Noise Boosted Neural Receptive Fields

Federico A. Galatolo
Dept. Information Engineering
University of Pisa
Pisa, Italy
federico.galatolo@ing.unipi.it

Mario G.C.A. Cimino
Dept. Information Engineering
University of Pisa
Pisa, Italy
mario.cimino@unipi.it

Alessandro Marincioni
Dept. Information Engineering
University of Pisa
Pisa, Italy
a.marincioni@studenti.unipi.it

Gigliola Vaglini
Dept. Information Engineering
University of Pisa
Pisa, Italy
gigliola.vaglini@unipi.it

***Abstract*—Conventional neural networks (NNs) for image classification make use of a convolutional layer and a feedforward (FF) classification layer. This paper presents a novel classification layer architecture and a training paradigm, in which the FF layer is split into small and specialized FF nets called Noise Boosted Receptive Fields (NBRFs), one per class. Each i-th NBRF provides three membership degrees: to the i-th class, to the super class made by its complementary classes, and to an extra class representing out-of-classes images. The training process artificially generates extra-class samples, via image transformation and noise addition. Experimental results, carried out on MNIST, KMNIST and FMNIST datasets show that, with respect to an FF layer, the NBRF layer improves robustness and accuracy of classification. The repository with the source code and experimental data has been publicly released to facilitate reproducibility and widespread adoption.**

Keywords—neural network, supervised learning, boosting, receptive field, image classification.

I. Introduction and Background

Convolutional Neural Networks (CNNs) are largely recognized as effective models for solving image classification tasks. CNNs employs convolutional hidden layers for feature extraction, i.e. for reducing data dimension and redundancy, generating feature maps. CNNs adopt feedforward (FF) neural networks to generate the output class from the feature space.

The explosion of connections needed by FF architectures for complex mappings leads to increasing difficulties in modeling and to inability to cope with highly nonlinear relationships [1]. To tackle this problem, in this paper a novel architecture is proposed, based on the concept of Receptive Field (RF) [1][2]. The concept of RF is related to local modeling, i.e., it relates to sub-models that focus predominantly on some selected regions of the entire modeling domain. In contrast to fully dense networks, appropriate RFs help the network to focus on local features of the input. Sequences of convolutional layers are an example of this method, which allows networks to extract complex, hierarchical features from increasingly large portions of the input [3]. This research work aims to adopt this design approach for the classification layer.

In the literature, an FF neural network architecture based on sub-models is known as modular neural network. It is made by a collection of neural networks moderated by a subsequent layer [4]. Each neural network serves as a module and operates on separate inputs to accomplish some subtask of the overall task. The moderator layer takes the output of each module and provides the output of the network as a whole. Recent works also focus on modular architectures to achieve model-intrinsic interpretability [4]. For example, different classes in a classification task may belong to a common superclass. This sort of category hierarchy can be exploited through specific network architectures as shown in [5]. In the literature, a type of interpretable neural architecture based on RF and computational stigmergy, called stigmergic RF, has been designed and successfully used to time series for behavioral analysis via wearable sensing [6][7][8]. Here, each RF is related to a different time series pattern. Another application field where an RF-based architecture has been successfully used to achieve interpretability is that of financial time series [9].

The novelty of the undertaken study relates to a new way in which RFs are being formed and optimized for image classification. Specifically, this work introduces the Noise Boosted Receptive Field (NBRF), a classification architecture and a training paradigm based on modular FF nets. With respect to a conventional FF classification layer with the same number of parameters, a layer of NBRFs is more accurate and robust, because it allows to recognize noise (extra-class) samples. Noise samples are artificially generated at training time via image transformation and noise addition. Experimental results, carried out on MNIST, KMNIST and FMNIST datasets, compare the FF and the NBRF layers, with different extra-class generation techniques. As a result, the NBRF layer improves robustness and accuracy of classification. The repository with the source code and experimental data has been publicly released to facilitate reproducibility and widespread adoption [11].

