Undergraduate Research: Deep Learning Based Plant Classifiers and Their Real- Life Research Applications

Dr. Deng Cao, Central State University

Dr. Deng Cao received his Ph.D. in Computer Science from West Virginia University in 2013. He also earned a master degree in Statistics and a master degree in Physics, both from West Virginia University. Dr. Cao currently serves as an associate professor of Computer Science at Central State University. His research interests includes Artificial Intelligence, Machine Learning, Computer Vision and Biometrics. His research has been supported by US Department of Agriculture, National Science Foundation, and US Air Force Research Laboratory.

Dr. Cadance Lowell, Central State University

Dr. Cadance Lowell is a Professor of Agriculture at Central State University and Chair of the Department of Agricultural and Life Sciences. She received a B.S. in Botany from Duke University, a M.S. in Botany from the University of Florida, Gainesville, and a Ph.D. in Horticulture from the University of Florida, Gainesville. She did post-doctoral work with the USDA in Peoria, IL as a biochemist in soybean oligosaccharides before joining Central State University in 1989. Dr. Lowell maintains a research program in directed energy to kill weeds as an integrated pest management strategy. She mentors undergraduate students in funded research projects who have gone on to present at local, state and national conferences.

Dr. Craig M. Schluttenhofer, Central State University

Dr. Craig Schluttenhofer received his doctorate in Plant Physiology from the University of Kentucky in 2016. In 2011, he obtained a master's degree in Plant Pathology from Purdue University. He received bachelor's degrees in Horticulture Science as well as Plant Genetics and Breeding from Purdue University. In 2019, he joined Central State University as a research assistant professor of natural products. Dr. Schluttenhofer specializes in the genetics and biochemistry of Cannabis used for agricultural and medical purposes. He started working with hemp in 2014 while at the University of Kentucky, operating under the state hemp research pilot program. Current research foci include developing new hemp varieties, understanding the plants' biochemistry, improving production and processing practices, and developing new uses for the crop. His research has been supported by the United States Department of Agriculture.

Dr. Augustus Morris, Central State University

Dr. Augustus Morris is an Associate Professor of Manufacturing Engineering at Central State University, Wilberforce, OH. He is also the Program Director of the NSF funded grant, Implementing Pathways for STEM Retention and Graduation (IPSRG). His research interests include robotic applications in agriculture, haptic devices, high altitude balloon payload design, and cellulose-based composite materials.

Mr. Austin R Erdman, Central State University

Austin Erdman worked as a software engineer at Global Neighbor Incorporated developing software and weed detection algorithms for robotic equipment before working at Central State University. At CSU he is a computer vision specialist designing code for autonomous farming equipment. He is a US patent holder, as well as having worked on projects funded by the USDA and United States Air force.

Torry Johnson
Mr. Jeffrey D Taylor Jr, Central State University

Undergraduate Research: Deep Learning Based Plant Classifiers and Their Real- Life Research Applications

Abstract

Deep learning structures, such as Convolutional Neural Networks (CNNs), have been introduced to the undergraduate students in Central State University for the past three years. Funded by an 1890 Land Grant Evans-Allen research program and a USDA Capacity Building Grant, a number of students with minimum deep learning background were trained to develop customized CNNs. After training, the students were able to solve given plant classification problems and develop plant classification apps to showcase the performance of the customized CNNs. In particular, two students' research projects were discussed in details in this work. One project's goal was to identify Soybean (Glycine max) in its Cotyledon (VC) and 1st -5th trifoliate stages, the other project's goal was to identify Hemp (Cannabis sativa) in its three variations. The databases used in these projects were built from real field images, which contain 9 common weed species. The students' achievement, as well as discovered issues, are assessed and reported in this work. The students' projects will be further used to support our 1890 Land Grant and CBG research.

