2023 Seventh IEEE International Conference on Robotic Computing (IRC) | 979-8-3503-9574-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IRC59093.2023.00011

Tactile Sensing with Contextually Guided CNNs: A Semisupervised Approach for Texture Classification

Olcay Kursun
Dept. of Computer Science
Auburn University at Montgomery
Montgomery, Alabama
okursun@aum.edu

Beiimbet Sarsekeyev
Dept. of Computer Science
University of Central Arkansas
Conway, Arkansas
bsarsekeyev@gmail.com

Mahdi Hasanzadeh
Dept. of Computer Systems Technology
North Carolina A&T State University
Greensboro, North Carolina
mhasanzadehhesar@aggies.ncat.edu

Ahmad Patooghy
Dept. of Computer Systems Technology
North Carolina A&T State University
Greensboro, North Carolina
apatooghy@ncat.edu

Oleg V. Favorov
Dept. of Biomedical Engineering
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina
favorov@email.unc.edu

***Abstract*—Texture classification plays a crucial role in applications ranging from object recognition and product design to surface exploration. Utilizing deep learning methods with sensors, such as accelerometers, offers a way to identify key surface features without the need to precisely replicate human touch. A Contextually Guided Convolutional Neural Network (CG-CNN) employs contextual guidance by developing auxiliary tasks during its training. These tasks offer implicit, yet rigorous, internal supervision signals. When trained with these subtasks, CG-CNN learns to represent the innate structure and patterns within the data, resulting in robust, transferrable, and local/contextual-neighborhood-preserving domain representations. This paper extends the CG-CNN framework for texture classification by integrating semisupervised learning. Empirical evaluations on the VibTac-12 texture dataset reveal that CG-CNN effectively generalizes to novel and unfamiliar textures, even when trained with scarce labeled examples. By harnessing vast amounts of unlabeled, contextually relevant data alongside the labeled samples, CG-CNN ensures robust and precise texture classification. Such advancements hold promise for applications in robotics, prosthetics, and haptic interfaces.**

***Index Terms*—Deep Learning for Texture Classification, Contextual Guidance, Semisupervised Auxiliary Tasks.**

I. Introduction

Semi-supervised learning bridges the gap between supervised and unsupervised methods, harnessing both labeled and unlabeled data during training [1], [2]. This approach becomes essential in areas like texture classification and tactile perception. Capturing the essence of a touch demands controlled environments to obtain clear labels. In our daily interactions, while humans touch myriad surfaces, rarely do we consciously recognize or categorize them in terms of supervised labels. This makes gathering accurate labeled tactile data challenging, often facing issues of noise, time constraints, or simple feasibility. Leveraging semi-supervised methodologies, we can better harness available data and also offer a more resource-efficient approach to learning.

While traditional deep Convolutional Neural Networks (CNNs) learn features through error backpropagation, which involves passing the error (e.g., classification or reconstruction) from the higher layers down to lower layers, the emergence of features in the visual cortex is local and self-organized. This emergence can be attributed to their local high transfer utility [3]. The Contextually Guided Convolutional Neural Network (CG-CNN) introduced by [3]–[5] offers a bottom-up approach distinct from the traditional top-down deep CNNs. While both employ error backpropagation, CG-CNN prioritizes preserving important data, ensuring the retention of neighborhood or contextual information throughout the learning process. CG-CNNs use local contextual information instead of solely relying on supervised backpropagation, offering an autoencoder-like approach to learning descriptive (pluripotent) features. This approach, which we derived from our computational neuroscientific studies of cerebral cortical networks [4]–[6], attributes an objective function to maximize for each convolutional area and, thus, eliminates the need for backpropagating the error from the top layers [4], [7], [8]. CG-CNN approach reduces the risk of mode collapse and vanishing gradients [9]. For each convolutional area, it uses a self-supervised shallow CNN, and thus, the first area learns its features from small, unlabeled datasets, instead of relying on large, manually labeled datasets. Higher areas can be built using the outputs of the previous CG-CNN areas. CG-CNN offers a natural

979-8-3503-9574-7/23/$31.00 ©2023 IEEE
DOI 10.1109/IRC59093.2023.00011

extension for semisupervised learning, which involves using both labeled and unlabeled data [1], [10]. In a semisupervised setting CG-CNN self-supervises when provided with unlabelled examples, and applies backpropagation to take advantage of supervised training when provided with labelled training examples.

