

LCANETS++: ROBUST AUDIO CLASSIFICATION USING MULTI-LAYER NEURAL NETWORKS WITH LATERAL COMPETITION

Sayanton V. Dibbo^{§†*}, Juston S. Moore[§], Garrett T. Kenyon[§], Michael A. Teti[§]

[§] Los Alamos National Laboratory, Los Alamos, NM, USA, [†]Dartmouth College, Hanover, NH, USA

ABSTRACT

Audio classification aims at recognizing audio signals, including speech commands or sound events. However, current audio classifiers are susceptible to perturbations and adversarial attacks. In addition, real-world audio classification tasks often suffer from limited labeled data. To help bridge these gaps, previous work developed neuro-inspired convolutional neural networks (CNNs) with sparse coding via the Locally Competitive Algorithm (LCA) in the first layer (i.e., LCANets) for computer vision. LCANets learn in a combination of supervised and unsupervised learning, reducing dependency on labeled samples. Motivated by the fact that auditory cortex is also sparse, we extend LCANets to audio recognition tasks and introduce LCANets++, which are CNNs that perform sparse coding in multiple layers via LCA. We demonstrate that LCANets++ are more robust than standard CNNs and LCANets against perturbations, e.g., *background noise*, as well as black-box and white-box attacks, e.g., *evasion* and *fast gradient sign (FGSM)* attacks.

Index Terms— Audio Classification, Robustness, Neural Networks, Adversarial Machine Learning

1. INTRODUCTION

Audio signal classification for the purposes of sound recognition (SR) or sound event detection (SE) has become an active area of research interest [1, 2]. This includes using the Convolutional Neural Network (CNN) models for understanding human speech words, e.g., ‘yes’, ‘stop’, etc., or classifying sound events like ‘baby cries’, and ‘barking’. However, standard CNNs are notoriously susceptible to perturbations or adversarial attacks [3, 4, 5, 6]. Standard audio classification models depend highly on large labeled datasets for better performances [7, 8], but large labeled datasets can be scarce for many common tasks, such as speaker identification. Generating augmented samples for audio data is one proposed approach to mitigate this challenge, but data augmentation can be time-consuming and expensive [9, 10]. Therefore, it is crucial to develop audio classifiers that can learn robust features with limited labeled samples.

Recent studies have shown that CNNs that are more similar to the primary visual cortex are more robust than standard CNNs [11]. Based on this, previous work developed CNNs in which the first layer performed sparse coding via the Locally Competitive Algorithm (LCA) [12, 13, 14], which is a biologically plausible model of the primate primary visual cortex [15]. These CNNs, which we refer to as LCANets, were shown to be more robust than standard CNNs on standard CV tasks. However, there are two issues with this approach we address here. First, sparse coding models were designed to model the visual cortex, so it is unclear how they will impact the performance of CNNs on audio classification tasks. Second, these LCANets were robust to natural corruptions, but they were susceptible to white-box adversarial attacks [13] unless the exact attack was known before hand [14].

Motivated by this, we introduce multi-layer LCANets, which perform sparse coding in multiple CNN layers. We refer to these multilayer LCANets as LCANets++ and train them on audio classification tasks. To test the robustness of LCANets++ relative to LCANets and standard CNNs, we first conducted experiments with different audio perturbations, e.g., *background noise*. In addition, we show that our proposed LCANets++ are more robust compared to the state-of-the-art (SOTA) models (e.g., ResNet18, standard CNN) and LCANets against white-box attacks, i.e., *fast gradient sign attack (FGSM)* [16] and *projected gradient descent attack (PGD)* [17], as well as black-box attacks, i.e., *evasion attack*.

