



Improving Classification with CNNs using Wavelet Pooling with Nesterov-Accelerated Adam

Allison M Rossetto and Wenjin Zhou

University of Massachusetts Lowell, Lowell, Massachusetts, U.S.A
Allison.Rossetto@student.uml.edu Wenjin.Zhou@uml.edu

Abstract

Wavelet pooling methods can improve the classification accuracy of Convolutional Neural Networks (CNNs). Combining wavelet pooling with the Nesterov-accelerated Adam (NAdam) gradient calculation method can improve both the accuracy of the CNN. We have implemented wavelet pooling with NAdam in this work using both a Haar wavelet (*WavPool-NH*) and a Shannon wavelet (*WavPool-NS*). The *WavPool-NH* and *WavPool-NS* methods are most accurate of the methods we considered for the MNIST and LIDC-IDRI lung tumor data-sets. The *WavPool-NH* and *WavPool-NS* implementations have an accuracy of 95.92% and 95.52%, respectively, on the LIDC-IDRI data-set. This is an improvement from the 92.93% accuracy obtained on this data-set with the max pooling method. The *WavPool* methods also avoid overfitting which is a concern with max pooling. We also found *WavPool* performed fairly well on the CIFAR-10 data-set, however, overfitting was an issue with all the methods we considered. Wavelet pooling, especially when combined with an adaptive gradient and wavelets chosen specifically for the data, has the potential to outperform current methods.

1 Introduction

Convolutional Neural Networks (CNN) have advanced image classification, object detection, and other image analysis applications. Wavelet pooling has the potential to improve the results of CNNs for some applications. Wavelets have been incorporated into CNNs to improve results before this work [5]. They are often used to do texture classification [9] or evaluate audio signals by adding wavelet analysis techniques as a part of the deep learning process [4]. They've also been used to improve image segmentation using CNNs [8]. Recently, it has been proposed that wavelet analysis can be used to improve CNNs by using wavelet analysis as a pooling method [27]. We have combined this recently proposed technique of wavelet pooling with the Nesterov-accelerated Adam (NAdam) [7] gradient descent method in order to improve classification. NAdam incorporates the Nesterov momentum [18] into the Adaptive moment estimation(Adam) stochastic optimization method [11, 7] for gradient descent. This provides us with an efficient and quick converging method for gradient descent. We apply this method to both lung cancer data, which is a real and critical application, as well as some of the commonly used testing data sets to show how combining these methods leads to improved classification.

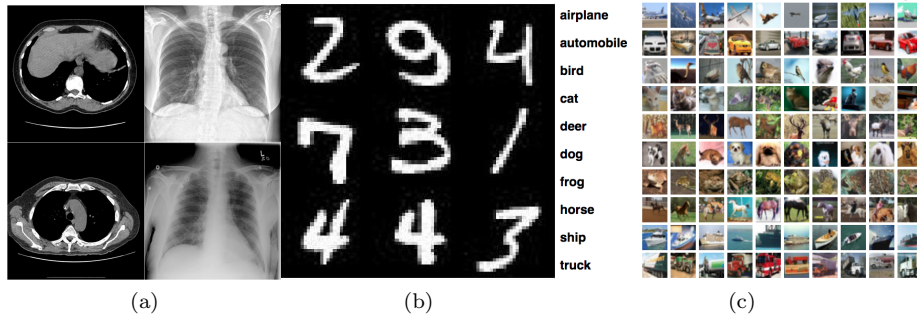


Figure 1: Example images from the (a) the LODC-IDRI lung tumor, (b) MNIST, (c) CIFAR10 [13] datasets.

We have chosen to also work with the LIDC-IDRI lung tumor data-set [2, 3]. Lung cancer is a leading cause of cancer death in the United States with over 150,000 deaths per year [23]. In fact, 19% of all cancer deaths worldwide are caused by lung cancer, with over 1.6 million deaths per year [25]. This is a pressing problem, one that deep learning is well equipped to help solve. This is because early detection and proper treatment is an essential part of effectively treating lung cancers [17]. However, lung cancer diagnoses have a false positive rate of over 96% [6, 10]. Lung cancer can also be diagnosed too late, as some cases are asymptomatic for long periods of time, sometimes even into late stages of cancer. One way to try to address this problem is by developing CNN techniques to improve the accuracy and decrease the false positive rates of identification. The method we cover here is a first step in a larger scheme to find and identify lung tumors. In this work, we apply wavelet pooling to improve the classification of patient scans as cancer being present or not using several different wavelet pooling techniques.