In this study, we conducted a comparative analysis of CG-CNN alongside other AI/ML methods, such as CNN and Autoencoders, to demonstrate its efficacy in texture classification using vibrotactile signals. The application of AI in tactile and vibration sensing systems holds great promise across diverse fields [11], [12]. Our study presents a novel approach by utilizing CNNs and Contextually Guided CNN (CG-CNN) for texture classification on the VibTac-12 dataset. This research is the first of its kind and demonstrates the feasibility of extending the CG-CNN algorithm through a semisupervised framework.

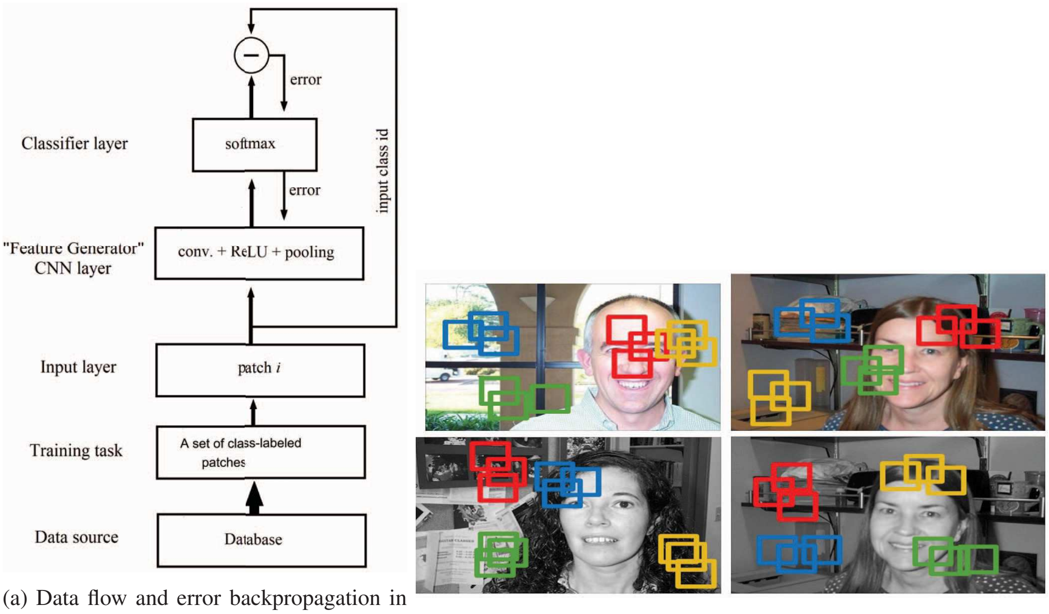
This paper is structured as follows: Section II provides a review of relevant literature, with a focus on CG-CNN. Section III introduces the proposed semisupervised extension of CG-CNN, called BeiimNet. In Section IV, we present the experimental results conducted on the VibTac-12 dataset for texture classification using vibrotactile signals. The experiments focus on evaluating the transferability of the extracted features to new texture classification tasks and assessing their effectiveness in accurately classifying previously unseen textures. Finally, Section V concludes the paper.

II. Background

Local invariance, the ability of CNNs to recognize objects regardless of their position within an image, and compositionality, their capability to assemble more complex features from simpler ones, are two significant advantages of Convolutional Neural Networks (CNNs) [13]. According to [13], in the initial layers, CNNs identify basic elements like edges by applying specialized filters, laying the groundwork for detecting more complex shapes in the subsequent layers. With every successive convolutional layer, CNNs learn to distinguish more specific features, which are then used to make predictions. CNNs use pooling layers to achieve representation of the image patches, enabling classification of objects in the image without considering their exact location. [14] presents a method for discriminative unsupervised feature learning using exemplar CNNs that only uses unlabelled data. The network discriminates between surrogate classes, which are generated from randomly generated image patches, called seeds, that are transformed using a family of transformations. The features generated from this method exhibit robustness to transformations that is not present in classic supervised approaches. The study is based on unsupervised learning of invariant features, and several instances of invariant feature generation/utilization are present in both unsupervised and supervised learning. However, while this method outperforms traditional unsupervised feature generation methods, it cannot achieve the same performance as classic supervised learning methods. The method starts by selecting a random sample of image patches from unlabelled images and applying various transformations to create surrogate training data. The CNN network is then trained to discriminate between these surrogate classes.

