

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/340282940>

# Streaming Networks: Increase Noise Robustness and Filter Diversity via Hard-wired and Input-induced Sparsity

Preprint · March 2020

CITATIONS

0

READS

144

2 authors:



[Sergey Tarasenko](#)

34 PUBLICATIONS 33 CITATIONS

SEE PROFILE



[Fumihiko Takahashi](#)

JapanTaxi Co.,Ltd.

6 PUBLICATIONS 3 CITATIONS

SEE PROFILE

# Streaming Networks: Increase Noise Robustness and Filter Diversity via Hard-wired and Input-induced Sparsity

Sergey Tarasenko\*, Fumihiko Takahashi†

Next-Gen Mobility Division, Mobility R&D Group, JapanTaxi  
Tokyo, Japan

Email: \*tarasenko.sergey@japantaxi.co.jp, †fumihiko.takahashi@japantaxi.co.jp

**Abstract**—The CNNs have achieved a state-of-the-art performance in many applications. Recent studies illustrate that CNNs’ recognition accuracy drops drastically if images are noise corrupted. We focus on the problem of robust recognition accuracy of noise-corrupted images. We introduce a novel network architecture called Streaming Networks. Each stream is taking a certain intensity slice of the original image as an input, and stream parameters are trained independently. We use network capacity, hard-wired and input-induced sparsity as the dimensions for experiments. The results indicate that only the presence of both hard-wired and input-induced sparsity enables robust noisy image recognition. Streaming Nets is the only architecture which has both types of sparsity and exhibits higher robustness to noise. Finally, to illustrate increase in filter diversity we illustrate that a distribution of filter weights of the first conv layer gradually approaches uniform distribution as the degree of hard-wired and domain-induced sparsity and capacities increases.

## I. INTRODUCTION

**Brief overview of the CNNs.** Since its first introduction in 1998 by Lecun et al. [13], the convolutional neural networks (CNNs) have proved their effectiveness by achieving state-of-the-art solutions for many tasks.

There is a vast variety of CNN architectures in the literature (AlexNet [12], LeNet [13], ResNet [9], GoogLeNet [25], VGG [22] etc). These are the networks that have a single stream structure.

Recently, CNNs with more than one processing streams have started to gain popularity. To our knowledge, the first two-stream network was introduced by Chorpa [5] and it is widely known as a “Siamese network”. The motivation behind two streams is that each of the streams carries information about a dedicated image. Images fed to the streams are different.

Most recently, two-stream networks have been used for the vast variety of recognition, segmentation and classification tasks such as similarity assessment (Siamese networks and pseudo-Siamese [5] [31]), change detection and classification [30], action recognition in videos [23], one-shot image recognition [11], simultaneous detection and segmentation [8], human-object interaction recognition [7], group activity recognition [1], etc.

**The CNNs and the primate brain.** Regarding the signal propagation in the brain networks, Thorpe et al. [27]–[29] argued that the stronger the response of a given neuron, the

faster such response should be produced, meaning that it takes less time to produce stronger output than a weaker one.

Thorpe et al. have suggested that neural outputs produced nearly at the same time form *waves of spikes*. So even information about the static single image is propagated through the neural network in time separated packets called waves of spikes, thus a static image is unfolded in time due to different response time for stronger and weaker outputs.

The idea of waves of spikes was then employed by Tarasenko [26]. Tarasenko continued work by Serre et al. [21] by proposing an on-line learning method for feature extraction and extending the pseudo-CNN to implement a predictive coding [19] mechanism.

The important peculiarity of work by Tarasenko is that to extract features images, containing complete information were used, while to enable mechanism of predictive coding after feature extraction, images were fed into the network by intensity slices (similar to waves of spikes). Examples of image intensity slices are presented in Fig. 1(a). Every single image with normalized pixel values was split into 10 images, which correspond to one of the intensity slices ranging from 0.0 to 1.0 with step 0.1. Then these slices were consecutively propagated into the network.

## II. RELATED WORK

**The CNNs and image distortions.** Recent studies have illustrated that CNNs’ performance is extremely fragile for distortions of the input images such as noise, image occlusions, rotation, scaling, etc. [24] [6]. In this paper, we approach the issue of robust recognition when images are corrupted with random zero-noise, i.e., a certain portion of pixels across the entire image is randomly set to zero intensity value.

The topic of robust recognition by CNNs under conditions of noise has been explored in [2], [3].

In work [2], authors analyzed the robustness (stability) of CNNs against image degradation due to noise. The same group of authors suggests a method to increase the stability of CNNs by introducing a denoising layers [3].

**Sparsity.** The networks in the brain exhibit very high sparsity degree [18]. The topic of sparsity is well established in studies by [18]. There are two approaches to sparsity: 1)

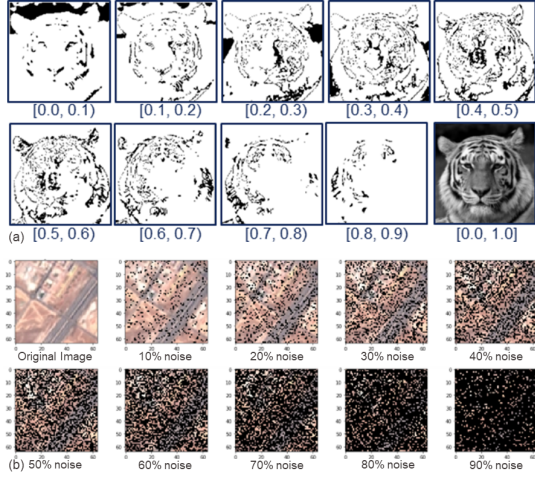


Fig. 1. Intensity slices and Noise.

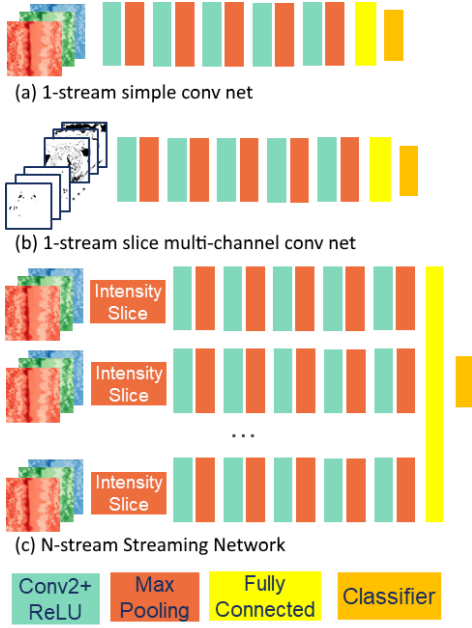


Fig. 2. One stream CNN and Streaming Network architectures.

sparsity of representation [17] [16] and 2) for network training [20], [15].

### III. STREAMING NETWORKS

**General Definition** Here we introduce a novel CNN architecture. We take a 1-stream CNN and add image intensity slicing module as an input. Then we clone new networks with intensity slices set to extract different intensity slices. Each such network constitutes a single stream. Finally, we concatenate outputs of all the streams to one fully connected layer, which is connected with a classifier. A number of fully connected layers after concatenated layer can vary.

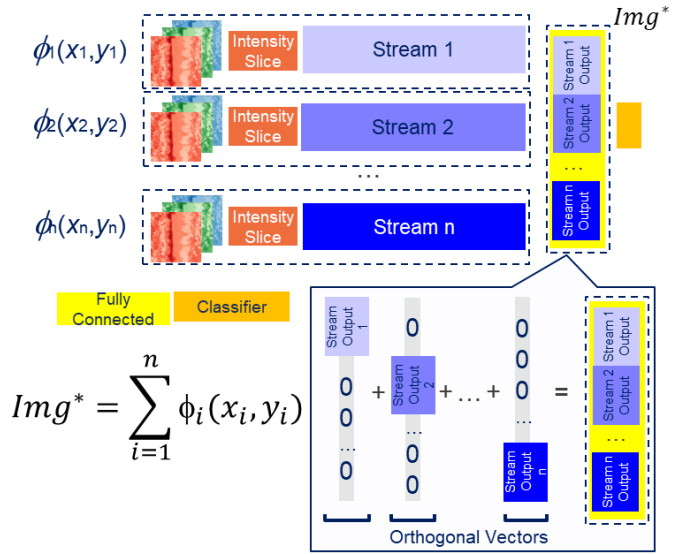


Fig. 3. Streaming Network as an orthogonal basis for image representation.

The weights and biases within every single stream are decoupled from the ones in other streams and are trained independently.

We call this architecture a Streaming Network (Fig. 2(c)).

#### Streaming Networks as an Orthogonal Basis for Image Representation

In this section, we explain the mathematical concepts behind Streaming Networks. Eq. (1) introduces an image representation by means of the orthogonal vectors, which form an orthogonal basis of some image representation space:

$$Img^* = \sum_{i=1}^n \alpha_i v_i \quad (1)$$

where  $\alpha_i$  is a weight coefficient for  $i$ -th orthogonal basis vector  $v_i$ ,  $i = 1, \dots, n$ , and  $Img^*$  is an image representation.

As the next step, we can generalize eq. (1) by providing orthogonal basis functions  $\phi_i(x_i, y_i)$  instead of basis vectors  $v_i$  [16]:

$$Img^* = \sum_{i=1}^n \alpha_i \phi_i(x_i, y_i) \quad (2)$$

where pair  $(x_i, y_i)$ ,  $i = 1, \dots, n$ , represents some part of an original image.

If now, we set all weighting coefficients  $\alpha_i$ ,  $i=1, \dots, n$ , to 1, we will get eq. (3):

$$Img^* = \sum_{i=1}^n \phi_i(x_i, y_i) \quad (3)$$

Eq. (3) perfectly describes the Streaming Network architecture (Fig. 3): functions  $\phi_i(\cdot)$ ,  $i=1, \dots, n$ , correspond to the streams of a Streaming Network, pairs  $(x_i, y_i)$ ,  $i=1, \dots, n$ , represent some transformation of parts of the original image, e.g., intensity slices.

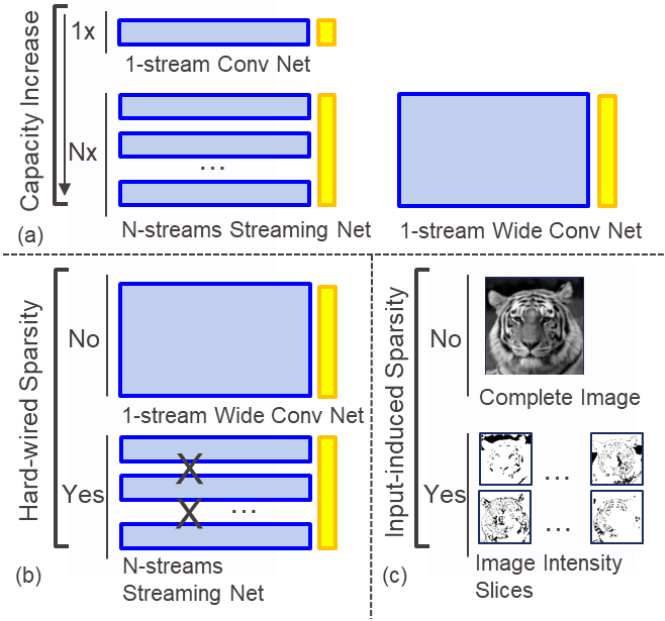


Fig. 4. Experimental dimensions explained: (a) network capacity increase; (b) hard-wired sparsity; (c) input-induced sparsity.