1. Introduction

In recent years, artificial intelligence (AI) has become one of the most intriguing research topics in the world. As one of the most successful AI structures, deep learning was applied to various fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, even lottery hypothesis [1] and drug design [2]. Today, deep learning is an attractive topic in higher education and many talented students in Science, Technology, Engineering, and Mathematics (STEM) programs are eager for hands-on experiences and applications that are related to deep learning.

Central State University (CSU) is a relatively small institution with a large diverse population of undergraduate students. Funded by an 1890 Land Grant Evans-Allen research program and a USDA Capacity Building Grant, we have been developing AIassisted plant classification and detection systems for precision weed control since 2017. A number of undergraduate research assistants (RAs) are needed and they are expected to carry out research work according to the project schedule, such as developing various plant recognition systems for various crops. We are particularly interested in Convolutional Neural Network (CNN), a deep learning structure that has been successfully applied to analyzing visual imagery. The advantage of this type of network is its exceptional robustness to changing weather condition and plant morphology, which is very common in crop fields. In general, CNN approaches require minimal preprocessing and make little assumption about the data distribution. Instead, a CNN will let the optimizer decide how to adjust the parameters of the network by feeding it with enormous amount of training data, such as hundreds of thousands of raw RGB images.

Unfortunately, this comes at a high cost. We therefore propose transfer learning, a commonly used method that utilizes the pre-trained networks and quickly transfers the pre-trained features (weights) to a new task, with a much smaller database. Transfer learning is well supported by several existing toolsets such as TensorFlow [3], Turi Create [4] and PyTorch [5]. These toolsets can simplify the development of customized deep learning models, making it feasible for undergraduate research that involves students with limited programming skills.

In 2017, deep learning was first integrated as a learning topic into CPS4420 Software Engineering, a computer science course that is designed for junior or senior students, or other students with sufficient programming backgrounds. We have two objectives: the first is to introduce deep learning to undergraduate students as a modern topic, and the second is to assess the students' performance and find suitable RA candidates for our research team.

2. Course Setup

CPS4420 Software Engineering is a major required course that offers in the fall semester every year. This course teaches students design and implementation issues for software systems, including software life cycle, requirements definition and specification, prototyping, verification, validation, testing, fault-tolerance, social and ethical issues of commercial software, and software management. The prerequisites for this course include CPS 1191 Computer Science I, CPS1192 Computer Science II, and CPS 2271 Data Structures, in which Computer Science I and II are C/C++ programming courses.

A fast-paced and practical term project is assigned to the students after the 2nd interim exam, which is approximately one month before the end of the semester. Each student is requested to design and implement an image classifier using CNN. The student is expected to use transfer learning to customize one or more pre-trained neural networks for new classes. The new network(s) should be able to distinguish at least two new classes that are not included in the pre-trained networks. Heavy coding is not required for this project, but each student needs to learn how to collect his (her) own database, install deep learning libraries and write proper code to re-train the existing network using transfer learning.

The students are allowed to use any deep learning structure for implementation, but the instructor uses TensorFlow as a demonstration, because it is open source, relatively easy to use, and can be implemented on different platforms.

The deliverable of the project is a real-time image classifier that can work on Linux, Windows, or Mac OS. Each student also needs to prepare a final project report that includes the motivation, the implementation details, the achieved goals and discovered issues, along with a 10-minutes PowerPoint presentation to explain the project in the classroom.

3. Learning Outcome Assessment

3.1. Assessment Methods

There were 18 students enrolled in CPS4420 between 2017 and 2019. The students' learning outcomes were evaluated based on their project reports and presentations. Also, anonymous questionnaires were collected from the students to assess the impact of the project. And a face-to-face interview was conducted for each student to discuss the discoveries and issues found in the project.

3.2. Assessment Results

The summary of the assessment is briefly listed as follows.

Proposal Phase:

In this phase, each student was asked to submit a proposal to identify the classes he or she plans to work on. The student also needed to justify why using an AI to classify the chosen classes was important in the real world. The proposal was then reviewed between the instructor and the student. Once the proposal was approved, the student then started to implement the proposed classifier.

