AGRICULTURAL HARVESTER SOUND CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS AND SPECTROGRAMS

Nioosha E. Khorasani[1], Gabriel Thomas[1,*], Simone Balocco[2], Danny Mann[3]

[1] Electrical and Computer Engineering, University of Manitoba Faculty of Engineering, Winnipeg, Manitoba, Canada.
[2] Department of Applied Mathematics and Analysis, University of Barcelona, Barcelona, Catalunya, Spain.
[3] Biosystems Engineering, University of Manitoba, Winnipeg, Manitoba, Canada.
[*] Correspondence: Gabriel.Thomas@umanitoba.ca

HIGHLIGHTS

- Automatic classification of harvester sounds.
- Final classification obtained using three convolutional neural networks.
- The results of the networks were combined via stacking and voting to achieve 100% accuracy.

ABSTRACT. *The use of deep learning in agricultural tasks has recently become popular. Deep learning networks have been used for analyzing images of crops, identifying paddy areas, distinguishing sick plants from healthy ones, to name a few applications. Besides visual systems, sound analysis of agricultural machinery is a time-sensitive task that can also be incorporated in decision making and can be done with the help of deep learning models. We propose a method to generate spectrogram images from the sound of a harvester and classify them into three working modes in real-time. We used three convolutional neural networks and use the outputs of these networks as inputs to a stacking ensemble method to improve the accuracy of the system. To achieve 100% classification accuracy, a final decision is made by voting based on several consecutive classifications made by the stacking step. We were able to perform classifications in less than 1 s which was the standard to be considered as a safe time for the harvester.*

Keywords. *Convolutional neural networks, Deep learning, Spectrograms, Stacking, Voting.*

To address the challenges in the complex multivariate agricultural ecosystem new technologies and analysis tools are being explored. Deep learning is one of the modern successful techniques that can be used for image processing and data analysis (Kamilari and Prenafeta-Boldú, 2018). Deep learning is a subset of machine learning that mainly uses artificial neural networks modeled after the brain. Albeit these methods gained a lot of recognition in image processing, they are not specifically an image processing tool but can be used in other applications as the one presented here. One of the deep learning well recognized features is that they do not require hand-crafted features and unlike machine learning methods is not highly dependent on data representation. These qualities are some of the factors that make deep learning a proper tool that has been used in various fields, including agriculture (Jiao and Zhao, 2019). For example, Chandy (2019) used a camera interfaced drone with embedded Nvidia Tegra SoC with ARM Cortex-based CPU and GPU to take pictures from a coconut farm and process the images with the help of deep learning. This system helped them to identify pests and infected coconut trees. Another utility of deep learning in processing the images taken from crops and agricultural fields is identifying paddy areas that need accurate land monitoring for food security control and support action purposes. In Nguyen et al. (2020), satellite images and an advanced spatial-temporal-spectral deep neural network were used to distinguish crop areas from non-crop areas.

In this article, we classify the sound of the S680 John Deere combine engine into three different operational sound classes: empty, engaged, and full. As in (Thomas et al., 2021), we also use the Short-Time Fourier Transform (STFT) method to convert the format of our data from audio to an image via a Spectrogram (SGs) (Simundsson et al., 2019). In this current work, we combined three Convolutional Neural Networks (CNNs) whose classification outputs were set as inputs to a meta learner (Stacking); then, after voting, we predict the class of the original audio segment. The outcomes show that we have obtained better results than the ones presented in Thomas et al. (2021).

Many operations in agriculture are time-sensitive and breakdowns can be very costly in terms of both time and service. It may be difficult to monitor operations with many moving parts, such as a harvester, without the benefit of

Submitted for review on 12 May 2021 as manuscript number ITSC 14668; approved for publication as a Research Article by Associate Editor Dr. Joshua Peschel and Community Editor Dr. Naiqian Zhang of the Information Technology, Sensors, & Control Systems Community of ASABE on 22 February 2022.

sensory information. A few seconds of delay in processing information may be the difference between preventing a failure and a serious harvest delay. Our goal then was to find the optimal network that yields the most accurate results by classification of the state in under 1 s.