In [15], the focus is on unsupervised learning and maximizing the mutual information between the input and output of deep encoders. However, computing mutual information is a complex and challenging task, and therefore, the proposed Deep Infomax method incorporates the input's locality into the objective to highlight the significance of the structure. The primary concept is to increase the mutual information between input and output, which is achieved by using an adversarial learning model consisting of an encoder and a decoder. While the Deep Infomax method outperforms many unsupervised learning tasks, it falls short of the performance levels expected from supervised learning.

Contextually-Guided CNN (CG-CNN) is an unsupervised approach that enables the extraction of highly discriminative and transferable features [3], [4]. In its application to natural images, the features developed in its first layer (the first convolutional area) resembled those of other state-of-the-art deep learning architectures [17], [18]. In its simplest form, the complete system consists of a single convolutional layer known as the Feature Generator connected to a linear classifier. Once a convolutional area is developed, then it can be used as input to other CG-CNN layers to form deep networks in a bottom-up manner. CG-CNN training utilizes transfer learning by creating different classification problems for self-supervision, allowing it to learn what to transfer. In other words, the Feature Generator gradually learns more discriminative features that the discriminator can adapt to its ever-changing classification problems, providing feedback to the Feature Generator (see Algorithm 1). While the modular nature of CG-CNN shares a resemblance with deep autoencoders, where networks can be stacked to form multiple convolutional areas, the current exploration of CG-CNN is purposefully confined to a single convolutional area (consisting of one convolutional layer followed by a ReLU activation). Although this initial setup with single-area has its limitations in capturing complex representations, it offers essential foundational insights, particularly in its self-supervised operation. CG-CNN self-supervision operates by presenting a selected input pattern — typically a small image window extracted from one of the internally generated contextual groups — to the network. This process allows the computation of feature values based on the current model architecture, subsequently feeding these extracted features into the classifier. The classifier is trained to differentiate all contextual groups of the current task from one another. The prediction error of the contextual group's input patterns is backpropagated to the classifier and the convolutional layer. The backpropagation algorithm alternately adjusts the connection weights of the softmax-classifier and the convolutional-layer. By minimizing the prediction error of



(a) Data flow and error backpropagation in the CG-CNN architecture.

(b) Contextual group demonstration.

Fig. 1: In its application to natural images, CG-CNN used small, e.g., 19×19, image patches for training. Each task contains one image only, and within that image, there are four contextual groups created (each group is shown with a different color and with three snipped image patches). In its application to tactile signals, these patches are snipped from sensor recordings that describe small regions of textures shown in Figure 2.

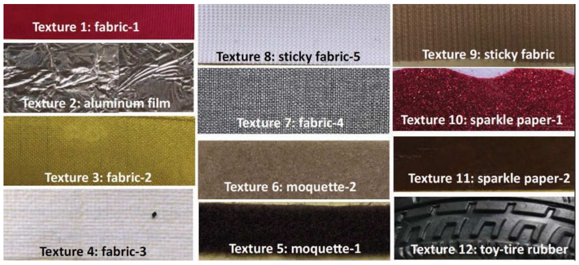


Fig. 2: Images of the 12 texture classes recorded in the VibTac-12 dataset [16].

these internally generated and ever-changing classification tasks, the features of CG-CNN gradually become more inferential than its inputs [4], [19]. The classes (contextual groups) of the tasks are internally generated (i.e., self-supervised and not related to any external supervision/class-labels) [14] (Figure 1). The training of the system involves multiple iterations, with each iteration using a unique set of contextual groups as training classes [19]. In each iteration, a small unsupervised set of training examples is drawn from the database, which includes many nearby image windows organized into (e.g., 50) contextual groups, and the system is trained to discriminate against them. After the completion of training, another small set of classes is selected, and training continues on this new set without resetting the already developed CNN connection weights. By the contextual guidance [4] principle, all the transformations of a given seed image patch are contextually related and are considered examples of a single class.

