Heliyon 9 (2023) e20522

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Machine learning approach in predicting GlutoPeak test parameters from image data with AutoML and transfer learning

Takehiro Murai [a,*], Yoshitaka Inoue [b], Assey Nambirige [a], George A. Annor [a,**]

[a] Department of Food Science and Nutrition, College of Food, Agricultural and Natural Resource Sciences, University of Minnesota, 1334 Eckles Avenue, St. Paul, MN, 55108, USA

[b] Department of Computer Science and Engineering, College of Science and Engineering, University of Minnesota, 200 Union St SE, Minneapolis, MN, 55455, USA

ARTICLE INFO

Keywords:
Gluten
GlutoPeak test
Machine learning
AutoML
Convolutional neural network
Transfer learning

ABSTRACT

This study introduces a novel machine learning methodology for predicting GlutoPeak test parameters from image data, leveraging AutoKeras and transfer learning. The GlutoPeak test is a tool used in the baking industry to evaluate the properties of flour, based on its gluten strength and elasticity. Our research aimed to devise an efficient and cost-effective technique for quantifying the gluten properties of wheat varieties. We aimed to accomplish this by predicting the GlutoPeak test results with convolutional neural network (CNN) models, utilizing the benefits of transfer learning and AutoKeras. AutoKeras is a public code repository capable of automating neural architecture search and hyperparameter tuning. The ResNet101 model, when trained with the Adam optimizer, achieved the highest accuracy of 0.5765 in the 2-class prediction. Meanwhile, the ResNet101 model trained with the SGD optimizer reached the highest accuracy of 0.4362 in the 4-class prediction. The outcomes of this study illustrate the possibility in using machine learning and deep learning techniques for predicting GlutoPeak test parameters from image data. This offers a faster and more cost-effective approach for evaluating gluten quality in wheat varieties.

1. Introduction

In recent years, the GlutoPeak test (GPT) developed by Brabender GmbH and CoKG in Duisburg, Germany, has been used as a fast shear-based method for evaluating gluten properties in addition to the mixograph and Farinograph. It measures dough mixing properties and evaluates gluten strength and elasticity, which are important factors in determining the baking properties of flour [1]. The test is primarily used to assess flour suitability for bread and other baked products. The GPT provides output values for aggregation energy (AE), maximum torque (MT), torque 15 s before MT (AM), torque 15 s after MT (PM), and the time required to reach the MT (PMT). Many researchers have explored predicting dough quality and rheology based on these parameters. Mecitoğlu and others reported significant correlations ($p < 0.01$) between all the output values (except PMT) and single-kernel characterization system-hardness, zeleny sedimentation test, Farinograph water absorption, alveograph energy, and bread loaf volume [2]. In another study, Bouachra and others observed a correlation ($r = 0.77$) between bread loaf volumes and AM measured by the GPT [3]. While the

* Corresponding author. 1334 Eckles Avenue, St. Paul, MN, 55108, USA.
** Corresponding author. 1334 Eckles Avenue, St. Paul, MN, 55108, USA.
E-mail addresses: murai021@umn.edu (T. Murai), gannor@umn.edu (G.A. Annor).

https://doi.org/10.1016/j.heliyon.2023.e20522
Received 28 June 2023; Received in revised form 21 September 2023; Accepted 27 September 2023
Available online 28 September 2023

GPT is already an efficient method requiring minimal flour samples, the assay procedure necessitates investment in high-cost equipment including the GPT unit, moisture analysis unit, milling unit, chemical reagents, and specialized personnel training. Utilizing convolutional neural networks (CNN) in a rapid screening of the GPT values could further expedite breeding efforts of wheat varieties and quality control of wheat products by reducing the processes.

The application of Artificial Neural Networks (ANN) and CNN has gained significant attention in various fields, including agriculture [4]. ANN and CNN are machine learning algorithms capable of analyzing complex patterns in large datasets. Their ability to learn from data makes them valuable tools in improving agricultural practices, optimizing crop production, and addressing challenges in the agricultural industry. Specifically in application to wheat crops, many studies have focused on the classification of wheat cultivars from image data. In the study conducted by Anagun and others, CNN was used to classify nine wheat varieties using the EfficientNet models where images were taken with a Scanning Electron Microscope (SEM) [5]. In another study, Lingwal and others explored building a CNN model from scratch by tuning various hyperparameters and successfully classified 15 varieties of wheat [6]. However, up to our knowledge, no attempt has been made in classifying images of wheat kernels according to their corresponding gluten quality.

