Ensemble Deep Learning

Emily Diegel

Jesse Adams and Margaret Lund, PhD Senior Scientists

Nevada National Security Site

Dr. Mihhail Berezovski, Embry-Riddle Aeronautical University

Neural networks are an emerging topic in the computer science industry due to their high versatility and efficiency with large datasets. Funded by the National Science Foundation, Embry-Riddle Aeronautical University is partnered with the Nevada National Security Site on the project, Ensemble Deep Learning, through the Research Experience for Undergraduates 2022 summer program. The Nevada National Security Site is seeking for deep learning techniques to analyze radiographic images of small-scale nuclear explosions in order to ensure that the United States nuclear stockpile remains safe, reliable and secure. In building a deep learning model, multiple convolutional neural network architectures are developed in parallel and combined to create an ensemble neural network. Neural networks are often referred to as a "black box algorithm" due to the uncertainty in the weights and biases applied to the model. This algorithm poses the uncertainty quantification, which is the question of how to correctly measure accuracy in a regressive convolutional neural network. The project's objective is to determine the comparative differences between the efficiency of the ensemble neural network versus each individual convolutional neural network through the uncertainty quantification. As model precision is a key aspect of machine learning, emphasis is placed on the efficiency of ensemble neural networks to produce an error bar alongside the predictions.

The Nevada National Security Site

The Nevada National Security Site is a research and development center that performs high hazard testing, waste disposal, threat detection and stockpile science of nuclear weapons. They provide nuclear and radiological emergency response capabilities and training to the nation. Their primary mission is to ensure the United States stockpile remains safe, secure, and reliable. They perform underground classified nuclear experiments along with high-tech computer simulations and detailed engineering analysis. They use gamma rays to produce radiographic images of mid-explosions to analyze the effectiveness and health of the weapons. The analysis of these radiographic images is made possible through deep learning, and the implementation of neural networks.

Project Scope

NNSS is working to create and introduce new deep learning techniques to be used to analyze radiographic images and provide the necessary error bars alongside, to ensure that the nation's stockpile remains effective and safe. Neural networks provide the opportunity to analyze large

amounts of image data to produce a regressive output. They are an algorithm modeled after the neurons in the human brain. Using a set of inputs and a set of desired outputs, the network is trained and tested in order to predict the desired outputs. The network trains itself through learning patterns and correlations between data and assigning weights and biases to certain values to produce a label. One problem that neural networks pose is the theory behind the black box algorithm. Essentially a neural network creates a black block algorithm because the weights and the biases are unknown in the network, leaving parts of the model unknown by the creator. This algorithm provides no real method to display an error bar on the predictions provided by the model. Without error bars, there is no telling of how accurate or dependable these results are. In the industry world, it is extremely important to provide an error bar alongside a model because these models are going to be used on real world applications and most likely will impact an abundance of lives. For this project, the goal is to determine a method to quantify uncertainty in the ensemble neural network, striving to be able to provide industry with the ability to place a number on validity in predictions made by neural networks.

The tasks of this project include:

- Implementation of convolutional neural network based on a given 2D dataset
- Implementation of ensemble neural network
- Implementation of the uncertainty quantification

Dataset Description:

Synthesis, optical imaging, and absorption spectroscopy data for 179072 metal oxides [1]

Input: Images of metal oxides (64 x 64 Pixels, 3 RGB Channels, 179072 Samples)

- Sample images were taken using a commercially available consumer flatbed scanner (EPSON Perfection V600) in refection configuration at 1200 dpi corresponding to a rate of 2.0 cm2 s−1 or 0.019 s per sample
- All images were rescaled to 64×64 pixels via the python image library (pillow)