Data Collection

The recordings correspond to the sound of a combine (model number S680, John Deere, Moline, Ill.) which was taken with a GoPro Hero Session camera during the canola harvest in East Selkirk, Manitoba, in 2017. The camera was placed at the rear of the combine near the straw chopper. The sampling rate was 48 kHz, and the recording was compressed and converted to waveform audio format. The audio files were recorded during three states of the combine. The three operating modes of the combine are *i*) empty: the combine is running, but mechanized threshing is not engaged (recording time 9 s), *ii*) engaged: the combine engine is running, mechanized threshing is engaged, but no actual threshing being done (recording time 5 s), and *iii*) full: the combine's engine is running, and mechanized threshing is engaged and utilized at approximately 80% capacity (recording time 5 s).

Image Generation using the STFT

The STFT is a powerful general-purpose tool used in audio signal processing. It is a sequence of 1D Fourier transforms of data selected by time-domain windowing of signals. The STFT provides time-localized frequency information for situations in which frequency components of a signal vary over time (Kehtarnavaz, 2008). SGs are two-dimensional images displaying the sequence of spectra with time and frequency as its axes. The plot's brightness and colors demonstrate the strength of the frequency in each time window (Wyse, 2017). The SG can be defined as an intensity plot of the STFT magnitude. SGs have been applied in many various audio processing purposes, such as automatic detection of speech deficits in cochlear implant users and phoneme class recognition to extract phone-attribute (Arias-Vergara et al., 2020).

The mathematical definition of the discrete STFT is:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, w(n - mR)\, e^{-j\omega n} \quad (1)$$

where $x(n)$ is the input signal at time index n, ω is the frequency, $w(n)$ is a time-domain window of length WL, and R is the number of samples the window is shifted to acquire the next block of samples to be Fourier transformed (Allen and Rabiner, 1977). With different window lengths, the number of images extracted from one specific audio file changes. A longer observational window WL yields a fewer number of images. To illustrate, if the audio file has 100,000 samples and we choose WL=1000 we can extract 100 blocks of data from that file. Then those 100 blocks will be converted to pictures so there will be 100 pictures. When generating these images and storing them in memory, we apply two different image compression values that result in images of sizes 107×134 and 67×54. As it was mentioned before, time is a vital parameter; hence, working with low-resolution images reduces processing time.

Here, for generating SGs from audio signals, we use two different window sizes, WL= 1000 and WL= 2000 samples, and two different compression percentages, 10%, and 20%. In this way, we can evaluate the effects of both the number of samples used in the Fourier transform and image resolution, on the classification result. Figure 1 shows three randomly selected pictures from each class extracted from the training dataset.

Table 1 shows the number of pictures in each dataset with different WLs and compression rates. We used 10% of data in each dataset as a testing dataset and 90% of data as a training dataset.

CNN Model Architecture

Convolutional neural networks have become prevalent in several computer vision and image processing tasks. As our approach requires a fast implementation, the CNN model implemented in this work was composed of few layers as described in table 2, whose goal is to learn various image features automatically (Yamashita et al., 2018). Note how the differences of the parameters chosen in the layers differ so that this allows for the three models to yield decisions that when combined would result in a better accuracy then the ones obtained by each CNN individually.

Applying the STFT to sound signals in order to generate images and using a CNN for classification of those SGs have been previously done in works such as Demir et al. (2019). in order to classify lung diseases. Another study that used the combination of STFT and CNN was presented in Huzaifah (2017), which compared the diverse time-frequency representations for classifying environmental sounds. In Huang et al. (2019), electrocardiograms of five different heartbeats were transformed into SGs, and the images of five arrhythmia types were used as inputs to a CNN to be identified and classified.