As expected, most of the enrolled students were juniors or seniors. They had certain C/C++ programming skills, and some statistic backgrounds since most students also took MTH 2001 Probability and Statistics I, a major required course for Computer Science and Mathematics students, in their second school years. But almost no student had Python programming and/or machine learning background. Since TensorFlow interface is mostly written in Python, a brief introduction of Python was presented in the classroom, followed by the introduction to deep learning and CNN that took 2-3 course hours. The concept of deep learning was attractive to the students. Almost all students demonstrated strong interests and adequate understanding of the concept, and proposed original classification tasks that are meaningful in the real world.

Programming Phase:

In this phase, the student needed to build a working CNN image classifier. A student's self-learning skill was extremely important in this phase. Which database, which toolset, which dependencies, which platform, which means and alternatives he or she chose would change the process and result significantly. And the instructor left these choices to the students intentionally. The students were expected to initiate their own desire to make these decisions and conquer the implementation.

A major challenge was the student's limited programming skill. To address this issue while maintain the student's motivation and self-learning interest, the following reference links were provided for the student to explore.

- Tensorflow for poets (Linux user), by Google Codelab [6]
- Installing Tensorflow on Windows (Windows user), by Tensorflow team [7]
- Build a TensorFlow Image Classifier in 5 Min (Mac user), by Siraj Raval [8]

Approximately 40% of the students were able to use the references and other supplementary resources (online or in the library) and implemented their own CNNs independently. 60% of the students completed their tasks with the instructor's assistance. Some of them struggled when trying to install TensorFlow and making it work on their personal computers, especially when they were using Windows. In this case, virtualization tools such as Dockers [9] and Anaconda [10] could be useful to solve the problem. Some students had trouble when creating an Ubuntu / Windows dual boot system, because the bootable USB drive was not booting. Solving this problem could be tricky, because computers usually have a different OS versions and BIOS settings. Some students needed help when they were facing an error message that they could not understand, or the accuracy is not high enough. In most cases, the instructor would explain the issue individually and give guidance on possible solutions, so that the students could eventually complete their tasks.

Report Phase:

In this phase, each student submitted his or her final project and presented the hard work via PowerPoint presentation. Most students showed adequate skill to write a proper project report and prepared a brief presentation to explain the project results and findings. We also observed that a student could develop more intellectual confidence as he or she succeeded in the programming phase.

At the end of the semester, anonymous questionnaires were collected from each student to assess the impact of the term project at individual, course, and program levels (see Table 1).

Total evaluated students	**18**
Individual Level: Students who agreed that the project progressively deepened and broadened their research skills.	18 (100%)

Course Level: Students who agreed that adding deep learning related projects to the course made it more interesting.	18 (100%)
Program Level: Students who agreed that they were better motivated and engaged to stay in the Computer Science program when modern research topics such as deep learning was offered.	16 (89%)

Table 1: Project Impact Evaluation

4. Plant Classification in Real Field

A number of students who showed sufficient self-learning abilities and strong interests in deep learning in the CPS 4420 class were hired as RAs. Before taken CPS 4420, they had very little or zero machine learning background. But they developed more confidence in deep learning in the classroom. They chose to continue exploiting the world of AI as they worked with their advisors in different research teams. One of our ongoing research projects is to design an automated weeding robotic system that can treat the weed on a per-plant level in the field. Such a system will significantly reduce the potential impacts of herbicides on human health, as well as on terrestrial and aquatic eco-systems, and make organic farming more affordable. The robotic system requires real-time plant classification with high accuracy, which requires customized CNNs that can work on weeds and crops. In this work, two plant databases were collected and used for customizing the CNNs. Under the guidance of the advisor, most of the implementation tasks were completed by the RAs.