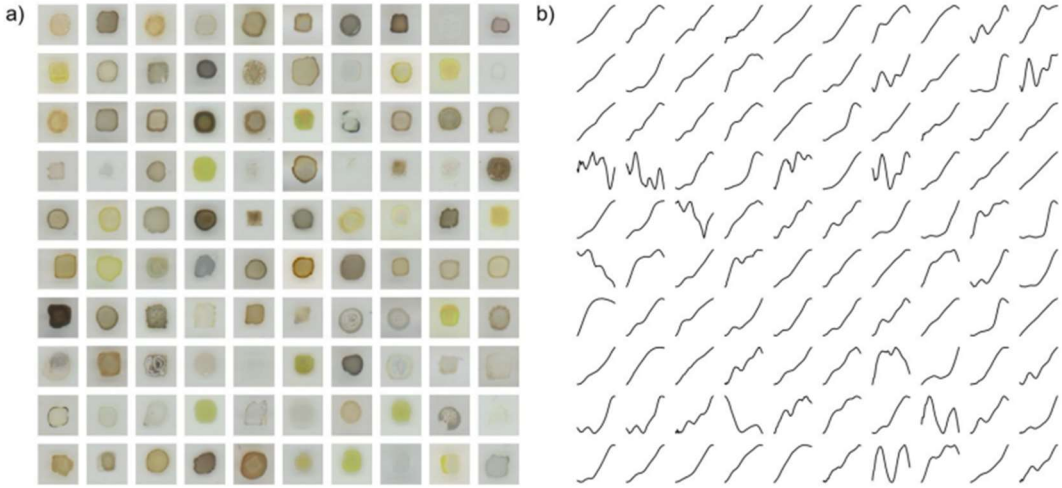
Output: Spectra graph (220 Fractional Absorb Coefficients, 179072 Samples)

- Energy range for all spectra is 1.32 eV (left end) to 3.1 eV (right end)
- Discretize into 220 photon energies
- Optical absorption spectra were measured using an on-the-fly scanning UV-Vis dual-sphere spectrometer

Figure 1. Summary of 2 attributes in the hdf5 container accompanying this manuscript. [1]

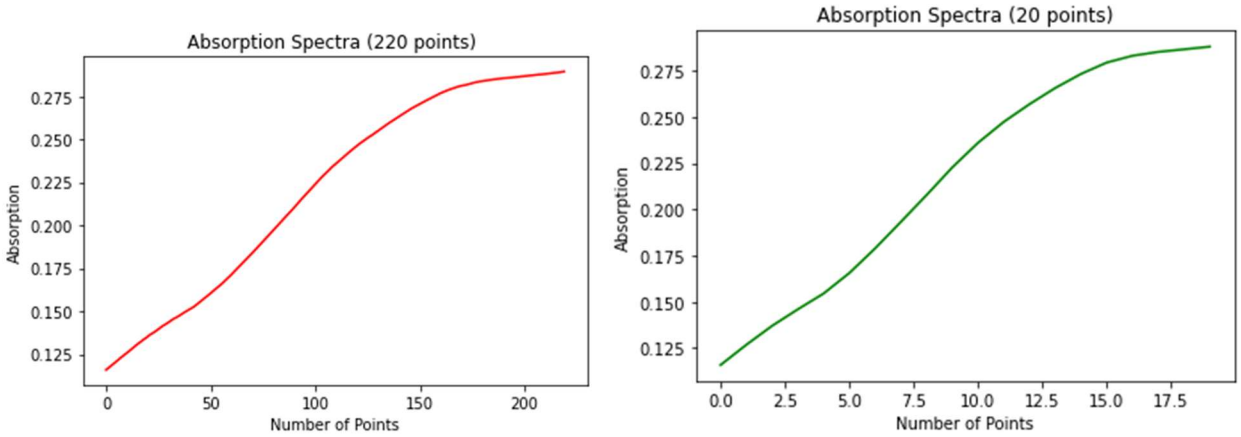
Dataset	Content Description	Data Range	Data Size	Physical Units	Method
Images	Sample Images	0-1 for every channel	(64, 64, 3, 180902)	Color Values for RGB	Platebead Scanner
Spectra	Fractional Optical Absorbance Spectrum	0-ca. 0.5	(220, 180902)	Fractional Absorb. Coefficient	Dual-Sphere Optical Spectrometer

Figure 2. Comparison of Images of Metal Oxides and their Spectra [1]



In the displayed spectra outputs, the graph uses 220 points to plot the shown spectrums. To simply this output, every 11th point was used in the spectra therefore only using 20 points on the graph rather than 220. Minimal accuracy was lost during this preprocessing of the data because the graphs maintained their original shape. This method was used in order to reduce computation of the outputs for the neural network. Rather than predicting 220 points on a graph, it produces 20 points which demonstrates practically the same curve.

