

Laboratory Glassware Identification: Supervised Machine Learning Example for Science Students

Arun K. Sharma

Department of Chemistry and Physics

Wagner College

Staten Island, NY

aksharma@wagner.edu

ABSTRACT

This paper provides a supervised machine learning example to identify laboratory glassware. This project was implemented in an Introduction to Scientific Computing course for first-year students at our institution. The goal of the exercise was to present a typical machine learning task in the context of a chemistry laboratory to engage students with computing and its applications to scientific projects. This is an end-to-end data science experience with students creating the dataset, training a neural network, and analyzing the performance of the trained network. The students collected pictures of various glassware in a chemistry laboratory. Four pre-trained neural networks, Inception-V1, Inception-V3, ResNet-50, and ResNet-101 were trained to distinguish between the objects in the pictures. The Wolfram Language was used to carry out the training of neural networks and testing the performance of the classifier. The students received hands-on training in the Wolfram Language and an elementary introduction to image classification tasks in the machine learning domain. Students enjoyed the introduction to machine learning applications and the hands-on experience of building and testing an image classifier to identify laboratory equipment.

KEYWORDS

Machine Learning, Object Identification, Laboratory Glassware, First-year

1 INTRODUCTION

Machine learning applications are increasingly common in the day-to-day interactions of students with technology. An increasing number of products from thermostats to recommendations for the next TV series or movie to watch use some form of machine learning to augment the user experience. Self-driving cars [1], victory in the game of Go over humans [28], and image classification [7] are some of the more high-profile applications of machine learning. However, in addition to these, such tools are also used in email spam filtering [6], credit score determination [9], as well as many others. An interactive history of machine learning, including references and major applications, has been developed by Google [4].

A recently released review article provides more detailed information on applications of machine learning to scientific domains and specifically to the area of material science research [24].

A variety of resources in the domain of machine learning, neural networks, and their applications is now available online that is accessible to people with a range of technical skills. A full review of machine learning is beyond the scope of this paper. However, the interested reader is referred to multiple freely available resources for additional information and background. Coursera hosts a very popular course on machine learning [11]. Wolfram Research provides multiple training videos and courses to get users started on machine learning basics [15, 17], image classification [20], and many more applications using the Wolfram Language [16]. Google also provides a course for developers to introduce them to machine learning, and the examples are accessible to beginners as well as those with more advanced skills [5].

Supervised learning corresponds to the family of approaches that train a neural network to learn from a training set of labeled examples. The trained network, after testing, is utilized in performing the specialized task on new samples of unlabeled data. Deep learning, based on multi-layer neural networks, has recently outperformed traditional approaches in computer vision and natural language processing. One of the major success stories of deep learning applications is image classification [12]. The goal in image classification is to classify a picture according to a set of possible categories. Transfer learning in the field of computer vision enables the construction and implementation of accurate models rapidly and without rebuilding the entire neural network architecture. In practice, a pre-trained model is adopted that was trained on a large benchmark dataset to solve a problem similar to the one under consideration. Such pre-built models are imported from published literature and then adapted for application to the problem of interest. A comprehensive review of the performance of pre-trained models for computer vision problems using the ImageNet data [23] challenge is provided [2].

A commonly implemented first example in image classification is that of distinguishing images of cats from dogs. A pre-trained neural network is provided with a labeled training set of images. The training is performed, and the trained network's performance is then tested using images that were not part of the training set. The success of training becomes quite evident with the results and can be measured in terms of accuracy of classification. The exercise is quite easy to construct and provides a good first example for students. Another exercise that is widely used is the identification of hand-written digits. The Modified National Institute of Standards and Technology (MNIST) database of hand-written digits

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2021 Journal of Computational Science Education
<https://doi.org/10.22369/jocse.2153-4136/12/1/2>

and its classification and identification are also widely used for assessment of potential image classification algorithms [10]. That database includes a training set of 60,000 examples and a test set of 10,000 examples and is also used quite commonly as one of the first examples in this domain.

Our main goals were to introduce students to machine learning applications, highlight the ease of creating such applications using the Wolfram Language, and encourage students to think about possibilities of applying such developments to scientific domains. This project was carried out with students in the author's "Introduction to Scientific Computing" course. The course philosophy and design have been previously described in this journal [26]. That course provides students with an introduction to programming in the Wolfram Language using the Mathematica notebook interface. The crux of the course is to provide students with hands-on experience in production, visualization, and analysis of technical data. Modifications of that course design were also successfully implemented to incorporate a course based undergraduate research style experience with large-scale data analysis [27]. The course has been taught at Wagner College for the last 6 years and has been highly successful in building an awareness of computational approaches in the sciences.