2. PROPOSED METHOD

2.1. LCA Layer

As presented in Fig. 1, LCA layer is the basic building block for the LCA frontend and our proposed LCANets++. LCA layer converts the input \mathcal{X} to a coding \mathcal{C} , i.e., representation of the input \mathcal{X} , leveraging the least number of active neurons (i.e., features). The goal of the reconstruction minimization problem applied here is to find the sparse coding representation \mathcal{C} closest possible to original input \mathcal{X} as follows:

$$\mathcal{L}_{re} = \min_{\mathcal{C}} \frac{1}{2} \|\mathcal{X} - \mathcal{C} \otimes \Phi\|_2^2 + \lambda \|\mathcal{C}\|_1 \quad (1)$$

* Author performed the work while working at the Los Alamos National Laboratory

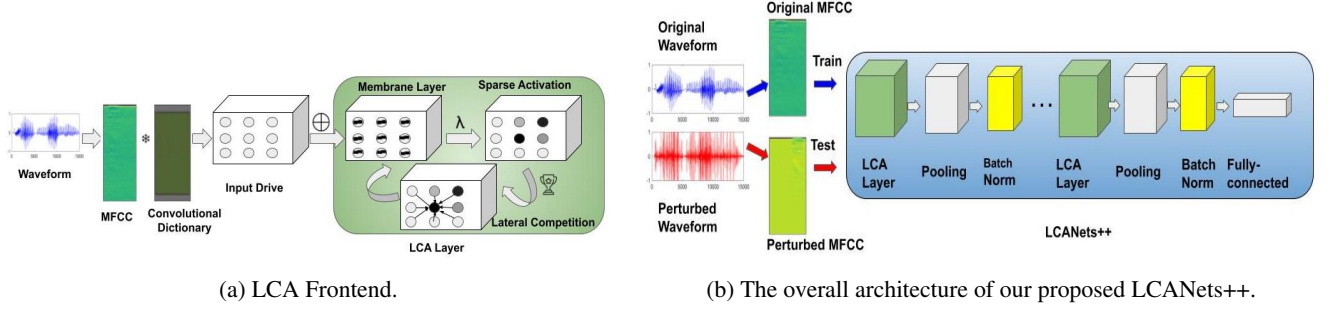


Fig. 1: An overview of (a.) LCA frontend and (b.) pipeline of our proposed LCA Nets++, utilizing sparse coding via multiple LCA layers in the state-of-the-art (SOTA) CNN backbone, enabling lower misclassification on perturbed test sets or attacks.

where \mathcal{L}_{re} denotes the reconstruction loss, \mathcal{X} is the original input, \mathcal{C} is the sparse code, \otimes is transpose convolution, Φ is dictionary components learned last iteration, and λ is the trade-off (i.e., regularization) constant. LCA layers perform lateral competitions to fire neurons and a neuron membrane follows the following ordinary differential equation [13]:

$$\hat{\mathcal{M}}(t) = \frac{1}{\gamma} [\mathcal{D}(t) - \mathcal{M}(t) - \mathcal{C}(t) * \mathcal{S} + \mathcal{C}(t)] \quad (2)$$

where γ is time constant, $\mathcal{D}(t)$ stands for neuron’s input drive obtained by convolution of inputs with dictionary, i.e., $\mathcal{X} * \Phi$; $\mathcal{M}(t)$ is neuron’s membrane potential, $\mathcal{S} = \Phi * \Phi$ is the pair-wise feature similarity, and $\mathcal{C}(t)$ is the neuron’s firing rate obtained by applying a soft threshold activation on the membrane potential $\mathcal{M}(t)$. Coordinate ascent is used to learn dictionary Φ , which solves for \mathcal{C} , given an input batch using LCA and then updates Φ with stochastic gradient descent (SGD).

2.2. LCA Frontend

LCA frontend is the unsupervised pre-training part of the LCA Net, as shown in Fig. 1a. It basically consists of the raw audio waveform converted to MFCCs, as input signal \mathcal{X} and the LCA layer to compute the sparse representation \mathcal{C} of the input \mathcal{X} , which can then feed to conventional CNN layers for the classification task.