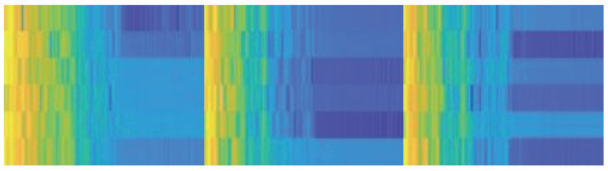

Figure 1. Spectrograms of a 1000 samples data block of class: (a) empty, (b) engaged, and (c) full. Compression is 10%.

Table 1. number of pictures in each dataset

WL	Compression Rate (%)	No. of Pics in the Dataset of "Empty" Class	No. of Pics in the Dataset of "Engaged" Class	No. of Pics in the Dataset of "Full" Class
1000	10	1111	789	926
2000	10	555	394	463
1000	20	1111	789	926
2000	20	555	394	463

Ensemble Methods

Model Averaging

Model averaging is an ensemble technique where multiple sub-models contribute equally to a combined prediction. Weighting the participation of each sub-model in the final prediction can make the model averaging better. This allows well-performing models to participate more, and less well-performing models to participate less in the process of final prediction. This approach is called the weighted average ensemble. The performance can be increased by training a completely new linear regression model to learn how to combine the contribution of sub-models predictions to have the best result. This method is called Stacked Generalization (or Stacking) and can result in better predictive performance than every single sub-model. In Stacking, an algorithm uses outputs of sub-models (predictions) as inputs and tries to learn how to combine these predictions in the best way to improve the result and achieve the best possible prediction (Brownlee, 2020). Figure 2 shows a block diagram of this stacking concept.

As mentioned before, we developed three deep learning models and trained them with our data. Then we used the testing dataset to obtain predictions of each of the three models. A meta learner model was trained with sub-models prediction and was learned to combine them in the best way possible. Ideally, the meta learner model would perform better than any single model. We implemented a logistic regression algorithm as a meta learner.

This technique achieved an accuracy of about 0.05% to 1.2% more than the highest accuracy among all CNN networks. Besides, if in one of the implementations, the accuracy of one of the CNNs decreases significantly, the meta learner takes care of it by reducing the effect of that particular CNN on the final result.

A block diagram depicting the sequence of the steps taken before voting is shown in figure 3.

To ensure that this improvement has not happened by chance and the difference between the accuracy of the models and the accuracy after stacking is significant, we used the t-test, a statistical hypothesis test that checks the means from two samples to see if they are significantly different from each other. In our case, the null hypothesis is the average of

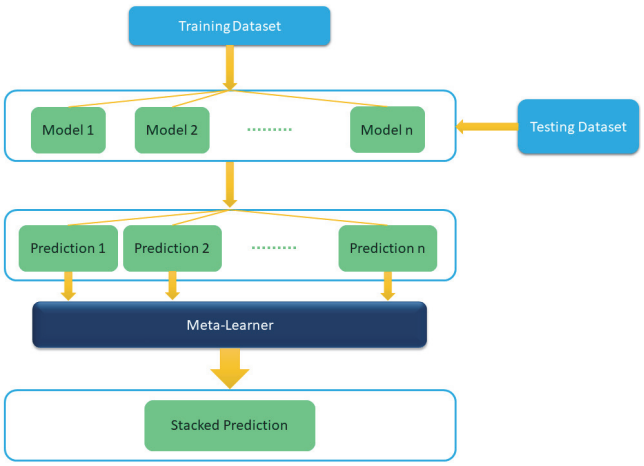

Figure 2. Stacking block diagram.

model accuracies being equivalent to the average of accuracy after stacking. We performed the test, pairwise, for the accuracy of each model and the accuracy after stacking. The significance level was set to $\alpha = 0.05$, and the t-statistic was calculated as:

$$t = \frac{M_x - M_y}{\sqrt{\frac{S_x^2}{n_x} + \frac{S_y^2}{n_y}}} \quad (2)$$

where M is the mean, n is the number of accuracies per group, and

$$S = \frac{\sum (x - M)^2}{n - 1} \quad (3)$$

where x is each accuracy.