2 METHOD

Students used their smartphones to click pictures of various laboratory glassware routinely used in a Chemistry laboratory. They uploaded the pictures into a shared Google folder directly from their phones. This procedure was adopted to simplify the data collection process. Most of the pictures were taken with the goal of having one main object in the image. A mixture of empty and filled glassware was used to mimic a typical Chemistry laboratory setting. For example, beakers of various volume capacities were used: 250 mL, 500 mL, etc. The chemical composition of the solutions was not important for this exercise. Our intention was to introduce colors into the beakers to increase the sample space of pictures. A collection of sample images from each category is shown in Figure 3. Table 1 displays the number of classes and the number of images in each class in the dataset. A recent publication by Eppel et al. [3] implemented crowdsourcing to collect pictures of glassware in a chemistry laboratory. Their report is of much larger scope, with identification of the phase of the substance present inside the glassware. Our end-to-end exercise is designed with the express purpose of acquainting undergraduate students with the entire process, from collection and organization of raw images to analysis of final results.

The overarching idea was to collect pictures of glassware in a typical laboratory setting. Toward this end, some variability was also introduced in each class by intentionally including some background clutter. However, we realize that this may not be a best practice in terms of a practical goal of achieving the highest possible accuracy or success metric in object identification. Our goal was to get students to think through some of these issues during the collection of pictures. For example, the test tube collection class has pictures of multiple test tubes organized in a test tube stand. In this case, some test tubes were left empty, while others were partially filled with some of the prepared solutions. The test tube stand

Table 1: Classes and number of images in each class

Class	Number of images
Beaker	30
Buchner funnel	20
Buret	11
Buret stand	5
Erlenmeyer flask	24
Flat bottom flask	18
Funnel	23
Graduated cylinder	47
Pipet	16
Round bottom flask	24
Separatory funnel	22
Standard measuring flask	36
Test tube	6
Test tube collection	17
Test tube stand	14
Viscometer	14
Wash bottle	8
Total	335



Figure 1: Erlenmeyer flask images collected by the students. Different colored solutions were used to fill up the flasks to various capacities.

class has pictures of empty test tube stands of different types as well as partially-filled and fully-filled stands. Clearly, this is a nebulous area of labeling in our problem. However, that is a question of semantics, and our interest in this exercise was to demonstrate identification between our assigned labels. Some glassware is routinely seen suspended: for example, burets, separatory funnels, etc. In all such cases, we collected pictures of the glassware by placing them on a laboratory bench and also with their stands or supporting structures.

Figure 1 shows the collection of pictures of Erlenmeyer flasks that were used in the exercise. Some of the flask pictures were taken with empty flasks or with water in the flask. As mentioned earlier, colored solutions were also used in some of the pictures. A concerted effort was made to ensure that the pictures covered different volumes and with some variations in the contents of the flasks. The location of the flasks was also varied, and some pictures were taken on the



Figure 2: Images of pipets of various volume capacities in the dataset. Pipets seem to be difficult to differentiate from the background, and some images used a piece of paper to highlight the object.

laboratory bench, while others involved a common surface that was used for many pictures. Students used their own phones to collect pictures, and consequently there is considerable variation in the brightness, clarity, and contrast between the pictures. Some clutter, like faucets or electrical sockets, is visible in some of the pictures. Figure 2 displays the collection of pipet pictures that were part of the dataset. Pipets were particularly difficult to distinguish from surroundings under the lighting conditions in the laboratory, and some pictures utilized a small piece of colored paper to provide a suitable contrast for the pipet. Some pictures included a rubber bulb attached to a pipet. Additionally, it was quite difficult to get a clear picture of the 5 mL pipets, and some images incorporated a small piece of paper for easier differentiation. An additional picture was added with a pipet suspended from a stand to get a vertical orientation against a neutral wall background. In a similar fashion, buret pictures were also taken with a stand in the picture frame. Similar considerations were applied to all of the images in the dataset.