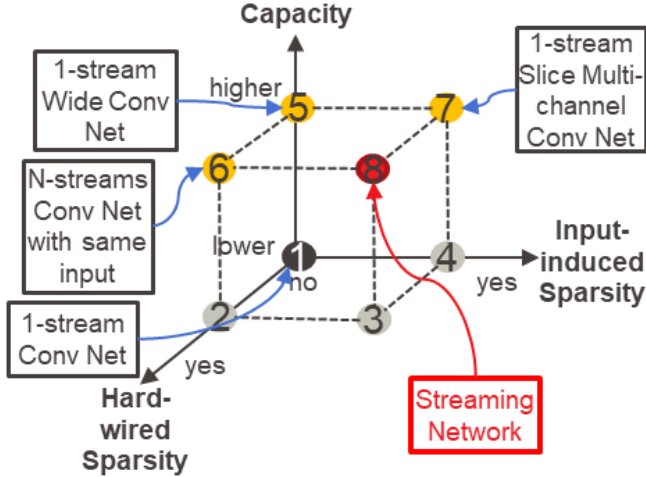


Fig. 5. Experimental design with corresponding network architectures.

#### IV. EXPERIMENTS

In the previous section, we have outlined the issue of accuracy reduction when images are corrupted in some way and touched the topic of sparsity, which is expected to produce robust encoding. In this section, we would like to investigate if it is possible to increase robustness of the CNNs introducing sparsity dimension.

The gist of sparsity control for network training is to make neurons to be activated with a certain frequency. Thus, providing a diversity of paths within a neural network. The concept of sparsity control makes a perfect sense in terms of binary neurons, which produce either 1, when active, or

0, when in-active. If neuron always produces 1 or 0, it is useless from the point of view of classification. To reduce the number of such neurons, usually, additional term, which penalized the loss function whenever activation frequency of neurons is deviating from the target frequency (usually set to 20%), is introduced. As mentioned above, the concept of sparsity makes perfect sense in terms of binary neurons. It can be extended to sigmoid or tahn neurons by introducing sum average value for activations, however, it is becoming very difficult to access in terms of CNNs with ReLUs, since the output of neurons is not limited on the interval  $(0, +\infty)$ .

To overcome the problem of sparsity control in CNNs with ReLU, we introduce two new concepts regarding sparsity in section Experimental Design IV-A.

##### A. Experimental Design

In this study, we introduce two approaches to control the sparsity: 1) *hard-wired sparsity*, when we eliminate all-to-all connections in the network by introducing independent streams; 2) *input-induced sparsity*, when we introduce a unique intensity slice of an image to each stream, which refrains independent streams from learning from the same input.

We also add the dimension of network *capacity*, i.e., a total number of filters inside each convolution layer of a given network.

Next we use these three dimensions to construct neural network architectures (Fig. 4). In this study, we qualitative evaluate capacities as *lower* or *higher*. The hard-wired and input-induced sparsities are considered to be either employed (*yes* value) or not employed (*no* values). Using this binary values of dimensions, it is possible to introduce our experimental space in the form of a cube in *Capacity - Hard-wired sparsity - Input-induced sparsity* 3D space (Fig 5).

Each vertex of the cube is related to a particular set of values for each dimension and has a corresponding neural network architecture.

Vertex (1) is characterize with *lower* capacity of the network, *no* hard-wired sparsity and *no* input-induced sparsity. The network architecture, which corresponds to vertex (1) is a *1-stream simple CNN* (Fig. 2(a)).

A 1-stream simple CNN has the following structure: 1) conv layer with 32 7x7 filters plus ReLU activation and 2x2 Max-pooling layers; 2) conv layer with 64 5x5 filters plus ReLU activation and 2x2 Max-pooling layers; 3) conv layer with 128 3x3 filters plus ReLU activation and 2x2 Max-pooling layers; 4) conv layer with 256 1x1 filters plus ReLU activation and 2x2 Max-pooling layers; 5) conv layer with 4<sup>1</sup> 1x1 filters plus ReLU activation and 2x2 Max-pooling layers; 6) fully connected layer; 7) SoftMax layer with the number of output neurons corresponding to the number of classes.

In our experiments, we consider network architectures with higher capacity, while we take 1-stream simple CNN as a bench-mark. Furthermore, vertices (2), (3) and (4) are lower

<sup>1</sup>For cifar10 dataset we use 10 1x1 filters

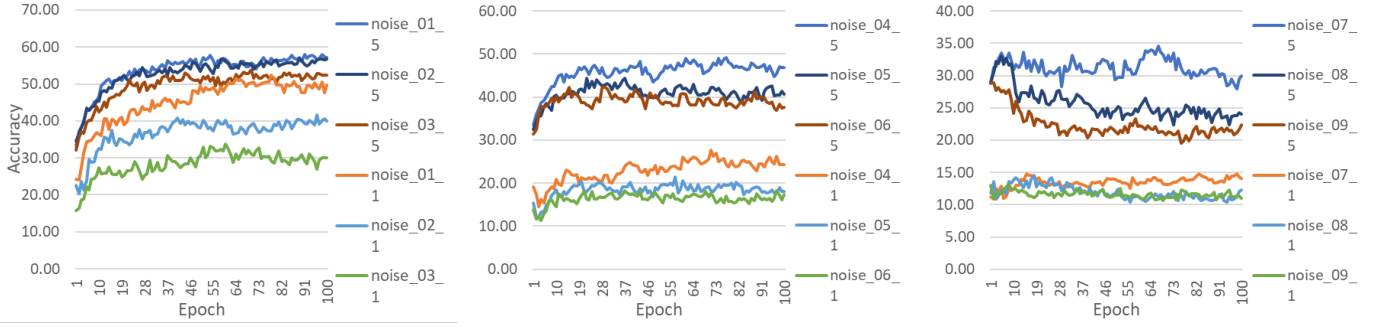


Fig. 6. Tests of Cifar10 dataset using Adam optimizer with 0.0001 learning rate for 1-stream simple CNN vs. 5-stream Streaming Net. The legend codes experimental conditions as noise\_{noise-rat}\_{network}. For example, noise\_01\_1 implies 1-stream simple network with noise 10%. By default we consider that networks with more than one stream are Streaming Networks. All curves are averaged across multiple runs.

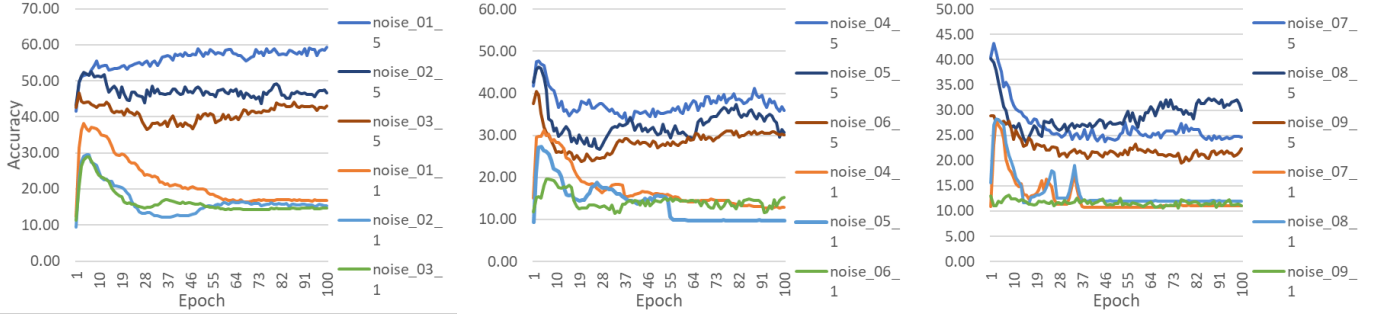


Fig. 7. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 1-stream simple CNN vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

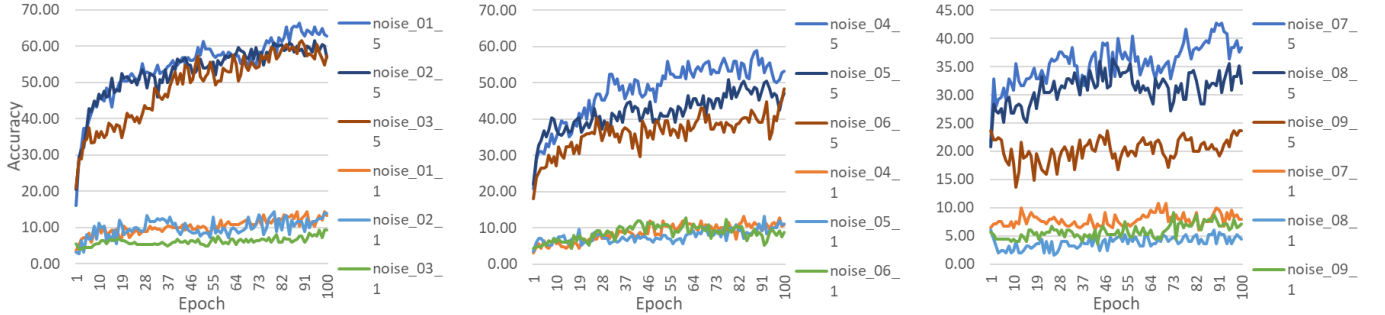


Fig. 8. Tests of UCmerced dataset using Adam optimizer with 0.0001 learning rate for 1-stream CNN and 5-stream Streaming Network. The notations are the same as in Fig. 6.

capacity analogs of vertices (5), (6), (7) and (8). Therefore, we exclude vertices (2), (3) and (4).

Hereafter, we introduce each vertex in the form of 3D vector  $\{Capacity - Hard-wired\ sparsity - Input-induced\ sparsity\}$ . Thus, vertex (1) is coded as {lower, no, no}.

Vertex (5) is coded as {higher, no, no} and the corresponding network architecture is a *1-stream wide CNN*, which has  $Nx$  filters in each conv layer, where is  $N$  an integer multiplier and  $x$  is a number of filters in corresponding conv layers of the 1-stream simple CNN. Vertex (6) is coded as {higher, yes, no} and the corresponding network architecture is *N-stream CNN with the same input for each stream*. Vertex (7) is

coded as {higher, no, yes}. We assume that the corresponding architecture is a *1-stream wide network*, where image slices are packed into different input channels. Thus, if we cut an original RGB image with three channels into five intensity slices, then input is combined from 15 channels - 3 channels per a single slice. We call such network architecture *1-stream slice multi-channel CNN*. We assume that a *1-stream slice multi-channel wide CNN* (Fig. 2(b)) corresponds to vertex (7). Finally, vertex (8) is coded as {higher, yes, yes} and the corresponding network architecture is a *Streaming Network*.