Afterward, we compared the critical t-values with the calculated t-statistic. In the t-test, if the computed t-statistic is greater than the critical t-value, the test concludes that there is a statistically significant difference between the two populations. Therefore, we reject the null hypothesis. In any other case, there is no statistically significant difference between the two populations, and the test fails to reject the null hypothesis.

Table 2. CNN architectures.

	CNN_1	CNN_2	CNN_3
Layer	Parameters and Description	Parameters and Description	Parameters and Description
Image input	Zero center normalization	Zero center normalization	Zero center normalization
Convolution	32 nodes, and filter size 3×3, no padding, Relu activation function, glorot kernel initializer	64 nodes, and filter size 3×3, no padding, Relu activation function	32 nodes, and filter size 3×3, no padding, Relu activation function, glorot kernel initializer
Max pooling	64 nodes, and filter size 3×3, Relu activation function	2×2 Maxpooling layer	64 nodes, and filter size 3×3, Relu activation function
Drop out	3×3 Maxpooling layer	0.1 Drop out layer	3×3 Maxpooling layer
Convolution	0.4 Drop out layer		0.4 Drop out layer
Drop out	32 nodes, and filter size 3×3, Relu activation function	128 nodes, Relu activation function	32 nodes, and filter size 3×3, Relu activation function
Flatten	0.25 Drop out layer	0.25 Drop out layer	0.25 Drop out layer
Dense		24 nodes, Relu activation	
Dropout	128 nodes, Relu activation function	0.25 Drop out layer	128 nodes, Relu activation function
Dense	0.5 Drop out layer	3 nodes Softmax activation function	0.5 Drop out layer
Output layer	3 nodes Softmax activation function	Cross entropy loss function, RMSprop optimizer, learning rate=0.001, batch size=64, epochs=50	3 nodes Softmax activation function

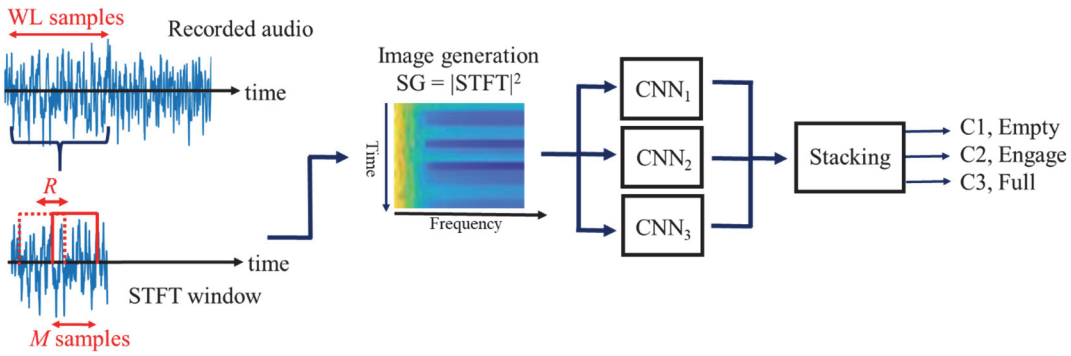

Figure 3. Steps taken by the proposed method to make a classification to be used in the final voting step.

VOTING

Based on the nature of the audio data recorded from the harvester machine engine, in a real situation, the operation modes of the machine do not shift suddenly from one state to the other. In other words, the classes do not change from empty to engaged, then empty again, then full, and so on. Instead, the samples and their correspondent classes are such that consecutive records correspond to the same class "empty," "empty," "empty," …, "engaged," "engaged," "engaged," …, "full," "full," "full," …, etc. This means that there is a sequence of one class and then a sequence of another one and so forth. After going through the mentioned steps and optimizing the network, we can benefit from this characteristic of our data and improve the prediction result.