The image classification task was carried out with four neural networks that have demonstrated excellent results with the ImageNet competition data [8]. This allowed comparative studies and group-based investigations. Inception v1 [13] and Inception v3 [14] released by Google and ResNet-50 [19] and ResNet-101 [18] released by Microsoft were implemented in our exercise. All of these networks were trained on the ImageNet Large Scale Visualization Challenge 2012 classification dataset [23] consisting of 1.2 million images with 1,000 classes of objects. The plug-and-play nature of the pre-trained neural networks was also emphasized by implementing multiple neural networks. These networks are quite recent and well-known in image classification tasks. A brief overview of these networks is provided in Table 2.

Table 2: Four neural networks used for the image classification task. The pre-trained networks were downloaded from the Wolfram Neural Net Repository.

Network	Year	Source	Layers	Parameters
Inception v1	2014	Google	147	6,998,552
Inception v3	2015	Google	311	23,885,392
ResNet-50	2015	Microsoft	177	25,610,216
ResNet-101	2015	Microsoft	347	44,654,504



Figure 3: A sample of thumbnail-sized pictures from each of the classes in the dataset used for the classification process.

These pre-trained neural networks were downloaded from the Wolfram Neural Net Repository [22] and set up according to the instructions provided on the Wolfram website [21]. The training was performed by removing the final classification layers and replacing them with a classifier corresponding to the number of classes, 17, and a SoftMax layer to compute probabilities. The function NetDrop was used to perform these operations, and the training was performed using NetTrain. The training was carried out on a system with dual consumer class Graphical Processing Units for a maximum of 10 training rounds. The training performance is shown in Figure 5. The collected images were labeled, and the dataset was split into training and testing sets. 80% of the images were used for training, and the other 20% were reserved for testing. Since the population of items in the dataset is not uniformly distributed, the splitting of data into training and testing sets was carried out at the level of each class. This ensured that the training and testing sets contained each item of laboratory equipment. The training rounds with augmented images, and thus much larger number of samples, drops down in error during training much more rapidly as compared to the dataset with no augmentation of image samples. In either case, 10 training rounds seem to be sufficient in achieving a very low error during the training phase of the neural networks.

The image classification task performs best with small images, so the first step was to take the thumbnail version of all the images in the dataset. The following four datasets were constructed from the collected pictures to carry out this activity:

- (1) Full color images captured by students
- (2) Enhancement of the full color image dataset by image augmentation methods
- (3) Grayscale images from the full color images
- (4) Enhancement of the grayscale images by image augmentation methods

The step of image augmentation can be carried out through a hidden layer in the neural network. However, we chose to explicitly perform image augmentation to lead students to think through the steps of modifying images to enhance the dataset. The following module was used to carry out the image augmentation.

```
imageSetAugmentation[objectImages_List]:=Module[
{detailEnhanced,blurredImages,noisyImages,
lightDarkImages,reflectedImages,
rotatedImages,augmentedImages},
detailEnhanced=ImageEffect[#, "DetailEnhancing"]
&/@objectImages;
blurredImages=Blur[#,RandomInteger[{1,3}]]&
/@objectImages;
noisyImages=ImageEffect[#, "Noise"]&
/@objectImages;
lightDarkImages=Join[Lighter[#,1]&/@objectImages,
Darker[#,1]&/@objectImages];
reflectedImages=ImageReflect[#,Left]&
/@Join[objectImages,detailEnhanced,blurredImages,
noisyImages,lightDarkImages];
rotatedImages=ImageRotate[#,RandomInteger[{-10,10}]]Degree]
&/@Join[objectImages,detailEnhanced,
blurredImages,noisyImages,lightDarkImages,
reflectedImages];
augmentedImages=Join[objectImages,
reflectedImages,rotatedImages];
Return[augmentedImages];
]
```

The Wolfram Language function ImageEffect was used to carry out detail enhancing and adding random noise effects to each image. Images were blurred using the Blur function with a randomly chosen pixel radius over which the blur was to be applied. Images were modified to appear lighter or darker using the appropriately named functions. Next, all of these images were collected and reflected from left to right. The final operation was to rotate all of these images with a randomly chosen rotation amount between -10 to 10 degrees. The result of all of these operations on one image taken from the set of Erlenmeyer flask images is shown in Figure 4. For every image in the raw dataset, 19 images were produced by the image augmentation procedure described above. The number of raw images in the dataset was 335, and after the implementation of the imageSetAugmentation module, the number of images increased to 6,365 for the two cases where image augmentation was applied. Thus, each network was trained and tested on 4 versions of the images. The versions without augmentation had 335 images in their complete dataset and the versions with augmentations had 6,365 images in their dataset.

