# TRAINABLE CO-OCCURRENCE ACTIVATION UNIT FOR IMPROVING CONVNET

*Takumi Kobayashi*

National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Ibaraki, Japan

## ABSTRACT

A deep neural network is one of the promising approach to produce state-of-the-art performance on various fields such as pattern recognition and signal processing. While the network architecture is intensively studied, as to the network components, non-linear activation functions are the main subject of research in the literature. Most of the activation functions, such as a rectified linear unit (ReLU), operate on each of feature channels in an element-wise manner and thus can be regarded as extracting *occurrence* characteristics from the input feature map. In this paper, we propose a *co-occurrence* activation unit to work across feature channels by extending the element-wise activation function. In contrast to the original co-occurrence formulation applied to hand-crafted feature extraction methods, the proposed co-occurrence unit is trainable by a gradient-based optimization through back-propagation learning and exploits the co-occurrence relationships among the feature channels. The experimental results on image classification datasets show that the proposed co-occurrence activation unit embedded into various types of ConvNets favorably improve classification performance.

***Index Terms—*** Neural network, Non-linear activation function, Co-occurrence

## 1. INTRODUCTION

Deep neural networks (DNN), such as deep convolutional neural networks (ConvNets), have been making great impact on pattern recognition and signal processing fields [1, 2, 3, 4]. The success of DNN lies in the end-to-end learning for the network parameters, though the network architecture is manually determined, which highly contrasts with the hand-crafted features, such as histogram of oriented gradients (HOG) [5].

The neural networks are basically built on several types of layers, such as linear projection (convolution), non-linear activation function and pooling. While the network architecture is intensively studied and various types of networks are proposed [6, 7], as to the network components, non-linear activation functions would be the main research subject. The legacy activation function is a sigmoid function, but it has the problem of vanishing gradients. Therefore, to cope with that problem, the rectified linear unit (ReLU) [8], which is currently the most popular activation function, and its variants [9, 10, 11, 12, 13] are proposed.

Such an activation function operates on each of feature channel, and it could be regarded as extracting the *occurrence* (mono-activation) of the input feature channel; the ReLU exploits the occurrence of the feature channel whose value exceeds zero. The local response normalization unit [1] accepts multiple feature channels, but it just normalizes the feature channels. Therefore, there might be room to further extend the activation function so as to work on multiple channels, and from the perspective that ReLU is related to occurrence, our goal is to extract *co-occurrence* across feature channels by an activation function.

Turning back into the hand-crafted image feature extraction, the occurrence characteristics are measured in the form of histogram such as by HOG [5], and the co-occurrence features are also well studied [14, 15, 16, 17, 18]. In those methods, the co-occurrence patterns are *hand-crafted*, usually exploiting all the combinations of feature channels in a form of rather higher dimensional features; the dimensionality is polynomially increased according to the number of input feature channels. To remedy the high dimensionality, a feature selection scheme is commonly adopted to reduce the feature dimensionality such as via AdaBoost [17, 18]; in most cases, the feature selection proceeds incrementally by picking up the informative one from the hand-crafted patterns. Thus, we can say that learning co-occurrence patterns from scratch, not selecting, is difficult and challenging.

In this study, we propose a novel activation unit based on co-occurrence across feature channels. In contrast to the hand-crafted co-occurrence feature extraction, the proposed co-occurrence unit is *trainable* in the framework of end-to-end learning which is the primary factor for successful DNNs; it is actually learned by a *gradient*-based optimization in the back-propagation. Note that, on the other hand, the feature selection approaches to learn the hand-crafted co-occurrence features [17, 18] are not compatible with the back-propagation in DNNs since the combinatorial selection process is not differentiable. For realizing the trainability, we rebuild the process to extract co-occurrence from the input feature map through decomposing the original formulation into three layers in a neural network manner and relaxing them to obtain the pseudo co-occurrence extraction.

## 2. TRAINABLE CO-OCCURRENCE UNIT

We first present a formulation to extract co-occurrence which is applied to hand-crafted features, and then propose the co-occurrence activation unit (CoOU) to be trainable and thus be embedded in neural networks through end-to-end learning.