Transfer learning is a concept in machine learning and deep learning where weights gained from one model is utilized to improve performance on another model [7]. In terms of neural networks, transfer learning involves taking a model typically trained on a large dataset and using it as a starting point for a new task. The pre-trained model can extract features, with the initial layers capturing low-level image features that are relevant across general image data, while the later layers are fine tuned to adapt to specific cases [8]. Transfer learning is particularly beneficial when the target task has limited data available or when training from scratch is computationally expensive [9]. It has been applied in various fields, including image recognition, natural language processing, and audio

Table 1
Gluten aggregation energy of wheat samples.

Sample #	Number of images	Gluten aggregation energy (AE)
1	160	1467.34
2	130	1432.94
3	120	1532.56
4	110	1366.11
5	130	1462.15
6	150	1278.56
7	120	1352.31
8	150	1391.08
9	80	1535.11
10	2	1549.85
11	120	1376.89
12	60	1273.92
13	140	1284.96
14	90	2138.01
15	100	1969.88
16	40	2134.05
17	180	1920.03
18	130	1907.66
19	150	2267.27
20	150	1959.26
21	100	2031.2
22	120	2203.13
23	100	2047.32
24	60	1816.95
25	160	1982.2
26	100	2102.44
27	80	1876.63
28	80	1984.45
29	120	1839.83
30	100	1920.15
31	150	1849.15
32	140	1706.63
33	100	1237.21
34	140	1507.08
35	120	1342.14
36	160	1365.02
37	100	1887.76
38	180	1748.46
39	130	1810.04
40	120	2124.32
41	120	1453.07
42	120	1455.64
43	130	1048.11
44	120	1644.37

analysis, allowing practitioners to achieve competitive results with less effort and resources [10,11].

AutoKeras is a popular open-source library that automates the process of neural architecture search (NAS) and hyperparameter tuning [12]. It simplifies the development of deep learning models by providing an easy-to-use interface for automatic model selection and optimization. With AutoKeras, users can quickly build and deploy custom neural network architectures without the need for extensive manual configuration. The library leverages NAS to automatically explore and select the optimal model architecture and hyperparameters for a given task. By automating these tedious and time-consuming tasks, AutoKeras allows researchers and practitioners to focus more on the problem at hand and accelerate the development of high-performing deep learning models.

In this study, we aim to develop a prototype of image acquisition and classification system that can be used in quantifying gluten properties of wheat varieties in effort to provide a more cost and time efficient method.

2. Materials and methods

2.1. Plant material

Forty-four spring wheat varieties harvested in St. Paul, Minnesota were used for this study. The wheat kernels were threshed in a Vogel thresher, dried for 72 h at 32 °C, and the debris were removed by screening and a fan. The wheat kernels were then stored under refrigeration at a temperature of 4 °C until the time of analysis.

2.2. GlutoPeak test

Whole grains of wheat were milled using a UDY cyclone mill (Fort Collins, CO, USA) to less than 0.5 mm in particle size. Flour moisture contents were taken on an Ohaus MB45 (Parsippany, NJ, USA) infrared balance within 24 h of the GlutoPeak test. Gluten properties of flours were measured in duplicate on a Brabender GlutoPeak (C. W. Brabender) based on the method reported by Chandi and Seetharaman [1]. AE values of each wheat variety are shown in Table 1.

2.3. Image acquisition and preprocessing

A compartment was designed to enable controlled image acquisition conditions. The compartment consisted of three components: a bottom sample tray, a middle section, and a top lid featuring two engraved lanes for positioning the UV-A LED strip lights (Fig. 1). Compartment parts were 3D printed on a Prusa Research Original Prusa i3 MK3S + unit (Prague, Czech Republic) with a 1.75 mm PLA Deep Black filament sourced from Atomic Filament. (Kendallville, IA) The UV-A LED strip lights with a wavelength of 365 nm were purchased from Waveform Lighting (Vancouver, WA). Images were captured with an iPhone® XR with an image acquisition software developed using Swift®. Users were instructed to position the sample tray within the designated orange box shown on the application screen and to tap the center of the red box to ensure consistent lighting focus across all acquired images (Fig. 2). The images underwent a cropping process, where a square region of 1500 × 1500 pixels was extracted around the sample tray area containing the wheat kernels (Fig. 3). Images were then preprocessed by rescaling it to a range of 0–1 by dividing each pixel value by 255. The quantity of images secured for each type depended on the accessible seed samples, with the goal of broadening the visual representation of each variety.