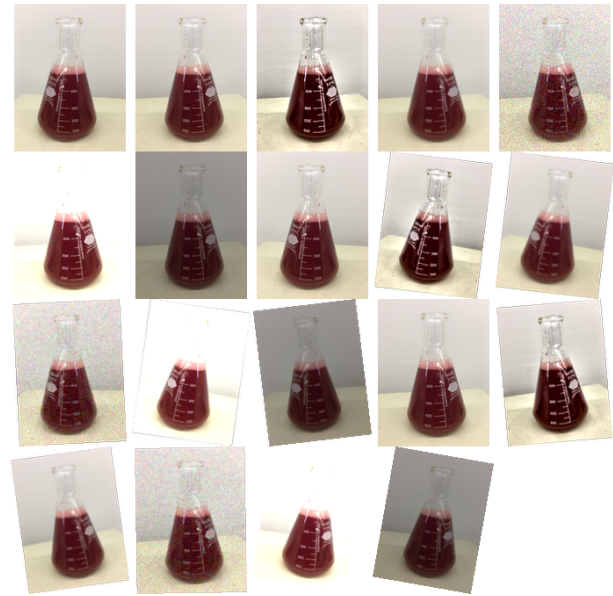


Figure 4: Image augmentation effects shown for an Erlenmeyer flask image. The images are subjected to blurring, rotation, reflection and changes in contrast as described in the text.

3 RESULTS AND DISCUSSIONS

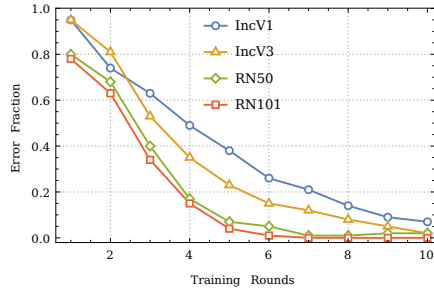
The training of each network resulted in a classifier trained to distinguish between the classes of the laboratory glassware in our training sample. These classifiers were then tested on the testing set generated for each set of images. The classification experiment for image sets without augmentation was carried out five times each, and the sets with augmentation were carried out three times each. The results of the classification performance on the testing set were compared using multiple metrics and are presented below.

3.1 Accuracy

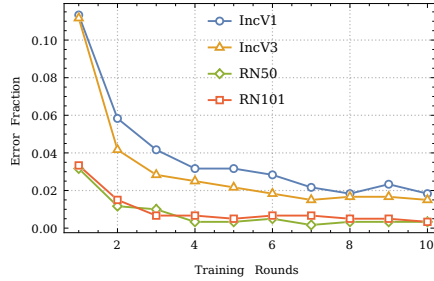
Accuracy is the fraction of correctly identified and labeled images from the testing set. The accuracy of classification is calculated as

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Examples}} \quad (1)$$

A graphical summary of the mean accuracy with standard error for the four networks and the four types of image datasets is shown in Figure 6. The plots show that there is essentially no difference in the training times for color images and grayscale images. The ResNet-50 network seems to provide a suitable trade-off between accuracy and training time in both cases, with and without image augmentation. There is a marked increase in accuracy with the application of image augmentation to increase the sample size for training. The highest classification accuracy of around 92% is lower than the least accuracy recorded, around 97% for the dataset enhanced with image augmentation methods. The image augmentation module increased the dataset size by a factor of 19, and a corresponding increase in training times can be seen from the plots. However, accuracy is not



(a) Validation error during training for the set of images of glassware.



(b) Validation error during training for the set of images augmented with image modifications of glassware.

Figure 5: Validation error during training of all four neural networks. The error is negligible within ten training rounds. The validation error decreases significantly with the larger dataset of augmented images.

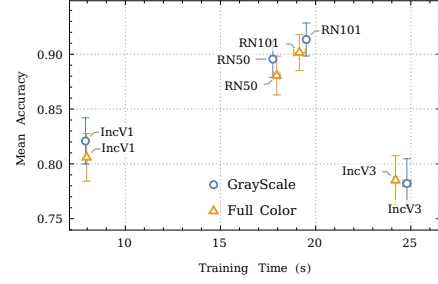
a very reliable metric for a class-imbalanced dataset as present in this exercise.

We also analyzed the accuracy and rejection rate of samples based on different indeterminate threshold values. The maximum rejection rate is seen at an indeterminate threshold of around 90%. A more reasonable value of indeterminate threshold around 30% or 0.3 leads to accuracy around 99%. Figure 7 provides a graphical summary of results from the dataset of colored images with augmentation effects for the ResNet-50 network.