2 Methods

The discussion of the methods used can be separated into four subsections: (a) Datasets, (b) Wavelet Pooling, (c) NAdam, and (d) Architectures of the CNNs. We have implemented all the CNNs, the *WavPool* methods, and NAdam in MATLAB 2016b [16] using the MatConvNet package [26].

2.1 Datasets

We use the LIDC-IDRI lung tumor dataset [2, 3] in this work. This dataset consists of the Computed Tomography (CT) scans and annotations of 1,018 patients. Figure 1(a) shows some sample images from the dataset. We use a 70% training and 30% testing rate with the lung tumor data.

The LIDC-IDRI lung tumor dataset is annotated to note nodule size and location for each patient. These annotations were created in two phases. The first phase was a "blind-read" phase by four radiologists. Each radiologist reviewed and noted any nodules they found in the CT scans, both cancerous and benign, classifying each nodule by size. The second phase had each radiologist review the work of the other three and either confirm or correct the annotations. This way as many nodules were identified as accurately as possible. This two phase annotation method tries to overcome any issues of false positives, false negatives, or missed modules. We

simplify the classification task in this work by looking at only whether or not cancerous nodules were identified in a scan.

In order to analyze our methodology, we also use two standard datasets for testing: MNIST [14] and CIFAR-10 [13, 12]. MNIST is a set of 70,000 images of handwritten digits (0 through 9). Figure 1(b) shows some example images from this dataset. The training set uses 60,000 of images and the rest for testing. CIFAR-10 is a set of 60,000 color images. Each of the images falls into one of 10 different object classes, all of which are the same size. Figure 1(c) shows some example images from this dataset. For this dataset, 50,000 of the images are used for testing and the rest for training. MNIST and CIFAR-10 are useful for this work since they are standards by which we can compare our methods with others in a meaningful way.

2.2 Wavelet Pooling

Wavelet pooling [27] uses wavelets as a method of reducing feature map dimensions during the pooling step. This differs from traditional pooling methods since it does not use a neighborhood sub-sampling method to reduce the dimensions of the feature maps. Neighborhood reduction [19] produces artifacts which wavelet pooling is specifically designed to reduce by discarding first-order sub-bands. This lessens the number of jagged edges and other artifacts that cause problems during image classification tasks and helps to better capture data compression.

The implementation of wavelet pooling we use in our work, which we will refer to as *WavPool*, has been implemented with both the Haar and Shannon wavelets. The baseline implementation uses a 2D Haar wavelet and Stochastic Gradient Descent (SGD) method. We have added the Shannon wavelet as well as implementing the SGD method here. Our two improved methods use the NAdam gradient descent methods with the two wavelets: *WavPool-NH* and *WavPool-NS*. The implementation of the wavelet pooling using the Haar wavelet with NAdam is denoted as *WavPool-NH*. *WavPool-NS* is the implementation of wavelet pooling using a Shannon wavelet with NAdam.

In order to use these wavelets for pooling, we need to define our forward and backward propagation methods using the wavelets. Forward propagation uses a second order decomposition in the wavelet domain of the 2D discrete wavelet transform (DWT) [22] while back propagation is achieved by reversing the forward propagation process. It should also be noted that after first order wavelet decomposition, the detail coefficient sub-bands are up-sampled by a factor of 2. The 2D DWT functions used for forward propagation are defined as follows:

$$W_{\Psi(j_0,k,l)} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \Psi_{j_0,k,l}(m,n)$$

$$W_{\phi(j,k,l)} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \phi_{j,k,l}^{i=H,V,D}(m,n)$$

where W_{Ψ} and W_{ϕ} are the wavelet and scaling coefficients, Ψ is the wavelet function and ϕ is the scaling function, $f(m,n)$ is the image feature at position m,n , k and l are the wavelet position, M and N are the feature dimensions, and H , V , and D are the horizontal, vertical, and diagonal scaling coefficients.