The paper is organized as follows. In Section II, the design of the NBRF is formally discussed. Experimental studies related to MNIST, KMNIST and FMNIST benchmarks are documented in Section III. Finally, Section IV draws some conclusions and future work.

II. Design of Noise Boosted receptive Fields

In this section, the NBRF classification model is formalized and discussed. Fig. 1 shows the reference architecture. Given an input image x, to determine its class $c(x) \in \{C_1, \ldots, C_n\}$, a CNN made up of convolutional and pooling layers is first used to extract the related feature vector y, as it is commonplace for image classification [10]. The feature vector is then processed by n small NBRFs, each made by an FF neural net specialized

on recognizing a domain class. Finally, a moderator component (MOD) takes the output of each NBRF to determine the overall output c.

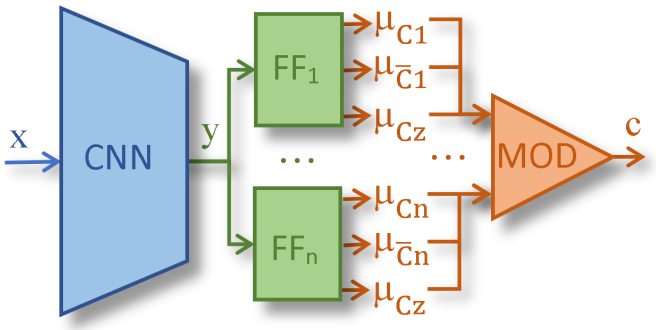

Fig. 1. NBRF architectural model. The CNN subnetwork is made up of convolutional and pooling layers, while each subnetwork FF_i is made up of feedforward and softmax layers, specialized to recognize whether the input image belongs to its class or not, or if it is a noise (extra-class) input.

A. *Receptive fields and moderator logic*

An *i*-th NBRF provides three real output values, i.e., μ_{C_i}, $\mu_{\bar{C}_i}$ and μ_{C_z}, representing the *membership degree* of x to C_i, $\bar{C}_i$, and C_z, respectively. Specifically, C_i is the i-th class, whereas $\bar{C}_i$ is the superclass made by the union of the classes complementary to C_i:

$$\bar{C}_i = \bigcup_{j \neq i} C_j \tag{1}$$

C_z is an extra class representing out-of-classes images. C_z provides robustness, i.e., the capability to cope with noise and undetermined input. More precisely, the outputs of an NBRF are provided via the softmax function. As a consequence, μ_{C_i}, $\mu_{\bar{C}_i}$, and μ_{C_z} are normalized, and their sum is 1.

The moderator employs the following inference formulas to compute the *strengthened membership degree* of the input image *x* to the *i*-th class:

$$M_{C_i}(x) = \mu_{C_i}(x) \cdot [1 - \mu_{\bar{C}_i}(x)] \cdot [1 - \mu_{C_z}(x)] \tag{2}$$

which combines three conditions beneficial to the membership of x to the *i*-th class: (a) the membership to the *i*-th class, (b) the non-membership to the complementary superclass, and (c) the non-membership to the extra class.

Given the strengthened membership degrees $M_{C_j}(x)$ of each *j*-th RF, the class of the input image x is assigned as follows:

$$c(x) = \begin{cases} C_i & if \ \max_j M_{C_j}(x) = M_{C_i}(x) > \mathcal{M} \\ C_z & otherwise \end{cases} \tag{3}$$

where $\mathcal{M}$ represents the limit membership degree for assigning a sample to a class. It is computed after the training process, as the optimum value minimizing the classification error, by applying Formula (3) to all the input images x of the training set and of the extra-class set. A good value of $\mathcal{M}$ can be easily found via a grid search on the interval [.5, .9] with step 0.1.