2.4. Labeling dataset

There was a total of 5162 images acquired for this study. Image data was stored as JPG files on a Google Drive folder. For training the 2-class model, the dataset was split into subcategories corresponding to over 1700 AE and under 1700 AE. For training the 4-class model, the dataset was split into subcategories corresponding to 1400–1740 AE, 1740–1950 AE, over 1950 AE, and under 1400 AE. Directory structure was constructed in a manner where the name of the sub-directories served as the class labels as seen in Fig. 4.

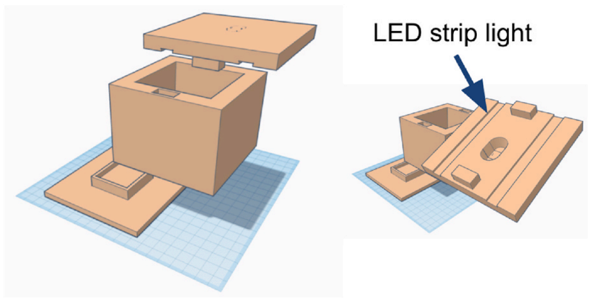

Fig. 1. Compartment for image acquisition.

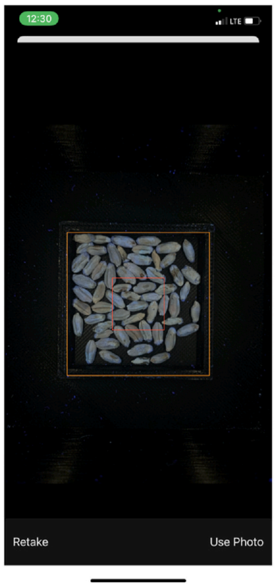

Fig. 2. Sample screen of image acquisition software in use.

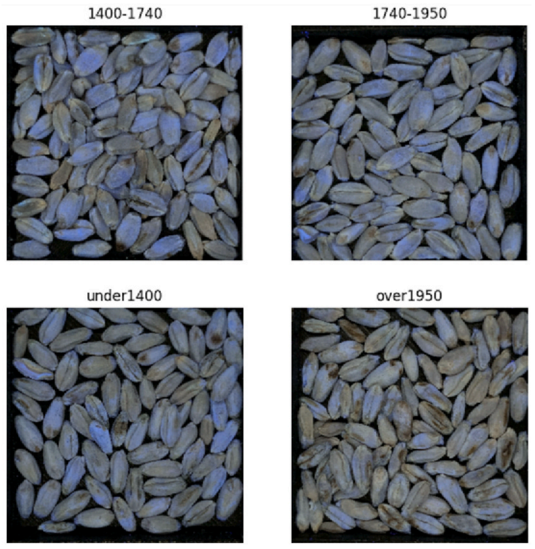

Fig. 3. Cropped images of wheat kernel samples
Labels correspond to the class of each image as defined in the section 2.4.

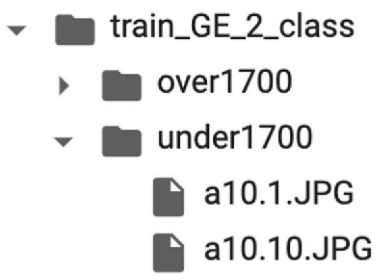

Fig. 4. Image dataset directory.

2.5. Training, validation, and test set

As discussed previously, it is known that CNN models can classify among wheat varieties at a high accuracy from past studies [5,6]. It is plausible that the CNN model could first categorize wheat images into varieties, and subsequently group these varieties into different AE value categories, instead of directly classifying wheat images into AE value categories (Fig. 5). Therefore, the preprocessed data was partitioned into training, validation, and test sets approximately at a ratio of 6:2:2 where each set contained different wheat varieties to prevent potential confounding effects (Fig. 6). Specific varieties used for the validation and test set in each classification task can be found in Table 2.

2.6. Computing environment

The CNN models were trained on a Google Colab platform with NVIDIA A100 Tensor Core GPU on a high memory runtime.