The image reconstruction of the second order wavelet sub-bands for the back propagation uses the inverse DWT function. The feature undergoes a first order wavelet decomposition. The detail coefficient sub-bands are then up-sampled by a factor of 2 to create the first level

decomposition which then becomes the second level decomposition. The second order decomposition is used to reconstruct the image feature in order to continue the back propagation. Again, note that this is the reverse of the forward propagation process. The 2D IDWT used for back propagation is defined as:

$$f(m, n) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\Psi(j_0, k, l) \Psi_{j_0, k, l}(m, n) +$$

$$\frac{1}{\sqrt{MN}} \sum_{j=j_0}^{\text{inf}} \sum_m \sum_n W_\phi^i(j_0, k, l) \phi_{j, k, l}^i(m, n), j = j_0 = 2$$

We have used the Haar wavelet, the simplest of the wavelets, here for two reasons: (1) because it is simple and computationally cheap, and (2) because it was used in the original wavelet pooling work. Using the same wavelet gives us a better understanding of how the addition of the NAdam algorithm for the gradient descent affects the CNNs. We also use the Shannon wavelet because it can be either discrete or complex and continuous. It is still simple wavelet but it has many real-world applications [1]. We have the wavelet pooling method with both the Haar wavelet and the Shannon wavelet in the MatConvNet package. The Haar wavelet is simple to implement, however, the Shannon wavelet is slightly more difficult. Both of the wavelet methods were tested outside the CNN to ensure proper implementation before integration into the CNN architectures.

The Haar wavelet is defined by a wavelet function [24], $\Psi(t)$, and a scaling function, $\phi(t)$. These are defined as follows:

$$\Psi(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2} \\ -1 & \frac{1}{2} \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

The Shannon wavelet is also defined by a wavelet function [15], $\Psi(t)$, and a scaling function, $\phi(t)$. These are defined as follows:

$$\Psi^{(Sha)}(t) = 2\text{sinc}(2t - 1) - \text{sinc}(t)$$

where $\text{sinc}(t) = \sin(\pi t)/\pi t$.

$$\phi^{(Sha)}(t) = \frac{\sin(\pi t)}{\pi t} = \text{sinc}(t)$$

2.3 NAdam

The Adam algorithm [11] uses a combination of classical momentum and adaptive learning rates for improved gradient calculations. Nesterov’s accelerated gradient (NAG) [18] is a momentum calculation that speeds up the process of the gradient calculation. NAG can also be used as the momentum calculation method in the Adam algorithm. NAdam, as proposed by Dozet [7], improves upon Adam by using NAG instead of SGD for momentum. This speeds up the calculation and can help to improve gradient calculation results. It also has the added advantage of giving us better bounds than stochastic gradient descent for convex, non-stochastic objectives. We use it here since NAdam outperforms other algorithms in reducing training and validation loss. It achieves good results even without tuning. We have implemented the method as a solver

option in the MatConvNet package since the integration of different solvers has been simplified to give users better options for customizing their application.

2.4 Architecture of the CNNs

We use three separate CNN architectures for each of the datasets used to test the wavelet pooling with NAdam methodology. The basic architecture is the same for all the implementations of each of the applications. The implementations we have chosen for MNIST and CIFAR-10 are the same as those used in the previous wavelet paper [27] to provide the best comparison possible. These architectures are similar to the examples provided in with the MatConvNet package [26]. The architecture used for the lung tumor data is based on our previous work[20, 21].

The major differences are in the pooling and gradient descent methods used. We test all the possible combinations of *WavPool* and the two gradient descent methods, SGD and NAdam, in order to better understand how the changes we make in the implementation affect our results. The variations applied to each of our CNN implementations are as follows: (1) CNN using Max Pooling: max pooling with SGD, (2) CNN using *WavPool-H*: wavelet pooling using the Haar wavelet with SGD. (3) CNN using *WavPool-S*: wavelet pooling using the Shannon wavelet with SGD, (4) CNN using *WavPool-NH*: wavelet pooling using the Haar wavelet with NAdam, and (5) CNN using *WavPool-NS*: wavelet pooling using the Shannon wavelet with NAdam.