2.3. LCA Nets

LCA Nets for the audio classification consist of the LCA frontend and followed by CNN layers. LCA frontend learns in unsupervised fashion and then passes the computed sparse code \mathcal{C} to the CNN layers to finally perform the classification task. One major difference is that, the sparse code \mathcal{C} does not need to recompute back to original input \mathcal{X} before feeding to CNN layer, as other reconstruction-based models usually do. This makes the LCA Nets more effective against perturbations, while reducing dependency on labeled audio samples.

2.4. LCA Nets++

We present the overview of our proposed LCA Nets++ in Fig. 1b. The basic building block of our proposed LCA Nets++ is the LCA layers. In this architecture, multiple LCA layers are inserted that learn in unsupervised fashion. The convolutional layers in SOTA CNN networks are replaced by the LCA layers, performing sparse coding in each layer. Similar to [18], in order to reduce over-sparsity, in between two consecutive sparse layers (i.e., LCA layers), a dense layer, i.e., batch normalization layer, is mounted (Fig. 1b) in our LCA Nets++.

3. EXPERIMENTS

In this section, details of the experimental setup, including dataset, pre-processing, and models, are described.

3.1. Dataset and Pre-processing

We experiment with Google Speech Commands v2 [19] dataset. This dataset has audio waveforms of 35 classes of human speech commands like “yes,” “no,” and “left,” “right.” We perform pre-processing on the raw waveforms of the three influential classes, i.e., “yes,” “no,” and “stop” to obtain the Mel-frequency cepstral coefficient (MFCC) [8] features and train all the models with the MFCCs of the waveforms.

3.2. Models

We experiment with regular CNN models with 2 convolutional layers. In our LCA Nets++ on CNN model, we replace both convolutional layers with the LCA layers. We also compare our LCA Nets++ with the larger SOTA model, i.e., ResNet18. In the ResNet18 model, we replace the alternative convolutional layers in the first block with LCA layers to obtain the ResNet18_LCA++ model. To experiment with the performance of LCA Nets++ against *white-box* or *black-box* attacks, we consider the regularization constant $\lambda = 1.00$ for better sparse representations and hence, improved robustness against perturbations or attacks.

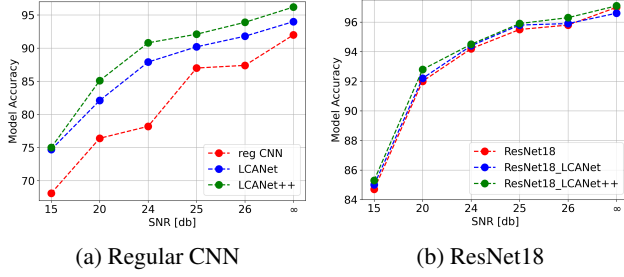


Fig. 2: Comparisons of our LCA nets++ and other SOTA models against perturbations with *background noise*.

3.3. Experimental Setup

We run all the experiments on 8 nodes NVIDIA A100-PCIE-40GB GPUs with 64-128 cores on the cluster. We use the Pytorch framework to develop the LCA class and LCA nets++ implementations. We consider a train test split of 70% and 30% for all the models experimented in this work. For the *background noise* experiment, we train models for 50 epochs, for rest of the experiments models are trained with 20 epochs. We consider 0.0001 learning rate. We use SGD optimizer with 0.9 momentum for optimization. In order to add background noise, we impose background noise on all test set raw waveforms of audio clips, tuning the SNR [db] values to obtain different perturbed test sets. Similarly, we consider perturbing MFCCs with different ϵ values.

4. RESULTS AND ANALYSIS

In this section, we illustrate the key results of our experiments on different perturbations and adversarial attacks.

4.1. Input Perturbations

We test the robustness of standard CNNs without the LCA layer(s), LCA nets, and our proposed LCA nets++ to perturbations. We experiment with two different cases of input perturbations: i) *background noise* on the raw audio clips and ii) *gaussian noise* on MFCCs to compare robustness against both perturbation scenarios.