B. *Noise boosting approaches*

According to the supervised learning paradigm, the NBRF layer is trained via labelled images. The three classes encoded by an NBRF are labelled with binary values, i.e., $(c_i, \bar{c}_i, c_z) \in \{0,1\}^3$. Two simpler variants can be also considered. The first variant, hereafter called RF, does not consider C_z, i.e., it is without noise boosting. In this case, $\mu_{C_z}(x) = 0$ in (2), and then $(c_i, \bar{c}_i) \in \{0,1\}^2$. Another variant, hereafter called $NBRF_2$, does not consider the complementary superclass $\bar{C}_i$. In this case, $\mu_{\bar{C}_i}(x) = 0$ in (2), and then $(c_i, c_z) \in \{0,1\}^2$.

For a given training set, for a better accuracy and robustness, a set of noisy inputs is artificially generated via data augmentation techniques. This *noise boosting* allows the weak learners represented by the NBRFs to generate a global strong learner [12].

Specifically, the following three techniques are considered effective:

1) *Statistical surrogate of the training set* (**S** for short): samples generated from a normal distribution with the mean and variance of the training set;
2) *Averaging of training subsets* (**T** for short): samples generated as the mean of randomly extracted samples of the training set;
3) *Averaging of the training batch* (**B** for short): samples generated as the mean of the samples of the current training batch.

The source of generation of the artificial samples can be of two different types:

1) *Input images x* (**I** for short), i.e., the source samples are sets of training images;
2) *Image feature vectors y* (**F** for short), i.e., the source samples are the feature vectors extracted from sets of training images.

By combining the different approaches, and considering $NBRF_2$, RF, and NBRF, the variants listed in Table I are considered effective.

C. *Loss function*

The overall architecture, i.e., feature extraction and classification layers, is trained using the cross-entropy as a loss function. More specifically, the overall objective function is the sum of the cross-entropy functions of all NBRFs.

For a given NBRF, the loss function is defined as the weighted average of the three cross-entropy functions related to the three outputs. The cross-entropy function of the *c*-th membership value, $c = c_i, \bar{c}_i, c_z$, provided by an NBRF is the following:

$$h_c(x) = -\sum_x \mu_c^t(x) \cdot log(\mu_c(x)) \tag{4}$$

where $\mu_c(x)$ and $\mu_c^t(x)$ are the membership value provided by the NBRF and the target value, respectively.

TABLE I. TYPES OF NEURAL NETS ARCHITECTURES

Acronym	Description
$NBRF_2$ – SF	Output: without complementary super-class; Noise sample generation: statistical surrogate; Noise sample source: feature vectors.
$NBRF_2$ – SI	Output: without complementary super-class; Noise sample generation: statistical surrogate; Noise sample source: input images.
$NBRF_2$ – TF	Output: without complementary super-class; Noise sample generation: averaging training set; Noise sample source: feature vectors.
$NBRF_2$ – TI	Output: without complementary super-class; Noise sample generation: averaging training set; Noise sample source: input images.
$NBRF_2$ – BF	Output: without complementary super-class; Noise sample generation: averaging training batch; Noise sample source: feature vectors.
$NBRF_2$ – BI	Output: without complementary super-class; Noise sample generation: averaging training batch; Noise sample source: input images.
RF	Output: without complementary super-class, and without noise extra-class.
NBRF – SF	Full output; Noise sample generation: statistical surrogate; Noise sample source: feature vectors.
NBRF – SI	Full output; Noise sample generation: statistical surrogate; Noise sample source: input images.
NBRF – TF	Full output; Noise sample generation: averaging training set; Noise sample source: feature vectors.
NBRF – TI	Full output; Noise sample generation: averaging training set; Noise sample source: input images.
NBRF – BF	Full output; Noise sample generation: averaging training batch; Noise sample source: feature vectors.
NBRF – BI	Full output; Noise sample generation: averaging training batch; Noise sample source: input images.
CNN – FF	Conventional convolutional and feed-forward layers.