### 2.1. Co-occurrence feature extraction

Suppose we have a (primitive) non-negative feature map $\boldsymbol{X} \in \mathbb{R}_+^{H\times W\times C}$ extracted from an input image where $H$ and $W$ are height and width of the map and $C$ indicates the number of feature channels. The co-occurrence features are generally computed on the map $\boldsymbol{X}$ by

$$F_{h,w,k} = \beta_k \prod_{(\Delta_h,\Delta_w,c)\in\mathcal{P}_k} X_{h+\Delta_h,w+\Delta_w,c}, \tag{1}$$

where $\beta_k\,(>0)$ is the positive scaling parameter, $\mathcal{P}_k$ indicates the $k$-th co-occurrence pattern consisting of the displacement position $(\Delta_h,\Delta_w)$ relative to the reference position $(h,w)$ and the feature channel $c$ to participate in the co-occurrence; for example, the co-occurrence of the channels $c$ and $c'$ at the same position is extracted by the pattern $\mathcal{P}=\{(0,0,c),(0,0,c')\}$. The $K$-dimensional co-occurrence features are produced by the $K$ patterns $\{\mathcal{P}_k\}_{k=1}^K$ which are defined *manually* in the hand-crafted feature extraction methods [14, 15, 16] where local (small $|\Delta_h|$ and $|\Delta_w|$) and two-point co-occurrence ($|\mathcal{P}_k|=2$) are usually considered for extracting all the $C^2$ combinations of the feature channels. A variety of co-occurrence features are extracted according to the types of the feature map $\boldsymbol{X}$ [14, 15, 16], and the co-occurrence feature is finally computed by pooling $F_{h,w,k}$ over the region of interest by $f_k=\sum_{(h,w)\in\mathcal{D}_{ROI}} F_{h,w,k}$.

### 2.2. Decomposition of co-occurrence formulation

For inducing tractable co-occurrence formulations, we rewrite (1) into the following form;

$$\begin{aligned} F_{h,w,k} &= \exp\Big[\log(\beta_k) + \sum_{(\Delta_h,\Delta_w,c)\in\mathcal{P}_k} \log(X_{h+\Delta_h,w+\Delta_w,c})\Big] \\ &= \exp\Big[b_k + \sum_{(\Delta_h,\Delta_w,c)\in\mathcal{N}} w^k_{\Delta_h,\Delta_w,c}\log(X_{h+\Delta_h,w+\Delta_w,c})\Big], \\ & s.t.\ X_{h,w,c}\geq 0\ \ \forall(h,w,c), \end{aligned} \tag{2}$$

where we regard $\log(0)=-\infty$ and $\exp(-\infty)=0$, and introduce the bias $b_k=\log(\beta_k)$ and the weight $\boldsymbol{w}^k$ such that $w^k_{\Delta_h,\Delta_w,c}=1$ for $(\Delta_h,\Delta_w,c)\in\mathcal{P}_k$, otherwise $w^k_{\Delta_h,\Delta_w,c}=0$, and $\mathcal{N}$ indicates a set of neighborhood patterns, $\mathcal{N}\supseteq\mathcal{P}_k\ \forall k$, on which the weight $\boldsymbol{w}^k$ is defined; for example, spatially local co-occurrence of $|\Delta_h|=|\Delta_w|=1$ shapes the weight $\boldsymbol{w}^k$ into $3\times 3\times C$ tensor, while that of no spatial extent $|\Delta_h|=|\Delta_w|=0$ produces the weight of $1\times 1\times C$.

The formulation (2) reveals that the co-occurrence feature extraction is composed of the following three steps; the input $\boldsymbol{X}$ is (i) non-linearly transformed via $\log$, (ii) convolved with the binary filter $\boldsymbol{w}^k$ which encodes the co-occurrence pattern, and finally (iii) non-linearly transformed via $\exp$. Therefore, this reformulation enables us to compute the co-occurrence features by *three layers* all of which are standard in neural networks, *i.e.*, one convolution and two non-linear activation layers, but note that it is only the case of feed-forward path.