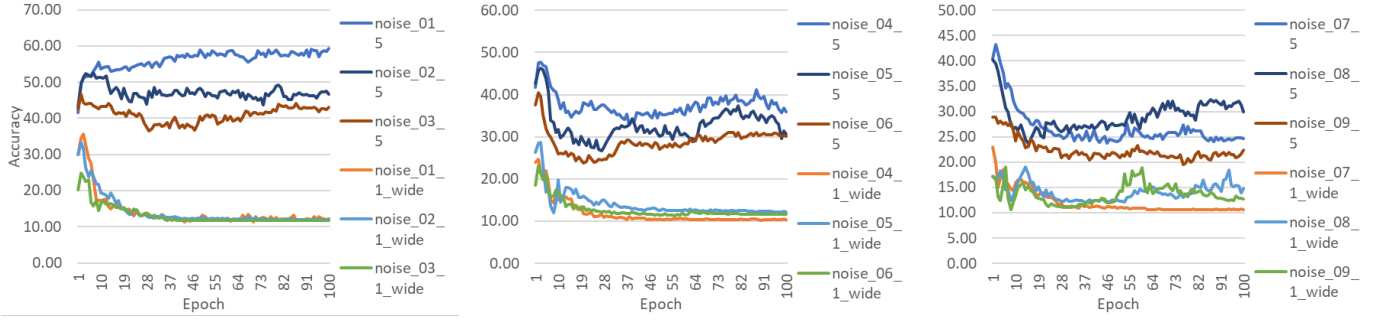


Fig. 9. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 1-stream wide CNN vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

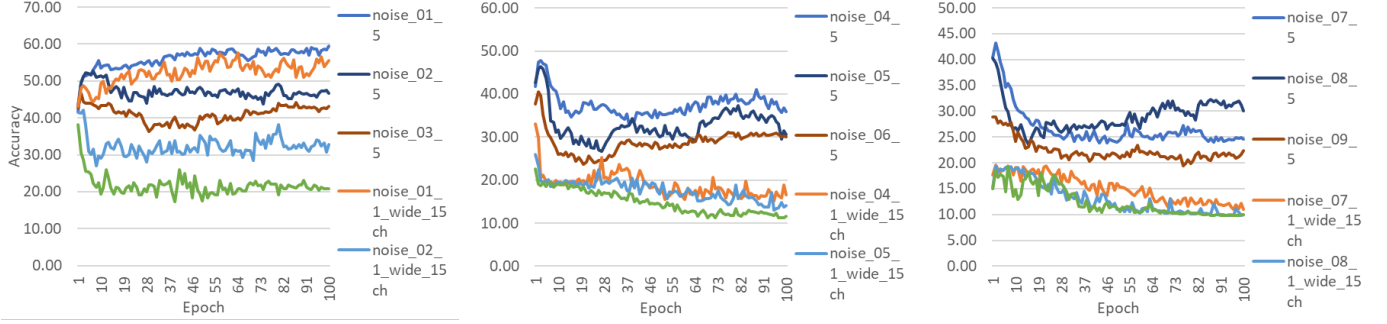


Fig. 10. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 1-streams 15-channel wide vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

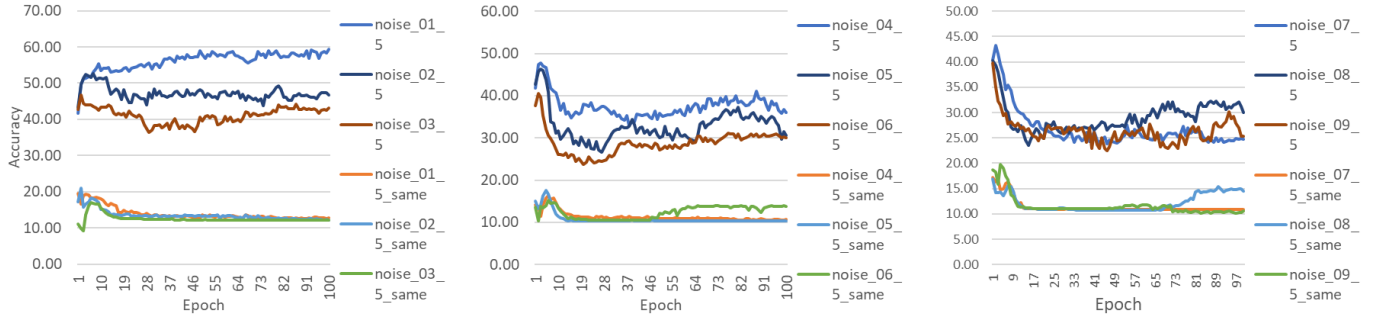


Fig. 11. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 5-streams with the same input vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

## B. Datasets and Learning Procedure

For our experiments we use three datasets. The selected datasets are cifar10<sup>2</sup>, Eurosat (rgb)<sup>3</sup> [10] and UCmerced land use<sup>4</sup>.

The reason to select these three dataset is to test our approach on different types of objects. Cifar10 dataset contains RGB 32x32 images of 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), while Eurosat and UCmerced consist of 10 and 21 classes, respectively, of aerial

satellite images. Thus, selecting these datasets we test images of various origins.

For all our experiments, we use Adam optimizer with learning rate of 0.0001 accompanied by  $\beta_1 = 0.99$ ,  $\beta_2 = 0.9$  and  $\epsilon = 1e-08$ , and run all the trainings for 100 epochs. Throughout the experiments, we use SoftMax classifier.

For each dataset we run the networks for noise level (ratio of pixels corrupted with noise) ranging from 0.1 to 0.9 with step 0.1, thus constituting 9 different levels. Examples of different noise levels for a selected Eurosat image are illustrated in Fig.1.

When we train the network after each iteration we compute

<sup>2</sup><https://www.cs.toronto.edu/~kriz/cifar.html>

<sup>3</sup><https://github.com/phelber/eurosat>

<sup>4</sup><http://weegee.vision.ucmerced.edu/datasets/landuse.html>

network accuracy for test data without noise and the test data corrupted with noise. In the figure, however, we only illustrates results for noise-corrupted images.

### C. Objective

Our objective is to investigate if it is possible to increase CNN robustness to noise by introducing two types of sparsity, i.e., input-induced and hard-wired. We do this by measuring recognition accuracy for noisy images for each type of five network architectures.

### D. Results

**Streaming Net vs. 1-stream simple CNN.** For all the dataset we computed accuracy for noise-corrupted images. The results for performance comparison of 1-stream simple CNN vs. 5-streams Streaming Net are presented in Figs. 6, 7 and 8 for cifar10, Eurosat and UCmerced datasets, respectively. These figures indicate that a 5-streams Streaming Net outperforms 1-stream simple CNN in recognition accuracy for noisy images.

**Streaming Net vs. 1-stream wide CNN.** Hereafter, we focus on the results for Eurosat datasets to illustrate performance networks, corresponding to vertices (5), (6), (7) and (8). The same results for cifar10 and UCmerced datasets are presented in Supplementary material.

Next, we compare 5-streams Streaming Net vs. 1-stream wide CNN. Fig. 21 indicates that the recognition accuracy of noise images degrades dramatically as noise ratio is increasing for 1-stream wide CNN, while it shows high robustness in the case of 5-streams Streaming Net.

**Streaming Net vs. 1-stream wide slice multi-channel CNN.** In Fig. 22 recognition accuracy for 1-stream slice 15-channel CNN is presented vs. performance of 5-stream Streaming Net. In terms of robustness to noise, a 1-stream slice 15-channel CNN is somewhat more stable to low levels of noise (up 30%) when 1-stream simple CNN, however, it shows no robustness for higher levels of noise. A 5-streams Streaming Network outperforms a 1-stream wide slice 15-channel CNN in noisy images recognition accuracy.

**Streaming Net vs. 5-stream CNN with same input.** A 5-streaming CNN with the same input is a very similar architecture to Streaming Net as it has multiple streams. However, the key difference is that Streaming Net takes different inputs into streams. Fig. 23 illustrates that noisy images recognition accuracy for a 5-stream CNN with the same inputs is dramatically worse than the one for a 5-streams Streaming Net.

Based on the comparison of neural network architectures for vertices (1), (5), (6), (7) and (8), the intermediate conclusion is that the Streaming Net is the only architectures robust to noise.

**Streaming Net with more streams.** Next, we investigate if it is possible to further improve noisy images recognition accuracy. We consider a 10-streams Streaming Network with intensity slices [0.0,0.1), [0.1,0.2), [0.2,0.3), [0.3,0.4), [0.4,0.5), [0.5,0.6), [0.6,0.7), [0.7,0.8), [0.8,0.9) and [0.9,1.1).

We compare performance of 10-stream Streaming Net vs. 1-stream wide networks and 5-stream Streaming Net. Fig. 24 illustrates that a 10-streams Streaming Net provides even higher robustness to noise compared to a 5-streams Streaming Net.

### E. Distribution of Trained Filter Values

In this section, we investigate distributions of values in the trained filters of the first conv layers. We perform our analysis from two perspectives: 1) distribution of filter weights<sup>5</sup> within a single convolution layer; 2) distribution of filter weights first conv layers for all streams.

Our hypothesis is based on the study by Lecun et al. [14], which implies that higher filter diversity enables faster learning and convergence. We hypothesize that emergence of the noise robustness, which we have observed in the experiments about originates from the higher diversity of filters produced by the Streaming Nets.

We assume that the filter weights are sampled from some restricted segment since there can be no infinitely big and infinitely small weights.

Since we sample filter weights from the segment, the most diversity of filter weights is expected from the distribution with the highest entropy. According to Bening and Korolev [4], the uniform distribution has the highest entropy value among all distributions spanned upon the restricted segment. Therefore, we expect uniform distribution to be a limit distribution, to which distribution of all filter weights is asymptotically approaching.

To measure how similar a given distribution is to the discrete uniform distribution, we compute Kullback–Leibler (KL) divergence between the distribution of filter weights and uniform distribution.

$$D_{KL}(p||q) = \sum_{i=1}^N p(x_i) \cdot \log \frac{p(x_i)}{q(x_i)} \quad (4)$$

By extending Lecun’s results, we proposed two hypotheses: introduction of more streams and slices will result in 1) an increase of filter weights diversity; and 2) that the distribution of all filter weights across all first layers of the Streaming Nets will gradually converge in terms KL divergence to a discrete uniform distribution.

**Distribution of the filter weights across individual conv layers.** Here we discuss distributions of filter weights for 1-stream simple CNN, 5- and 10-streams Streaming networks in the case of Eurosat data set. In Fig. 13, we illustrate the distribution of filter weight for the first conv layer of the 1-stream CNN and 5- and 10-stream Streaming Network separately for each color channel. As we hypothesized above, the KL divergence value decreases as we increase the number of channels. Therefore, the distribution of filter weights is gradually approaching discrete uniform distribution. The

<sup>5</sup>we consider values within filters as weights. To distinguish the weights within filters from weights between fully connect layer and classifier layer, we call filter values to be the filter weights.

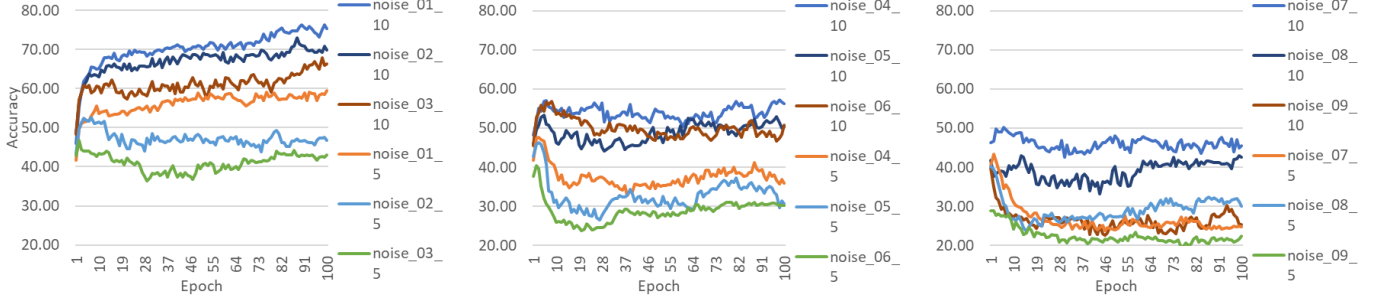


Fig. 12. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 5-streams vs. 10-stream Streaming Net. The notations is the same as in [fig. 6](#).

