Democratizing the access to artificial intelligence solutions for underrepresented and non-expert communities

R. E. P. Ferreira[1], T. Bresolin[2], J. C. F. Silva[1] and J. R. R. Dórea[1,3,*]
[1]*Department of Animal and Dairy Sciences, University of Wisconsin-Madison, WI, 53706, USA*
[2]*Department of Animal Science, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA*
[3]*Department of Biological Systems Engineering, University of Wisconsin-Madison, WI, 53706, USA*
[*]Corresponding author: João R. R. Dórea, joao.dorea@wisc.edu

Abstract

Artificial intelligence (AI) has become one of the most important technologies in different fields of science during the last decades. However, widespread access to AI algorithms and solutions is yet to become a reality for non-expert users. Open-source repositories host several deep learning algorithms used in AI, but they still require users to have programming experience to benefit from them. Tools designed using non-coding policies would positively impact communities of non-experts in many scientific domains, create new means for teaching in K-12 and higher education, and benefit underrepresented groups in minority serving institutions. In this project we developed an open-source software to democratize the access to deep learning algorithms. Such software will be used to perform image classification by training new and customized deep leaning algorithms on datasets provided by the user. An easy-to-use user interface was developed to allow users with no previous programming experience to train their own image classification neural networks and use such networks to perform predictions on newly chosen images. The software suite was implemented in Python using TensorFlow and Keras deep learning libraries. Developing easy-to-use open-source tools for Image Classification will help a large number of individuals who aim to learn and experiment with AI technology. Our goal is to create more accessible user interfaces so that coding ability is not a barrier.

Keywords: artificial intelligence, democratization, image analysis, machine learning

Introduction

Artificial intelligence (AI) has become one of the most important technologies in different fields of science during the last decades. While easy-to-use platforms for certain computer science fields such as data visualization have been progressively democratized in the last few years (e.g., Tableau, Plotly, PowerBI) for non-expert communities, widespread access to AI algorithms and solutions is yet to become a reality for such group of users. Open-source repositories host several deep learning algorithms used in AI, but they still require users to have a background in computer science, mathematics, and statistics to implement and benefit from them. Although such background requirements are critical to ensure trustworthy solutions, tools designed using non-coding policies can positively impact communities of non-experts in other scientific domains, create new means for teaching in K-12 and higher education, and benefit underrepresented groups in minority serving institutions. The tools available to date are either pay-to-use (e.g., Custom Vision from Microsoft, Amazon Recognition, IBM Watson Visual Recognition, and Clarifai) or not designed for scalability and research (Carney et al., 2020), and in some cases require understanding complex concepts on deep learning techniques, cloud platform, a specific programming language, among other factors. Given the potential of AI and more specifically state-of-the-art deep learning models, such technologies should be accessible for global communities, reducing entry barriers for individuals to start experimenting with deep learning. Currently, the biggest challenge for non-expert users is to have access to deep learning techniques through a more user-friendly interface to perform basic AI tasks, such as identifying an object in an image, without requiring extensive expertise in the corresponding areas (other than their own domain area expertise).

In this project, we developed an open-source software to democratize the access to deep learning techniques. Such software can be used to perform image classification by training new and customized deep neural networks on datasets provided by the user. The specific goals of this project were to (1) develop an open-source software to facilitate training deep neural networks for image classification tasks, focusing on users with no previous experience with the topic; (2) implement a framework for using previously trained models to perform image classification on datasets provided by the user and enabling model exporting for further use; (3) make the software suitable for research prototyping and preliminary exploration by focusing on scalability and accessibility; and (4) develop an online platform to host the application, and organize workshops and trainings to disseminate information about the tool and introduce it to potential new users. The main contributions of this work are the development of such software platform, education of potential users through online training and workshops, and a stronger focus on facilitating not only educational use of deep neural networks but also research prototyping and preliminary analyses.