### 2.3. Differentiable co-occurrence formulation

As shown in Sec. 2.2, learning co-occurrence features is reduced to an optimization of the binary filter $\boldsymbol{w}^k\in\{0,1\}^{|\mathcal{N}|}$, though having two difficulties. First, it results in a combinatorial optimization to search which filter weights should be activated, being regarded as an NP-hard problem. The second difficulty is that $\log(x)$ function is not differentiable at $x=0$, making it hard to apply the gradient-based optimization which is necessary in back-propagation to train the neural networks. Therefore, for embedding the co-occurrence extraction into the neural networks in the framework of end-to-end learning, it is required to approximate the formulation (2) so as to cope with the above-mentioned difficulties.

Our approximation is first applied to the filter $\boldsymbol{w}^k$ through relaxation into $\boldsymbol{w}^k\in[0,1]^{|\mathcal{N}|}\ s.t.\ \mathbf{1}^\top\boldsymbol{w}^k=1$; the filter weight is linearly relaxed and the unit-sum constraint is imposed for avoiding the exponential explosion of the feature magnitude. Second, the two non-linear functions $\log$ and $\exp$ are approximated by their upper/lower-bound functions represented in the following piece-wise linear forms:

$$\log(x)\approx \mathtt{l}(x)=\begin{cases} x-1 & x\geq\frac{\log(\alpha)}{\alpha-1} \\ \alpha x-1-\log(\alpha) & x<\frac{\log(\alpha)}{\alpha-1}\end{cases}, \tag{3}$$

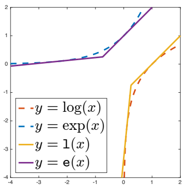
$$\exp(x)\approx \mathtt{e}(x)=\begin{cases} x+1 & x\geq\frac{\log(\alpha)}{\alpha-1}-1 \\ \frac{1}{\alpha}x+\frac{1+\log(\alpha)}{\alpha} & x<\frac{\log(\alpha)}{\alpha-1}-1\end{cases}, \tag{4}$$

where $\alpha>1$ is the fixed parameter for the slope, say $\alpha=10$ in this study; these functions are depicted in Fig. 1a. Applying these approximations to the co-occurrence extraction (2), we obtain the *pseudo* co-occurrence activation function as

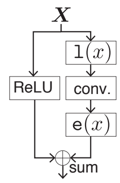
$$\hat{\boldsymbol{F}}_k=\mathtt{e}\Big[\boldsymbol{w}^k * \mathtt{l}(\boldsymbol{X})+b_k\Big],\quad s.t.\ \boldsymbol{w}^k\geq 0,\ \mathbf{1}^\top\boldsymbol{w}^k=1, \tag{5}$$

where the functions $\mathtt{l}$ and $\mathtt{e}$ operate elementwisely and $*$ is the convolution operator. Note that the proposed co-occurrence formulation accepts *any* feature map $\boldsymbol{X}$ of which value is even negative; due to the functionality of $\mathtt{l}$, the negative input significantly brings down the output of $\boldsymbol{w}*\mathtt{l}(\boldsymbol{X})$, which eventually results in the smaller output $\hat{F}$ close to zero as is the case with the co-occurrence containing zero input. This formulation facilitates the end-to-end learning since the input feature map $\boldsymbol{X}$ can also be optimized via $\frac{\partial}{\partial\boldsymbol{X}}\hat{\boldsymbol{F}}$. The constrained filter weights $\boldsymbol{w}^k$ are updated via SGD as

$$\tilde{\boldsymbol{w}}=\boldsymbol{w}^k+(\boldsymbol{I}-\frac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top)\boldsymbol{\nabla},\ \boldsymbol{w}^k\leftarrow\frac{\max[\tilde{\boldsymbol{w}},0]}{\sum_{\Delta_h,\Delta_w,c}\max[\tilde{w}_{\Delta_h,\Delta_w,c},0]}, \tag{6}$$



(a) Approximated functions




(b) Co-occurrence unit (CoOU)

**Fig. 1**. The approximated functions $\mathtt{l}(x)$ and $\mathtt{e}(x)$ in (a) compose the proposed co-occurrence unit (CoOU) in (b).

where $\boldsymbol{I}$ is an identity matrix, $\mathbf{1}$ is a vector whose elements are all 1, and $\boldsymbol{\nabla}^k$ indicates the updating vector[1] for $\boldsymbol{w}^k$ produced such as by a SGD method based on the learning rate, (back-propagated) gradient w.r.t $\boldsymbol{w}^k$, momentum and so on. The above update formula ensures the constrained filter $\boldsymbol{w}$ satisfying non-negativity and unit-sum.