4.1.1. Background Noise

In Fig. 2a, we present the performance of regular CNN, LCA net, and our proposed LCA net++, tested on different perturbed test sets with *background noise*, varying SNR [db] values. Observe that the regular CNN model performance drastically goes down as more perturbation is applied to original waveforms (i.e., lower SNRs). Whereas, LCA net goes down slowly with increasing perturbations, and our proposed LCA net++ shows the most robustness compared to LCA net and regular CNN models, as presented in Fig. 2a. This is

Table 1: Performance comparisons against perturbations with Background Noise on waveforms

Model	SNR = 15db	20db	24db	25db	∞
CNN	0.692	0.788	0.793	0.858	0.920
LCA net	0.760	0.840	0.876	0.904	0.940
LCA net++	0.768	0.847	0.903	0.914	0.962
ResNet18	0.847	0.920	0.942	0.955	0.970
ResNet18_LCA	0.850	0.922	0.944	0.958	0.966
ResNet18_LCA++	0.853	0.928	0.945	0.959	0.971

Table 2: Performance comparisons against perturbations with Gaussian Noise on MFCCs

Model	ϵ = 0	0.01	0.02	0.03	0.04	0.05
CNN	0.866	0.864	0.863	0.863	0.858	0.856
LCA net	0.939	0.938	0.935	0.925	0.909	0.883
LCA net++	0.950	0.943	0.939	0.927	0.914	0.900

attributed to the fact that LCA layers learn in an unsupervised fashion, reducing the numbers of the neurons activated through lateral competitions. These fewer activated neurons represent the most relevant input features, which are less impacted by slight perturbations.

We also test the robustness of LCA nets++ on larger models, i.e., ResNet18 model with 18 layers. As presented in Fig. 2b, we observe that the ResNet18 with multilayer LCAs, i.e., ResNet18_LCA net++ outperforms regular ResNet18 and ResNet18 with LCA in the first layer, i.e., ResNet18_LCA net. From Table 1, we find that for the ResNet18 architecture, LCA nets++ slightly improves the robustness on perturbed test sets than regular ResNet18 without LCA layers, as opposed to significantly higher robustness LCA nets++ exhibited on regular CNN model. Larger model with more layers and parameters make ResNet18 inherently more robust than regular CNNs, resulting in LCA nets++ to boost up only slightly in ResNet18 than regular CNNs.

4.1.2. Gaussian Noise

We impose *Gaussian noise* on the MFCCs varying ϵ values. As presented in Table. 2, with increasing the ϵ (more perturbations), performance of the regular CNN model goes down. Also, LCA net and LCA net++ performance slightly goes down, but still, the models with LCA layers show more robustness compared to the model without LCA layers, i.e., regular CNN model. This shows that our LCA nets++ are more robust not only against perturbations on raw waveforms, but also against perturbations on the feature space, i.e., MFCCs.

Table 3: Comparisons against *white-box* attacks

Attack	Model	ϵ	$= 0$	0.01	0.016	0.02	0.03
	CNN		0.866	0.439	0.196	0.108	0.017
FGSM	LCANet		0.939	0.261	0.123	0.092	0.062
	LCANet++		0.950	0.679	0.418	0.417	0.414
	CNN		0.866	0.382	0.147	0.073	0.025
PGD	LCANet		0.939	0.028	0.005	0.005	0.005
	LCANet++		0.950	0.588	0.585	0.579	0.567

4.2. Adversarial Attacks

We experiment with adversarial attacks having different capabilities. For experimental purposes, we consider both the *white-box* and *black-box* adversarial attacks.