2.7. Transfer learning

Transfer learning was utilized in this study by leveraging pre-trained models from Keras Applications. The weights of these models were initialized with the popular “imagenet” dataset [13]. (Fig. 7) To fine-tune the models for our specific task, a common approach was followed, as demonstrated in the Keras Applications documentation [14]. The final layer of the base model was excluded, and three additional layers were added: a Global Average Pooling layer, “relu” activation layer, and an output layer. For the two-class models, the activation function of the output layer was set to “sigmoid”, while for the four-class models, the activation function was set to “softmax”. This configuration allows the new models to adapt the base models to the new image classes while retaining the pre-learned features from the “imagenet” dataset. Image size used for the training was 224 $\times$ 224 and the batch size, 32. Models were trained using both the Adam optimizer and the Stochastic Gradient Descent (SGD) for comparison [15]. Both optimizers had a learning rate of 0.001 and the SGD optimizer’s momentum was configured to 0. For models with two classes, the BinaryCrossentropy was used as the loss function, and for those with four classes, CategoricalCrossentropy was chosen. Correspondingly, we utilized binary accuracy as the metric for the two-class models and categorical accuracy for the four-class models to evaluate their performance. Models were set to train for 100 epochs with early stopping implemented to terminate the model training if there was no improvement of the validation loss after 20 epochs. Seed was set prior to training for reproducibility. Transfer learning was employed with Python 3.10 programming language with TensorFlow version 2.12.0 embedded with Keras. Summary of the hyperparameters for transfer learning process is as seen in Table 3.

2.8. Stochastic gradient descent

Stochastic Gradient Descent (SGD) is a frequently utilized optimization method in neural network training. It is an extension of the traditional Gradient Descent (GD) algorithm but with some key differences [16]. The SGD method iteratively adjusts the network’s weights and biases by selecting a random subset of training samples to progressively reduce the loss. This random sampling introduces a level of stochasticity into the optimization process. Secondly, SGD converges faster than GD since it performs frequent weight updates based on the gradients computed on mini batches. Thirdly, SGD introduces more noise and variance into the optimization process compared to GD which can help SGD to escape local minimums.

2.9. Adam optimizer

The Adam optimizer is an optimization algorithm with an adaptive learning rate that combines the advantages of both AdaGrad and RMSProp algorithms [15]. The Adam optimizer maintains adaptive learning rates for each parameter in the network by calculating a progressively diminishing average of past gradients and squared gradients. It incorporates both first-order moments (the average of gradients) and second-order moments (the average of squared gradients) to improve the parameters. Advantages of the Adam optimizer include computing individual learning rates for each parameter to handle sparse gradients and noisy data, adapting the learning rates based on the intensity of gradients for quicker convergence, and incorporating momentum to accelerate the optimization process

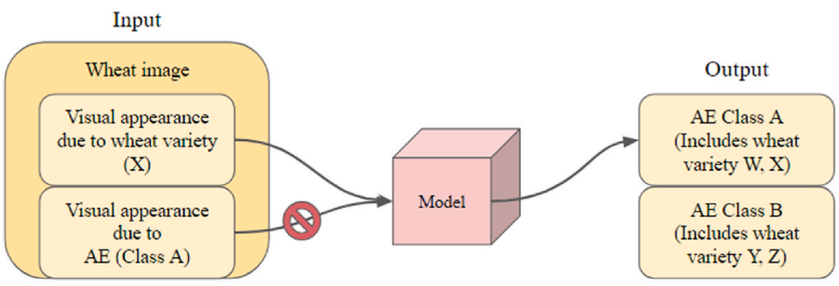
Input
Wheat image
Visual appearance due to wheat variety (X)
Visual appearance due to AE (Class A)
Model
Output
AE Class A (Includes wheat variety W, X)
AE Class B (Includes wheat variety Y, Z)

Fig. 5. Plausible confounding effect.