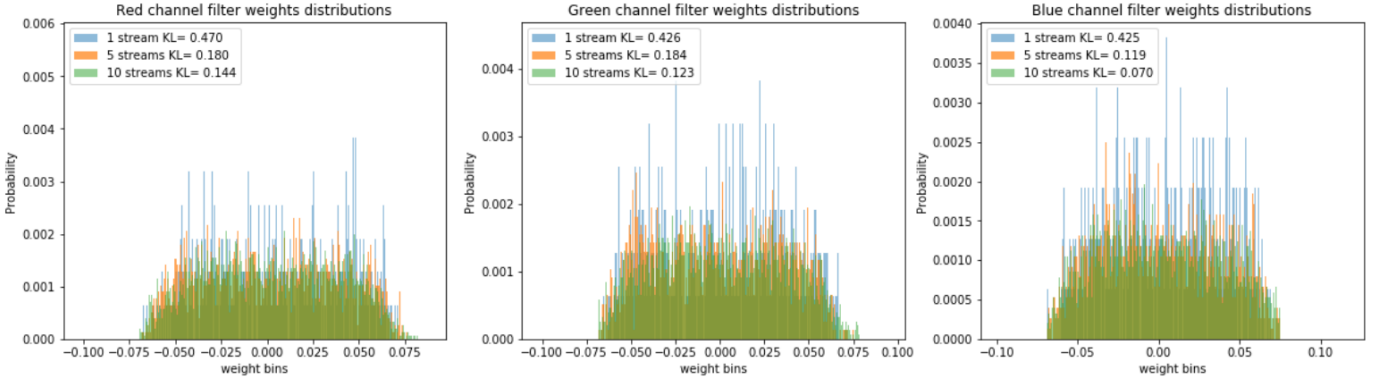


Fig. 13. Distribution of weights Eurosat dataset with 0.0001 learning rate.

Network Architecture	Capacity	Hard-wired Sparsity	Input-induced Sparsity	No Noise Accuracy Up	Noise Robustness
(1)	Lower	No	No	N/A	No
(5)	Higher	No	No	Yes	No
(6)	Higher	Yes	No	Yes	No
(7)	Higher	No	Yes	Yes	No
(8) <b>Streaming Network</b>	Higher	Yes	Yes	Yes	Yes

TABLE I  
EXPERIMENTAL RESULTS: COMPARISON OF THE NETWORK ARCHITECTURES.

distribution for 10-stream Streaming Net is extremely close to a uniform distribution in [Fig. 13](#). The weight distribution is derived from single runs of the network. In total, we have analyzed eight triplets of networks. The figures for remaining triplets are presented in Supplementary materials.

Furthermore, we compare the filter distributions across all first conv layers for each color channel with the corresponding distributions for a 1-stream wide CNN ([Fig. 15](#)), a 1-stream wide slice 15-channel CNN ([Fig. 16](#)) and a 5-streams same input CNN ([Fig. 17](#)). The results indicate that the distribution for a 5-streams Streaming Net is closer to the uniform distribution than all other distributions.

**Distribution of the filter weights in the first conv layer across each stream.** Here we compare filter weight distribution in the first conv layer of a 1-stream simple CNN vs. each stream of a 5-stream network. The results are presented in [Fig. 36](#). In this particular case, filter weights distribution

of each particular stream of 5-stream Streaming Network is more similar to the uniform distribution than the distribution for a 1-stream CNN

In general, for Streaming Networks filter weights distribution for a single stream can be more different from the uniform than the distribution for a 1-stream CNN, however, the overall distribution of weights within the layer is more similar to a uniform and this similarity increases as we increase the number of streams. We provide more examples in the Supplementary materials.

## V. DISCUSSION AND CONCLUSION

Our results indicate that Streaming Net is the only network architecture that enables noise robustness. Furthermore, for the higher capacity models, noise robustness appears only in the case when both hard-wired and input induced-sparsity are employed. However, architectures employing either type of



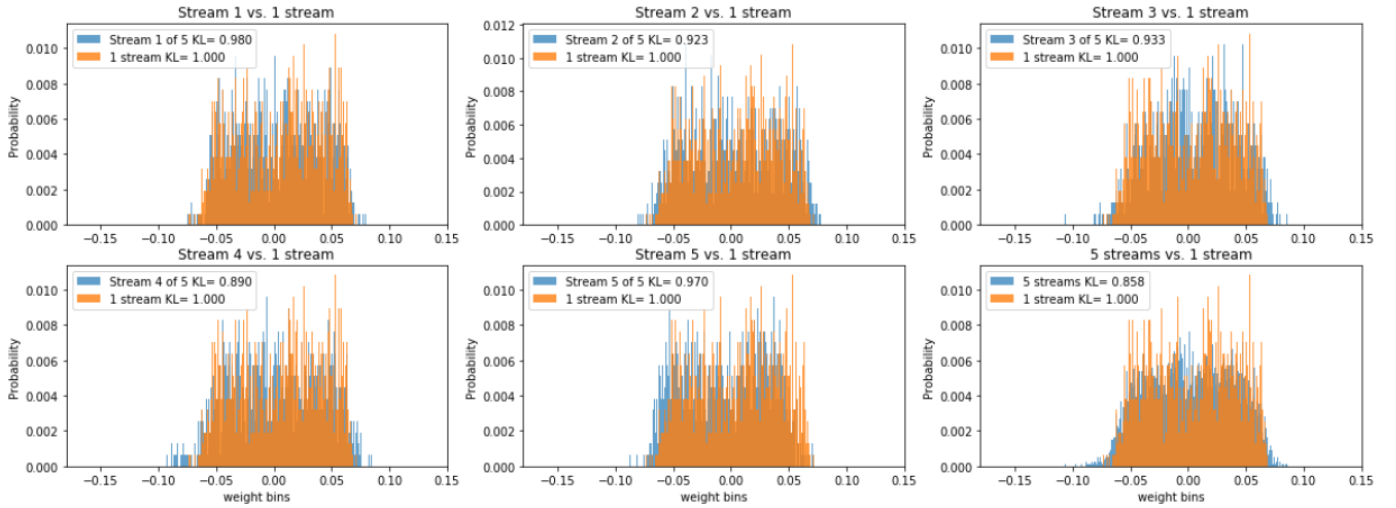


Fig. 14. Distribution of filter values across all layers for each stream obtain for Eurosat dataset with 0.0001 learning rate.

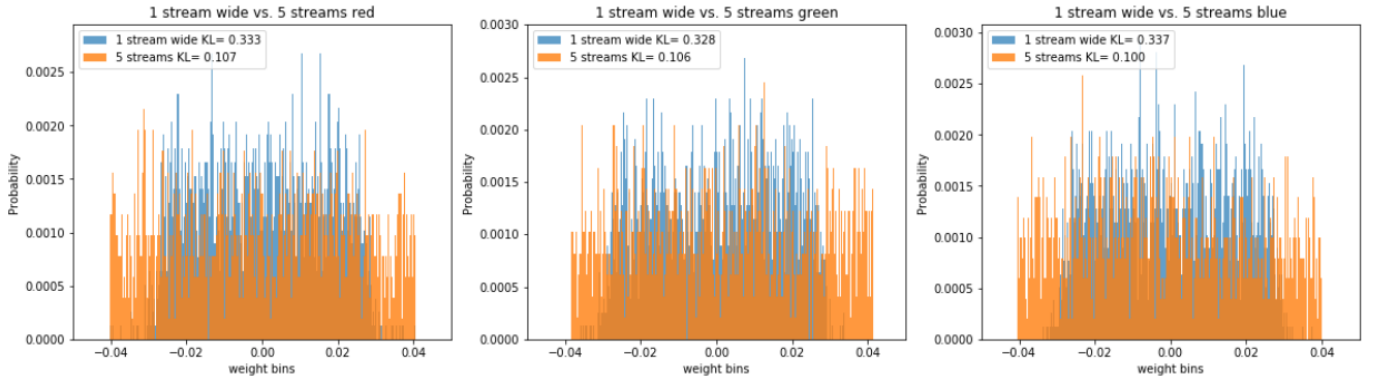


Fig. 15. 5 streams Streaming Network weights vs. 1 stream wide CNN.

sparsity exhibit no noise robustness. This is summarized in Table I.

Regarding filter weights distributions, we have illustrated that filter weights diversity increases and distribution in the first conv layer approaches uniform distribution as hard-wired (number of streams) and input-induced (number of slices) sparsity and capacity grow.

We also speculate that robust recognition of noisy data (previously unseen data) is achieved by a greater diversity of filters created by interplay of two types of sparsity.

To summarize, through a series of tests, we found that hard-wired and input-induced sparsity taken together enable noise robustness and introduce Streaming Nets as a new simple method for robust recognition of noisy images.

## REFERENCES

- [1] S. M. Azar, M. G. Atigh and A. Nickabadi (2018) A Multi-Stream Convolutional Neural Network Framework for Group Activity Recognition, *ArXiv, abs/1812.10328* 1
- [2] H. H. Aghdam and E. J. Heravi and D. Puig (2016) Analyzing the Stability of Convolutional Neural Networks against Image Degradation, *VISIGRAPP*. 1
- [3] H. H. Aghdam, E. J. Heravi and D. Puig (2016) Increasing the Stability of CNNs using a Denoising Layer Regularized by Local Lipschitz Constant in Road Understanding Problems, *VISIGRAPP*. 1
- [4] V.E. Bening and V. Yu. Korolev (2002) Generalized Poisson Models and their Applications in Insurance and Finance. *VSP*. 6
- [5] S. Chopra and R. Hadsell and Y. LeCun (2005) Learning a similarity metric discriminatively, with application to face verification, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 05)*, p. 539-546. 1
- [6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno and D. Song (2018) Robust Physical-World Attacks on Deep Learning Models, *CVPR 2018*. 1
- [7] G. Gkioxari, R. B. Girshick, P. Dollár and K. He (2018) Detecting and Recognizing Human-Object Interactions, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 8359-8367. 1
- [8] B. Hariharan and P. Andrés Arbeláez and R. B. Girshick and J. Malik (2014) Simultaneous Detection and Segmentation, *ECCV 2014* 1
- [9] K. He, X. Zhang, S. Ren and J. Sun (2015) Deep Residual Learning for Image Recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 16)*, p. 770-778. 1
- [10] P. Helber, B. Bischke, A. Dengel and D. Borth (2017) EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12, p. 2217-2226. 5, 10
- [11] G. R. Koch (2015) Siamese Neural Networks for One-Shot Image Recognition 1
- [12] A. Krizhevsky, I. Sutskever and G. E. Hinton (2012) ImageNet Classi-

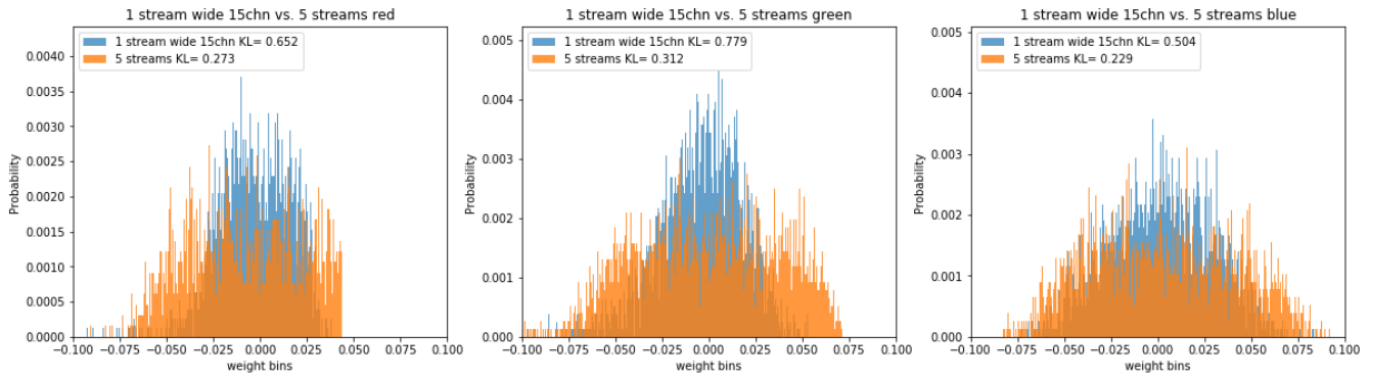


Fig. 16. 5 streams Streaming Network weights vs. 1 stream wide 15-channel CNN.