4.1. The Soybean Database

The first plant database used in this study is a soybean database built in 2018. The United States is the first in the world for soybean production. Today farmers in more than 30 states grow soybeans, making soybean the second largest crop in cash sales and the top agricultural export in the US. At USDA's February 2019 Agricultural Outlook Forum, the average of the analysts' estimates was 86.2 million acres of soybeans. Soybeans generated over $2 billion in annual cash receipts in the last decade [11].

The soybean database consists of 2,000 RGB images in two classes: Soybean (*Glycine max*) and Weed. The Soybean class contains 1000 images of soybeans in its cotyledon and 1st - 5th trifoliate stages, along with some soil and dirt images. The Weed class contains 1000 images of 9 common US weed species (Figure 1):

- Dandelion (*Taraxacum officinale F.H.Wigg*.)
- Hairy Crabgrass (*Digitaria 150 sanguinalis L.*)
- Eastern Cottonwood (*Populus deltoides W. Bartram ex Marshall*)
- American Pokeweed (*Phytolacca americana L.*)
- Broadleaf Plantain (*Plantago major L.)*
- Buckhorn Plantain (*Plantago lanceolata L.*)
- Carpetweed (*Mollugo verticillata L.*)
- Yellow Woodsorrel (*Oxalis stricta L.*)

Figure 1. Top line: soybean seedlings at various stages. Bottom two lines: common US weed species.

The raw images in the database were collected from a soybean field outside of our university in 2018. A customized WC800-DM4 robot platform [12] was used for image acquisition. Two Logitech HD webcam C270 are mounted face-down at the front bar of the robot to take top down plant images, plus one face-forward webcam for obstacle avoidance. A regular laptop PC was used to control the cameras and collect images automatically (Figure 2).