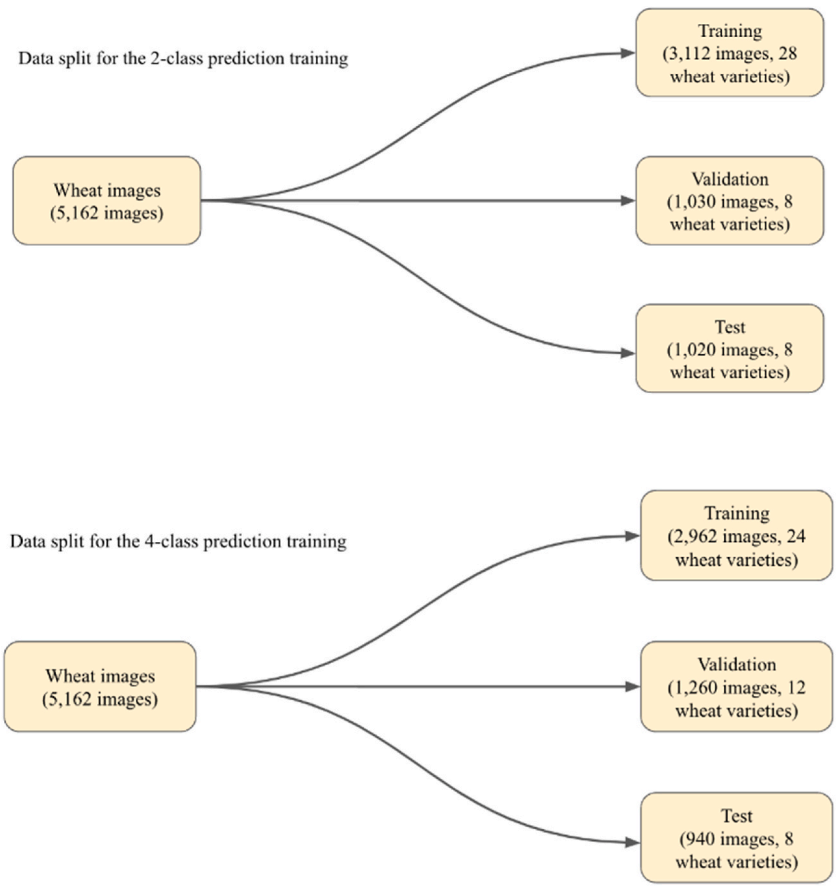
Data split for the 2-class prediction training
Wheat images (5,162 images)
Training (3,112 images, 28 wheat varieties)
Validation (1,030 images, 8 wheat varieties)
Test (1,020 images, 8 wheat varieties)
Data split for the 4-class prediction training
Wheat images (5,162 images)
Training (2,962 images, 24 wheat varieties)
Validation (1,260 images, 12 wheat varieties)
Test (940 images, 8 wheat varieties)

Fig. 6. Splitting scheme of image dataset.

Table 2
Samples used for validation and test set.

Number of class	Set	Sample #s
2	Validation	29, 30, 31, 32, 33, 34, 35, 36
2	Test	37, 38, 39, 40, 41, 42, 43, 44
4	Validation	3, 9, 11, 26, 27, 28, 29, 33, 35, 37, 40, 41
4	Test	4, 7, 14, 18, 19, 30, 42, 44

and overcome local minimums.

2.10. Categorical cross-entropy and binary cross-entropy

Categorical cross-entropy is a widely used loss function in multi-class classification tasks [15]. It quantifies the dissimilarity between the predicted probabilistic distribution and the true probabilistic distribution, often transformed into one-hot encoded labels in the context of classification. In neural networks, categorical cross-entropy measures the discrepancy between the class probabilities predicted by the model and the actual probabilistic distribution. Larger discrepancies between the probabilistic distributions predicted by the model and the actual probabilistic distributions result in higher loss values. In this study, we compute loss for the 4-class prediction models using categorical cross-entropy, and for the 2-class prediction models, we employ binary cross-entropy. The formula for calculating cross-entropy in multi-class classification is defined as.

$\text{Loss} = -\sum_{i=1}^{n} y_i \cdot \log(\hat{y}_i)$, while the formula for binary cross-entropy is defined as

$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n} \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$, where y_i is defined as expected value and $\hat{y}_i$ is defined as the predicted value.

2.11. ReLU function

The Rectified Linear Unit (ReLU) function is a type of activation function in neural networks that introduces non-linearity to the

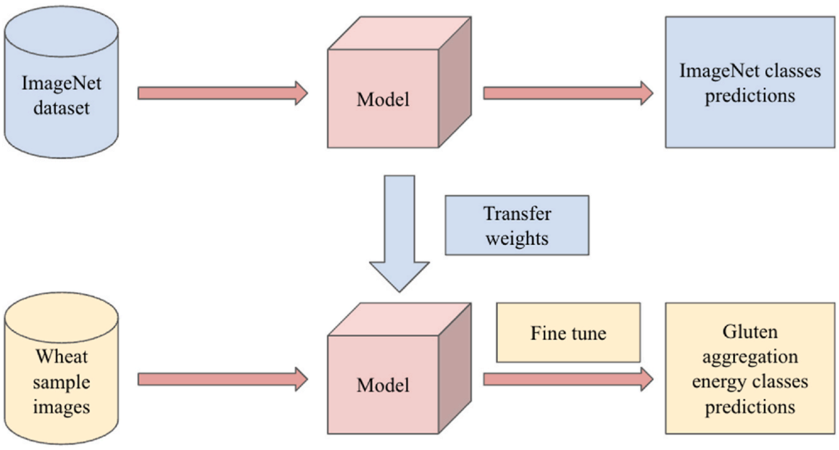
ImageNet dataset
Model
ImageNet classes predictions
Transfer weights
Wheat sample images
Model
Fine tune
Gluten aggregation energy classes predictions

Fig. 7. Diagram of transfer learning.

