude of LTP actions, and ΔG_d is the magnitude of LTD actions. α_p, α_d, β_p, β_d, G_{max} and G_{min} are parameters that are tuned based on other network configurations. This algorithm also dynamically adjust the probability of LTP/LTD based on spike timing and input signal frequency. τ_{dep} and τ_{pot} are time constant parameters. Δt is determined by subtracting the arrival time of the pre-synaptic spike from that of the post-synaptic spike ($t_{post} - t_{pre}$). Probability of LTP P_{pot} is higher with smaller Δt, which indicates a stronger causal relationship. The probability of LTD P_{dep} is higher when Δt is larger. γ_{pot} and γ_{dep} controls the peak value of probabilities.

3 Motivation behind SAFE-DNN

The gradient descent based weight update process in a DNN computes the new weight as $W' = W - \eta \nabla L$, where the gradient of loss function L is taken with respect to weight: $\nabla_w L = \langle \frac{\partial L}{\partial W_i}, ..., \frac{\partial L}{\partial W_k} \rangle$. Consider cross entropy loss as an example for L, weight optimization of element i is described by:

$$W_i' = W_i - \eta \frac{-\frac{1}{N}\partial\{\sum\limits_{n=1}^{N}[y_n log(\hat{y}_n)]\}}{\partial W_i} \tag{4}$$

Here η is the rate for gradient descent; N is the number of classes; y_n is a binary indicator for the correct label of current observation and $\hat{y}_n$ is the predicated probability of class n by the network. For equation (4), gradient is derived based on the output prediction probabilities $\hat{y}$ and ground truth. Such information is available only at the output layer. To generate the gradient, the output prediction (or error) has to be back-propagated from the output layer to the target layer using chain rule. As $\hat{y} = g(W, X)$ with g being the logistic function and X the input image, the prediction probabilities are the outcome of the entire network structure.

Consider the low level feature extraction layers in a deep network. Equation 4 suggests that gradient of the loss with respect to a parameter is affected by all pixels in the entire input image. In other words, the back-propagation makes weight update sensitive to interactions between non-neighboring pixels; which facilitates global learning and improve accuracy of higher level feature detection and classification.

However, the global learning also makes it difficult to strongly impose local constraints during training. Hence, the network *does not learn to ignore* local perturbations during low-level feature extraction as it is trained to consider global impact of each pixel for accurate classifications. In other words, although a noisy pixel is an outlier from the other pixels in the neighbourhood, a DNN must consider that noise as *signal* while extracting low-level features. The resulting perturbation from pixel level noise propagates through the network, and degrades the classification accuracy.

The preceding discussion suggests that, to improve robustness to stochastic input perturbation (noise), the low level feature extractors must learn to consider local spatial correlation. The local learning will allow network to more effectively “ignore” noisy pixels while computing the low-level feature maps and inhibit propagation of input noise into the DNN pipeline.

The motivation behind SAFE-DNN comes from the observation that STPD in SNN enables local learning of features. Compared to conventional DNN, SNN conductance is not updated through gradient descent that depends on back propagation of global loss. Consider a network with one spiking neuron and n connected input synapses, a spiking event of the neuron at time t_{spike} and timing of closest spikes from all input spike trains T_{input}, the modulated conductance is given by:

$$G_i' = G_i + \text{sign}(\Delta t_i) \cdot r(G_i) \cdot p(\Delta t_i, f_i) \tag{5}$$

Here $\Delta t_i = t_{spike} - T_{input}^i$ is spike timing difference, r is the magnitude function (Equation 2) and p is the modulation probability function (Equation 3). The value of t_{spike} is a result of the neuron’s response to the collective sum of input spike trains in one kernel. Hence, the modulation of weight of each synapse in a SNN depends only on other input signals within the same (local) receptive field. Moreover, as the correlation between the spike patterns of neighboring pre-synaptic neurons controls and causes the post-synaptic spike, STDP helps the network learn the expected spatial correlation between pixels in a local region. During inference, if the input image contains noise, intensity of individual pixel can be contaminated but within a close spatial proximity the correlation is better preserved. As the SNN has learned to respond to local correlation, rather than individual pixels, the neuron’s activity experiences less interference from local input perturbation. In other words, the SNN “learns to ignore” local perturbations and hence, the extracted features are robust to noise.