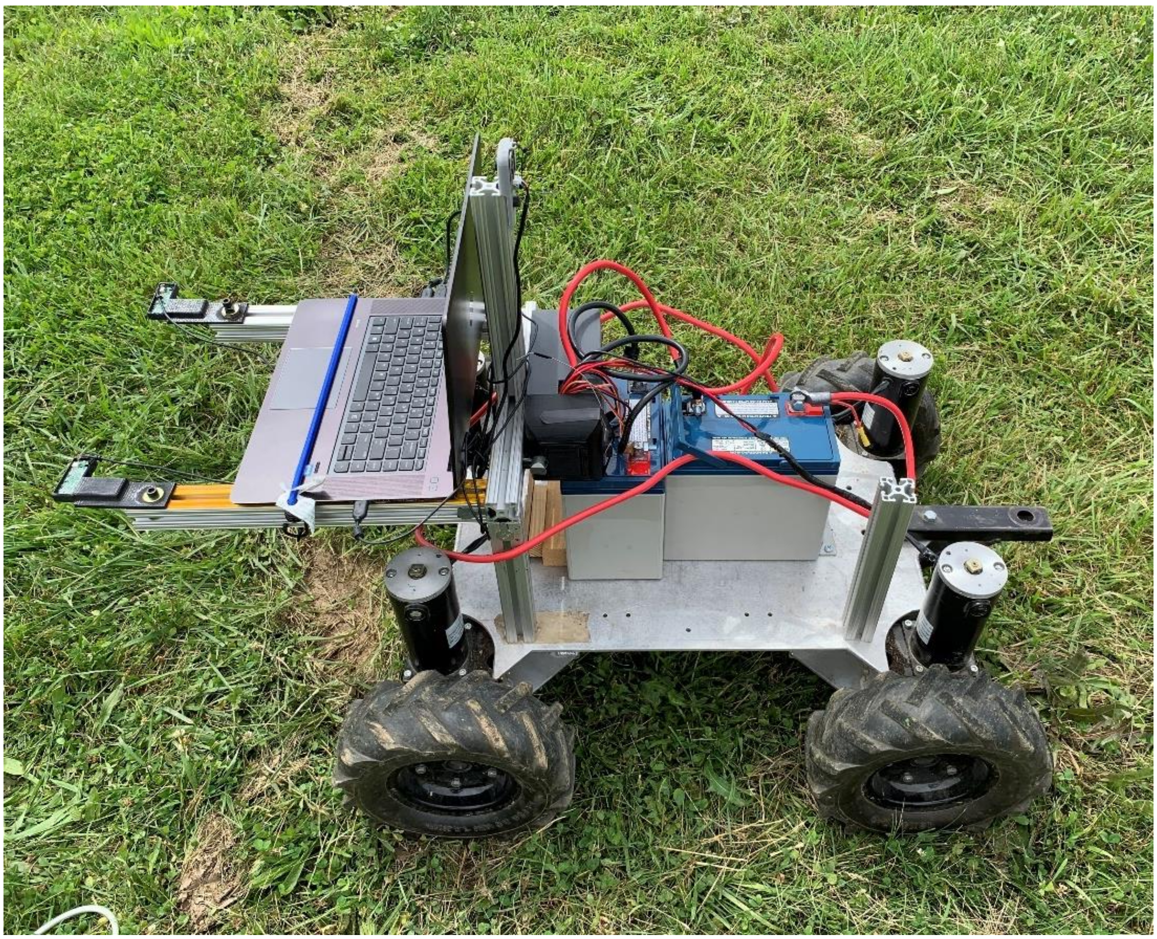

Figure 2. The customized WC800-DM4 robot platform for image acquisition.

An 80% | 20% split was used for training set and test set, respectively. The experiment was repeated 10 times on each network and the mean accuracy was calculated using the well-known and widely used formula, Equation (1):

$$Mean\ Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{1}$$

where *TP* is the number of True Positives, *TN* is the number of True Negatives, *FP* is the number of False Positives, *FN* is the number of False Negatives, and *N* = 10.

Three pre-trained network models were re-trained using transfer learning: SqueezeNet_v1.1 [13], ResNet-50 [14], and Apple's machine learning model Vision Feature Print (VFP) [15]. Our study showed that customized weed classifier with high accuracy can be achieved using transfer learning. Quantitative results were compared and analyzed to support our conclusion (Figure 3).