As in the hand-crafted co-occurrence features [14, 15, 16], it is preferable to take into account not only the (higher-order) co-occurrence features but also the simple occurrence ones. In the neural network of our interest, the occurrence features are simply extracted by ReLU [8] and can be combined with the proposed co-occurrence activation function (5) by means of the *residual* model [7] (Fig. 1b), in which the number of the co-occurrence patterns is the same as that of the input feature channels, $K = C$. As discussed in [7], this co-occurrence activation unit (*CoOU*) makes it possible to effectively extract the higher-order co-occurrence characteristics which are *residual* compared to the occurrence based on ReLU.

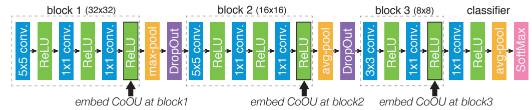
For training CoOU, the co-occurrence function (5) is simply initialized as an *identity* mapping; $w^k_{\Delta_h=0,\Delta_w=0,k}=1$, otherwise 0.

## 2.4. Connection to Related Works

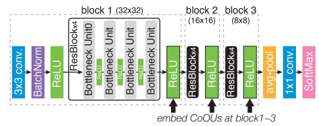
The layers of pseudo log $\mathtt{l}(x)$ and pseudo exp $\mathtt{e}(x)$ are actually variants of leaky ReLU [11, 12]; $\mathtt{l}(x)$ uses significantly leaky slope ($\alpha = 10$) while $\mathtt{e}(x)$ employs similar leaky slope ($1/\alpha = 0.1$) to those used in [11, 12]. While those ReLUs have been applied mainly for remedying the gradient vanishing problem in the sigmoid-based activation unit and due to the biological insights [19], we derive $\mathtt{l}(x)$ and $\mathtt{e}(x)$ from a co-occurrence feature extraction process. Thus, the ReLU can be interpreted from the viewpoint of the occurrence/co-occurrence. Note that the whole CoOU (Fig. 1b) exhibits clear difference from the ordinary ReLUs; the pair of large ($\alpha$) and small ($1/\alpha$) leaky ReLUs have not been applied.

Our co-occurrence function is also related to MEX function implemented in SimNet [20] which consists of $\log$ and $\exp$ functions. The main difference is the order of those non-linear functions; in MEX, $\exp$ is first applied to the input feature map $\boldsymbol{X}$ and then $\log$ is finally applied to provide the functionality similar to $\max$ [20]. It is far from the co-occurrence characteristics and a different feature extraction process from ours. In addition, those non-linear functions are directly applied without any approximation like $\mathtt{l}(x)$ and $\mathtt{e}(x)$, inducing unfavorable convergence in learning ConvNets which are different architectures from SimNets, as shown in Sec. 3.1.

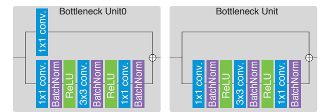
[1]Here, the tensor wight $\boldsymbol{w}^k$ is unfolded into a vector of $\mathbb{R}^{|\mathcal{N}|\times 1}_+$.




(a) NiN [2]




(b) ResNet-38 [7]




(c) Building components of ResNet

**Fig. 2**. ConvNet architectures used in the experiment.

## 3. EXPERIMENTAL RESULTS

The proposed CoOU is applied to ConvNets on image classification tasks. In the experiment, we apply the simplest CoOU that has no spatial extent by $\{\boldsymbol{w}^k \in \mathbb{R}^{1\times 1\times C}\}_{k=1}^C$; the convolution layer in the CoOU (5) results in $1 \times 1$ convolution. We implement all the ConvNets by using MatConvNet toolbox [21] and basically follow the learning parameters provided in the toolbox for training the networks from scratch.

### 3.1. Performance analysis on NiN

We analyze the performance of the CoOU from various aspects, by embedding it into Network-in-Network (NiN) model [2] on Cifar100 dataset [22]. The performance is evaluated by the classification error rate averaged over 5 trials with different random seeds for the initialization of NiN.