It is important to note that the maximum epochs are set for 20 epochs for MNIST while CIFAR-10 and LIDC-IDRI each use a maximum of 50 epochs. MNIST has a lower maximum epoch setting because it is easy to overfit with this dataset.

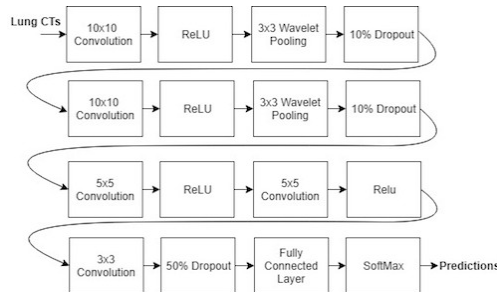


Figure 2: Architectures of the CNN for Lung Tumor classification with *WavPool*.

We developed a basic architecture in our previous work [21, 20] which classified lung CT scans as having cancer present. We have improved the accuracy by incorporating *WavPool* into our methodology. It is also important to note in this architecture that the sequence of a 10×10 convolutional layer, a ReLU layer, the wavelet pooling layer, and a 10% dropout layer are repeated twice. We also repeat the sequence of 5×5 convolutional layer followed by a ReLU layer. While not overly complex, we have found this architecture does well as a simple classifier for whether each scan contains tumors or not.

3 Results and Discussion

The CNNs were implemented in MATLAB [16] using the MatConvNet package [26]. This package is easily extendable and allowed us to implement the wavelet pooling, NAdam gradient, and both the Haar and Shannon wavelets. Two Nvidia GeForce GTX 970 GPUs with 16GB of video

RAM were used for training and testing the CNNs. We compare the results of these implementations in the following sections the results of using max pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS*.

3.1 MNIST

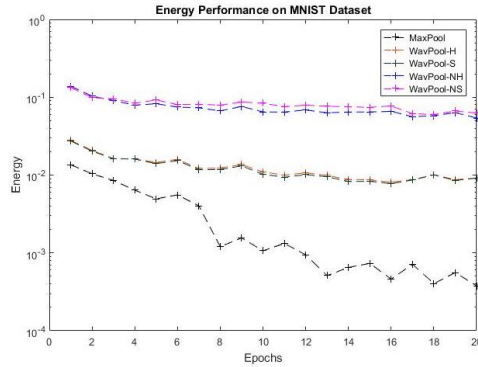


Figure 3: MNIST implementation energy performances.

Table 1: Comparison of Max Pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS* on MNIST dataset

MNIST Dataset		
Method	Accuracy	Time
Max Pooling	98.35%	699.81s
<i>WavPool-H</i>	97.35%	654.82s
<i>WavPool-S</i>	98.29%	673.23s
<i>WavPool-NH</i>	99.98%	537.82s
<i>WavPool-NS</i>	99.99%	599.71s

Table 1 shows the accuracy and time measures for the CNN with max pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS*. We can see that for this dataset the *WavPool-NH* and the *WavPool-NS* give us very similar results. *WavPool-H* and *WavPool-S* also give similar results. In fact, we find it interesting to note that *WavPool-H* and *WavPool-S* are extremely close in energy performance. All methods do fairly well in classifying this dataset. We can see from Figure 3 the max pooling overfits the data. We also see that all four implementations of the wavelet pooling have fairly smooth energy performances for the MNIST data. *WavPool-NH* and *WavPool-NS* have very similar results with this dataset with their energy performances and accuracy being very close. *WavPool-NH* is also the fastest of the classifiers. This is not unexpected with this dataset as this dataset does well with many types of pooling methods.