Table 3
Hyper-parameters used for training the 2-class and 4-class, transfer learning models and AutoKeras models.

Model	Image size	Batch size	Learning rate	Max trials	Epochs	Early stopping (epochs)
Transfer learning	224 × 224	32	0.001	N/A	100	20
AutoKeras	224 × 224	32	N/A	50	50	20

network by mapping the input values to the output based on the following equation:

ReLU $\sigma(x) = \max\{0, x\}$ where x is the input value [17].

By introducing non-linearity, the ReLU function allows neural networks to learn complex patterns and propagate important features of the network onto the next layers.

2.12. *Sigmoid function*

The sigmoid function is a type of activation function in neural networks that maps the input values between 0 and 1, providing a non-linear transformation making it suitable for binary classifications defined as:

Sigmoid $(x) = \frac{1}{1+e^{-x}}$ where x is the input value [16].

The result from the sigmoid function can be understood as the likelihood that the input is associated with a specific class.

2.13. *Softmax function*

The softmax function is a type of activation function widely utilized for multiclass classification problems in neural networks, that takes an input vector and normalizes it into a probabilistic distribution over multiple classes defined as:

Softmax $(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ where z is the input vector, e^{z_i} is the standard exponential function for the input vector, e^{z_j} is the standard exponential function for the output vector, and K is the total number of classes in the classifier [16].

The softmax function transforms the input values into a probabilistic distribution, amplifies the differences between the input values, and produces outputs that are non-negative and mutually exclusive.

2.14. *AutoKeras*

In this experiment, both the two class and four class models were trained over a total of 50 trials, with each trial involving the training of the model for 50 epochs. Image size of 224 × 224 and batch size of 32 were used for both models. The hyperparameters are as summarized in Table 3. Seed was set prior to training for reproducibility. AutoKeras version 1.1.0 was employed for training the models.

2.15. *Measure of performance*

Performance of the models will be evaluated on the two test sets by evaluating the test accuracy, precision, recall, and F1-score. The equation for each are as follows:

$$\text{Test accuracy} = \frac{\text{Number of predictions correct}}{\text{Number of predictions made}}$$

$$\text{Precision} = \frac{\text{Number of TP}}{\text{Number of TP} + \text{Number of FP}}$$

$$\text{Recall} = \frac{\text{Number of TP}}{\text{Number of TP} + \text{Number of FN}}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP is true positives, FP is false positives, and FN is false negatives [18].

In multiclass predictions, precision, recall, and F1 score are calculated for each class individually, often considering that class as the "positive" class and the other classes as the "negative" class, then a weighted average is usually considered for all classes to provide a single measure. For our study, we used macro-average recall, precision, and F1 score to treat all classes equally by averaging the metrics for each class without considering their size, thus ensuring that performance on smaller classes will have an equal impact on the final score as larger ones.

3. Results & discussion

Trained models were evaluated based on their test accuracy, precision, recall, and F1 score. The results for the 2-class prediction models can be found in Table 4, while Table 5 presents the results for the 4-class prediction models.

In the two-class prediction tasks, ResNet101, trained with the Adam optimizer, outperformed other models with the highest test accuracy, precision, recall, and F1-score of 0.5765, 0.5787, 0.5700, and 0.5612, respectively (Table 4). In the four-class prediction tasks, ResNet101 trained with the SGD optimizer excelled with the highest test accuracy, recall, and F1-score, yielding values of 0.4362, 0.4348, and 0.3168 (Table 5). However, when it comes to precision, DenseNet121, trained with the SGD optimizer, led with a score of 0.3072. Considering that these values exceed the probability of correct classifications with random chance, they hint at a moderate correlation between certain physical attributes and the AE value of various wheat varieties.

In the context of this study, where equal importance is given to the correctness of all classes, test accuracy emerges as the most appropriate performance measure [19]. Therefore, ResNet101 would be the preferred model for both tasks. On the other hand, when the goal is to identify a specific AE class with minimal false-positive predictions, DenseNet121 trained with the SGD optimizer becomes the better choice due to its superior precision [20].

When deploying neural networks, computational intensity becomes a significant consideration, alongside evaluation metrics such as test accuracy [21,22]. Although the ResNet101 models outperform others in both classification tasks, they also contained the greatest total number of total parameters. Although the relationship between the parameter size and computational cost isn't always straightforward, models with more parameters are typically more computationally intensive. This is largely due to the need to store each parameter—a floating-point number—in memory, which requires more resources [23]. Thus, it's essential to find a balance between computational speed and the effectiveness of the model, particularly in scenarios where resources are limited, or deployment speed is a critical factor.