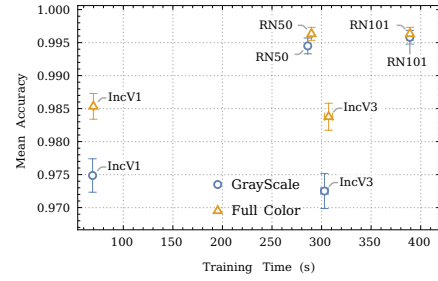
3.2 F₁ Score

The F₁ score is the harmonic mean of the precision and recall for the classification task. A high score implies that the classification produces a low number of false positives and false negatives. The values reported in Figure 8 are averages of the microaveraged F₁ score from each of the iterations. The microaverage F₁ score was calculated for each iteration to account for the differences in class frequencies. It is clear from Figure 8 that training on the augmented dataset enlarged with image effects gives rise to the highest values of F₁ scores for each case, full color images and grayscale images. The difference between ResNet performance and Inception performance is larger when the dataset is small. The calculation of the F₁ score is carried out as follows.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$



(a) Accuracy of classification and training times for the set of images of glassware.



(b) Accuracy of classification and training times for the set of images augmented with image modifications of glassware.

Figure 6: Accuracy of all four neural networks for each dataset. The ResNet-50 neural network provides the best trade-off between accuracy and training time. The datasets with image augmentation lead to much higher accuracy in classification.

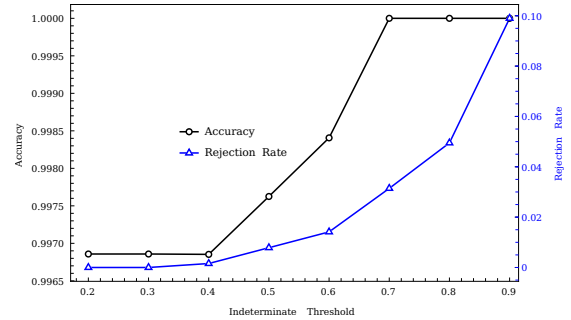


Figure 7: Accuracy and Rejection rate as functions of the threshold for indeterminate classification. Very high accuracy is observed for a wide range of classification thresholds for the dataset with augmented images.

However, since this is a multi-class problem, we computed the micro-averaged F₁ score. The micro-F₁ score is calculated as:

$$\text{Micro } F_1 \text{ score} = 2 \times \frac{\text{Micro precision} \times \text{Micro recall}}{\text{Micro precision} + \text{Micro recall}} \quad (3)$$

The calculation of the micro precision and micro recall are carried out as shown below. The acronyms have their usual meaning, TP



Figure 8: Microaveraged F₁ score for all datasets and neural networks in this exercise. The classification on augmented datasets gives rise to high F₁ scores in each case.

stands for true positives, FP is for false positives, and FN represents false negatives. The sums end at 17, because that is the number of classes in this classification problem.

$$\text{Micro precision} = \frac{TP_1 + \dots + TP_{17}}{TP_1 + \dots + TP_{17} + FP_1 + \dots + FP_{17}} \quad (4)$$

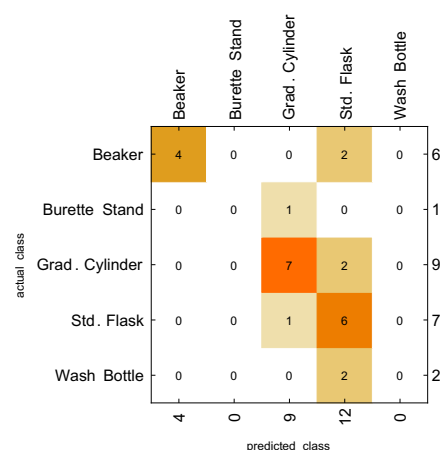
$$\text{Micro recall} = \frac{TP_1 + \dots + TP_{17}}{TP_1 + \dots + TP_{17} + FN_1 + \dots + FN_{17}} \quad (5)$$