Materials and methods

Software overview

Developing easy-to-use open-source tools for Image Classification can help many individuals who aim to learn and experiment with AI technology. Our goal was to create more accessible user interfaces so that coding ability would not be a barrier anymore. A snapshot of the developed software is shown in Figure 1. In the proposed system, users can train their own deep learning algorithms for image classification by performing simple drag-and-drop operations, visualize the performance of the trained model on a validation set, use previously trained models to perform inference on new images, and export their model to a standard format so it can be used in external applications.

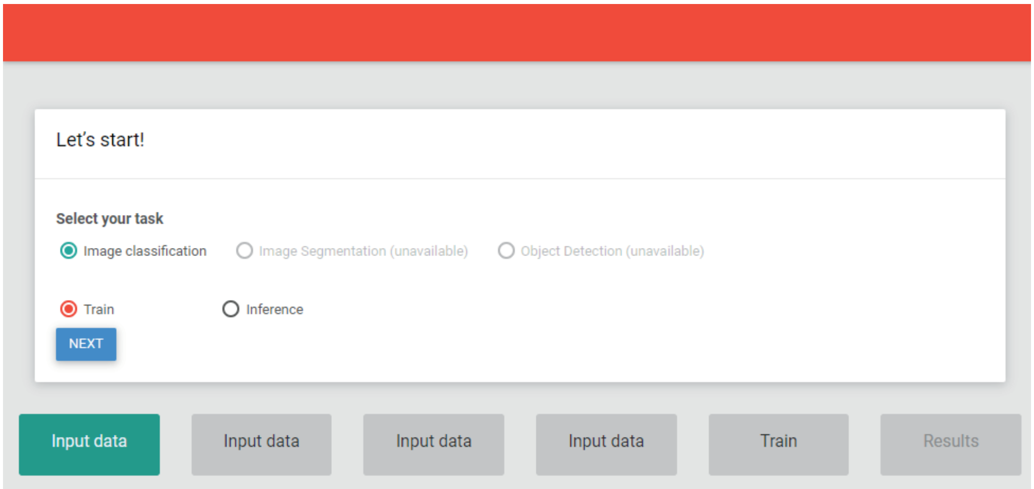

Figure 1: Snapshot of the developed software. Users can navigate through an intuitive interface to train customized neural networks using their own uploaded image datasets.

The tool is intuitive and self-guide the user through the full process of training a deep neural network. In the first (landing) page the user chooses whether they want to train a new model or use a previously trained model to run inference on new images. After choosing the training option, they are redirected to the next page, where they choose the number and name of the classes that the model will be able to classify, and upload both training and validation images for each of the classes. Upon user confirmation, the network starts the training process, showing log outputs on the screen. After the network is trained, results are shown on the screen, with metrics such as Recall, Precision, and Accuracy for each individual class, and the users can evaluate the performance of the trained model. On the same screen there are options for downloading the trained model in a standard HDF5 format and running inference using that model. The user is then directed to upload images that they want the trained model to classify. Finally, a page shows a list of each uploaded image, the predicted class, and the confidence level output from the trained model. This allows users to visually access the predictions of their newly trained model on multiple chosen examples. A flow of the steps the user goes through when using the tool is shown in Figure 2.