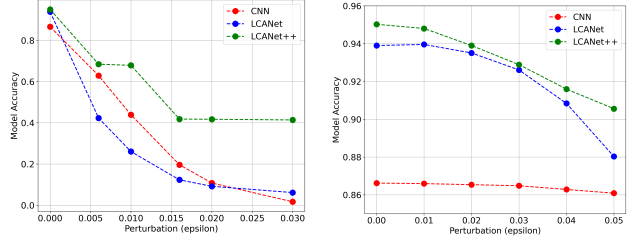
4.2.1. White-box Attacks

In *white-box* attacks, an adversary has more capabilities like having access to the model architectures, including model parameters, weights, and gradients. We consider two different types of *white-box* attacks, i.e., *FGSM* [16] and *PGD* [17]. In both attacks, the adversary utilizes the gradients to perturb the MFCCs of test sets to misclassify them during inference.

We present the performances of the regular CNN model and LCANets, as well as our proposed LCANets++, against the FGSM attack in Fig. 3a. We find that the regular CNN is not very robust, and its performance goes down, as perturbations (ϵ) go higher. We observe that, the single-layer LCA, i.e., LCANets are not robust against the *white-box* FGSM attack, which is consistent to findings in [13] for CV tasks. However, our proposed multi-layer LCANets++ outperforms the CNN model and LCANets on audio classification against the FGSM attack. In Fig. 3a, we observe that LCANets++ performance decreases comparatively slowly as attack becomes stronger with higher perturbations (ϵ). We also experiment with another *white-box* attack, i.e., PGD attack, where LCANets++ consistently show more robustness than SOTA models and LCANets, as shown in Table 3.

4.2.2. Black-box Attacks

We experiment with the *black-box* evasion attack, where the adversary has no access to the model gradients. In this attack, an adversary only has query access to the model and can get predictions from the model utilizing the query access. In our setup, the adversary is able to make queries to the original target model and get predictions from the model. The adversary utilizes the predictions and input queries to develop a surrogate model. The surrogate model generates the perturbed samples, varying perturbations (ϵ), and we tested the performance of the original models on these perturbed test sets. We present the performances of regular CNN, LCANet,

(a) FGSM (*white-box*) Attack (b) Evasion (*black-box*) Attack**Fig. 3:** Comparisons of LCANets++ and SOTA models on L_∞ norm *white-box* attacks.**Table 4:** Comparisons against *black-box* (Evasion) attack

Model	ϵ	$= 0$	0.01	0.02	0.03	0.04	0.05
CNN		0.866	0.865	0.865	0.864	0.862	0.860
LCANet		0.939	0.939	0.935	0.926	0.908	0.880
LCANet++		0.950	0.948	0.939	0.928	0.915	0.905

and our proposed LCANet++ against the *black-box* evasion attack in Fig. 3b. We observe that, in *black-box* evasion attack, LCANet shows more robustness compared to CNN and LCANet++ outperforms all the models on perturbed test sets (i.e., $\epsilon > 0$). Note that, models are trained for 20 epochs with three audio classes (i.e., limited samples), which might lead to a significant performance gap among the regular CNN and LCA-based models on unperturbed test sets $\epsilon = 0$ in Table. 4.

5. CONCLUSIONS

In this work, we developed CNNs with sparse coding in multiple layers, referred to as LCANets++. We showed that LCANets++ can be easily implemented using regular CNNs like ResNet18. Our empirical analysis shows that LCANets++ can be used in audio classifiers to increase robustness to noise and adversarial attacks relative to LCANets and standard CNNs. In addition, we observe how the unsupervised training with LCA and number of LCA layers impacts clean and robust test accuracy. Overall, our work sheds light into future directions in designing privacy-preserving robust audio classifiers.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge support from the Advanced Scientific Computing Research (ASCR) program office in the Department of Energy's (DOE) Office of Science, award #77902, as well as the Center for Nonlinear Studies and the Cyber Summer School at Los Alamos National Laboratory.