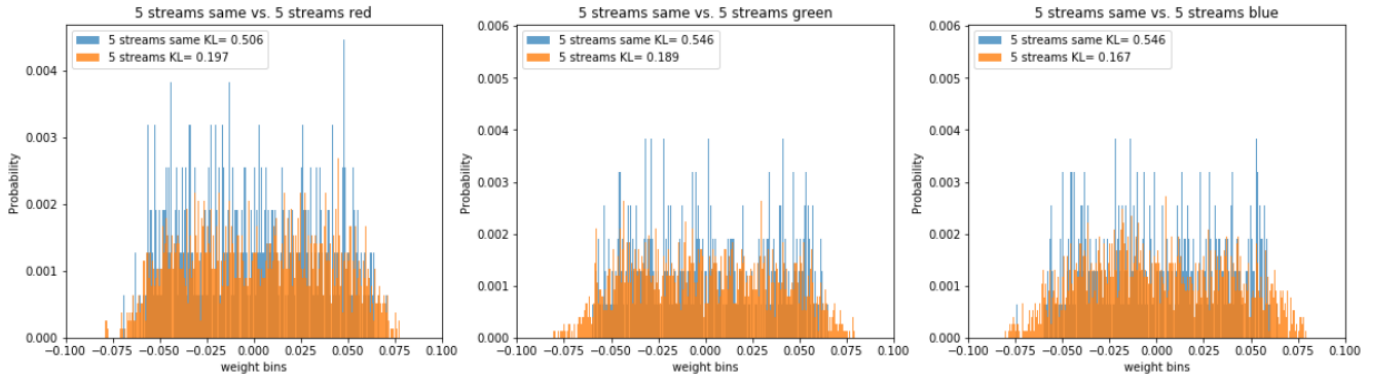


Fig. 17. 5 streams Streaming Network weights vs. 5 streams same input CNN.

- fication with Deep Convolutional Neural Networks, *NIPS 2012*. 1
- [13] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner (1998) Gradient-based learning applied to document recognition. *Proceeding of the IEEE*, 86 (11), p. 2278-2324. 1
- [14] Y. A. Lecun, L. Bottou, G. B. Orr and K.-R. Müller (1998) Efficient backprop, In *Eds. Orr, G. and Müller K. Neural Networks: Tricks of the trade*, Springer, p. 9-48. 6
- [15] Masters T. (2015) Deep Belief Nets in C++ and CUDA C: Volume 1: Restricted Boltzmann Machines and Supervised Feedforward Networks, 1st Edition. CreateSpace Independent Publishing Platform. 2
- [16] B. A. Olshausen and D. J. Field (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature*, 381, p. 607-609. 2
- [17] B. A. Olshausen and D. J. Field (1997) Sparse coding with an over-complete basis set: A strategy employed by V1?, *Vision Research*, 37, p. 3311-3325. 2
- [18] M.'A. Ranzato, Y.-L. Boureau and Y. LeCun (2007) Sparse Feature Learning for Deep Belief Networks, *NIPS*. 1
- [19] R. P. N. Rao and D. H. Ballard (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects, *Nature Neuroscience*, 2, p. 79-87. 1
- [20] R. Salakhutdinov. [http://www.cs.toronto.edu/~rsalakhu/code\\_DBM/dbm\\_mf.m](http://www.cs.toronto.edu/~rsalakhu/code_DBM/dbm_mf.m) 2
- [21] T. Serre, L. Wolf, S. M. Bileschi, M. Riesenhuber and T. A. Poggio (2007) Robust Object Recognition with Cortex-Like Mechanisms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29, p. 411-426. 1
- [22] K. Simonyan and A. Zisserman (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition, *Arxiv, abs/1409.1556*. 1
- [23] K. Simonyan and A. Zisserman (2014) Two-Stream Convolutional Networks for Action Recognition in Videos, *ArXiv, abs/1406.2199*. 1
- [24] C. Szegedy and W. Zaremba and I. Sutskever and J. Bruna and D. Erhan and I. J. Goodfellow and R. Fergus (2013) Intriguing properties of neural networks, *Arxiv, abs/1312.6199*. 1
- [25] Ch. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich (2014) Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 15)*, p. 1-9. 1
- [26] S. S. Tarasenko (2011) A General Framework for Development of the Cortex-like Visual Object Recognition System: Waves of Spikes, Predictive Coding and Universal Dictionary of Features, *Proceedings of The 2011 International Joint Conference on Neural Networks*, p. 1515-1522. 1
- [27] S. I. Thorpe and D. Fize and C. Marlot (1996) Speed of processing in the human visual system, *Nature*, 381, p. 520-522. 1
- [28] S. J. Thorpe, A. Delorme and R. van Rullen (2001) Spike-based strategies for rapid processing, *Neural networks : the official journal of the International Neural Network Society*, 14 (6-7), p. 715-725. 1
- [29] R. van Rullen and S. J. Thorpe (2002) Surfing a spike wave down the ventral stream, *Vision Research*, 42, p. 2593-2615 1
- [30] A. Varghese and J. Gubbi and A. Ramaswamy and P. Balamuralidhar (2018) ChangeNet: A Deep Learning Architecture for Visual Change Detection, *ECCV 2018 Workshops*. 1
- [31] S. Zagoruyko and N. Komodakis (2015) Learning to compare image patches via convolutional neural networks, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 15)*, p. 4353-4361.

### A. Additional Experimental Results

For our experiments we use three datasets. The selected datasets are cifar10<sup>6</sup>, Eurosat (rgb)<sup>7</sup> [10] and UCmerced land use<sup>8</sup>.

#### B. Cifar10

Cifar10 dataset contains RGB 32x32 images of 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). The total number of images is 60,000 with 6,000 images for each class. To train and test the networks, we use 50,000 and 10,000 images respectively.

The results of performance comparison are presented as follows:

- 1-stream simple conv net vs. 5-stream Streaming Net (Fig. 6);
- 1-stream wide conv net vs. 5-stream Streaming Net (Fig. 18);
- 1-stream slice 15-channel wide conv net vs. 5-stream Streaming Net (Fig. 19);
- 5-stream the same inputs conv net vs. 5-stream Streaming Net (Fig. 20).

#### C. Eurosat

Eurosat dataset contains Sentinel-2 satellite images covering both 13 spectral bands and RGB (3-channel) 64x64 images and consisting of 10 classes (AnnualCrop, Forest, Herbaceous, Vegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake) with in total 27,000 labeled and geo-referenced images. We use only RGB images for this study.

The results of performance comparison are presented as follows:

- 1-stream simple conv net vs. 5-stream Streaming Net (Fig. 7);
- 1-stream wide conv net vs. 5-stream Streaming Net Fig. (Fig. 21);
- 1-stream slice 15-channel wide conv net vs. 5-stream Streaming Net (Fig. 22);
- 5-stream the same inputs conv net vs. 5-stream Streaming Net (Fig. 23);
- 5-stream vs. 10-stream Streaming Net (Fig. 24).

#### D. UCmerced

The UCmerced land-use dataset contains 256x256 RGB satellite images of 21 class of land use. There are 100 images for each of the following classes: agricultural, airplane, baseballdiamond, beach, buildings, chaparral, denseresidential, forest, freeway, golfcourse, harbor, intersection, mediumresidential, mobilehomepark, overpass, parkinglot, river, runway, sparseresidential, storagetanks, tenniscourt.

The results of performance comparison are presented as follows:

- 1-stream simple conv net vs. 5-stream Streaming Net (Fig. 8);
- 1-stream wide conv net vs. 5-stream Streaming Net Fig. (Fig. 25);
- 1-stream slice 15-channel wide conv net vs. 5-stream Streaming Net (Fig. 26);
- 5-stream the same inputs conv net vs. 5-stream Streaming Net (Fig. 27).

### E. Additional Experimental Results: Filter Weights Distributions for the first conv layer using Eurosat dataset

In this section, we introduce the results of the analysis of distributions of filter weights after training neural networks using Eurosat dataset. We introduce results for both overall filter weights distributions and stream-to-stream distributions comparison. Overall filter weights distributions imply that all the filter weights are collected from all the streams and filters of the first conv layers. Stream-to-stream distributions comparison means that we compare all the filter weights from all the filters in the first conv layer of a 1-stream simple conv net vs. all the filter weights from all the filters in the first conv layer of for each stream of the corresponding Streaming Network.

#### Distribution of the filter weights across individual conv layers

Here we discuss distribution of filter weights for a 1-stream simple conv net, 5- and 10-stream Streaming networks in the case of Eurosat data set.

We performed an analysis of 8 different sets of weights for 1-stream simple conv net, 5- and 10-stream Streaming Net. We illustrate the distribution of filter weight for the first conv layer of the 1-stream conv net and 5- and 10-stream Streaming Network separately for each color channel for each of 8 triplets in Figs. 28 through 35. In each figures (a), (b) and (c) sections introduce distributions of weights for red, green and blue channels, respectively.

One can infer that KL divergence value decreases as we increase the number of streams. Therefore, the distribution of filter weights is gradually approaching a discrete uniform distribution.

**Distribution of the filter weights in the first conv layer across each stream** Here we compare filter weight distribution in the first conv layer of a 1-stream simple conv net, each stream of 5- and 10-stream Streaming Network. The results of the comparison between 1-stream simple conv net and 5-stream Streaming Net are presented in Fig. 36. The results of the comparison between 1-stream simple conv net and 10-stream Streaming Net are presented in Fig. 37. In Figs. 36 and 37, we introduce a stream-to-stream comparison only for red channel for one of the tests.

Figs. 36(a)-(e) and Figs. 37(a)-(j) introduce stream-to-stream comparison of filter weights distributinos between each stream of 5-stream Streaming Net and a 1-stream simple conv net, and 10-stream Streaming Net and a 1-stream simple conv net, respectively.

<sup>6</sup><https://www.cs.toronto.edu/~kriz/cifar.html>

<sup>7</sup><https://github.com/phelber/eurosat>

<sup>8</sup><http://weegee.vision.ucmerced.edu/datasets/landuse.html>

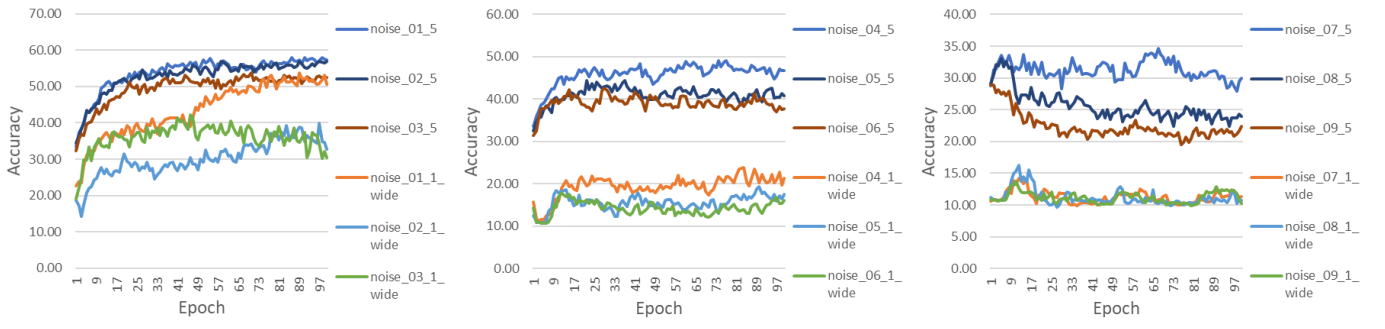


Fig. 18. Tests of Cifar10 dataset using Adam optimizer with 0.0001 learning rate for 1-stream wide conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