In both classification tasks, the models constructed with AutoKeras, which were trained for 50 trials and 100 epochs each, exhibited the fewest total parameters: 243 for the two-class prediction task, and 46,507 for the four-class prediction task. Even so, its performance was comparable to other transfer learning models such as DenseNet, VGG19, and Xception, all of which have significantly more parameters than the models built with AutoKeras. Other research has also shown that there isn't a direct linear relationship between the number of parameters and the model's efficacy [24]. The result of this study also underscores the intricate relationship between the number of parameters, model architecture, and the characteristics of the dataset being classified.

Adam typically outperforms SGD in real-world applications, which often results in its adoption as the standard optimizer in a multitude of deep learning frameworks [15]5]. However, the selection between Adam, SGD, or any other optimizer, is contingent upon the specific characteristics of the problem and dataset being considered. As evidenced in this study, there are instances where SGD may outperform Adam in certain tasks or types of network architectures [25]. This underlines the importance of empirical testing in machine learning model development.

Some limitations exist for this study. Samples of wheat kernels were limited to spring wheat harvested from St. Paul Minnesota. As such, the results demonstrated in this study may not transfer to a setting where there is a wider range of wheat varieties to classify. Results of this study are also specific to the conditions of image acquisition as described in the methodology section. Similarly, results are limited to the versions of the libraries, TensorFlow and AutoKeras, and computing environment in which the neural networks were employed. Therefore, these findings should be considered as a preliminary guide rather than definitive conclusions.

4. Conclusion

This study offers insights into leveraging machine learning for predicting GlutoPeak test parameters, serving as a preliminary guide for subsequent studies in this field. The ResNet101 model, trained with the Adam optimizer, achieved the highest accuracy, precision,

Table 4
Summary of 2-class prediction models.

Model	Number of total parameters	Optimizer	Test accuracy	Precision	Recall	F1 Score
DenseNet121	8,088,129	SGD	0.3598	0.3603	0.3602	0.3598
Xception	22,960,681	SGD	0.5196	0.5187	0.5187	0.5187
VGG19	20,550,721	SGD	0.3490	0.3274	0.3429	0.3295
ResNet101	44,757,377	SGD	0.5235	0.5450	0.5335	0.4961
DenseNet121	8,088,129	Adam	0.3911	0.2598	0.5000	0.3419
Xception	22,960,681	Adam	0.3843	0.3845	0.3873	0.3820
VGG19	20,550,721	Adam	0.3353	0.3340	0.3341	0.3341
ResNet101	44,757,377	Adam	**0.5765**	**0.5787**	**0.5700**	**0.5612**
AutoKeras	243	N/A	0.3569	0.3461	0.3526	0.3469

Highest evaluation metrics are bolded.

Table 5
Summary of 4-class prediction models.

Model	Number of total parameters	Optimizer	Test accuracy	Precision	Recall	F1 Score
DenseNet121	8,091,204	SGD	0.2755	**0.3072**	0.2774	0.2751
Xception	22,963,756	SGD	0.2287	0.2495	0.2292	0.2284
VGG19	20,553,796	SGD	0.2309	0.2479	0.2288	0.2350
ResNet101	44,760,452	SGD	**0.4362**	0.2872	**0.4348**	**0.3168**
DenseNet121	8,091,204	Adam	0.3266	0.2571	0.3277	0.2863
Xception	22,963,756	Adam	0.2681	0.2663	0.2670	0.2659
VGG19	20,553,796	Adam	0.2053	0.1637	0.2038	0.1809
ResNet101	44,760,452	Adam	0.3521	0.2446	0.3457	0.2558
AutoKeras	46,507	N/A	0.2287	0.2055	0.2324	0.2102

Highest evaluation metrics are bolded.

recall, and F1 score for the 2-class prediction. The same ResNet101 model also achieved the highest accuracy, recall, and F1 score for the 4-class prediction, while the DenseNet121 model trained with the SGD optimizer demonstrated the highest precision. The results of this study are specific to the wheat varieties used, the conditions of image acquisition, the library versions, and the computing environment. Future research should expand upon these findings by exploring a wider range of wheat varieties with diverse appearances and gluten properties. Moreover, researchers should consider exploring various libraries, computing environments, and hyper-parameters for the training of machine learning models.

Author's notes

Codes for transfer learning, AutoKeras, Stl files for the compartment, and the image acquisition software are available at https://github.com/murai021/GlutoPeakTest_CNN.