4 SAFE-DNN Architecture and Learning Process

4.1 Network Architecture

Discussion in section 3 motivates the design of SAFE-DNN architecture. Fig. 1 (a) shows an illustrative implementation of SAFE-DNN. The network contains spiking layers placed contiguously to form the spiking convolution module, along with conventional CNN layers. The spiking convolution module is placed at the front to enable robust extraction of local and low-level features. Further, to ensure that the low-level feature extraction also consider global learning, which is the hallmark of gradient back-propagation as discussed in section 3, we also place several conventional CNN layers of smaller size in parallel with the spiking CNN module. This is called the auxiliary CNN module. The output feature map of the two parallel modules is maintained to have the same height and width, and concatenated along the depth to be used as input tensor to the remaining CNN layers, referred to as the main CNN module. Main CNN module is responsible for higher level feature detection as well as the final classification. The main CNN module can be designed based on existing deep learning models. The concatenation of features from auxilary CNN and Spikining convolutional module helps integrate global and local learning.

Table 1: Network Complexity

Model	Params (M)	MACs (G)
Baseline MobileNetV2	3.50	0.33
Baseline ResNet101	44.55	7.87
Baseline DenseNet121	7.98	2.90
SAFE-MobileNetV2	3.57	0.36
SAFE-ResNet101	44.62	7.90
SAFE-DenseNet121	8.04	2.94

Fig. 2 shows the process of implementing SAFE-MobileNetV2 based on the original MobileNetV2. The first convolution layer and the following one block from the original network architecture are dropped and the remaining layers are used as the mian CNN module of SAFE-MobileNetV2. We show that SAFE-DNN is a versatile network by testing three configurations in this work, which have the main CNN module based on MobileNetV2 (Sandler et al. (2018)), ResNet101 (He et al. (2015)) and DenseNet121 (Huang et al. (2016)), respectively. The storage and computational complexity of the networks are shown in Table 1. It can be observed that SAFE-DNN implementations do not introduce a significant overhead to the baseline networks.

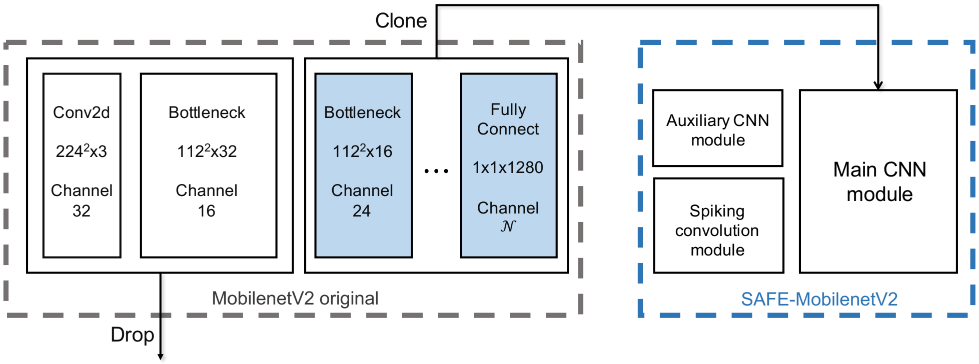

Figure 2: Creating SAFE-MobileNetV2 from the original MobileNetV2

4.2 Training Process

In the dynamical system of SNN, neurons transmit information in the form of spikes, which are temporally discrete events that spread across multiple simulation time steps. This requires input signal intensity to be converted to spike trains (She et al. (2019a); Diehl & Cook (2015)), and a number of time steps for neurons to respond to input stimulus. Such mechanism is different from that of the conventional DNN, which takes only one time step for data to propagate through the network. Due to this reason the native SNN model can not be used in spiking convolution module of SAFE-DNN. Two potential solutions to this problem are, run multiple time steps for every input, or, adapting the spiking convolution module to single-time-step response system. Since the first slows down both training and inference by at least one order of magnitude, we choose the latter.