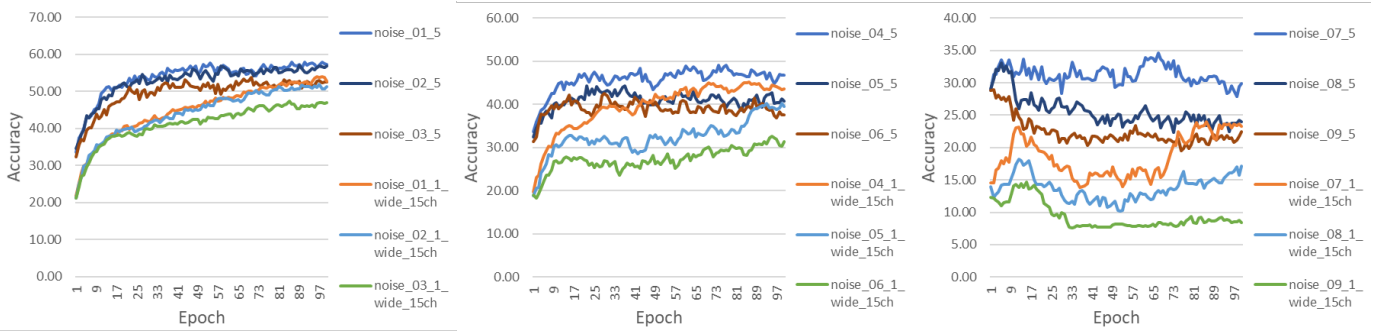


Fig. 19. Tests of Cifar10 dataset using Adam optimizer with 0.0001 learning rate for 1-stream slice 15-channel wide conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

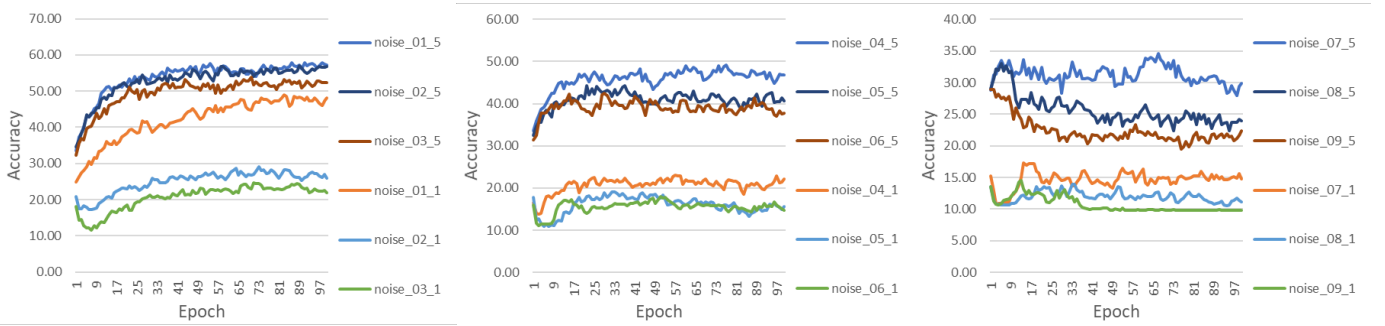


Fig. 20. Tests of Cifar10 dataset using Adam optimizer with 0.0001 learning rate for 5-stream the same inputs conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

In general, filter weights distribution of each particular stream of 5-stream Streaming Network is more similar to the uniform distribution than the distribution for a 1-stream conv net. However, for 10-stream Streaming Net there are cases when filter weights distribution of particular single stream is more different from the uniform distribution than the distribution for 1-stream network.

Nevertheless, the overall weight distribution across all the streams for both 5- and 10-stream Streaming Nets are more similar to the uniform distribution than a corresponding distribution derived from 1-stream simple conv net as shown in Fig. 36(f) and Fig. 37(k), respectively.

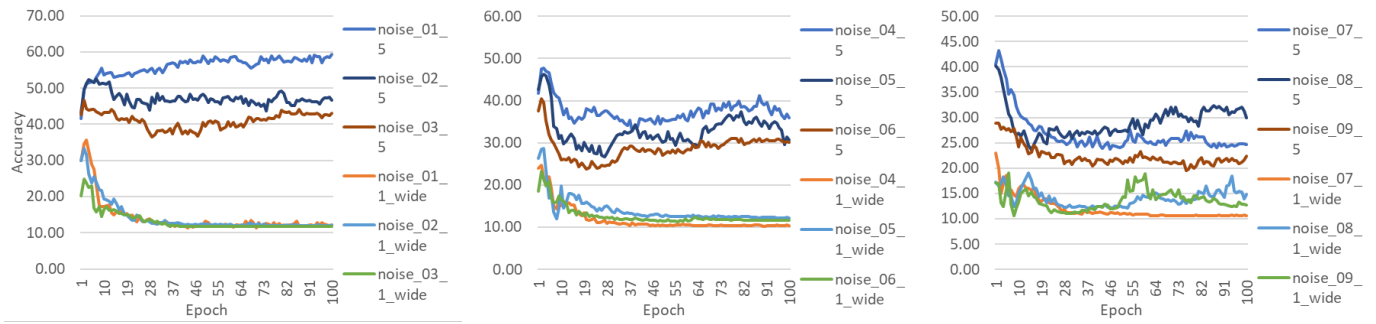


Fig. 21. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 1-stream wide conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

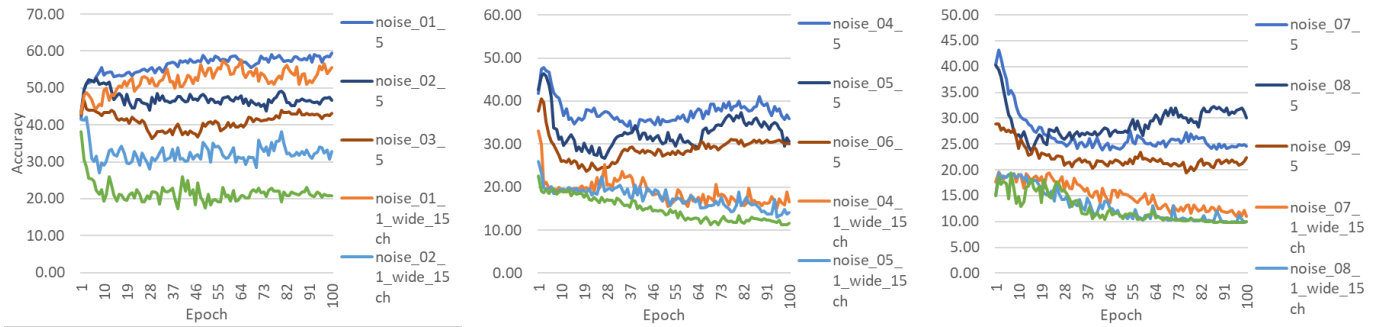


Fig. 22. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 1-stream 15-channel wide conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

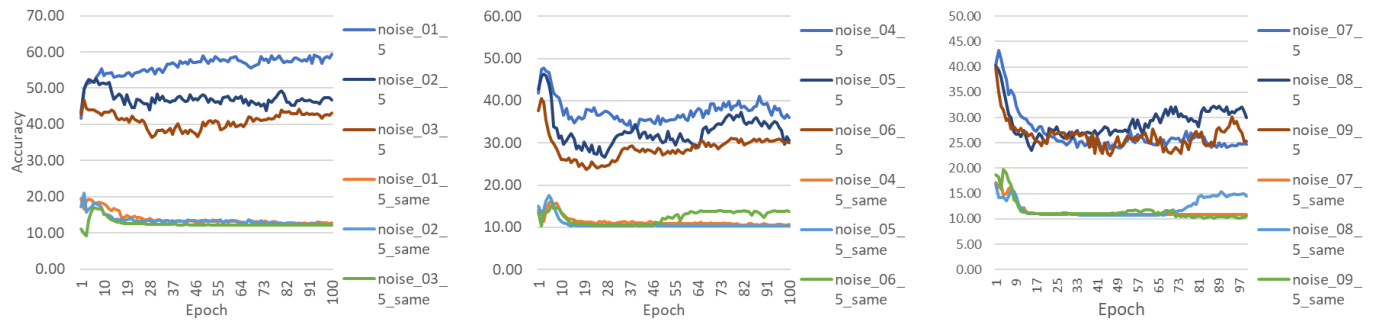


Fig. 23. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 5-streams with the same inputs conv net vs. 5-stream Streaming Net. The notations are the same as in Fig. 6.

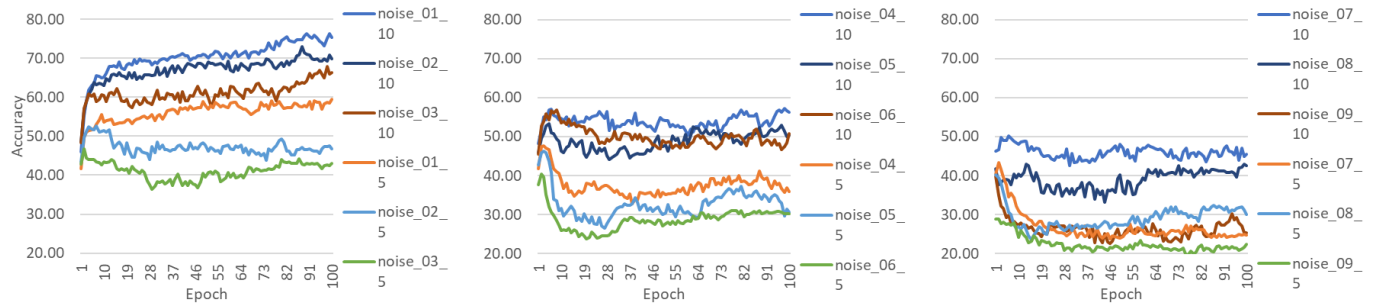


Fig. 24. Tests of Eurosat dataset using Adam optimizer with 0.0001 learning rate for 5-streams vs. 10-stream Streaming Net. The notations is the same as in Fig. 6.



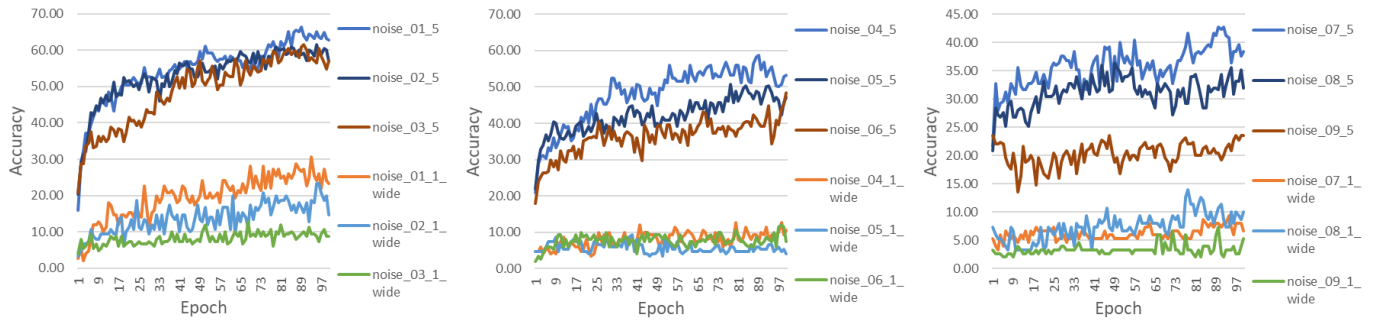


Fig. 25. Tests of UCmerced dataset using Adam optimizer with 0.0001 learning rate for 1-stream wide conv net vs. 5-stream Streaming Network. The notations are the same as in Fig. 6