**Depth** (Table 1a). The CoOUs are embedded in the NiN by replacing ReLUs at the end of blocks of various depths (Fig. 2a) to exploit co-occurrence of diverse-leveled features. The CoOU favorably improves the performance of NiN compared to the baseline performance (i). And, the performance improvement is affected by the depth at which the CoOU is embedded. The CoOUs at the deeper blocks (iii,iv) exhibit better performance than that of the first block (ii). This is related to the abstract levels of the input features over which the co-occurrence characteristics are extracted by CoOU. Co-occurrence generally exhibits discriminative power on the features of the moderately higher abstract levels as shown in the hand-crafted features [15, 16]. While the first block in NiN renders lower-level features, the deeper blocks can extract the more semantic features of high abstraction which are favorably fed into the CoOU. The multiple CoOUs further improve performance (v$\sim$vii), and the best performance is achieved by embedding CoOUs at three blocks (vii).

**Comparison** (Table 1b). The *pseudo* co-occurrence formulation (5) is compared with the *original* formulation (2) directly using $\log$ and $\exp$ functions; practically, $\log(\max(0, x) +$

(a) Depth

| | CoOU is embedded at block 1 | block 2 | block 3 | Error rate (%) |
|---|---|---|---|---|
| i | | | | 35.42±0.31 |
| ii | ✓ | | | 35.14±0.24 |
| iii | | ✓ | | 34.69±0.27 |
| iv | | | ✓ | 34.63±0.27 |
| v | ✓ | ✓ | | 34.37±0.16 |
| vi | | ✓ | ✓ | 34.17±0.21 |
| vii | ✓ | ✓ | ✓ | **34.03**±0.31 |

(b) Comparison

| | Unit type | Error rate (%) |
|---|---|---|
| i | CoOU (pseudo) | **34.03**±0.31 |
| ii | CoOU (original) | 75.19±29.18 |
| iii | MEX [20] (pseudo) + ReLU | 35.28±0.35 |
| iv | MEX [20] (original) + ReLU | 99.00±0 |
| v | $1 \times 1$ Convolution + ReLU | 35.14±0.33 |

**Table 1**. Performance analysis using NiN [2] on Cifar100.

(a) Cifar100

| ConvNet | ResNet-38 | ResNet-74 | ResNet-110 |
|---|---|---|---|
| original | 28.76±0.13 | 27.51±0.22 | 27.31±0.54 |
| with CoOU | 28.24±0.11 | 26.75±0.14 | 26.40±0.27 |

(b) ImageNet

| ConvNet | AlexNet [1] | VGG-M [23] |
|---|---|---|
| original | 16.58 | 12.96 |
| with CoOU | 15.57 | 12.43 |

**Table 2**. Performance results on the other ConvNets.

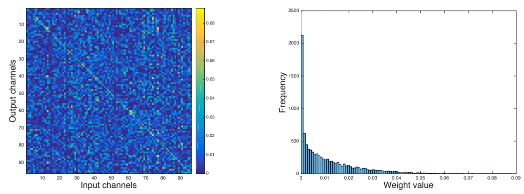
$1e^{-10}$) for log. These CoOUs are embedded in the three blocks as in Table 1a(vii). As expected, the ConvNet containing the CoOUs of the *original* formulation (ii) does not favorably converge in the training. On the other hand, the proposed CoOU using pseudo co-occurrence (5) works to facilitate the end-to-end learning while favorably extracting the co-occurrence characteristics. The proposed CoOU is also compared to the other types of units; MEX function [20] and $1 \times 1$ convolution followed by ReLU. For fair comparison, those units are implemented in the residual path with ReLU (Fig. 1b). As is the case with (i,ii), we apply the pseudo formulation (4) to log and exp functions of the MEX unit in addition to the original formulation, and similarly observe that the MEX of the original formulation hampers learning the ConvNet (iv), while the pseudo formulation remedies it (iii). The MEX, however, is inferior to the proposed CoOU since the MEX function is originally designed for the SimNet [20], not for the ConvNet. It is noteworthy that CoOU can boost the performance of ConvNets which are widely applied in an image analysis. As to the $1 \times 1$ convolution, the performance (v) is slightly improved from the baseline (35.42±0.31), but is still inferior to that of CoOU. This comparison shows that the non-linearity in CoOU is suitable for boosting the performance of ConvNet as an activation unit.