7. REFERENCES

- [1] Roman A Solovyev, Maxim Vakhrushev, Alexander Radionov, Irina I Romanova, Aleksandr A Amerikanov, Vladimir Aliev, and Alexey A Shvets, “Deep learning approaches for understanding simple speech commands,” in *2020 IEEE 40th international conference on electronics and nanotechnology (ELNANO)*. IEEE, 2020, pp. 688–693.
- [2] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., “Cnn architectures for large-scale audio classification,” in *IEEE international Conf. on acoustics, speech and signal processing*, 2017, pp. 131–135.
- [3] Nicholas Carlini and David Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in *IEEE security and privacy workshops*. IEEE, 2018, pp. 1–7.
- [4] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan, “Enabling fast and universal audio adversarial attack using generative model,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 14129–14137.
- [5] Sayanton V Dibbo, Dae Lim Chung, and Shagufta Mehnaz, “Model inversion attack with least information and an in-depth analysis of its disparate vulnerability,” in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2023, pp. 119–135.
- [6] Sayanton V Dibbo, “Sok: Model inversion attack landscape: Taxonomy, challenges, and future roadmap,” in *IEEE 36th Computer Security Foundations Symposium*. IEEE Computer Society, 2023, pp. 408–425.
- [7] Yu Wang, Nicholas J Bryan, Mark Cartwright, Juan Pablo Bello, and Justin Salamon, “Few-shot continual learning for audio classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 321–325.
- [8] Sayanton V Dibbo, William Cheung, and Sudip Vhaduri, “On-phone cnn model-based implicit authentication to secure iot wearables,” in *5th Intl Conf. on Safety and Security with IoT*. Springer, 2022, pp. 19–34.
- [9] Yuichiro Koyama, Kazuhide Shigemi, Masafumi Takahashi, Kazuki Shimada, Naoya Takahashi, Emiru Tsunoo, Shusuke Takahashi, and Yuki Mitsufuji, “Spatial data augmentation with simulated room impulse responses for sound event localization and detection,” in *IEEE Intl Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8872–8876.
- [10] John Lorenzo Bautista, Yun Kyung Lee, and Hyun Soon Shin, “Speech emotion recognition based on parallel cnn-attention networks with multi-fold data augmentation,” *Electronics*, vol. 11, no. 23, pp. 3935, 2022.
- [11] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David Cox, and James J DiCarlo, “Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13073–13087, 2020.
- [12] Dylan M Paiton, Charles G Frye, Sheng Y Lundquist, Joel D Bowen, Ryan Zarccone, and Bruno A Olshausen, “Selectivity and robustness of sparse coding networks,” *Journal of vision*, pp. 10–10, 2020.
- [13] Michael Teti, Garrett Kenyon, Ben Migliori, and Juston Moore, “Lcanets: Lateral competition improves robustness against corruption and attack,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 21232–21252.
- [14] Mingyang Li, Pengyuan Zhai, Shengbang Tong, Xingjian Gao, Shao-Lun Huang, Zhihui Zhu, Chong You, Yi Ma, et al., “Revisiting sparse convolutional model for visual recognition,” *Advances in Neural Information Processing Systems*, pp. 10492–10504, 2022.
- [15] Bruno A Olshausen and David J Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [16] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al., “Adversarial attacks and defences competition,” in *NIPS’17 Competition: Building Intelligent Systems*, 2018, pp. 195–231.
- [17] Ping-Yeh Chiang, Jonas Geiping, Micah Goldblum, Tom Goldstein, Renkun Ni, Steven Reich, and Ali Shafahi, “Witchcraft: Efficient pgd attacks with random step size,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3747–3751.
- [18] Yubei Chen, Zeyu Yun, Yi Ma, Bruno Olshausen, and Yann LeCun, “Minimalistic unsupervised representation learning with the sparse manifold transform,” in *11th International Conference on Learning Representations*, 2022.
- [19] Pete Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” *arXiv preprint arXiv:1804.03209*, 2018.