To implement this approach, we separate STDP-based learning and DNN training into two stages. In the first stage, the spiking convolution module operates in isolation, learns all images in the training set without supervision. The learning algorithm follows STDP as defined in section 2. The overall learning process for spiking convolutional module is discussed next in section 4.3.

In the second stage, network parameters are first migrated to the spiking convolution module of SAFE-DNN. The network building blocks go through a conversion process shown in Fig. 1 (b). Here, the input signal to spike train conversion process is dropped, and conductance matrix is re-scaled to be used in the new building block. Batch normalization is inserted after the convolution layer. A special activation unit (SAU) replaces the basic spiking neuron model. The SAU is designed to fit the non-linearity of spiking neurons, and discussed later in section 4.4. The entire SAFE-DNN is then trained using statistical method, while weights in the spiking convolution module are kept fixed to preserve features learned by SNN. The inference is performed using the network architecture created during the second stage of training i.e. instead of the baseline LIF, the SAU is used for modeling neurons.

4.3 Spiking Convolutional Module

The overall architecture of the spiking convolutional module is shown in Fig. 3. This architecture shares similarity with conventional DNN with a few differences. First, the 8-bit pixel intensity from

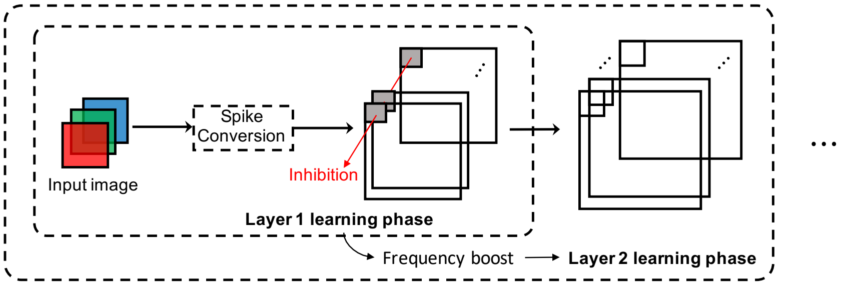

Figure 3: The architecture of the spiking convolutional module for feature extraction and layer-by-layer learning process.

input images is converted to spike train with frequency over a range from f_{min} to f_{max}. The input spike trains connect to spiking neurons in the convolution layer in the same way as conventional CNN. When a neuron in the convolution layer spikes, inhibitory signals are sent to neurons at the same (x,y) coordinate across all depth in the same layer. This prevents all neurons at the same location from learning the same feature.

Due to the diminishing spiking frequency of multiple-layer SNN, a layer-by-layer learning procedure is used. When the first layer completes learning, its conductance matrix is kept fixed and cross-depth inhibition disabled. Next, all neurons in the first layer are adjusted to provide higher spiking frequency by lowering the spiking threshold as illustrated in Fig.4. The neurons in the first layer receive input from input images and produce enough spikes that can facilitate learning behavior of the second layer. The same process is repeated until all layers complete learning.