Figure 3 and 4. Absorption Spectrums for Metal Oxide



These figures show the minimum difference between only plotting every 11th point or using all 220 points provided in the original dataset. A model can produce 20 outputs more accurately

than 220 points. The curves maintain their shape, allowing this method to be utilized in the neural network.

Convolutional Neural Networks

Convolutional neural networks represent the connectivity of the neurons of the human brain. They produce outputs from image-based data through convolutional, dense, max pooling and flatten layers. The images of the metal oxides are the input values of the network, and the output values are the values of the spectra. This network produces a regressive output of 20 points on the spectra.

Figure 5. Convolutional Neural Network

```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_16 (Conv2D)          (None
, 62, 62, 64)        1792

 dense_64 (Dense)            (None, 62, 62, 64)        4160

 max_pooling2d_16 (MaxPoolin  (None, 31, 31, 64)       0
 g2D)

 flatten_16 (Flatten)        (None, 61504)             0

 dense_65 (Dense)            (None, 128)               7872640

 dense_66 (Dense)            (None, 64)                8256

 dense_67 (Dense)            (None, 20)                1300

=================================================================
Total params: 7,888,148
Trainable params: 7,888,148
Non-trainable params: 0
```

Figure 5 shows the architecture of the convolutional network that was implemented for the spectroscopy dataset. In total there were seven layers in the sequential model. The input shape for our model was a 64 by 64 (pixels) by 3 (RGB) image with an output of 20 points on a spectra.

Layer Summary

Conv2D Layer - Applies a filter to an area of the image and a dot product is calculated between the input pixels with the filter. The dot product is fed through an output array. The filter shifts then by a stride to every kernel in the image to reduce the size of the overall image matrix. [2]

Dense - Computes a weighted average of the input and pass through a non-linear function also known as an activation function.

Max Pooling2D - Calculates the maximum value of each patch in each of the feature maps. Highlights the most important feature of the patch through pooling down into a smaller matrix.

Flatten - Converts the data into a 1D array for inputting into the next layer.

Training

The batch size represents the number of samples in a singular batch. Based on the size of the training data, and the batch size, the number of batches can be calculated. The number of epochs represents the number of times each batch is trained. For example, if there are 1000 batches with 100 epochs, each of the 1000 individual batches of data will be trained different times which is different than training all the data in at once in one batch. Batch size allows the network to train smaller samples of data multiple times. This allows the model to learn more patterns in smaller samples of data.

For the implemented convolutional neural network, an epoch of 100 was used to train the model. Originally 50 epochs were used, but once predictions were run, the graphs produced were not able to learn the patterns of the dataset due to undertraining. With the spectroscopy dataset, for the model to preform well and produce close to the desired results, more epochs were necessary. The batch size for the implemented CNN is 10. Through model testing, the larger the batch size, the more the predictions varied from the label. Lowering the batch size allowed the model to learn more patterns within the inputs and the outputs and assign more accurate weights and biases to produce the output.

As the model is trained, it is tested in parallel to evaluate if the network is training properly. The subset size used to train this model was 10,000 images. 80 percent of the images were used for training and 20 percent was used for validation.

- Number of training batches: 800 with 10 images in each batch
- Number of validation batches: 200 with 10 images in each batch

Testing data is used to ensure that the network can predict values from images that it has not seen before. The images that are used to train the data should not be the same to test if the network is able to function as intended. The purpose of testing data is to introduce new inputs to the network to then have the network produce the output that is unknown by the model.

Loss Function

Loss functions are used to optimize the model in order to minimize the loss function. The goal is to achieve zero loss with any loss function, yet that is not probable for certain models. This function allows the model to define the best weights for the given data. The loss function is directly related to the activation function used in the output layer of the convolutional neural network. The output layer is a choice about the framing of the prediction problem, and the choice of the loss function is a way to calculate the error for a given framing problem.[3]

Loss functions are applied after a batch has been trained on one epoch. Based on the calculation of the loss function, the network will evaluate and adjust the weights applied to data to minimize the loss function. Then the network will train another batch and adjust the weights according to the loss function.