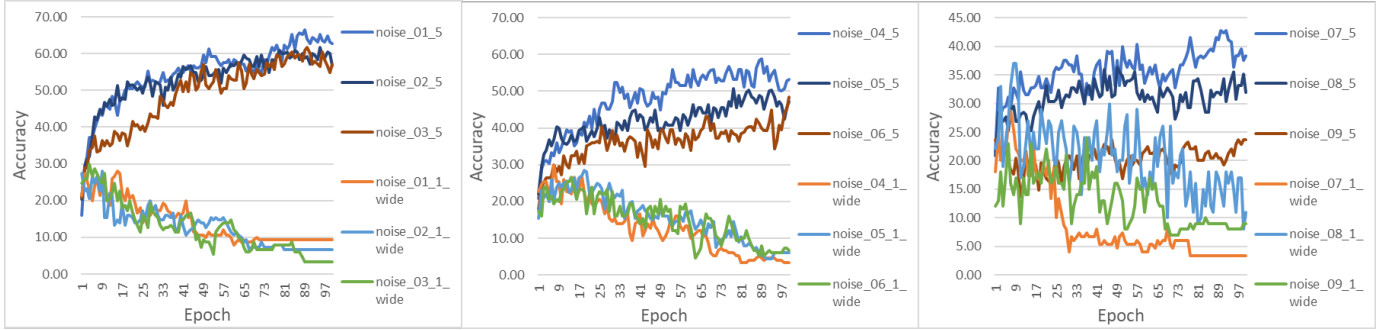


Fig. 26. Tests of UCmerced dataset using Adam optimizer with 0.0001 learning rate for 1-stream slice 15-channel wide conv net vs. 5-stream Streaming Network. The notations are the same as in Fig. 6

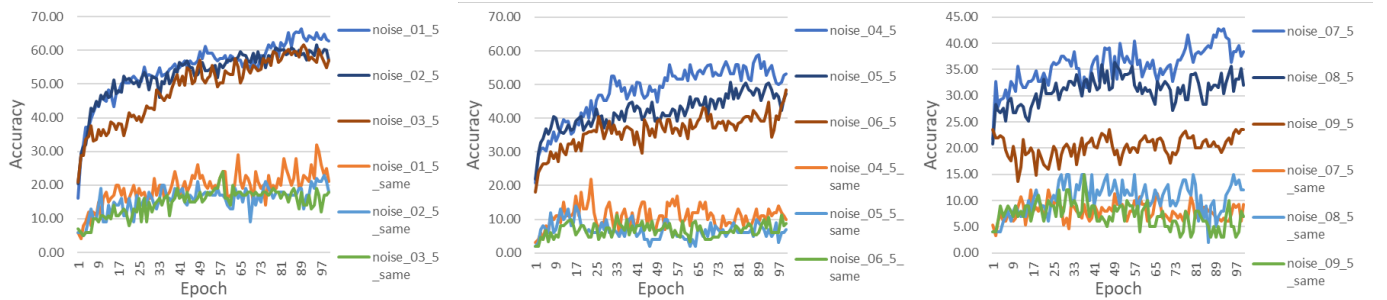


Fig. 27. Tests of UCmerced dataset using Adam optimizer with 0.0001 learning rate for 5-stream with the same input conv net vs. 5-stream Streaming Network. The notations are the same as in Fig. 6

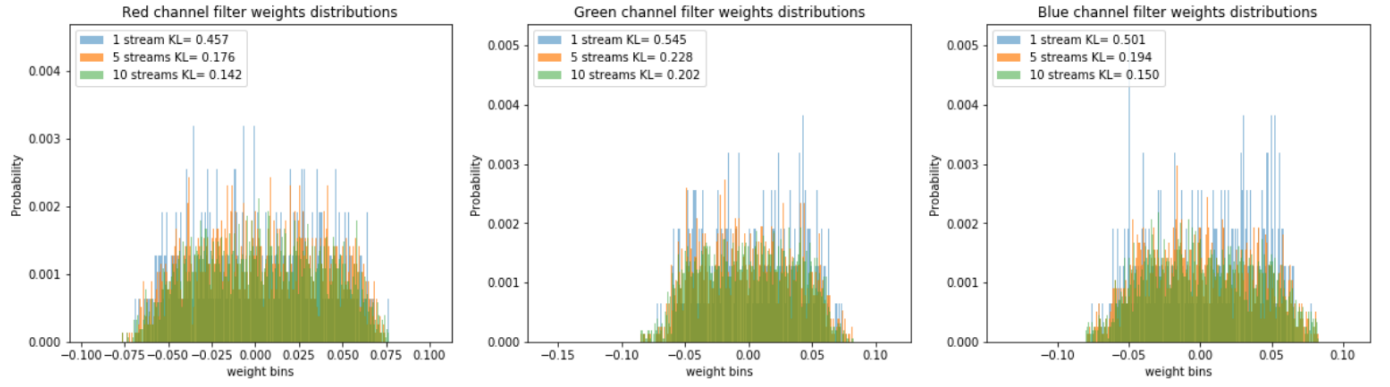


Fig. 28. Test 1: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notation: “1 stream” implies 1-stream simple conv net, “5 streams” implies 5-stream Streaming Network, “10 streams” implies 10-stream Streaming Network. KL is a short for Kullback–Leibler divergence.