As discussed on the labeling process, target values are binary, i.e., $\mu_c^t(x) \in \{0,1\}$. The cross-entropy function of the NBRF is the following:

$$H_{NBRF}(x) = \sum_{c=c_i,\bar{c}_i,c_z} \frac{w_c}{3} \cdot h_c(x) \tag{5}$$

where w_c is the weight used for class balancing. The weight w_c is calculated according to the cardinality ratio between each i-th class and the related i-th superclass or the noise extra-class, in order to tackle class unbalancing, as follows:

$$w_c = \begin{cases} |\mathrm{C_i}|/|\mathrm{C_i}| & if\ x \in \mathrm{C_i} \\ |\mathrm{C_i}|/|\bar{\mathrm{C}}_\mathrm{i}| & if\ x \in \bar{\mathrm{C}}_\mathrm{i} \\ |\mathrm{C_i}|/|\mathrm{C_z}| & if\ x \in \mathrm{C_z} \end{cases} \tag{6}$$

Indeed, batch size and number of extra-class samples per batch are two independent hyperparameters. As a consequence, the number of generated samples can lead to unbalanced training. In order to balance the impact of labelled and classless images, the losses corresponding to each type of image are weighted in Formula (5). In the case of RF and $NBRF_2$, Formula (5) and Formula (6) must be simplified, by removing the noise extra-class and the complementary super class, respectively. In particular, the denominator value of Formula (5) becomes 2, and the number of cases in Formula (6) become two.

III. EXPERIMENTAL STUDIES

To investigate the effectiveness of the NBRF architecture, some experiments have been carried out on three real-world problems used for benchmarking machine learning algorithms: MNIST [12], Fashion-MNIST [14], and K-MNIST [15]. MNIST is made of handwritten digits images, Fashion-MNIST is a dataset of fashion article images, whereas K-MNIST is made of handwritten Japanese characters. All the datasets are made by a training set of 60,000 examples, and a test set of 10,000 examples. Each example is a 28×28 image, with pixels in 0–255 grayscale values, associated with a class label of 10 possible classes. The task is to classify a given image into one of such 10 classes.

A. Architectural settings

All the types of architecture listed in Table I have been developed and compared. A repository with the source code and experimental data has been publicly released to facilitate reproducibility and widespread adoption [11].

To guarantee a fair comparison, the convolutional layers are identical for all the networks. The classification layers have been structured so as to have a similar number of weights. Table II shows the number of weights in the FF layers for each type of architecture. The first FF layer is equipped with the same number of weights, whereas the second FF layer is slightly different because of the structural differences in the number of outputs.

TABLE II. NUMBER OF WEIGHTS IN THE FF LAYERS

Architecture	1st FF Layer	2nd FF layer	Total
$NBRF_2$ or RF	4.000 M	10 K	4.010 M
NBRF	4.000 M	15 K	4.015 M
CNN – FF	4.000 M	50 K	4.050 M

More specifically, the convolutional subnetwork scales the input through a batch normalization layer, and then it applies two iterations of convolution, nonlinearity, and pooling layers. Both convolution layers use 5×5 kernels. The first convolution produces an output of 20 channels, while the second one produces one of 50. The nonlinearity function used is a Leaky ReLU. The pooling operation applies max pooling over 2×2 subregions. The resulting output is flattened, and corresponds to 800 values per image.

The ten NBRFs are made up of two FF layers. The first layer is equipped with 500 units. According to Table II, the number of connections of the first layer is then 800×500=400K per NBRF. The second layer of an NBRF is made by 3 nodes, which become 2 nodes for $NBRF_2$ and RF. Overall, 500×3 = 15K for NBRF, and 500×2 = 10K for $NBRF_2$ and RF, according to Table II. The first layer is followed by a nonlinearity layer

that applies the Leaky ReLU, while the second layer is followed by a softmax layer to normalize the outputs.

The CNN – FF network has two FF layers, equipped with 5000 and 10 nodes, respectively. According to Table II, the total number of weights is 800×5000 + 5000×10 = 40M + 50K.