4.4 Special Activation Unit

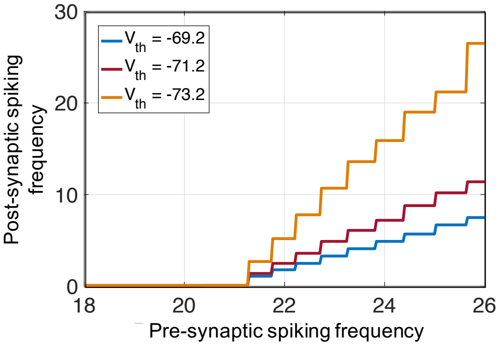

Figure 4: Post-synaptic spiking frequency (Hz) vs. pre-synaptic spike frequency (Hz)

Consider the spike conversion process of SNN, given an input value of $X \in [0, 1]$ and input perturbation ξ, conversion to spike frequency with range $\epsilon \in [f_{min}, f_{max}]$ is applied such that $F = Clip_{\epsilon}\{(X + \xi)(f_{max} - f_{min})\}$. For the duration of input signal T_{input}, the total received spikes for the recipient is $N_{spike} = \lfloor F * T_{input} \rfloor$. Also consider how one spiking neuron responses to input frequency variation, which is shown in Fig.4: it can be observed that flat regions exist throughout spiking activity as its unique non-linearity. Therefore, for $|\xi| \leq \frac{\delta}{T_{input}(f_{max}-f_{min})}$ perturbation does not cause receiving neuron to produce extra spikes. While the exact value of δ changes with different input frequency, it is small only when original input frequency is near the edges of non-linearity. This provides the network with extra robustness to small input perturbations. Based on this, we design the Special Activation Unit (SAU) to be a step function in the form of $f(x) = \sum_{i=1}^{n} \alpha_i \chi_i(x)$ where α_i and χ_i are pre-defined multiplication parameter and interval indicator function.

5 Experimental Results

5.1 CIFAR10 Dataset

Three baseline networks: MobileNetV2 (Sandler et al. (2018)), ResNet101 (He et al. (2015)) and DenseNet121 (Huang et al. (2016)), are tested in comparison with SAFE-DNN. We also studied two enhancement methods for baseline networks, namely, training with noisy input (30 dB) and using average filter (2x2) for image pre-processing. Note SAFE-DNN is never trained with nosiy images; it is only trained with clean images and only tested with noisy images. Fig. 5 shows training and test loss (top), and training accuracy and test accuracy (bottom) for the training process SAFE-MobileNetV2.

Visualization of the Embedding Space: We compare the capability of SAFE-MobileNetV2 in clustering noisy input with three networks. First, we consider the standard (baseline) MobileNetV2. The second one, referred to as MobileNetV2-μ, has the same architecture as SAFE-MobileNetV2, but the spiking convolution module is replaced with regular

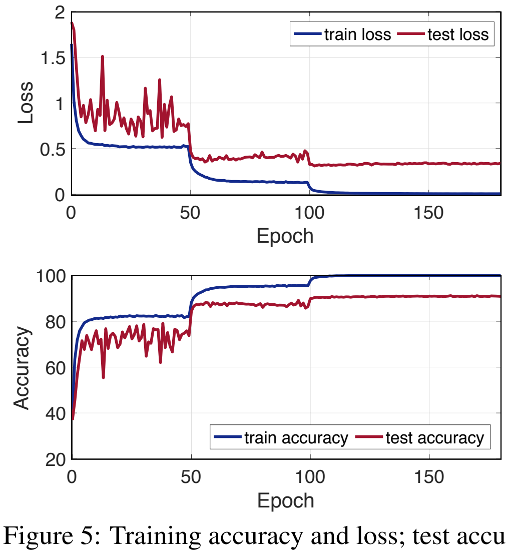

Figure 5: Training accuracy and loss; test accuracy and loss for SAFE-MobileNetV2.