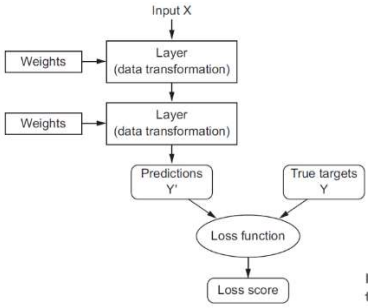


Figure 6. Loss Function Diagram [3]

The loss function used in the implemented convolutional neural network is mean squared logarithmic error.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(x_i+1) - \log(y_i+1))^2}$$

Figure 7. Equation of mean squared logarithmic error

MSLE is used for regression outputs such that the values are continuous. This loss function penalizes overestimates more than underestimates. This function focusses on percentile differences rather than big values versus much smaller values. For instance, a small difference between small data values with be treated similarly to a large difference with large data values.

Figure 8. Mean Squared Logarithmic Error for Implemented CNN

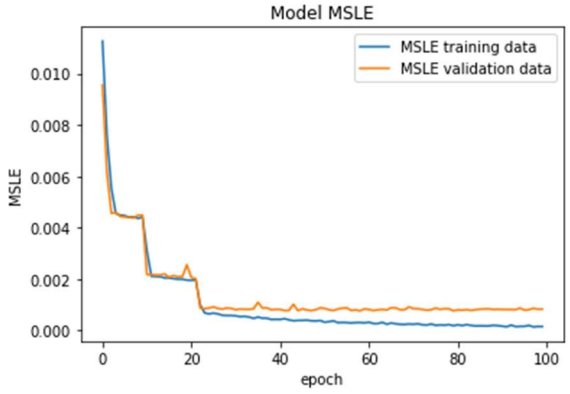


Figure 8 demonstrates the loss of the CNN as the network is trained over 100 epochs. It reaches an overall min of 1.9546e-04 after the training is completed. As the model is trained, the loss decreases due to the new weights and biases that are being assigned to the data.

Figure 9. Predictions from Implemented Neural Network

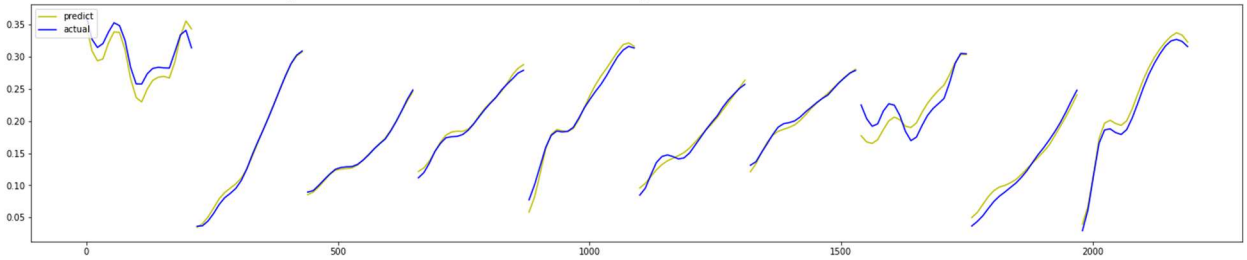


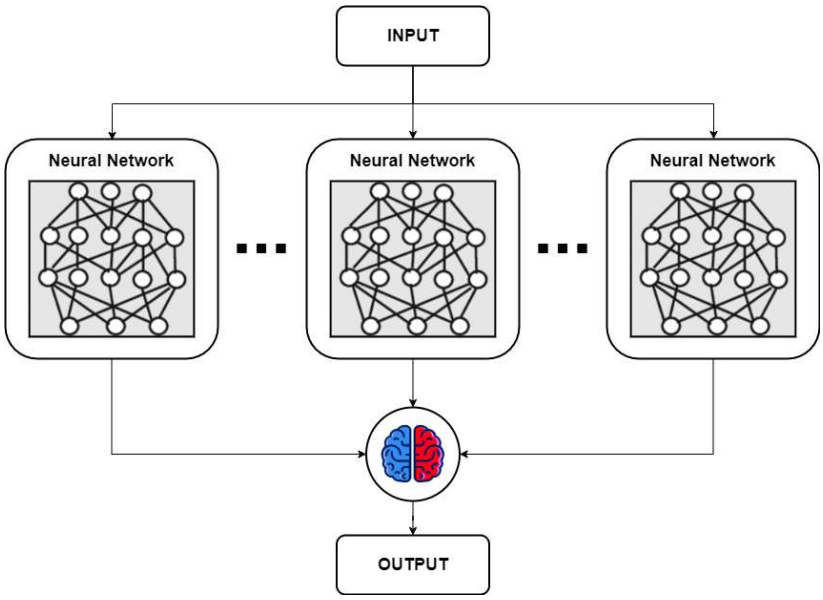
Figure 9 demonstrates 10 different spectra predictions beside the actual spectra values from the original dataset. The yellow is the predicted spectra from the neural network and the blue is the actual spectra.