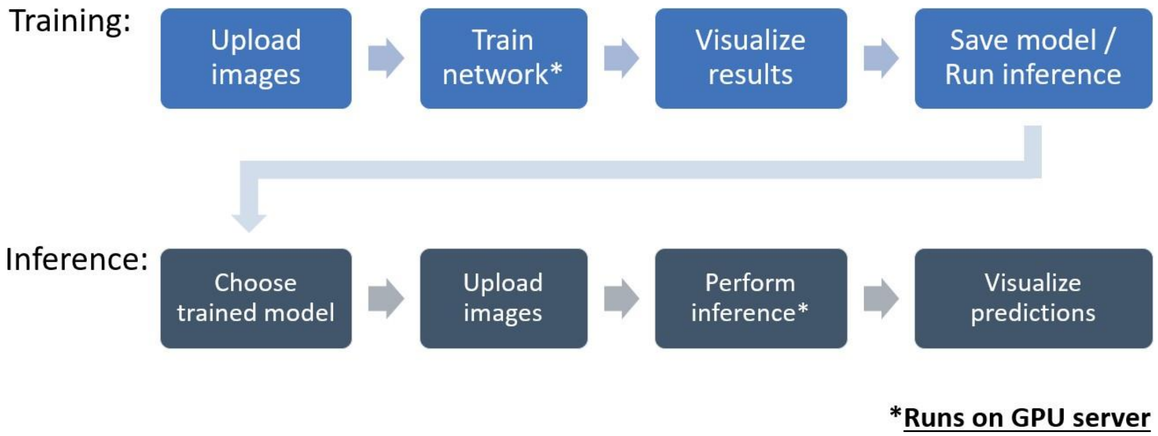

Figure 2: Steps that the user performs when using the tool to train a new network or run inference using a model previously trained. All steps are guided through an intuitive user interface and require no programming knowledge. The neural network is trained and performs inference using servers equipped with Graphic Processing Units (GPUs), which are effective at training neural networks.

Implementation details

The neural network training procedure was implemented in Python using Keras (Chollet, 2015) and TensorFlow (Abadi et al., 2016). The trainable network follows the MobileNetV2 architecture (Sandler et al., 2018), which is designed for use in mobile applications and is more efficient both in floating-point operations (FLOPS) and memory usage than other state-of-the-art architectures, and that is the reason why we chose this architecture for our application. The last fully connected (FC) layer of the network contains softmax activation with the number of outputs corresponding to the number of classes chosen by the user. All layers except for the last FC layer are initialized with weights from a MobileNetV2 network trained on ImageNet (Deng et al., 2009), an open image dataset containing more than 1 million examples of diverse objects and environments. Such a strategy of initializing weights with those optimized for large generic datasets is known as transfer learning (Weiss et al., 2016), and it accelerates the training process when compared to random weight initialization. This technique helps the new trained networks learn generic features, such as textures, edges, corners, and shapes, previously learned in a different task using a much larger dataset and is especially efficient when the final task consists of real-world images.

The training procedure is split into two consecutive stages: feature extraction and fine-tuning. In the feature extraction stage, the neural network is trained for 30 epochs keeping the weights of all but the last FC layer frozen. This allows features previously learned through transfer learning to be used and retained. In the fine-tuning stage, weights from earlier layers are unfrozen and the network is trained for 60 epochs with a smaller learning rate and early-stopping with patience parameter equal to 10, allowing it to further learn features that are more specific to the user's task. The network is trained using Adam (Kingma et al., 2014) optimization with a learning rate of 10^{-3} in the feature extraction stage and 10^{-5} in the fine-tuning stage.

Workshop results and discussion

As the proposed tool is not yet publicly available, we are measuring the preliminary results achieved so far through a virtual workshop held on December 9th, 2022 to demonstrate the developed tool. During this workshop, we provided a lecture on core concepts related to machine learning, computer vision, and image classification, gathered feedback, and discussed future directions for the development of the tool. The workshop started with a 45 minutes long lecture on image analysis, machine learning and computer vision to introduce users to how the developed tool works, its capabilities and why it is useful. It then proceeded to a 20-minutes long demonstration of the tool in real-time and a 15-minutes discussion on future directions. Both during and after the end of the workshop, we asked participants questions regarding current challenges and potential usage of the tool. The questions answered by the participants are shown in Figures 3, 4, and 5. In total, 181 people from 52 different countries registered for the workshop.