Table 2: Accuracy (%) results for CIFAR10 with noise

		SNR					
Model	Clean	40 dB	30 dB	25 dB	20 dB	15 dB	12 dB
Baseline MobileNetV2	91.30	90.86	84.85	66.25	35.13	18.50	14.26
Baseline ResNet101	93.57	89.74	86.39	78.32	55.47	26.33	15.53
Baseline DenseNet121	93.00	92.87	89.84	82.59	60.42	27.10	16.88
Noise trained MobileNetV2	90.54	90.64	90.16	86.36	62.22	25.37	16.51
Noise trained ResNet101	92.41	92.51	92.26	90.92	77.81	35.97	19.85
Noise trained DenseNet121	91.88	91.86	91.71	90.74	75.35	33.89	19.35
MobileNetV2 with average filter	58.91	55.12	48.37	42.56	38.79	33.36	29.88
ResNet101 with average filter	60.18	57.06	49.64	45.02	39.50	34.88	32.79
DenseNet121 with average filter	59.58	58.86	51.00	46.89	42.05	35.06	34.09
SAFE-MobileNetV2	91.33	91.25	90.01	90.68	87.88	64.95	39.22
SAFE-ResNet101	93.59	93.43	92.13	92.11	90.47	70.85	43.25
SAFE-DenseNet121	93.03	92.86	92.70	91.35	88.00	62.99	33.19

trainable DNN layers. The third one, referred to as the MobileNetV2-λ, is constructed by replacing the activation functions in the first three layers of a trained MobileNetV2-μ with the SAU (without any re-training). The comparisons with MobileNetV2-μ and MobileNetV2-λ show whether benefits of SAFE-MobilenetV2 can be achieved by only architectural modifications or new (SAU) activation function, respectively, without local STDP learning. All networks are trained with CIFAR10 dataset. Fig. 6 shows embedding space visualizations of all four networks with clean and noisy (signal-to-noise ratio or SNR equal to 25dB) images. The embedding space is taken between the two fully connected layers and each color represents one class. We observe that with clean input images, the vectors in embedding space of the baseline MobileNetV2 are distributed into ten distinct clusters. As noise is added to the images the clusters overlap which leads to reduced classification accuracy. On the other hand, SAFE-MobileNetV2 is able to maintained good separation between feature mappings for each class from no noise to 25 dB. We further observe that clusters for noisy images also heavily overlap for MobileNetV2-μ and MobileNetV2-λ, showing that only using architectural modification or spiking activation function, without STDP learning, cannot improve noise robustness of a DNN.

Accuracy Comparison: Table 2 shows accuracy of all network variants for CIFAR-10. For the baseline DNNs, noise in images significantly degrades classification accuracy. The networks that are trained with noise (30dB noise is used during training) show higher robustness to noise, and the improvement is more prominent when inference noise is at similar level (30 dB) with training noise. For clean images the accuracy is degraded. Average filtering provides accuracy gain over the original networks in highly noisy conditions (less than 20 dB signal to noise ration (SNR)); but major performance drop is observed under mild to no noise. This is expected as average filtering results in significant loss of feature details for input images in the CIFAR-10 dataset.

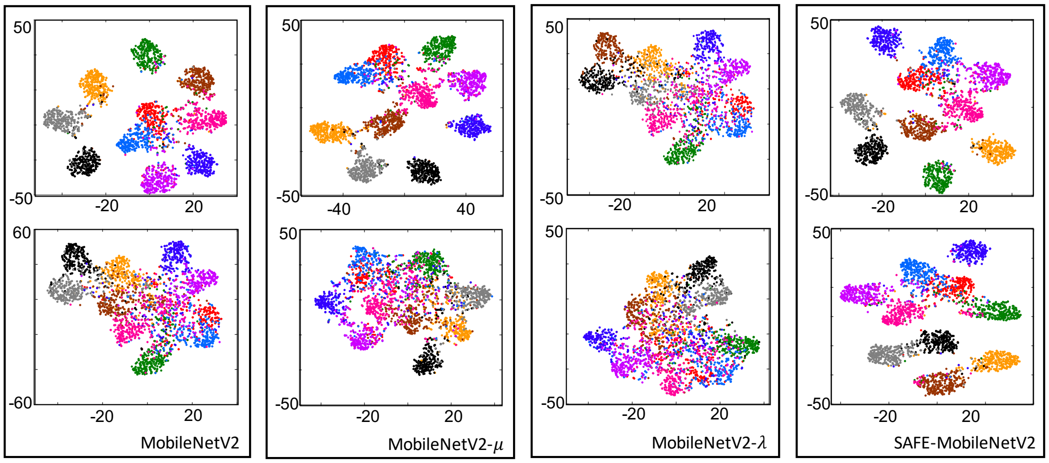

Figure 6: Visualization of the embedding space for clean (top) and noisy (bottom) input images.