The Adaptive Moment Estimation (Adam) method [16] has been used to compute adaptive learning rates for each parameter of the gradient descent optimization algorithms, carried out with batch method [17]. Early stopping is used as stopping criterion for the training loop. The validation loss is monitored using a patience value of 3. The optimum value found for $\mathcal{M}$ is 0.9. Each architecture has been trained 10 times to get average performance measurements and confidence intervals.

B. *Analysis of Results*

Fig. 2 shows the classification capabilities of the different network architectures. Let us assume that the training set and the corresponding test set belong to the same dataset. Although CNN-FF has been equipped with 35-40K additional weights, it is apparent that there are NBRF models, such as NBRF-TI, NBRF-SI, and NBRF$_2$-SI, performing better than the conventional CNN-FF on the three datasets.

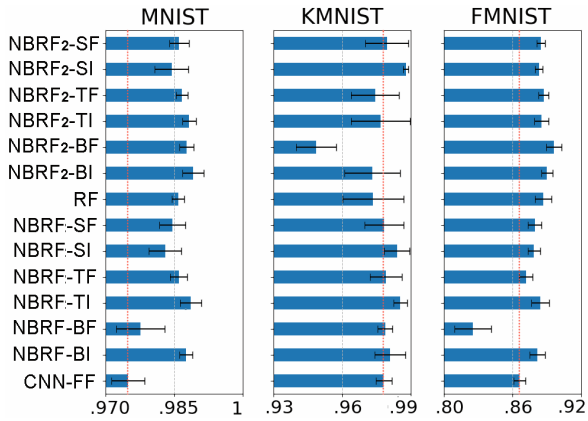

Fig. 2. Average and confidence intervals of the classification accuracy, for different architecture variants over different datasets.

The impressive advantage of the proposed architecture is clear when considering robustness. Let us consider the CNN – FF. Fig. 3 shows the percentage of samples classified as noise extra-class, i.e., samples whose membership to all classes is lower than $\mathcal{M}$ = 0.9. For each cell, a related CNN – FF architecture has been trained with the dataset in the ordinates and tested with the dataset in the abscissae. As expected, diagonal cells achieve a classification close to zero, because the training and testing datasets are the same. Thus, no extra-class samples are available for that cells. However, the Figure clearly shows that the architecture is unable to adequately recognize extra-class samples in the other cells. Indeed, a very high noise percentage is expected to be found in non-diagonal cells, in which the architecture has been trained with a dataset and tested with a completely different dataset. However, the non-diagonal cells show a very low noise percentage, of about 8-16%. Fig. 4 and Fig. 5 show the same type of matrix for NBRF$_2$–TI and NBRF$_2$–TF. Here, it can be easily noted that the NBRF-based architecture sensibly outperforms the conventional CNN-FF in terms of robustness. Indeed, both NBRF$_2$–TI and NBRF$_2$–TF are able to recognize a considerable fraction of another dataset as a set of noise extra-class samples.

Finally, Fig. 6 shows the performance of an RF net. It is worth noting that without noise boosting there is a noticeable decrease of performance. This proves the effectiveness of the proposed approach.

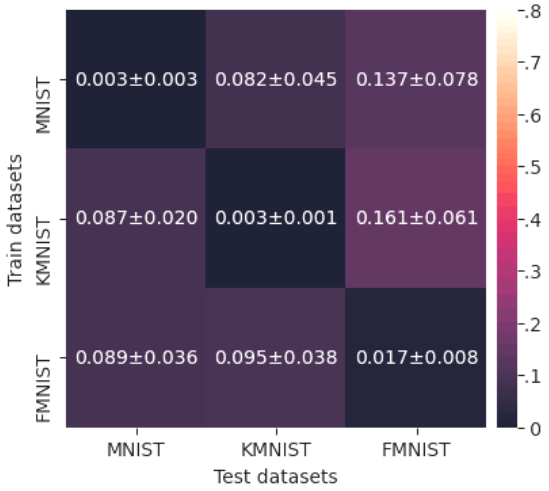

Fig. 3. CNN-FF net: percentage of samples classified as noise extra-class, with training and testing sets on the ordinates and abscissae, respectively.