3.3 Confusion Matrix

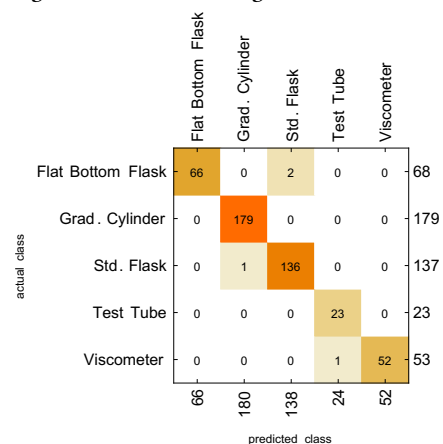
The confusion matrix is a succinct graphical representation of the confusions in the classes encountered by the classifier. Since the performance of ResNet-50 seems to be the most optimal, we highlight the confusion matrix for the top five confusions of this network for the case of augmented and unaugmented full color images. Figure 9 shows the confusion matrix, and it is evident that some of the confusions can be rationalized on the basis that the items in those classes indeed look quite similar to the human eye. For instance, a graduated cylinder is confused with a standard measuring flask, and a beaker is misclassified as a standard measuring flask. Such confusions, on a much smaller scale, also persist in the case of the dataset with augmented images. Another interesting example is that of a viscometer misclassified as a test tube. However, it is important to note that it is one misclassification out of 53 such images tested.

3.4 Geometric Mean Probability

Finally, the average and standard error of the geometric mean probability for the trials are shown in Figure 10. The geometric mean of the class probabilities provides an insight into the overall classification performance. Larger values of the geometric mean signify



(a) Confusion matrix for top five confusions for the set of unaugmented full color images.



(b) Confusion matrix for top five confusions for the set of augmented full color images.

Figure 9: Confusion matrix plots for the classifier from the ResNet-50 network. The numbers on the bottom of each frame represent the number of correctly identified images, and numbers on the right edge of the frame are the total number of images for that class.

uniformly high confidence in the probabilities reported by the classifier during the testing phase. Figure 10 highlights the importance of augmentation and the resulting larger dataset for each case. The geometric mean probability appears insensitive to the color spectrum of the images and increases to values approaching 0.9 – 1.0 with the augmented datasets.

4 TEACHING IDEAS

The images of the dataset and sample notebooks used for training of networks and data analysis are freely available as Supporting Information. Short student projects to investigate the performance of classification for smaller number of classes may be constructed using the dataset. Students could be tasked with specific classes

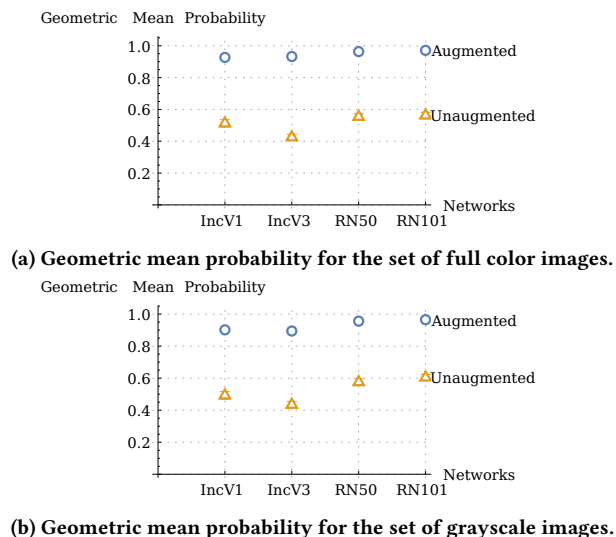


Figure 10: Geometric mean of probabilities of actual class predictions. Part (a) is the set of full color images with and without augmentation, and part (b) is the set of grayscale images with and without augmentation. The augmented datasets in each case display a much larger value of the geometric mean, indicating stronger performance across the different classes.

of glassware images and asked to compare accuracy of classification. Another extension that could be implemented is to carry out image augmentation using different types of transformations and/or subsets of transformations from those that have been used in our implementation. The students could then investigate the effectiveness of those transformations towards the final classification performance. The exercise can also be extended by adding more glassware images and investigating classification performance. An interesting and possibly more advanced application would be to identify the text on glassware that annotates the volume, especially on beakers or Erlenmeyer flasks. Another application would be to identify the piece of glassware and to identify the hand-written or printed chemical species from the label attached to the glassware. This would integrate image classification and hand-writing recognition. The training of neural networks on the dataset with image enhancement is best carried out on systems with a GPU. The training times shown in this manuscript are result from execution on a dual-GPU workstation. However, the smaller datasets without image augmentation can be easily processed on workstations or laptops without a dedicated GPU. We imagine that instructors with limited resources could choose to carry out the training of augmented datasets on a dedicated workstation with a GPU, and students would work with the unaugmented datasets on their personal computing devices.