Table 3: Top 1 Accuracy (%) results for ImageNet subset with noise

	SNR				
Model	Clean	25 dB	15 dB	10 dB	5 dB
Baseline MobileNetV2	70.80	67.41	57.92	45.35	34.48
Baseline ResNet101	71.02	67.81	64.04	46.27	35.47
Baseline DenseNet121	70.92	67.60	63.28	44.34	27.50
Noise trained MobileNetV2	66.30	68.12	59.71	46.70	34.66
Noise trained ResNet101	68.91	69.20	65.71	52.60	41.32
Noise trained DenseNet121	69.13	70.51	66.47	52.76	36.32
MobileNetV2 with average filter	67.44	66.91	61.69	51.69	40.43
ResNet101 with average filter	68.18	68.25	65.14	53.09	41.50
DenseNet121 with average filter	65.38	64.56	62.40	50.65	39.40
SAFE-MobileNetV2	71.05	67.86	65.91	53.82	42.33
SAFE-ResNet101	71.14	70.67	67.24	55.04	42.87
SAFE-DenseNet121	70.81	69.44	65.47	54.30	40.84

For SAFE-DNN implemented with all three DNN architectures, performance in noisy condition is improved over the original network by an appreciable margin. For example, at 20 dB SNR SAFE-MobileNetV2 remains at good performance while the original network drops below 40% accuracy, making a significant (50%) gain. Similar trend can be observed for other noise levels. Compared to networks trained with noise, SAFE-DNN shows similar performance at around 30 dB SNR while its advantage increases at higher noise levels. Moreover, for clean images accuracy of SAFE-DNN is on par with the baseline networks.

5.2 Test on ImageNet Subset

Considering the use case scenario of autonomous vehicles, we conduct test on a subset of ImageNet that contains classes related to traffic (cars, bikes, traffic signs, etc)[1]. The subset contains 20 classes with a total of 26,000 training images. The same baseline networks as in the CIFAR10 test are used. Here 25 dB SNR images are used for noise training. The accuracy result is shown in Table 3. All networks achieve around 70% top 1 accuracy on clean images. Noise training shows robustness improvement over the baseline network but still negatively affects clean image accuracy. In this test the average filter shows less degradation under no noise condition than for the CIFAR10 test, due to higher resolution of input images. DensNet121 shows more noise robustness than MobileNetV2 and ResNet101 when noise training is used, while for average filtering ResNet101 benefits the most. SAFE-DNN implementations of all three networks exhibit same or better robustness over all noise levels. Clean image classification accuracy is also unaffected. Comparing top 5 accuracy result for SAFE-MobileNetV2 and its baselines, as shown in Table 4, SAFE-MobileNetV2 is able to maintain above 80% accuracy even at 5 dB SNR, outperforming all three baselines.

Table 4: Top 5 Accuracy (%) results for MobileNetV2 on ImageNet subset with Noise

Model	Clean	10 dB	5 dB
Baseline	94.72	79.20	67.15
Noise trained	92.43	81.63	68.36
Average filtering	92.81	85.37	78.95
SAFE-MobileNetV2	95.91	89.57	83.92

6 Conclusions

In this paper we present SAFE-DNN as a deep learning architecture that integrates spiking convolutional network with stochastic STDP based learning into a conventional DNN for robust low level feature extraction. The experimental results show that SAFE-DNN improves robustness under noisy input while maintaining performance on clean images. SAFE-DNN is compatible with various DNN designs, making it an attractive candidate for real-time and autonomous systems that require accurate image classifications even under noisy input.