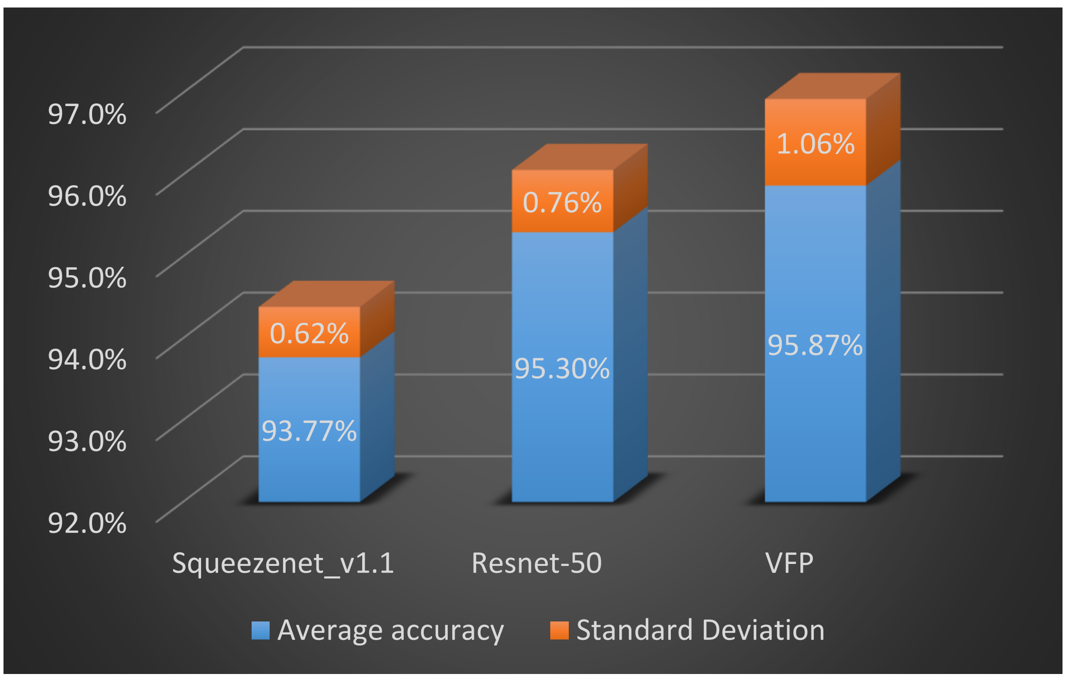

Figure 3. Compared network performance (Soybean Database)

4.2. The Hemp Database

The second plant database used in this study is a hemp database built in 2019. After decades of prohibition, the United States legalized the commercial production of hemp in 2018. Grown for its fibers, grains, and metabolites, the long proscription of hemp production in the US has hindered research to advance the crop. As with most crops, hemp plants encounter stress from weed pressure, especially during early growth stages. Currently, no herbicides are registered for use on hemp, making weed control challenging for a number of growers. Autonomous vehicles provide an opportunity for delivering efficient non-chemical weed control methods.

The hemp database has 1,000 RGB images in two classes: Hemp (*Cannabis sativa*) and Unidentified. The Hemp class contains 500 images of seedling and young plants for three European dual-purpose, *i.e.* used for grain or fiber, ('Felina 32', 'Futura 75', and 'USO-31') and one Chinese fiber ('HanMa') varieties. 'HanMa' plants have broader leaflets (broad-leaf type) compared to the dual-purpose varieties (narrow-leaf type). The Unidentified class contains common US weed species we found in the field in 2019, and bare ground with dead grass or rubble. The database is used to aid development of methods for detecting hemp plants by robotic systems (Figure 4).

Figure 4. Top line: hemp plants with broad leaf type and narrow leaf type. Bottom line: common US weed species and bare ground.

Two pre-trained network models were used here: Inception-v3 [16] and MobileNet 1.0 [17]. An 90% | 10% split was used for training set and test set, respectively. The experiment was repeated 10 times on each network and the mean accuracy was calculated using Equation (1) (Figure 5).