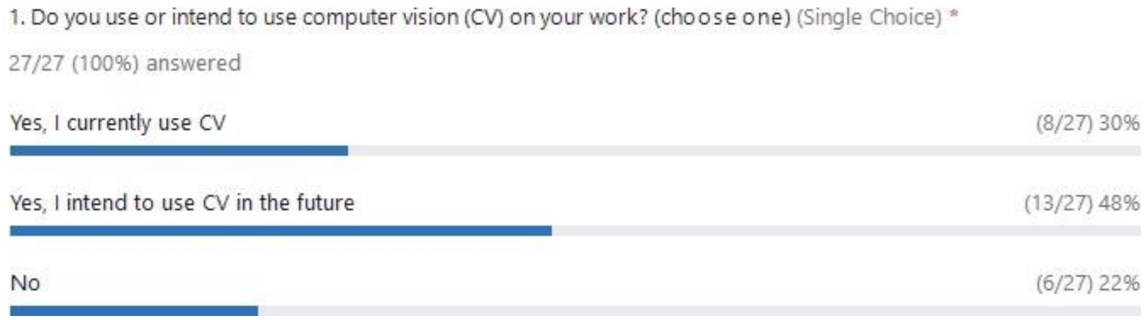

Figure 3: First question asked during the workshop. Participants were asked to answer this multiple-choice question at the start of the workshop, before the lecture.

It is worth noting that most participants that answered the poll (78%) currently use or intend to use computer vision (CV) in the future, which is not surprising as computer vision is currently a hot topic in artificial intelligence research and great potential has been demonstrated in multiple scientific and commercial applications. Regarding Graphics Processing Unit (GPU) availability, 32% of respondents do not currently have access to any GPU resources, which makes it very difficult to train deep neural networks that achieve state-of-the-art results for image analysis in their own computer that is not equipped with a GPU, due to the time it takes to train such models in a Central Processing Unit (CPU) compared to a GPU. While training language models locally using a CPU might be unfeasible, there are alternative GPU solutions available such as Google Colab and Domino Data Lab. However, it is important to note that these solutions may have limited free versions and can become costly when training multiple models for extended periods of time. A majority of the respondents (64%) answered that coding expertise is one of the main challenges faced when implementing deep learning for image analysis, aligning with our motivation of enabling the development of deep learning-based applications for agriculture and livestock through a tool that does not require any coding. Furthermore, lack of computational resources was answered by 46% of respondents as a big challenge, which is also a problem that is solved by our tool through enabling model training to be performed on a powerful computer server, instead of the user's own computer. Finally, 100% of respondents answered that they think the developed tool is useful for their work, with 89% stating that it would be useful for research, which is one of the main focuses of this project. However, we would like to acknowledge the fact that the use of the developed tool for research should be done with caution, as the lack of knowledge about the underlying mechanisms of the software might lead to inconclusive or wrong answers to the corresponding research questions. Instead, the proposed software is designed to simplify research prototyping and preliminary analyses, making it easier for users to obtain valuable preliminary results for their computer vision-related research questions. Therefore, it is recommended that final research and hypothesis testing be conducted with greater control over model training and validation, and a deep understanding of important image analysis and machine learning concepts. Regarding future directions, we aim to incorporate new functionalities into the tool, such as models for object detection and image segmentation, and further improve usability and user experience based on feedback and past experiences.