In this study, the approach is to make the final decision based on not only just one particular vote but also on the two previous and two following ones. After that, we predict the class of these five consecutive decisions. The predicted class is computed as the majority of these five votes. Figure 4 shows an example.

RESULTS

We use the STFT on audio signals with two WL=1000 and WL=2000 to obtain SG images. These values lead to acquisition times of less than 5 ms leaving enough time for the computational operations needed to obtain classifications within one second. The accuracy of the three first CNN models was, on average, 95% to 96%. After performing model averaging, the accuracy improved between 1% and 2%. As it was mentioned, to evaluate the true effect of the averaging method on this improvement, we performed the t-test two by two on the accuracy we obtained from each model and the accuracy after averaging. We set the significance level α=0.05. The results are shown in table 3. In all of them, the two-tailed p-value is positive and smaller than the significance level (α=0.05). Hence, we reject the null hypothesis of equal averages and concluded that the improvement in the accuracy caused by the model averaging was not accidental.

We implemented the code on four different datasets with different window lengths and compression sizes. In two datasets, every image is generated from 1000 audio samples, and in the other two, images are generated from 2000 samples. Each of the two datasets generated from 1000 samples has different compression sizes, one with the image dimension of 67×54 and one with 134×107. This is the same for datasets produced with 2000 samples.

The CNNs need more time to predict classes according to datasets, which include bigger images with version two compression (134×107). Besides, the number of images in datasets with pictures generated from 1000 samples is doubled in comparison with the dataset composed of pictures with WL=2000. The running time for each single data sample block was significantly less than 1 second; hence we accomplished our goal, which was being able to run this algorithm

Table 3. t-tests results.

	CNN_1 Accuracy and Stacking Accuracy	CNN_2 Accuracy and Stacking Accuracy	CNN_3 Accuracy and Stacking Accuracy
p-value	0.00077	0.00080	$7.7 \times e^{-10}$
t statistics	3.4739	3.4641	6.8455

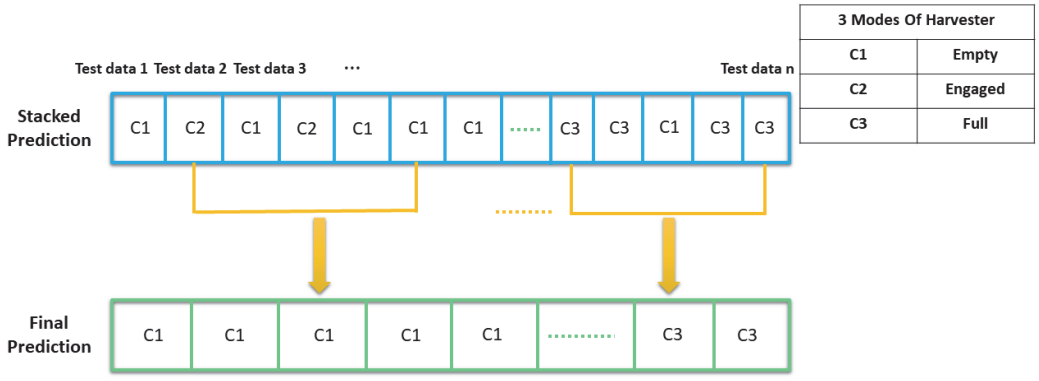

Figure 4. Making the decision based on a number of consecutive classifications.

Table 4. Results and running times.

WL	size	CNN_1 Accuracy (%)	CNN_2 Accuracy (%)	CNN_3 Accuracy (%)	Accuracy after stacking (%)	Accuracy after voting (%)	Average execution time (ms)
1000	67×54	97.92	97.71	95.11	98.37	100	3.7
1000	134×107	96.68	96.66	94.37	97.98	100	13.6
2000	67×54	97.57	95.03	95.09	98.18	100	4.6
2000	134×107	96.24	97.84	95.58	98.28	100	10

on agricultural machinery in real-time. The work was done with an Asus laptop with a 64 bit Intel i5 CPU @ 1.6 GHz and 12 GB of RAM. We used the Matlab platform version 2019b for producing images and python version 3.5.4, and Keras library for implementing the deep learning part.