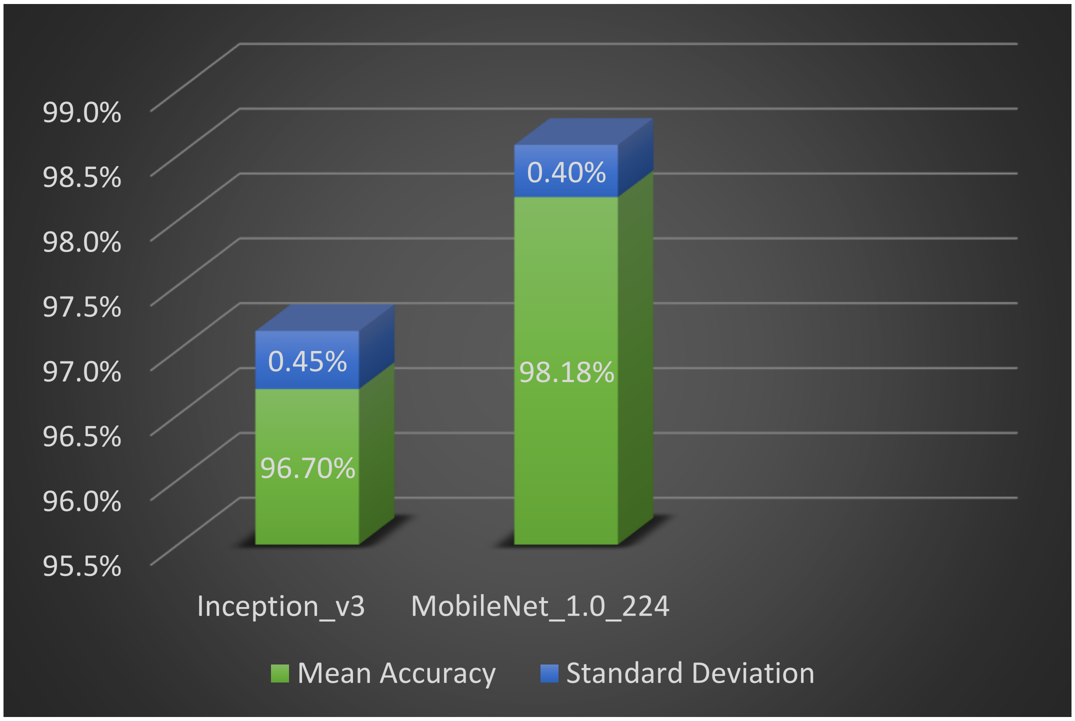

Figure 5. Compared network performance (Hemp Database)

An iOS app named “SoybeanNet” (Figure 6) and an Android app named “HempNet” (Figure 7) were made by two of the RAs in order to demonstrate the performance of the customized CNNs, using Core ML [18] and Android Studio [19], respectively.

Figure 6. iOS app “SoybeanNet”

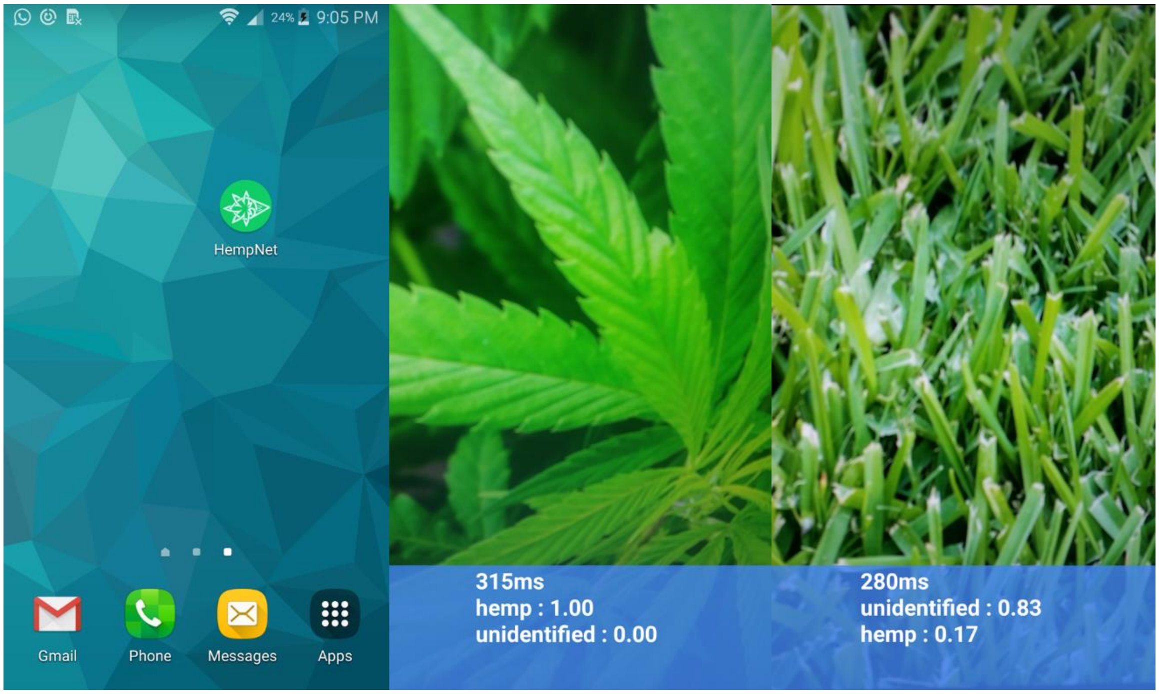

Figure 7. Android app “HempNet”

The customized neural networks were evaluated in real crop fields and the results showed high accuracies that are comparable to the current state-of-the-art image classification accuracy in general, suggesting that these pre-trained networks are transferable to plant classification tasks without significant performance loss. The customized CNNs can be used for real time classification (>=30fps) on a regular computer or a smart phone.