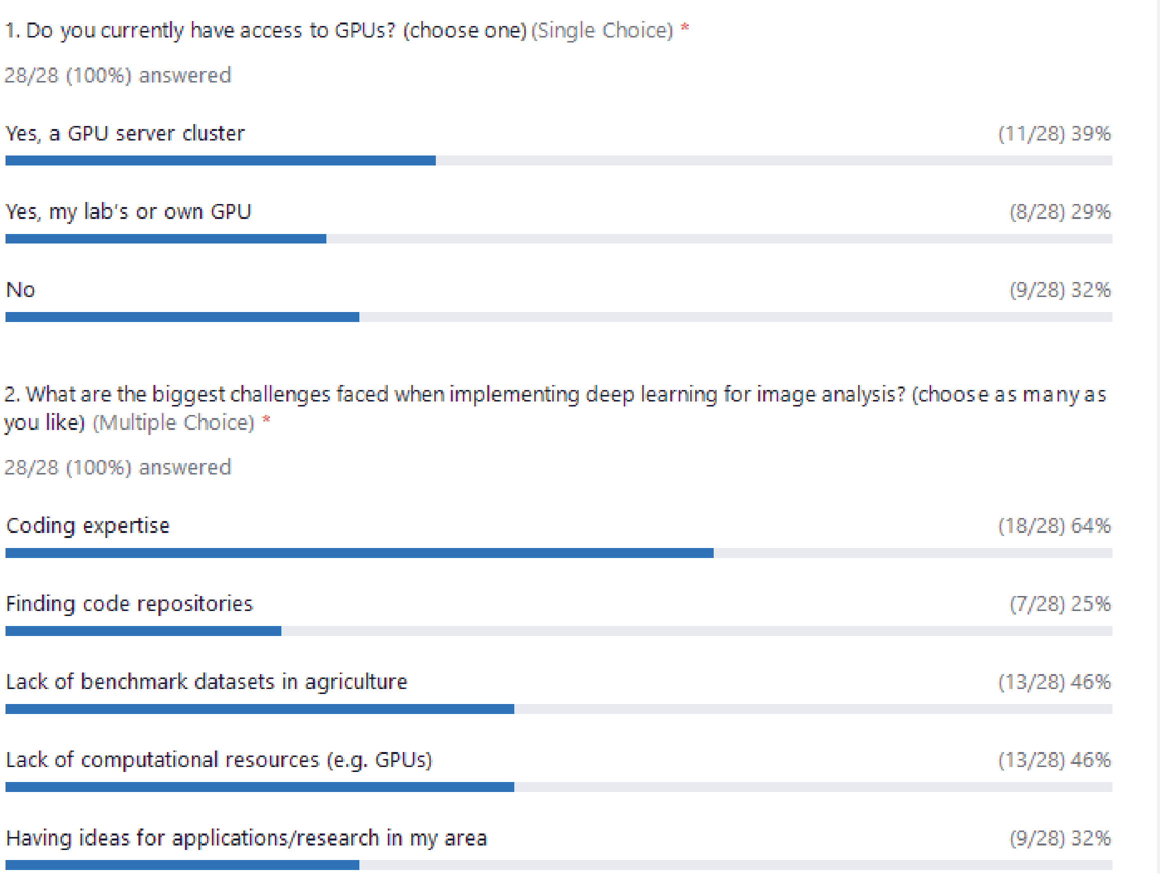

Figure 4: Second and third questions asked during the workshop. Participants were asked to answer these multiple-choice questions between the end of the lecture and the start of the tool demonstration.