****graph to show accuracy****

Ensemble Neural Networks

Ensemble neural network combine the decisions of multiple neural networks in parallel to produce a single output for the ensemble network. Outputs may be combined using different methods including taking the average or using a neural network to combine the outputs. Ensemble networks provide the opportunity to use different sections of the dataset in different networks. Each individual neural network in the model can have a slightly different architecture and different training data to output varying results instead of using one singular output from a neural network. This concept can be seen below in figure 4.

Figure 10. Diagram of Ensemble Neural Network [4]



The brain displayed on figure 10 is where the outputs are combined using a chosen method to produce a singular desired output. If a neural network is used to combine the results of the

individual models, the outputs of the individual models would then become in the inputs of the neural network. The outputs used to train the final neural network would be the original outputs from the given dataset. The final neural network would be strictly regressive because the inputs and the outputs are numerical. Only dense layers will be used in this network to produce the desired output.

Implemented Ensemble Neural Network

For the implemented ensemble neural network, five identical convolutional neural networks were trained in parallel. Each was trained on 10,000 of the same images. The architecture of the individual networks include: one convolutional layer was utilized, one max pooling layer, one flatten layer, and a total of four dense layers. Each individual model in the ensemble neural network initializes differently even with the same architecture and training data therefore results in slightly varied predictions.

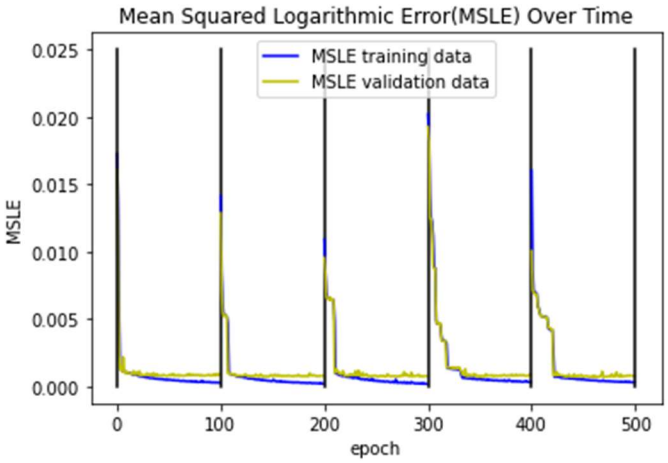
Training:

- 80 percent training
- 20 percent validation
- Epochs = 100
- Batch Size = 10

Loss function:
Mean squared logarithmic error
Metrics:
Mean Absolute Percentage Error