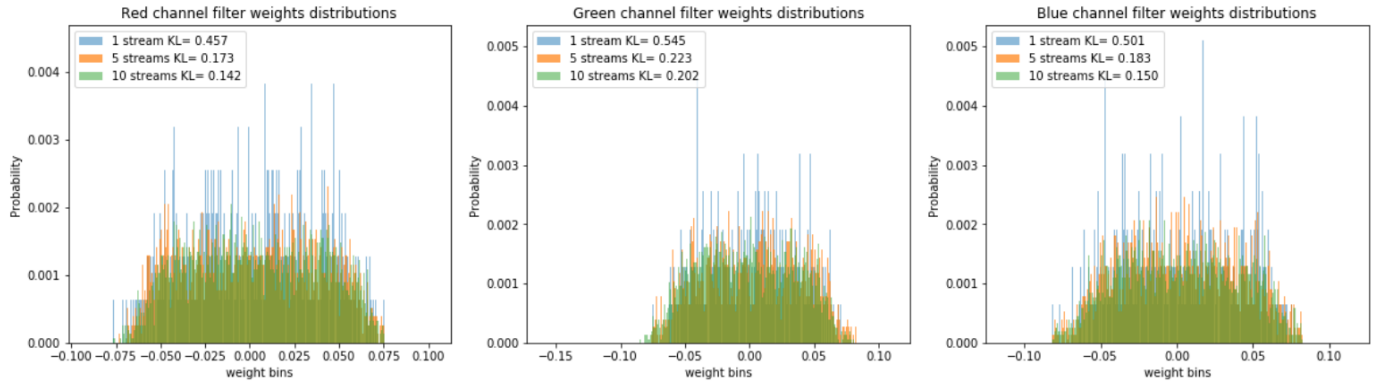


Fig. 29. Test 2: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

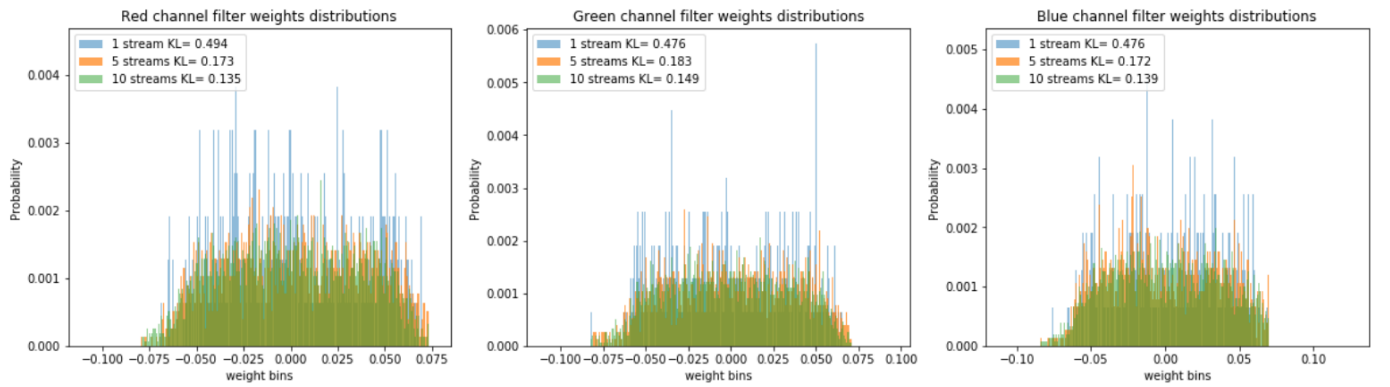


Fig. 30. Test 3: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

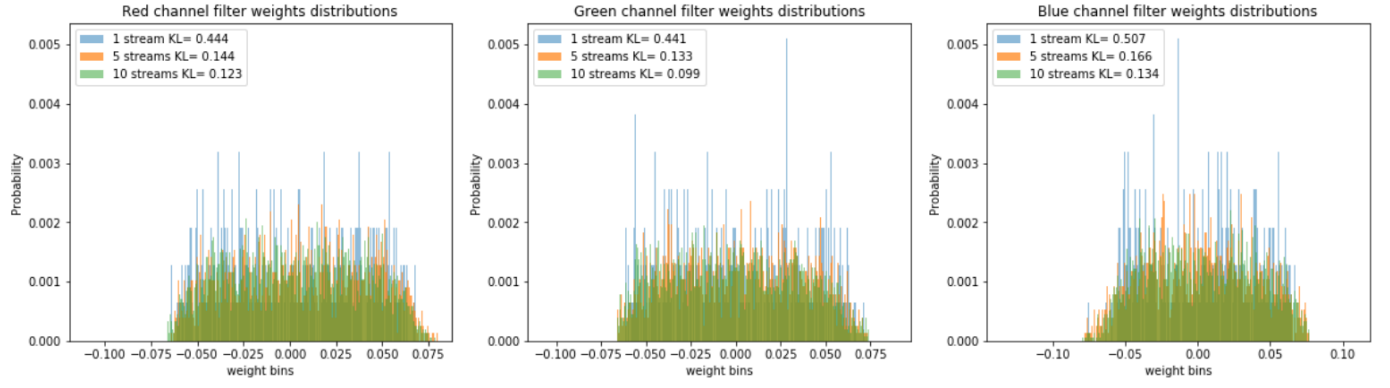


Fig. 31. Test 4: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

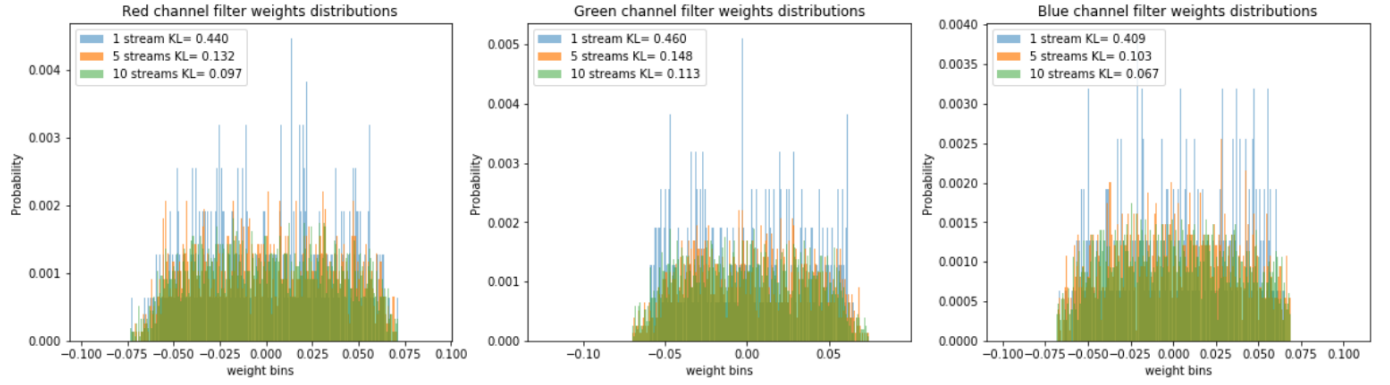


Fig. 32. Test 5: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

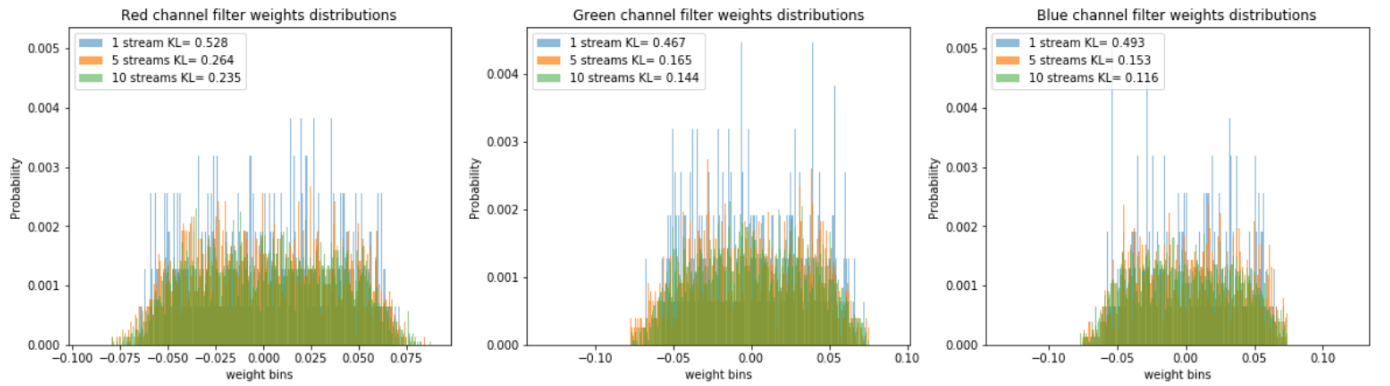


Fig. 33. Test 6: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

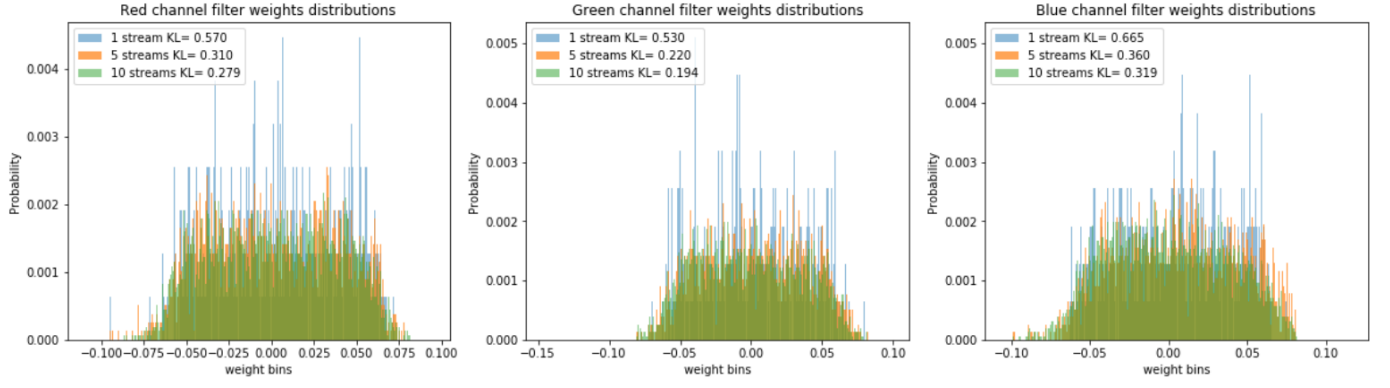


Fig. 34. Test 7: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

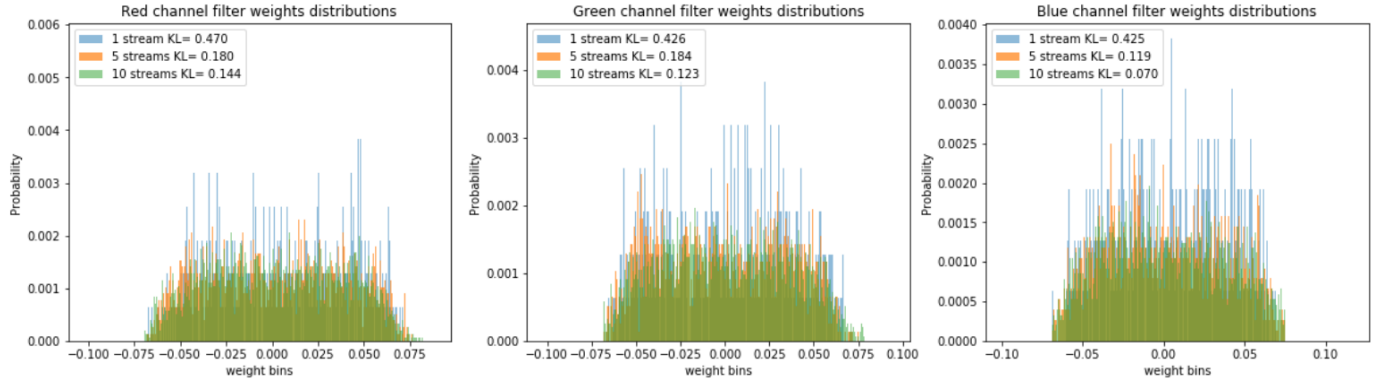


Fig. 35. Test 8: the overall distribution of the filter weights across all the layers and filters of the first conv layers for Eurosat dataset. Notations are the same as in Fig. 28.

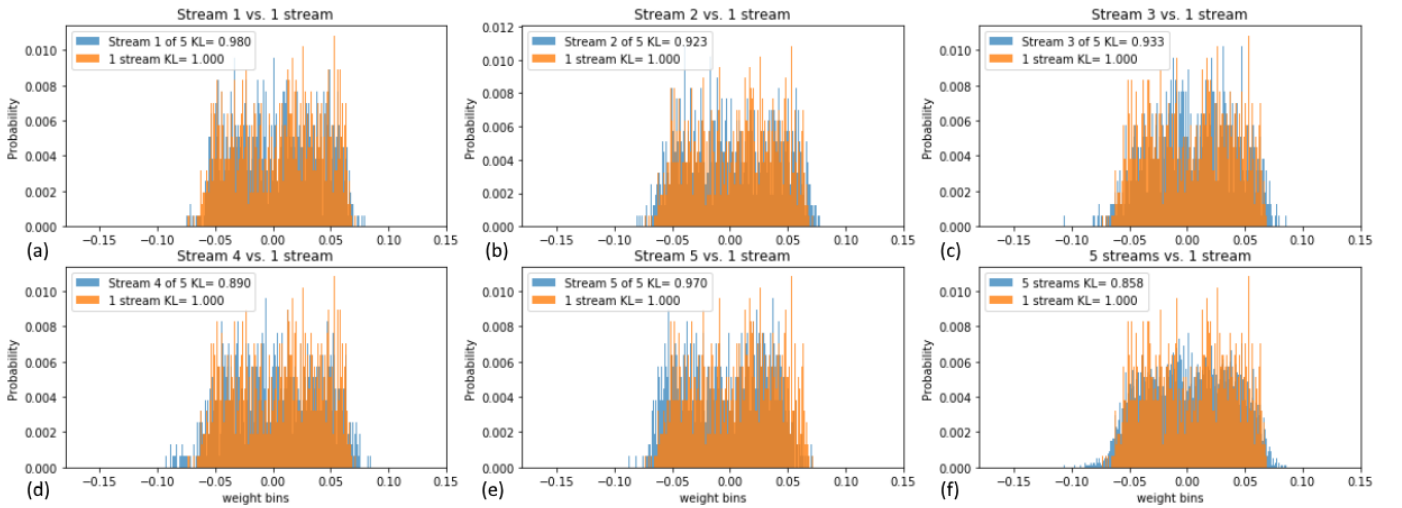


Fig. 36. Distribution of filter weights across all layers for each stream of 5-stream Streaming Net vs. a 1-stream simple conv net obtained for Eurosat dataset.

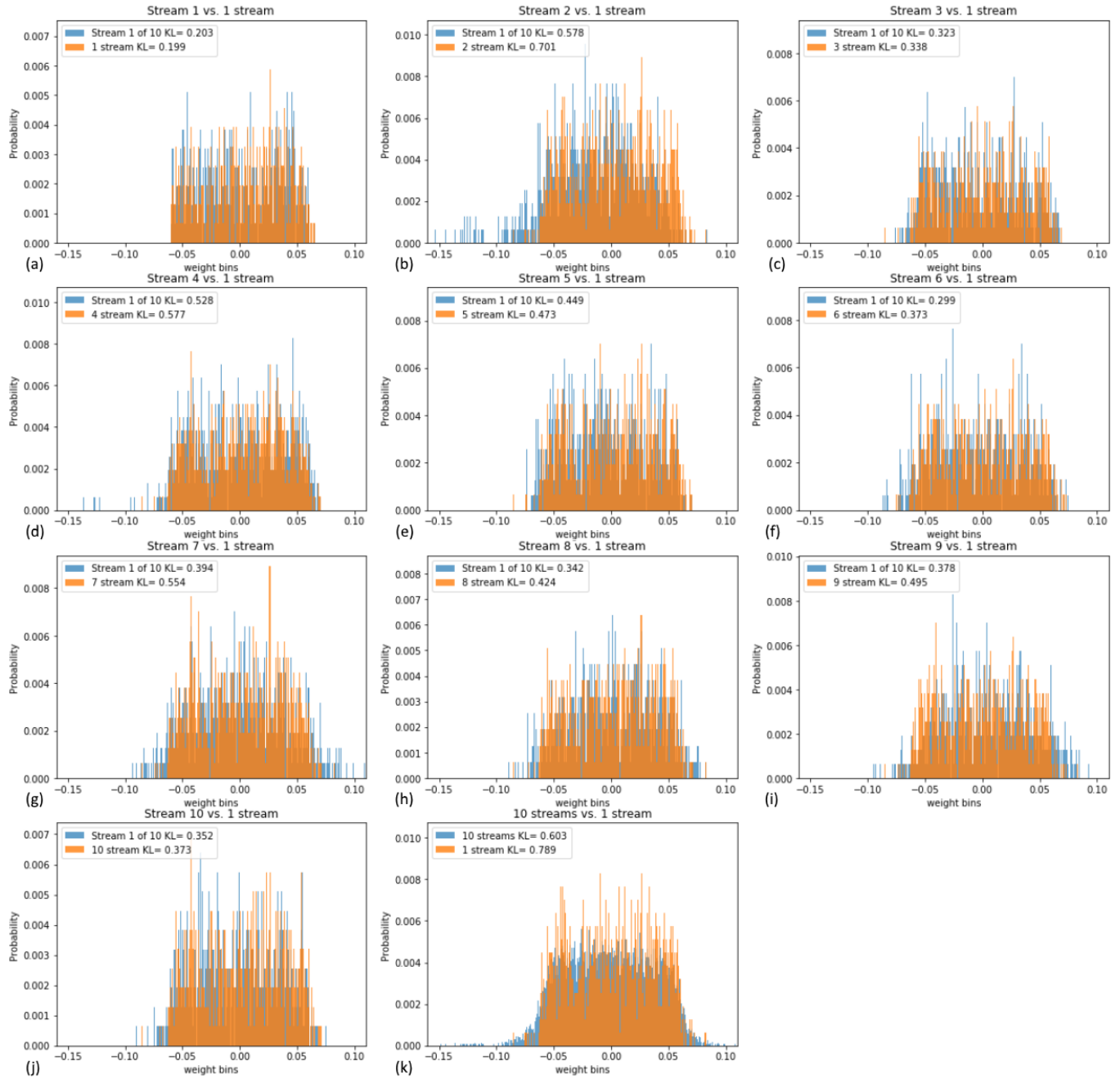


Fig. 37. Distribution of filter weights across all layers for each stream of 10-stream Streaming Net vs. a 1-stream simple conv net obtained for Eurosat dataset.