III. The Proposed Semi-Supervised CG-CNN

The original CG-CNN method [3], [4] trains a convolutional layer so that the feature set gradually becomes more pluripotent for discriminating any set of contextually related input patterns. In the initial CG-CNN model, only unlabeled examples were used. This is particularly relevant at the first layer, where neurons' small receptive fields render supervised class labels less effective. The network snipped the image patches from the large unlabelled images, and all examples within close proximity in that image (i.e., contextually-related image patches) were given a unique group label that was to be discriminated maximally from other such groups (hence the name "pluripotent"). However, we believe that in a stack of CG-CNN layers, higher-order features should utilize more global (more expansive) receptive fields at the higher CG-layers. As these features get more sophisticated, the feature tuning in higher convolutional layers should gradually benefit from supervised examples to exhibit more utility for supervised classification tasks. This step will help develop better features instead of preserving/transforming the data while maintaining all the contextual regularities.

Our method, selects a new small task (supervised or unsupervised) from a large pool of data containing both labeled

ALGORITHM 1
CG-CNN for semi-supervised texture feature extraction

```
CG-CNN=[
   //First, the Input layer with 2 sec window (400 samples) and 3 sensors (X, Y, Z)
     Layer-1: InputLayer(input_size = (400 \times 3)
   //Then, the CNN layers of Feature Generator with 3 areas (Conv+ReLU+MaxPool):
     Layer-2: ConvLayer (kernel_size = (10 \times 3), out_channels = 25)
     Layer-3: ReLULayer
     Layer-4: MaxPool(kernel_size = $3 \times 3$)
     Layer-5: ConvLayer (kernel_size = (5 \times 1), out_channels = 50)
     Layer-6: ReLULayer
     Layer-7: MaxPool(kernel_size = $3 \times 3$)
     Layer-8: ConvLayer (kernel_size = (3 \times 1), out_channels = 100)
     Layer-9: ReLULayer
     Layer-10: MaxPool(kernel_size = $3 \times 3$)
     Layer-11: $Features$ = GlobalAveragePooling()
   //Last, the Classifier layer (Discriminator):
     if supervised: //Use $D = 12$ texture classes
         Layer-12: $Posterior$ = SoftmaxClassificationLayer(output_size = 12)
      elseif unsupervised: //Use $C$ auxilary classes
         Layer-12: $Posterior$ = Softmax(output_size = $C$) ]
Randomly initialize weights $W$ of Layers 2, 5, and 8
Repeat E-M iterations
   //New task:
     Create a new task (data subset) while alternating between supervised&unsupervised
     Split the task dataset into the E-dataset and M-dataset
   //E-step:
     Set learning rate to 0 for the convolutional weights $W$
     Randomly initialize the new Classifier weights $V$ of Layer-12
     Train $V$ on the E-dataset
   //Compute the class&group accuracy (Figure 5) on the test (M-dataset)
     Use the new $V$ and the existing $W$ on the M-dataset to check convergence
   //M-step:
     Set learning rate to 0 for the weights $V$ of Layer-12
     Restore learning rate for $W$ of Layers-2, 5, and 8.
     Continue updating (retrain) the existing weights $W$ by using the M-dataset
End E-M
```

and unlabeled examples, as shown in Figures 1 and 2. The training task (dataset) for each Expectation-Maximization (EM) iteration is either supervised or unsupervised (see Algorithm 1 and Figures 3 and 4). In the unsupervised case, C classes are used and a seed image patch is chosen for each class, generating input patches with additional data augmentations. In the supervised case, D classes are used. Like the original CG-CNN, the classifier layer in the contextually guided network is trained to differentiate the C contextual groups using existing features, referred to as the E-step of the EM algorithm. Conversely, for a supervised task, the method selects labeled training examples from D classes. Typically, we can use all classes, say $D = 12$ for the VibTac-12 dataset, but D can be a subset of classes as well. This way, new classes can be continuously added for continuous/online learning [20]. Data augmentation methods can be utilized to enhance convergence in both unsupervised and supervised cases [21], [22]. Next, the task resets the discriminator (now with D output units for the D classes) and initiates the E-step of the EM optimization. In the E-step, the method trains the discriminator (SoftMax) using the class-labeled examples in the task, while keeping the feature generator/convolutional layers fixed. The M-step then begins at which the discriminator is frozen and the feature generator is allowed to learn from the weights backpropagated from the previous E-step. Additionally, the M-step minimizes the feature generator's error through backpropagation. This is where the learned features develop to better support supervised classification. The end of the M-step marks the start of another EM iteration in which the proposed method creates a new task and alternates between supervised and unsupervised learning. The number of iterations can be set or continue until convergence is reached. The proposed method monitors the transfer utility, which is group accuracy for unsupervised tasks and class accuracy for supervised tasks and tracks their fluctuation task by task and over time, these accuracies are expected to converge as the iterations progress (as shown in Figure 5).