**Qualitative analysis.** Fig. 3a shows the learned filter weights $\{\boldsymbol{w}^k \in \mathbb{R}_+^{96}\}_{k=1}^{96}$ in the CoOU embedded at the first block of NiN, representing the co-occurrence patterns. The $k$-th co-occurrence pattern is learned so as to be compatible with the $k$-th feature component due to the residual model (Fig. 1b), and thus we can roughly observe the diagonal weight activation in Fig. 3a. Fig. 3b shows that the learned weights are sparse due to the non-negativity and unit-sum constraints in the co-occurrence activation function (5).

### 3.2. Application to other ConvNets

We apply the CoOU to the other ConvNets, ResNet [7] on Cifar100 dataset, and AlexNet [1] and VGG-M [23] on ImageNet dataset [24] for demonstrating its generality.

As in NiN, the ResNet that we use also contains three blocks and the CoOUs are embedded at those three blocks. The depth of the ResNet is simply controlled by changing the number of building components called *Bottleneck Unit* in Fig. 2c; for example, we show in Fig. 2b the ResNet-38 in which four units are stacked in each block. In this experiment, we employ ResNet-38/74/110 which are constructed by stacking 4/8/12 units, respectively. The classification performance on Cifar100 dataset is measured in the same manner as in Sec. 3.1. In the ConvNets of AlexNet [1] and VGG-M [23] which are composed of five convolution blocks, the CoOUs are embedded by replacing ReLU layers at those five blocks. Since the fully-connected layers are regarded as a classifier (multi-layered perceptron), we focus only on the feature extractors of convolution layers `conv1~5`. The performance on ImageNet dataset [24] is evaluated by top-5 error rate.




(a) $\{\boldsymbol{w}^k \in \mathbb{R}_+^{96}\}_{k=1}^{96}$ (b) Histogram of filter weight value

**Fig. 3**. Learned co-occurrence filter weights.

The performance results are shown in Table 2 demonstrating that the proposed CoOU favorably improves the performance of those ConvNets. These results show that CoOU works well in various types of ConvNets as an activation unit.

## 4. CONCLUSION

We have proposed a novel activation unit to extract co-occurrence characteristics among the feature channels. The original co-occurrence formulation employed in hand-crafted feature extraction has difficulty in the optimization especially by a gradient-based approach (back-propagation). We decompose the co-occurrence extraction process into three layers and approximate them to make the co-occurrence function trainable in the neural networks through the end-to-end learning. Then, we propose the co-occurrence activation unit (CoOU) by integrating the co-occurrence and occurrence (ReLU) activation functions in the residual model. In the experiments on Cifar100 and ImageNet datasets, it is shown that the proposed CoOU is favorably learned, improving the classification performance of various ConvNets. Our future work includes applying the CoOU to the other DNNs than ConvNets since the simple CoOU of no spatial extent accepts various types of multi-channel input.

## 5. REFERENCES


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[2] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR,* abs/1312.4400, 2013.

[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *CVPR Workshop*, 2015, pp. 36–45.

[4] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, Springer, 2015.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015.

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *ICLR*, 2016.

[11] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013.

[12] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolution network," *CoRR,* abs/1505.00853, 2015.

[13] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," in *ICML*, 2013.

[14] N. Otsu and T. Kurita, "A new scheme for practical flexible and intelligent vision systems," in *IAPR Workshop on Computer Vision*, 1988.

[15] T. Kobayashi and N. Otsu, "Image feature extraction using gradient local auto-correlations," in *ECCV*, 2008, pp. 346–358.

[16] T. Kobayashi and N. Otsu, "Color image feature extraction using color index local auto-correlations," in *ICASSP*, 2009, pp. 1057–1060.

[17] T. Mita, T. Kaneko, B. Stenger, and O. Hori, "Discriminative feature co-occurrence selection for object detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1257–1269, 2008.

[18] H. Ren and Z.-N. Li, "Object detection using generalization and efficiency balanced co-occurrence features," in *CVPR*, 2015, pp. 46–54.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.

[20] N. Cohen, O. Sharir, and A. Shashua, "Deep simnets," in *CVPR*, 2016, pp. 4782–4791.

[21] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *BMVC*, 2014.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.