3.2 CIFAR-10

Table 2 shows the accuracy and time measures for the CNN with max pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, *WavPool-NS*. We see from this table that all of the implementations

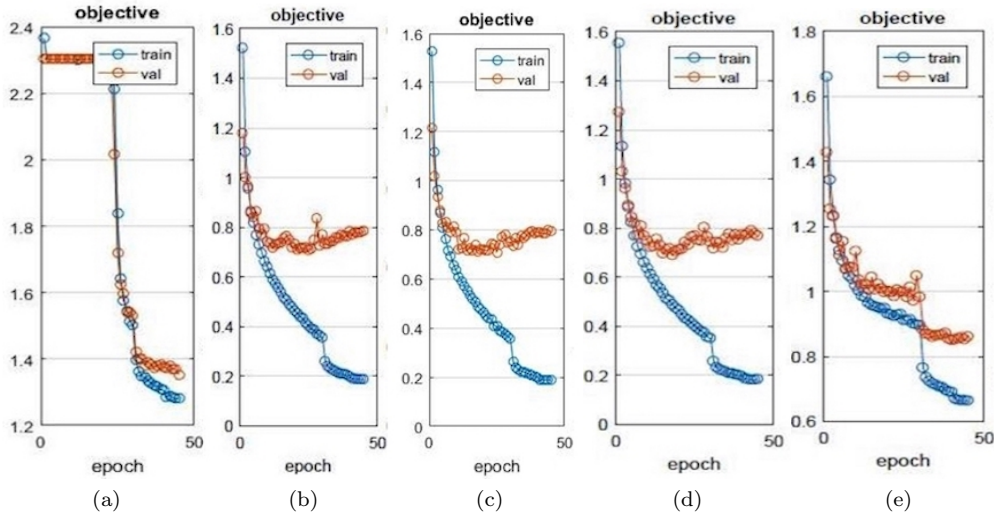


Figure 4: CIFAR-10 energy performance for (a) Max Pooling, (b) *WavPool-H*, (c) *WavPool-S*, (d) *WavPool-NH*, (e) *WavPool-NS*.

Table 2: Comparison of Max Pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS* on CIFAR-10 dataset

CIFAR-10 Dataset		
Method	Accuracy	Time
Max Pooling	76.08%	916.76s
<i>WavPool-H</i>	81.01%	1023.81s
<i>WavPool-S</i>	79.10%	1018.32s
<i>WavPool-NH</i>	80.59%	940.57s
<i>WavPool-NS</i>	85.68%	944.72s

have similar accuracies. However, Figure 4(a) shows that the max pooling overfits quickly with this dataset, though it is the fastest. Overfitting is also a problem with the wavelet pooling implementations as we can observe in Figures 4(b), (c), (d), and (e). This is not surprising as previous work [27] also found that overfitting was a problem when applying wavelet pooling to the CIFAR-10 dataset. We can see the advantage of using the more complex wavelet since here the Shannon wavelet improves the results and lessens overfitting.

3.3 LIDC-IDRI

Figure 5 shows the energy performance and error rates of the networks. We can see that we get relatively good performance from all of these methods. Overfitting is also not a problem with the *WavPool* implementations in this application, as we can see in 5 (b), (c), (d), and (e).

The accuracy of the max pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS* for the LIDC-IDRI dataset are shown in Table 3. The classification here is a simple binary one: cancerous tumors are either present (a positive case) or they are not (a negative case). Simplification of this problem is a useful first step for working with this data because of the