IV. Experimental Evaluations

Although CG-CNN was originally designed to work with image data, it can also be applied to any types of data that have contextual regularities, such as data collected from sensors in IoT devices. The evaluation of the proposed method is done using a dataset of vibrotactile signals collected by authors in [12]. This dataset consists of 12 classes of textures and comprises 20 seconds of recordings from a 3D accelerometer sensor attached to a probe rubbing against

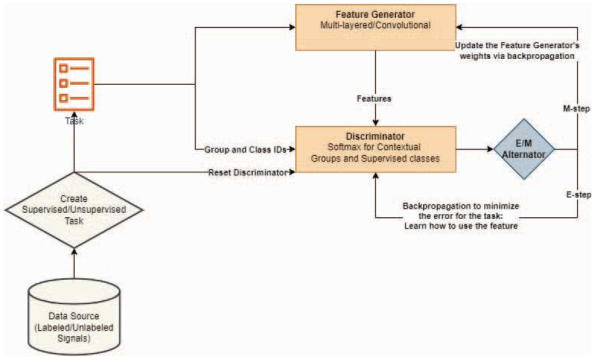


Fig. 3: Proposed method diagram. Feature Generator gradually learns more discriminative features that Discriminator can adapt to its ever-changing classification problems, providing feedback to Feature Generator.

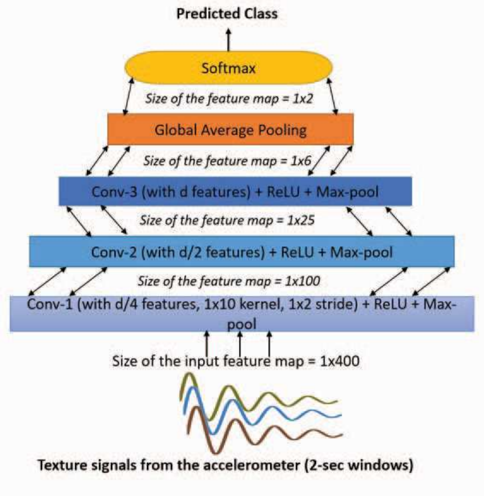

Predicted Class
Softmax
Size of the feature map = 1x2
Global Average Pooling
Size of the feature map = 1x6
Conv-3 (with d features) + ReLU + Max-pool
Size of the feature map = 1x25
Conv-2 (with d/2 features) + ReLU + Max-pool
Size of the feature map = 1x100
Conv-1 (with d/4 features, 1x10 kernel, 1x2 stride) + ReLU + Max-pool
Size of the input feature map = 1x400
Texture signals from the accelerometer (2-sec windows)

Fig. 4: 3-layered CNN architecture used in the experiments. While the size of the receptive fields of the neurons doubles from one convolutional area to the next, the number of feature maps is doubled as well (e.g., for the 3-layer CNN, 25, 50, and 100 neurons were used in the first, second, and third layers, respectively.

a rotating drum covered with textured materials. Figure 2 displays photographs of segments of the texture materials utilized in the data collection process. The 3D accelerometer helps record data in X, Y, and Z channels, which were resampled at 200 Hz, resulting in a 3-by-4000 dataset.

A. Experimental Setup

We have defined a sampling window that randomly slides over the first and second 10-seconds of sensor data to create training and test set signals, respectively.

As for the proposed architecture, which utilizes semi-supervised learning, its training process involved alternating between supervised and unsupervised learning in each iteration of the algorithm, as discussed in section III. For supervised tasks, the algorithm leverages the first 10 seconds of the recordings along with the available class labels. For unsupervised tasks, the algorithm employs the remaining 10 seconds of the recordings and relies on the contextual guidance principle. With regards to the tactile dataset, the use of 30 contextual groups was found to be nearly optimal for training the proposed architecture. Increasing the number of contextual groups had limited impact on accuracy, while reducing the number of contextual groups resulted in decreased transferability of features and decreased accuracy.