Figure 11. Loss Function for ENN

Figure 12. Loss Function Metric for ENN



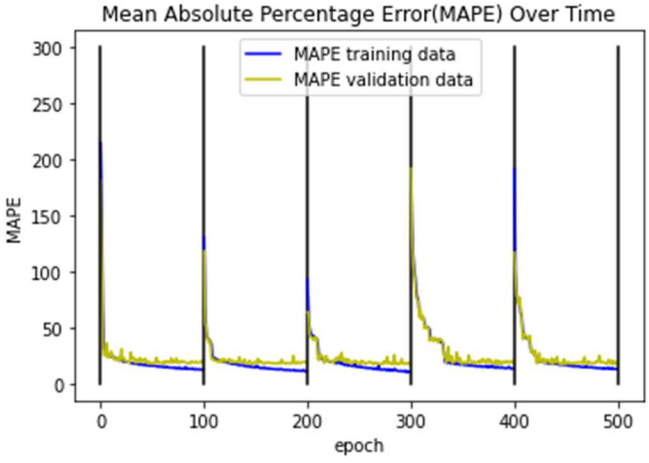
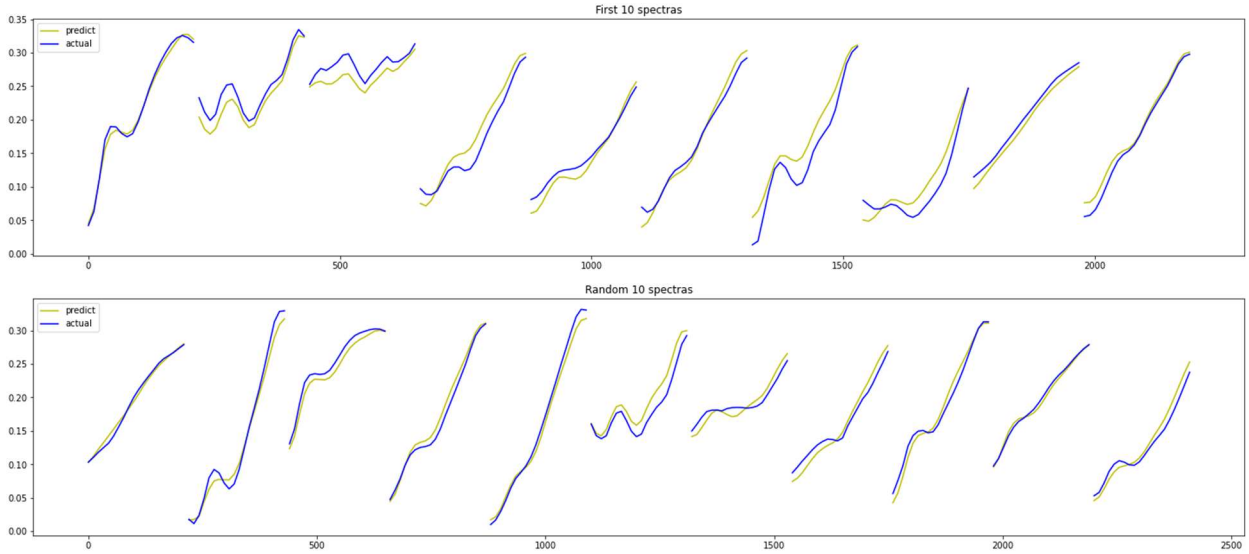


Figure 11 demonstrates the MSLE over the 5 neural networks in the ensemble. This loss function was used to optimize the model during the training whereas the metric, MAPE, was measured during the training but had no affect on the optimization of the network. The training and validation data was tracked for both metrics. The yellow representing the validation data and the blue representing the training data. The loss functions decrease as the network is trained because weights and biases are adjusted according to the value of MSLE. The smaller the MSLE, the better the network is preforming.

WILL PROABALY CHANGE



****graph to show accuracy***
show how ENN is better than NN
UQ

References

Saha, S., 2022. *A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way*. [online] Medium. Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> [Accessed 26 May 2022].

http://www.math.utah.edu/~palais/pcr/spike/Evaluating%20the%20Goodness%20of%20Fit%20--%20Fitting%20Data%20(Curve%20Fitting%20Toolbox)%20copy.html
- good reference for goodness of model

[1] H. S. Stein, E. Soedarmadji, P. F. Newhouse, Dan Guevarra, and J. M. Gregoire, "Synthesis, optical imaging, and absorption spectroscopy data for 179072 metal oxides," *Scientific Data*, vol. 6, no. 1, Mar. 2019, doi: 10.1038/s41597-019-0019-4.

[2] By: IBM Cloud Education, "What are convolutional neural networks?," IBM. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Accessed: 09-Jun-2022].

[3] A. Rakhecha, "Importance of loss function in machine learning," Medium, 16-Sep-2019. [Online]. Available: https://towardsdatascience.com/importance-of-loss-function-in-machine-learning-eddaaec69519. [Accessed: 09-Jun-2022].

[4] M. Cerliani, "Neural Networks Ensemble," *Medium*, Jul. 01, 2020. https://towardsdatascience.com/neural-networks-ensemble-33f33bea7df3 (accessed Jun. 01, 2022).