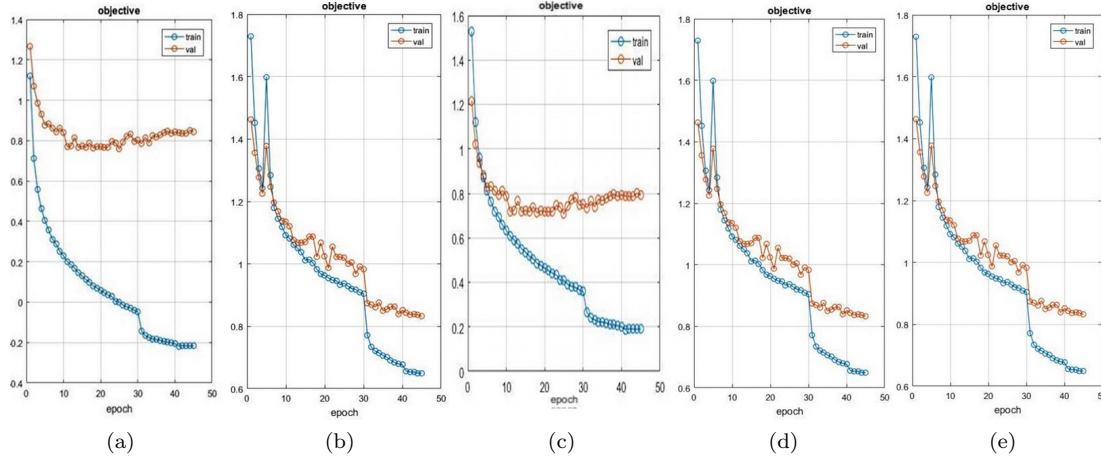


Figure 5: LIDC-IDRI energy performance for (a) Max Pooling, (b) *WavPool-H*, (c) *WavPool-S*, (d) *WavPool-NH*, (e) *WavPool-NS*.

Table 3: Comparison of Max Pooling, *WavPool-H*, *WavPool-S*, *WavPool-NH*, and *WavPool-NS* on LIDC-IDRI Lung Tumor dataset

LIDC-IDRI Dataset		
Method	Accuracy	Time
Max pooling	92.93%	56773.20s
<i>WavPool-H</i>	94.84%	59987.71s
<i>WavPool-S</i>	94.51%	60193.34s
<i>WavPool-NH</i>	95.92%	68371.22s
<i>WavPool-NS</i>	95.52%	69214.49s

difficult nature of identification and classification of tumors. Complex classification and location identification, such as RCNNs, require accurate CNNs as their base. Thus, improved accuracy, even in extremely simplified cases, is important.

We can see that the *WavPool-NS* implementation has the highest accuracy for this dataset. The wavelet pooling methods work well with this application, which we expected. One interesting thing to note is that all three implementations of the wavelet pooling are within 2% of each other in accuracy. Even though max pooling is the fastest implementation we consider here, the increase in accuracy is worth the extra time cost in using the *WavPool* implementations. Further, the slowest implementation, the *WavPool-NS*, is only 21.91% slower than the max pooling implementation. Since this cost is up front in the training phase, we consider it to be a reasonable trade-off for the improved accuracy. We also note that we do expect *WavPool-NS* to be slightly slower than *WavPool-NH* since the Shannon wavelet is computationally more expensive. We believe that the large time difference is due to the full implementation not being fully optimized.

More complex classification problems with this dataset, such as using a Regions with Convolutional Neural Network(RCNN) that gives both the location and type of tumor, could be improved by using the *WavPool* implementations. Continuous wavelets, even a continuous

Shannon wavelet, may also be a good implementation here, although it will be even more computationally expensive. For the more simple task of classifying patient scans as either having tumors present or not, the *WavPool* implementations outperform the max pooling. The increase in accuracy is not overly large but, because of the importance of the application, we think this is enough to justify the minor increased cost in time.

4 Conclusion

We have improved the accuracy of the *WavPool* method in this work. We have also worked on increasing the speed of the methodology using the NAdam gradient technique. Also, since NAdam can improve on the results of gradient descent without intensive tuning, it is beneficial to combine this method with *WavPool* to help improve the overall process of the classification.

We considered both the Haar wavelet and the Shannon wavelet to compare how changing the wavelet method impacts the results. We have found that for the lung tumor classification, we were able to achieve a small improvement by using the Shannon wavelet rather than the Haar wavelet. The *WavPool-NS* method performed the best with the LIDC-IDRI dataset with a 95.52% accuracy. However, there was also a trade-off in time by changing the wavelet method. Further research needs to be done to determine the best wavelet for the lung tumor detection application. It also needs to be determined whether the wavelet pooling gives a significant enough improvement to justify the increased computational cost and if further optimization methods can improve to overall time. *WavPool* methods show promise in addressing the problem of lung tumor classification and the technique may help improve the results for other applications. It is important to note that improving pooling methods alone is not enough to make major improvements to most applications. *WavPool* and similar methods need to be used in conjunction with other techniques to create better CNN pipelines.