As illustrated in Figure 5, the supervised classification accuracy of CG-CNN, an unsupervised network, is lower than that of BeiimNet, a semi-supervised network. This can be attributed to the improved performance of semi-supervised learning through contextually guided training. On the other hand, the supervised network, which does not incorporate contextual guidance, exhibits a lower generalization ability compared to BeiimNet.

V. Conclusions

Contextually Guided Convolutional Neural Networks (CG-CNN) use unlabeled examples for contextual guidance in auxiliary classification tasks. By treating temporally adjacent sensor windows as similarly labeled, they form classes for internal supervision. By generating and solving these subtasks, CG-CNN learns the inherent structure and patterns present in the data, resulting in robust and transferrable representations. In this paper, we extended the application of CG-CNN to texture classification by incorporating semi-supervision. We achieved this extension by alternating between supervised and unsupervised cycles, which encouraged CG-CNN to develop increased sensitivity to the labeled examples during representation learning. Our experiments on the VibTac-12 texture dataset demonstrated that CG-CNN features generalize well to new and unseen textures, even with limited labeled training examples. By harnessing the wealth of unlabeled contextual data alongside the labeled examples, CG-CNN generated comprehensive and discriminative representations and performed favorably on VibTac-12 for texture classification with simple tactile sensors, such as accelerometers. By combining tactile sensing, self-supervision, and semi-supervised learning, we obtain robust and transferrable representations that have the potential to enhance various applications in robotics, prosthetics, material science, and haptic interfaces and to enable improved perception of textures in real-world scenarios.

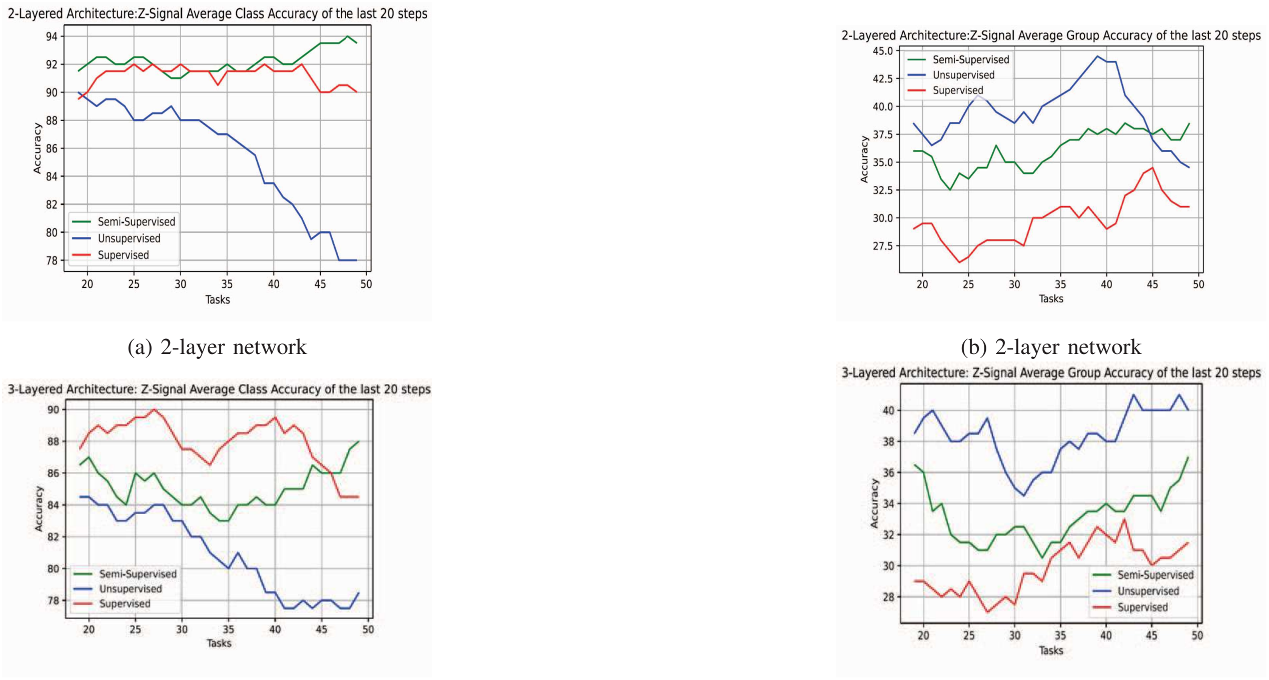
Acknowledgment

This work was supported, in part, by the National Science Foundation under grants No. 2003740 and No. 2302537.