[1] The test on entire ImageNet is currently under progress

REFERENCES

Guo-qiang Bi and Mu-ming Poo. Synaptic Modification by Correlated Activity: Hebb's Postulate Revisited. *Annual Review of Neuroscience*, 2001. ISSN 0147-006X. doi: 10.1146/annurev.neuro.24.1.139.

T. V P Bliss and A. R. GardnerMedwin. Longlasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 1973. ISSN 14697793. doi: 10.1113/jphysiol.1973.sp010274.

Peter Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9(August):99, 2015. ISSN 1662-5188. doi: 10.3389/fncom.2015.00099. URL `http://journal.frontiersin.org/article/10.3389/fncom.2015.00099`.

Wulfram Gerstner, Raphael Ritz, and J. Leo van Hemmen. Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics*, 1993. ISSN 03401200. doi: 10.1007/BF00199450.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

D O. Hebb, Fred Attneave, and M. B. The Organization of Behavior; A Neuropsychological Theory. *The American Journal of Psychology*, 1950. ISSN 00029556. doi: 10.2307/1418888.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL `http://arxiv.org/abs/1608.06993`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. AlexNet. *NIPS*, 2012. ISSN 10495258. doi: 10.1145/3065386.

Benjamin James Lansdell and Konrad Paul Kording. Spiking allows neurons to estimate their causal effect. *bioRxiv*, pp. 253351, 2019.

Yixin Luo and Fan Yang. Deep learning with noise. *hp://www. andrew. cmu. edu/user/fanyang1/deep-learning-with-noise. pdf*, 2014.

S. Milyaev and I. Laptev. Towards reliable object detection in noisy images. *Pattern Recognition and Image Analysis*, 27(4):713–722, Oct 2017. ISSN 1555-6212. doi: 10.1134/S1054661817040149. URL `https://doi.org/10.1134/S1054661817040149`.

Rubén Moreno-Bote and Jan Drugowitsch. Causal Inference and Explaining Away in a Spiking Network. *Scientific Reports*, 2015. ISSN 20452322. doi: 10.1038/srep17531.

Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Noise-robust and resolution-invariant image classification with pixel-level regularization. In *International Conference on Acoustics, Speech and Signal Processing,(ICASSP)*, 2018.

Taesik Na, Minah Lee, Burhan A. Mudassar, Priyabrata Saha, Jong Hwan Ko, and Saibal Mukhopadhyay. Mixture of pre-processing experts model for noise robust deep learning on resource constrained platforms. In *2019 IEEE International Joint Conference on Neural Network*, 2019.

Tiago S Nazaré, Gabriel de Barros Paranhos da Costa, Welinton A Contato, and Moacir Ponti. Deep Convolutional Neural Networks and Noisy Images. In *CIARP*, 2017.

Damien Querlioz, Olivier Bichler, Philippe Dollfus, and Christian Gamrat. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, 12(3):288–295, 2013. ISSN 1536125X. doi: 10.1109/TNANO.2013.2250995.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. URL `http://arxiv.org/abs/1801.04381`.

Xueyuan She, Yun Long, and Saibal Mukhopadhyay. Fast and Low-Precision Learning in GPU-Accelerated Spiking Neural Network. *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019a.

Xueyuan She, Yun Long, and Saibal Mukhopadhyay. Improving robustness of reram-based spiking neural network accelerator with stochastic spike-timing-dependent-plasticity. *arXiv preprint arXiv:1909.05401*, 2019b.

Gopalakrishnan Srinivasan, Abhronil Sengupta, and Kaushik Roy. Magnetic Tunnel Junction Based Long-Term Short-Term Stochastic Synapse for a Spiking Neural Network with On-Chip STDP Learning. *Scientific Reports*, 6, 2016. ISSN 20452322. doi: 10.1038/srep29545.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.