Lung cancer deaths are a serious problem worldwide. Early detection methods with good accuracy rates are needed to help increase the patient survival rate. Works such as ours contributes to the advancement of techniques which leads towards a truly efficient and useful early detection system.

Future work needs to focus on more complex classification that include both tumor type and location identification. This can be done by incorporating this work into Regions with Convolutional Neural Networks (RCNNs). This method can also be improved by testing further, more complex wavelets for the wavelet pooling.

References

- [1] Zaheer Abbas and Princess Raina. Shannon wavelet analysis with applications: A survey. *IOSR Journal of Mathematics (IOSR-JM)*, 12:08–20, 09 2016.
- [2] Samuel G Armato, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.
- [3] S Armato III, Geoffrey McLennan, Luc Bidaut, M McNitt-Gray, C Meyer, A Reeves, L Clarke, et al. Data from lidc-idri. the cancer imaging archive, 2015.
- [4] Shadnaz Asgari, Alireza Mehrnia, and Maryam Moussavi. Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. *Computers in biology and medicine*, 60:132–142, 2015.

- [5] Chiraz Ben Chaabane, Dorra Mellouli, Tarek M Hamdani, Adel M Alimi, and Ajith Abraham. Wavelet convolutional neural networks for handwritten digits recognition. In *International Conference on Health Information Science*, pages 305–310. Springer, 2017.
- [6] Jennifer M Croswell, David F Ransohoff, and Barnett S Kramer. Principles of cancer screening: lessons from history and study design issues. In *Seminars in oncology*, volume 37, pages 202–215. Elsevier, June 2010.
- [7] Timothy Dozat. Incorporating Nesterov Momentum into Adam. *ICLR 2016*, (1):2013–2016, 2016.
- [8] Yiping Duan, Fang Liu, Licheng Jiao, Peng Zhao, and Lu Zhang. Sar image segmentation based on convolutional-wavelet neural network and markov random field. *Pattern Recognition*, 64:255–267, 2017.
- [9] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv:1707.07394*, 2017.
- [10] Jeffrey P Kanne. Screening for lung cancer: what have we learned? *American Journal of Roentgenology*, 202(3):530–535, 2014.
- [11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [12] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.
- [13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: <http://www.cs.toronto.edu/kriz/cifar.html>*, 2014.
- [14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [15] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.
- [16] Mathworks. *Matlab 2016b*. Mathworks Inc., Natick, MA, 2016.
- [17] Kimberly D Miller, Rebecca L Siegel, Chun Chieh Lin, Angela B Mariotto, Joan L Kramer, Julia H Rowland, Kevin D Stein, Rick Alteri, and Ahmedin Jemal. Cancer treatment and survivorship statistics, 2016. *CA: a cancer journal for clinicians*, 66(4):271–289, 2016.
- [18] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.
- [19] J Anthony Parker, Robert V Kenyon, and Donald E Troxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on medical imaging*, 2(1):31–39, 1983.
- [20] Allison M Rossetto and Wenjin Zhou. Deep learning for categorization of lung cancer ct images. In *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2017 IEEE/ACM International Conference on*, pages 272–273. IEEE, 2017.
- [21] Allison M Rossetto and Wenjin Zhou. Ensemble convolution neural network with a simple voting method for lung tumor detection. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 729–734. ACM, 2017.
- [22] Mark J Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.
- [23] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.
- [24] Radomir S Stanković and Bogdan J Falkowski. The haar wavelet transform: its status and achievements. *Computers & Electrical Engineering*, 29(1):25–44, 2003.
- [25] Lindsey A Torre, Rebecca L Siegel, and Ahmedin Jemal. Lung cancer statistics. In *Lung cancer and personalized medicine*, pages 1–19. Springer, 2016.
- [26] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.
- [27] Travis Williams and Robert Li. Wavelet Pooling for Convolutional Neural Networks. *ICLR 2018*, pages 1–11, 2018.