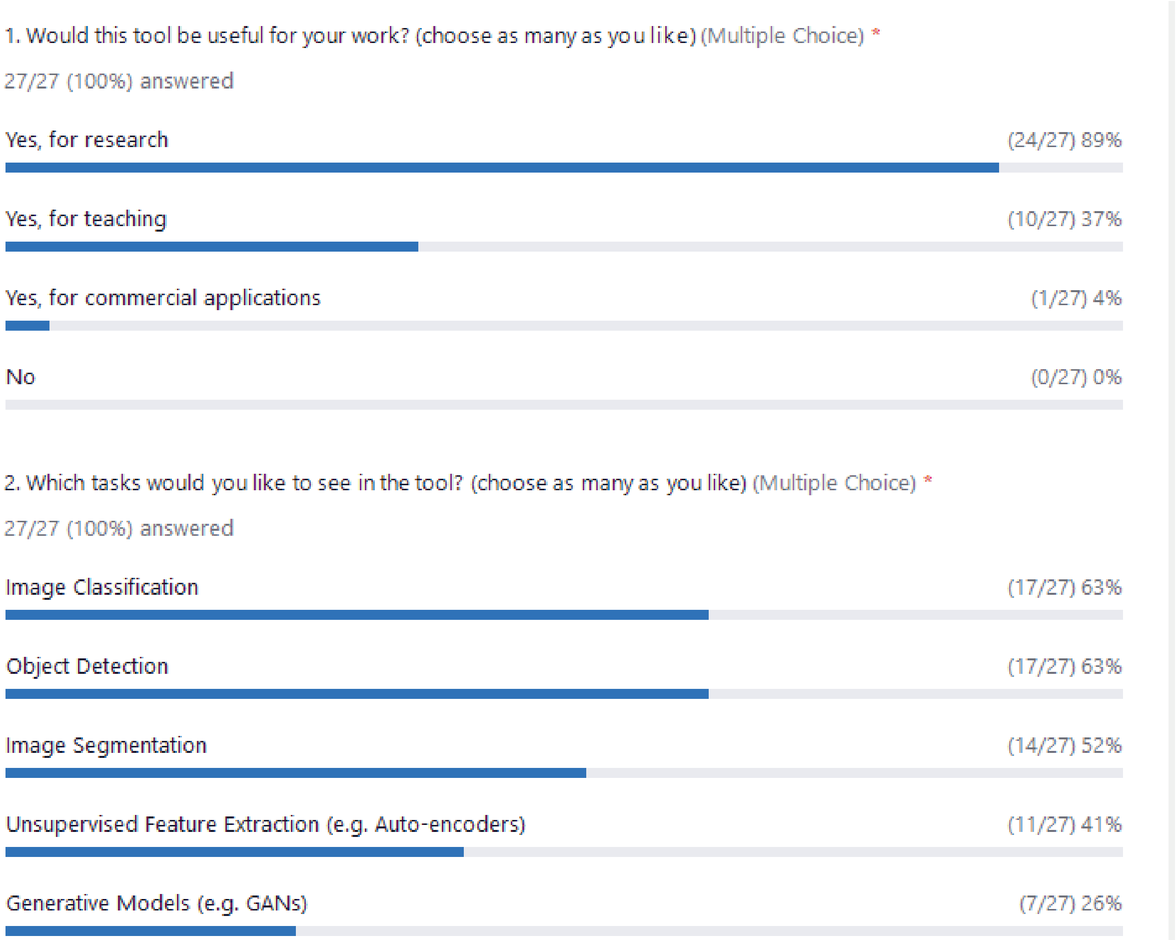

Figure 5: Last two questions asked during the workshop. Participants were asked to answer these multiple-choice questions after the end of the demonstration.

Conclusions

The open-source software proposed in this study has the potential to advance the democratization of AI technology for non-expert communities, by allowing users to experience such technology without a programming expertise. The developed tool and its future functionalities provide means for users to develop their own novel solutions and research prototypes for agriculture problems using powerful deep learning algorithms and dedicated infrastructure, without requiring the extensive knowledge and hardware that other currently available deep learning tools and resources do. By collecting constant feedback from users through open communication channels and workshops, we expect to continuously improve the tool and include more functionalities.

Acknowledgements

This project was funded by Agricultural Genome to Phenome Initiative (AG2PI) 2021 Seed Grant.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., and Isard, M. (2016) Tensorflow: A system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation*, Savannah, GA, USA, 265-283.

Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., and Chen, A. (2020). Teachable machine: Approachable web-based tool for exploring machine learning classification. In: *2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 1-8.

Chollet, F. (2015) keras, Github. *https://github.com/fchollet/keras.*

Deng, J., Dong, W., Socher, R., Li, L.J., Kai, L., and Li, F.-F. (2009) Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 248-255.

Kingma, D.P., and Ba, J. (2015). Adam: A method for stochastic optimization. In: *3rd International Conference for Learning Representations*, San Diego, CA, USA.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, 4510-4520.

Weiss, K., Khoshgoftaar, T.M., and Wang, D. (2016) A survey of transfer learning. *Journal of Big Data* 3(1), 9.