IV. Conclusion and Future Work

In this paper, the concept of modular NBRFs has been discussed and developed, as an alternative to a monolithic FF classification layer. The proposed architecture is characterized by the capability of detecting extra-class samples, thanks to noise boosting.

With respect to an FF classification layer having the same number of parameters, a classification layer of NBRFs is more accurate and robust. It allows by design to recognize noise extra-class samples. For this purpose, noise samples are artificially generated at training time via image transformation and noise addition.

Experimental results have been carried out on MNIST, KMNIST and FMNIST datasets, to compare the FF and the NBRF layers, with different extra-class generation techniques.

Results show the high potential of the proposed approach, encouraging further comparative research. The source code has been publicly released to facilitate reproducibility and widespread adoption.

Acknowledgment

Work partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

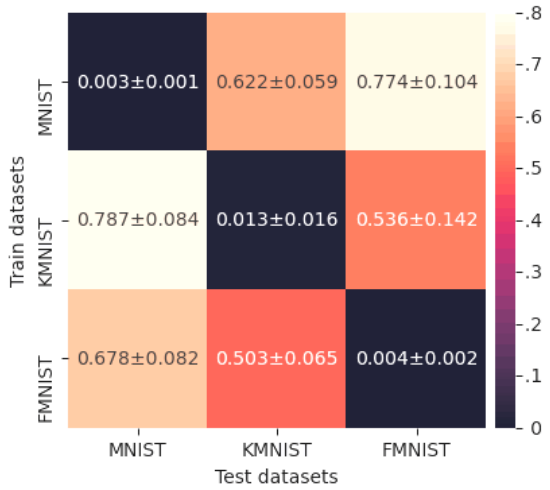

Fig. 4. $NBRF_2$ – TI net: percentage of samples classified as noise extra-class, with training and testing sets on the ordinates and abscissae, respectively.

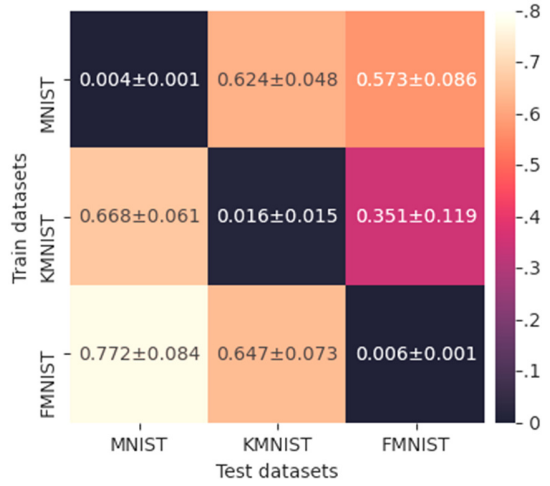

Fig. 5. $NBRF_2$ – TF net: percentage of samples classified as noise extra-class, with training and testing sets on the ordinates and abscissae, respectively.

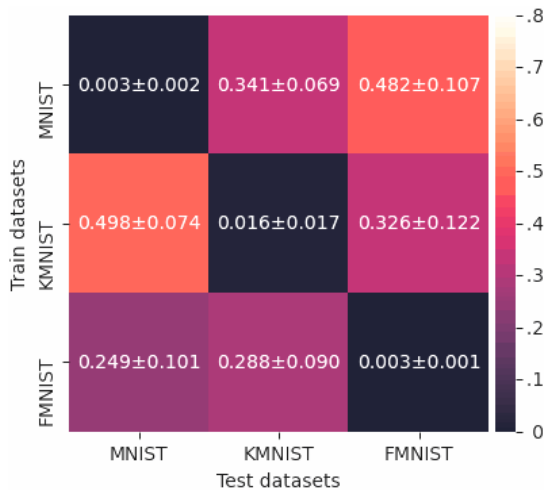

Fig. 6. RF net: percentage of samples classified as noise extra-class, with training and testing sets on the ordinates and abscissae, respectively.