References

[1] J. Enguehard, P. O'Halloran, and A. Gholipour, "Semi-supervised learning with deep embedded clustering for image classification and segmentation," *IEEE Access*, vol. 7, pp. 11 093–11 104, 2019.

[2] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-supervised learning*. MIT press, 2010.



(a) 2-layer network

(b) 2-layer network

(c) 3-layer network

(d) 3-layer network

Fig. 5: Accuracy of Supervised, Unsupervised, and Semi-Supervised networks on tactile dataset (Z-sensor). 'Class accuracy' refers to the network's ability to correctly predict the texture class of an input. 'Group accuracy' denotes the network's success in accurately identifying the contextual classes associated with the input.

[3] O. Kursun, S. Dinc, and O. V. Favorov, "Contextually guided convolutional neural networks for learning most transferable representations," in *24th IEEE International Symposium on Multimedia (IEEE-ISM), Naples, Italy*, December 2022.

[4] O. Kursun and O. V. Favorov, "Suitability of features of deep convolutional neural networks for modeling somatosensory information processing." [Online]. Available: https://doi.org/10.1117/12.2518573.

[5] O. Kursun, S. Dinc, and O. V. Favorov, "Contextually guided convolutional neural networks for learning most transferable representations," arXiv:2103.01566 [Cs], March. [Online]. Available: http://arxiv.org/abs/2103.01566.

[6] O. V. Favorov and O. Kursun, "Neocortical layer 4 as a pluripotent function linearizer," *Journal of neurophysiology*, vol. 105, no. 3, pp. 1342–1360, 2011.

[7] O. Kursun, E. Alpaydin, and O. V. Favorov, "Canonical correlation analysis using within-class coupling," *Pattern Recogn. Lett*, vol. 32, no. 2, p. 134–44, 2011. [Online]. Available: https://doi.org/10.1016/j.patrec.2010.09.025.

[8] J. Hawkins, S. Ahmad, and Y. Cui, "A theory of how columns in the neocortex enable learning the structure of the world," *Frontiers in neural circuits*, p. 81, 2017.

[9] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus, H. Arshad, A. A. Kazaure, U. Gana, and M. U. Kiru, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158 820–158 846, 2019.

[10] B. Sarsekeyev, "Beiimnet: Semi-supervised contextually guided convolutional neural networks," Master's thesis, University of Central Arkansas, Dept. of Computer Science, 4 2021, advisor: Your Name.

[11] J. Zhou, L. Zheng, Y. Wang, and C. Gogu, "A multistage deep transfer learning method for machinery fault diagnostics across diverse working," *Conditions and Devices."IEEE Access*, vol. 8, p. 80879–98, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2990739.

[12] O. Kursun and A. Patooghy, "An embedded system for collection and real-time classification of a tactile dataset," *IEEE Access*, vol. 8, p. 97462–73, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2996576.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[15] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," 2018. [Online]. Available: https://arxiv.org/abs/1808.06670

[16] O. Kursun and A. Patooghy, "Vibtac-12: Texture dataset collected by tactile sensors," 2020. [Online]. Available: https://dx.doi.org/10.21227/kwsy-x398

[17] M. Aminolroaya and S. Nahavandi, "Cg-cnn: a convolutional neural network with compact feature representation," *Neural Computing and Applications*, vol. 33, no. 16, pp. 9551–9567, 2021.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" arXiv:1411.1792 [Cs], November. [Online]. Available: http://arxiv.org/abs/1411.1792.

[19] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[20] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, no. May, p. 54–71, 2019. [Online]. Available: https://doi.org/10.1016/j.neunet.2019.01.012.

[21] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," arXiv:1807.05511 [Cs], April. [Online]. Available: http://arxiv.org/abs/1807.05511.

[22] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0.