5 STUDENT FEEDBACK

This exercise was carried out with a cohort of eight first-year students in the author's Introduction to Scientific Computing course.

The students expressed enthusiasm and interest toward more applications of machine learning following this exercise. Although there were no formal surveys, through informal feedback and one-on-one interviews, students pointed out that they enjoyed the project. They specifically enjoyed bringing their computing knowledge into the wet laboratory. Students with interest in biological sciences started discussions on applications of machine-learning methods to images obtained from microscopes. A majority of comments indicated that the activity helped them feel less intimidated about approaching machine learning or artificial intelligence related literature. They also reported increased interest in exploring computation as a tool toward their scientific domains of interest. A significant outcome of the informal feedback process was the realization from students that machine learning and advanced approaches are not limited to computer science majors or large technology companies.

6 CONCLUSIONS

We developed and implemented an end-to-end data science exercise with an application of machine learning experience for STEM students using their laboratory surroundings and equipment as the source of the project. Classification of images based on supervised learning is a common example in the machine learning domain, and the students adapted that into the chemistry laboratory. First-year students collected pictures of various glassware in the chemistry laboratory and implemented the training and testing of classifiers based on four pre-trained neural networks. These neural networks were chosen due to their wide availability and well-known performance on image classification tasks. The glassware images were split into two categories of full color images and grayscale images. Each set of images was enlarged with an image augmentation routine that resulted in a 19-fold increase in the size of the dataset. The students then compared the performance of the classification of glassware among the four networks and for each of the four types of datasets. The performance of the classifiers on the augmented datasets seems to be the most reliable, irrespective of using color images or grayscale images. Our analysis shows that ResNet-50 provides the best trade-off between accuracy and training time for the datasets considered in this activity. We believe that this activity provides students with an accessible and empowering introduction to advanced techniques in the data science domain through the lens of typical glassware in a chemistry laboratory.

7 SUPPORTING INFORMATION

We have provided the dataset of images and some of the Mathematica notebooks used to train the neural networks and to analyze the performance of the classifiers. The components are:

- (1) Chemistry-Glassware-ML-no-augmentation-run1.nb: This notebook provides the code for setup and training of all four neural networks mentioned in the Methods for the dataset of full color images without image augmentation.
- (2) Chemistry-Glassware-ML-with-augmentation-run1.nb: This notebook provides code for setup and training of the aforementioned neural networks for the dataset of full color images augmented with image modification effects.
- (3) A folder called Glassware-Images contains images of the various glassware organized by name.

- (4) Training and testing datasets for the first iteration of the experiment with no image augmentation with filenames training-Set-edison-2020-06-10T05:24:59.mx and testing-Set-edison-2020-06-10T05:24:59.mx
- (5) Binary data export of neural networks trained on the laboratory glassware data. These files all have the .mx extension and the names start with trainedNet-*.mx. The name of the network is included in the filename string.

These resources are located in a shared Google drive folder. A copy of these resources is also hosted on Zenodo [25]. The dataset provides our trained networks with the extension “.mx,” and the notebook entitled, “Analysis-run1-no-augmentation.nb” is set up with the correct filenames to load the trained networks and the testing and training dataset used for that iteration.

ACKNOWLEDGMENTS

We would like to record our appreciation and gratitude to students in the Introduction to Scientific Computing course. We would also like to thank Dr. Tuseeta Banerjee, Dr. Joshua Schrier, and Dr. Rishabh Jain for their helpful suggestions.