REFERENCES

[1] M.G.C.A. Cimino, W. Pedrycz, B. Lazzerini, and F. Marcelloni, "Using multilayer perceptrons as receptive fields in the design of neural networks", Neurocomputing, vol. 72, n. 10–12, pp. 2536-2548, 2009.

[2] W. Pedrycz, M.G. Chun, and G. Succi, "N4: computing with neural receptive fields", Neurocomputing, vol. 55, I. 1–2, pp. 383-401, 2003.

[3] Y. Bengio, and Y. LeCun, "Convolutional Networks for Images, Speech, and Time-Series", The handbook of brain theory and neural networks, pp. 255–258, MIT Press, MA, USA, 1998.

[4] W. Li, M. Li, J. Qiao, X. Guo, "A feature clustering-based adaptive modular neural network for nonlinear system modeling", ISA Transactions, vol. 100, pp. 185-197, 2020.

[5] W. Han, C. Zheng, R. Zhang, J. Guo, Q. Yang, J. Shao, "Modular neural network via exploring category hierarchy", Information Sciences, vol. 569, pp 496-507, 2021.

[6] M Avvenuti, C Bernardeschi, M G C A Cimino, G Cola, A Domenici, and G Vaglini, "Detecting elderly behavior shift via smart devices and stigmergic receptive fields", Proc. of The 6th EAI International Conference on Wireless Mobile Communication and Healthcare (MOBIHEALTH 2016), vol. 192, pp. 398-405, 2017.

[7] A.L. Alfeo, M.G.C.A. Cimino, and G. Vaglini, "Measuring Physical Activity of Older Adults via Smartwatch and Stigmergic Receptive Fields", Proc. of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017), pp. 724-730, 2017.

[8] A.L. Alfeo, P. Barsocchi, M.G.C.A. Cimino, D. La Rosa, F. Palumbo, G. Vaglini, "Sleep behavior assessment via smartwatch and stigmergic receptive fields", Personal and Ubiquitous Computing, vol. 22, pp.227-243, 2018.

[9] M.G.C.A Cimino, F. Dalla Bona, P. Foglia, M. Monaco, C.A. Prete, G. Vaglini, "Stock price forecasting over adaptive timescale using supervised learning and receptive fields", Mining Intelligence and Knowledge Exploration, vol. 11308, pp. 279-288, 2018.

[10] F.A. Galatolo, M.G.C.A. Cimino, G. Vaglini, "Generating Images from Caption and Vice Versa via CLIP-Guided Generative Latent Space Search", In Proc.of the International Conference on Image Processing and Vision Engineering (IMPROVE 2021), pp.166-174, 2021.

[11] F.A. Galatolo, Github repository on Noise Boosted Receptive Fields, https://github.com/galatolofederico/noise-boosted-receptive-fields.

[12] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, J. Song , "RBoost: Label Noise-Robust Boosting Algorithm Based on a Nonconvex Loss Function and the Numerically Stable Base Learners", IEEE Trans Neural Netw Learn Syst. 2016, vol. 27, n. 11, pp. 2216-2228.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", In Proc. IEEE, vol. 86, n. 11, pp. 2278–2324, 1998.

[14] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms", CoRR, vol. abs/1708.07747, 2017.

[15] T. Clanuwat, M. Bober-Irizar , A. Kitamoto, A. Lamb, K. Yamamoto and D. Ha, "Deep learning for classical japanese literature", CoRR, vol. abs/1812.01718, 2018.

[16] D.P. Kingma and J.L. Ba, "Adam: a method for stochastic Optimization.", Proc. of International Conference on Learning Representations, pp. 1–13, 2015.

[17] S. Ioffe and C.Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," CoRR, vol. abs/1502.03167, 2015.