Data availability statement

Data will be made available on request.

CRediT authorship contribution statement

Takehiro Murai: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yoshitaka Inoue:** Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Assey Nambirige:** Data curation. **George A. Annor:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to correct for grammatical errors. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Gurpreet Kaur Chandi, Koushik Seetharaman, Optimization of gluten peak tester: a statistical approach: optimization of gluten peak tester, J. Food Qual. 35 (1) (2012) 69–75, https://doi.org/10.1111/j.1745-4557.2011.00425.x.
[2] Mecitoğlu Güçbilmez, Çiğdem, Mehmet Şahin, Aysun Göçmen Akçacık, Seydi Aydoğan, Berat Demir, Sümeyra Hamzaoğlu, Sadi Gür, Enes Yakışır, Evaluation of GlutoPeak test for prediction of bread wheat flour quality, rheological properties and baking performance, J. Cereal. Sci. 90 (2019), 102827, https://doi.org/10.1016/j.jcs.2019.102827.
[3] Safia Bouachra, Jens Begemann, Lotfi Aarab, Alexandra Hüsken, Prediction of bread wheat baking quality using an optimized GlutoPeak® -test method, J. Cereal. Sci. 76 (2017) 8–16, https://doi.org/10.1016/j.jcs.2017.05.006.
[4] Sebastian Kujawa, Gniewko Niedbała, Artificial neural networks in agriculture, Agriculture 11 (6) (2021) 497, https://doi.org/10.3390/agriculture11060497.
[5] Yildiray Anagun, Isik Sahin, Murat Olgun, Okan Sezer, Zekiye Budak Basciftci, Nazife Gozde Ayter Arpacioglu, The classification of wheat species based on deep convolutional neural networks using scanning Electron microscope (SEM) imaging, Eur. Food Res. Technol. 249 (4) (2023) 1023, https://doi.org/10.1007/s00217-022-04192-8, 34.
[6] Surabhi Lingwal, Komal Kumar Bhatia, Manjeet Singh Tomer, Image-based wheat grain classification using convolutional neural network, Multimed. Tool. Appl. 80 (28–29) (2021), https://doi.org/10.1007/s11042-020-10174-3, 35441–65.
[7] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, et al., A comprehensive survey on transfer learning, Proc. IEEE 109 (2020) 43–76.
[8] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis (Eds.), Artificial Neural Networks and Machine Learning—ICANN 2018, Lecture Notes in Computer Science, vol. 11141, Springer, Cham, 2018, pp. 270–279.
[9] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[10] R. Razavi-Far, B. Wang, M.E. Taylor, Q. Yang (Eds.), Federated and Transfer Learning, Springer, 2023.
[11] F. Yu, X. Xiu, Y. Li, A survey on deep transfer learning and beyond, Mathematics 10 (19) (2022) 3619, https://doi.org/10.3390/math10193619.
[12] H. Jin, Q. Song, X. Hu, Auto-Keras: an Efficient Neural Architecture Search System. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1946–1956.
[13] ImageNet dataset. [Online], Available: http://www.image-net.org. (Accessed 21 May 2023).
[14] Keras, Keras Applications, 2023. https://keras.io/api/applications/.
[15] Diederik P. Kingma, Jimmy Ba Adam, A Method for Stochastic Optimization, 2014, https://doi.org/10.48550/ARXIV.1412.6980.
[16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*. Adaptive Computation and Machine Learning, The MIT Press, Cambridge, Massachusetts, 2016.
[17] Richard H.R. Hahnloser, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, H. Sebastian Seung, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, June 22, Nature 405 (6789) (2000) 947, https://doi.org/10.1038/35016072, 51.
[18] Mark "Christopher D. Manning Sanderson, 13 978-0-521-86571-5, Xxi + 482, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008, pp. 100–103, https://doi.org/10.1017/S1351324909005129. Natural Language Engineering 16, no. 1 (January 2010).
[19] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (4) (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.
[20] D.M. Powers, Evaluation: from precision, recall and F-measure to ROC, informedness, markedness, and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.
[21] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.
[22] V. Sze, Y.-H. Chen, T.-J. Yang, J.S. Emer, Efficient processing of deep neural networks: a tutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329, https://doi.org/10.1109/JPROC.2017.2761740.
[23] S. Han, H. Mao, W.J. Dally, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, 2015 arXiv preprint arXiv:1510.00149.
[24] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, in: Proceedings of the 5th International Conference on Learning Representations (ICLR), 2017.
[25] A.C. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht, The marginal value of adaptive gradient methods in machine learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4148–4158.