REFERENCES

- [1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. 2016. End to End Learning for Self-Driving Cars. (apr 2016). arXiv:1604.07316 <http://arxiv.org/abs/1604.07316>
- [2] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An Analysis of Deep Neural Network Models for Practical Applications. arXiv:1605.07678 <http://arxiv.org/abs/1605.07678>
- [3] Sagi Eppel, Haoping Xu, Mor Bismuth, Alan Aspuru-Guzik, and Cifar Lebovic Fellow. 2020. Computer vision for recognition of materials and vessels in chemistry lab settings and the Vector-LabPics dataset. (apr 2020). <https://doi.org/10.26434/CHEMRXIV.11930004.V3>
- [4] Google. 2017. Explore the history of machine learning. <https://cloud.withgoogle.com/build/data-analytics/explore-history-machine-learning/>
- [5] Google. 2020. Machine Learning Crash Course | Google Developers. <https://developers.google.com/machine-learning/crash-course>
- [6] Thiago S. Guzella and Waldir M. Caminhas. 2009. A review of machine learning approaches to Spam filtering. , 10206–10222 pages. <https://doi.org/10.1016/j.eswa.2009.02.037>
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. CoRR abs/1502.01852 (2015). arXiv:1502.01852 <http://arxiv.org/abs/1502.01852>
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2016-Decem. IEEE, 770–778. <https://doi.org/10.1109/CVPR.2016.90> arXiv:1512.03385
- [9] Cheng Lung Huang, Mu Chen Chen, and Chieh Jen Wang. 2007. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications* 33, 4 (nov 2007), 847–856. <https://doi.org/10.1016/j.eswa.2006.07.007>
- [10] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. <http://yann.lecun.com/exdb/mnist/>
- [11] Andrew Ng. 2020. Machine Learning by Stanford University | Coursera. <https://www.coursera.org/learn/machine-learning>
- [12] Waseem Rawat and Zenghui Wang. 2017. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation* 29, 9 (sep 2017), 2352–2449. https://doi.org/10.1162/neco_a_00990
- [13] Wolfram Research. 2020. Inception V1 - Wolfram Neural Net Repository. <https://resources.wolframcloud.com/NeuralNetRepository/resources/Inception-V1-Trained-on-ImageNet-Competition-Data>
- [14] Wolfram Research. 2020. Inception V3 - Wolfram Neural Net Repository. <https://resources.wolframcloud.com/NeuralNetRepository/resources/Inception-V3-Trained-on-ImageNet-Competition-Data>
- [15] Wolfram Research. 2020. Machine Learning Basics Video Series: Wolfram U. <https://www.wolfram.com/wolfram-u/machine-learning-basics/>
- [16] Wolfram Research. 2020. Machine Learning Courses and Classes: Wolfram U. <https://www.wolfram.com/wolfram-u/catalog/machine-learning/>
- [17] Wolfram Research. 2020. Overview of Machine Learning in the Wolfram Language: Wolfram U Class. <https://www.wolfram.com/wolfram-u/catalog/wl030/>
- [18] Wolfram Research. 2020. ResNet-101 - Wolfram Neural Net Repository. <https://resources.wolframcloud.com/NeuralNetRepository/resources/ResNet-101-Trained-on-ImageNet-Competition-Data>
- [19] Wolfram Research. 2020. ResNet-50 - Wolfram Neural Net Repository. Retrieved June 22, 2020 from <https://resources.wolframcloud.com/NeuralNetRepository/resources/ResNet-50-Trained-on-ImageNet-Competition-Data>
- [20] Wolfram Research. 2020. Supervised Machine Learning: Input & Output: Wolfram U Class. <https://www.wolfram.com/wolfram-u/catalog/wl031/>
- [21] Wolfram Research. 2020. Train a Custom Image Classifier: New in Wolfram Language 12. <https://www.wolfram.com/language/12/machine-learning-for-images/train-a-custom-image-classifier.html?product=mathematica>
- [22] Wolfram Research. 2020. Wolfram Neural Net Repository of Neural Network Models. <https://resources.wolframcloud.com/NeuralNetRepository/>
- [23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (dec 2015), 211–252. <https://doi.org/10.1007/s11263-015-0816-y> arXiv:1409.0575
- [24] Jonathan Schmidt, Mário R. G. Marques, Silvana Botti, and Miguel A. L. Marques. 2019. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials* 5, 1 (dec 2019), 83. <https://doi.org/10.1038/s41524-019-0221-0>
- [25] Arun Sharma. 2020. Glassware images and code samples for training and identification of glassware by neural networks. <https://doi.org/10.5281/zenodo.4019356>
- [26] Arun K Sharma. 2017. A model Scientific Computing course for freshman students at liberal arts Colleges. *The Journal of Computational Science Education* 8, 2 (jul 2017), 2–9. <https://doi.org/10.22369/issn.2153-4136/8/2/1>
- [27] Arun K Sharma, Michelle Hernandez, and Vinh Phuong. 2019. Engaging Students With Computing And Climate Change Through A Course In Scientific Computing. *Journal of STEM Education* 20, 2 (2019), 5–13. <https://www.jstem.org/jstem/index.php/JSTEM/article/view/2409>
- [28] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (jan 2016), 484–489. <https://doi.org/10.1038/nature16961>