After voting, we achieved the final testing accuracy of 100% when using five consecutive blocks. Table 4 summarizes the results.

CONCLUSION

In this work, we developed three deep learning networks trained with spectrograms generated from the recorded audio. The predictions obtained from each of these three networks were used as an input to a meta learner, which is a linear regression algorithm. Weighting the prediction of each network reached a prediction better than every single network prediction, and we proved this hypothesis with the help of the t-test. Considering the nature of our data, we made the final decision based on five consecutive audio segments. After all these steps, we were able to achieve 100% accuracy in real-time using our dataset. In future work, we will explore the effects of audio noise coming from unexpected sources such as nearby traffic relying on the CNN flexibility for adjusting to those changes having enough examples during the training of these networks.

ACKNOWLEDGMENTS

This work was partially supported by the MICINN Grant RTI2018-095232-B-C21, 2017 SGR 1742.

REFERENCES

Allen, J. B., & Rabiner, L. R. (1977). A unified approach to short-time fourier analysis and synthesis. *Proc. IEEE, 65*(11), 1558-1564. https://doi.org/10.1109/PROC.1977.10770

Arias-Vergara, T., Klumpp, P., Vasquez-Correa, J. C., Noth, E., Orozco-Arroyave, J. R., & Schuster, M. (2020). Multi-channel spectrograms for speech processing applications using deep learning methods. *Pattern Analysis Appl., 24*(2), 423-431. https://doi.org/10.1007/s10044-020-00921-5

Brownlee, J. (2020). Stacking ensemble for deep learning neural networks in python. Retrieved from https://machinelearningmastery.com/stacking-ensemble-for-deep-learning-neural-networks

Chandy, A. (2019). Pest infestation identification in coconut trees using deep learning. *J. Artificial Intelligence Capsule Networks, 1*(1), 10-18. https://doi.org/10.36548/jaicn.2019.1.002

Demir, F., Sengur, A., & Bajaj, V. (2019). Convolutional neural networks based efficient approach for classification of lung diseases. *Health Information Sci. Syst., 8*(1), 4. https://doi.org/10.1007/s13755-019-0091-3

Huang, J., Chen, B., Yao, B., & He, W. (2019). ECG Arrhythmia classification using STFT-based spectrogram and convolutional neural network. *IEEE Access, 7*, 92871-92880. https://doi.org/10.1109/ACCESS.2019.2928017

Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arXiv preprint arXiv:1706.07156.

Jiao, L., & Zhao, J. (2019). A survey on the new generation of deep learning in image processing. *IEEE Access, 7*, 172231-172263. https://doi.org/10.1109/ACCESS.2019.2956508

Kamilaris, A., & Prenafeta-Boldu, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric., 147*, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

Kehtarnavaz, N. (2008). *Digital signal processing system design.* Amsterdam: Academic Press.

Nguyen, T. T., Hoang, T. D., Pham, M. T., Vu, T. T., Nguyen, T. H., Huynh, Q.-T., & Jo, J. (2020). Monitoring agriculture areas with satellite images and deep learning. *Appl. Soft Comput., 95*, 106565. https://doi.org/10.1016/j.asoc.2020.106565

Simundsson, A., Thomas, G., & Mann, D. (2019). A neural network to classify auditory signals for use in autonomous harvester control systems.

Thomas, G., Balocco, S., Mann, D., Simundsson, A., & Khorasani, N. (2021). Intelligent agricultural machinery using deep learning. *IEEE Instru. Meas. Mag., 24*(2), 93-100. https://doi.org/10.1109/MIM.2021.9400957

Wyse, L. (2017). Audio spectrogram representations for processing with convolutional neural networks.

Yamashita, R., Nishio, M., Do, R. K., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Imaging, 9*(4), 611-629. https://doi.org/10.1007/s13244-018-0639-9