5. Conclusion

Although not statistically conclusive, the assessment results and the performance of the RAs suggested that hands-on AI project is an intriguing experience for undergraduate students. It could benefit a diverse population of students by motivating, engaging, and enhancing their self-learning. Further, it could advance the student’s research skill and enhance the student’s confidence in public presentation. The RA who worked on the soybean database presented his work at the Association of 1890 Research Directors (ARD) symposium in 2019, the other RA who worked on the hemp database presented his work at the Internship and Research Review Workshop at CSU in 2019. Both of them were minority undergraduate students who had good C/C++ programing backgrounds, but no deep learning backgrounds or research experiences before. We believe this work demonstrated a positive example of integrating modern technology and research into undergraduate education.

To some of the students, the term project was a bit too fast-paced, and it was not always easy to maintain a student's self-learning interest. When the students faced coding challenges, they sometimes turned to the instructor instead of working on the issues by themselves, even the issues were not beyond their skillsets. We are considering a longer project period that involves more preparation materials, such as review sessions of intro-level algebra, basic statistics, Python programming basics, and usage of some dependencies such as Matplotlib [20], pandas [21], and Numpy [22].

6. Acknowledgement

This study was funded by the following grants:

- USDA Evans-Allen program: Enhanced Crop Production Efficiency through Mechanized Integrated Pest Management Strategies, 2016 - Present.
- USDA Capacity Building Grant: All-in-One Organic Weed and Crop Disease Management Using Directed Energy and Convolutional Neural Networks, 2019 - 2022.

References:

1. F Jonathan and C Michael, The Lottery Ticket Hypothesis: Training Pruned Neural Networks, in ICLR conference, 2019
2. F Ghasemi, AR Mehridehnavi, A Fassihi, H Perez-Sanchez. Deep Neural Network in Biological Activity Prediction using Deep Belief Network, Applied Soft Computing, 2017.
3. Available at: https://www.tensorflow.org/
4. Available at: https://apple.github.io/turicreate/docs/api/
5. Available at: https://pytorch.org/
6. Google Codelab, Tensorflow for poets. https://codelabs.developers.google.com/codelabs/tensorflow-for-poets

7. TensorFlow Team, Installing Tensorflow on Windows. https://www.tensorflow.org/install/install_windows
8. S Raval. Build a TensorFlow Image Classifier in 5 Min. https://github.com/llSourcell/tensorflow_image_classifier
9. Available at: https://docs.docker.com/
10. Available at: https://www.anaconda.com/
11. Available at: https://data.ers.usda.gov.
12. Available at: https://www.superdroidrobots.com/
13. F N Iandola, S Han, M W Moskewicz, K Ashraf, W J Dally, and K Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. arXiv preprint arXiv:1602.07360, 2016.
14. K He, X Zhang, S Ren, and J Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
15. Available at: https://apple.github.io/coremltools/coremlspecification/sections/VisionFeaturePrint.html
16. C Szegedy, V Vanhoucke, S Ioffe, J Shlens, Z Wojna. Rethinking the Inception Architecture for Computer Vision, Computer Vision and Pattern Recognition, 2015.
17. A G. Howard, M Zhu, B Chen, D Kalenichenko, W Wang, T Weyand, M Andreetto, H Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Computer Vision and Pattern Recognition, 2017.
18. Available at: https://developer.apple.com/machine-learning/core-ml/
19. Available at: https://developer.android.com/studio
20. Available at: https://matplotlib.org/
21. Available at: https://pandas.pydata.org/
22. Available at: https://numpy.org/