Indian J. Genet. Plant Breed., (2025); 85(1): 78-86
https://doi.org/10.31742/ISGPB.85.1.9
ISSN: 0975-6906
www.isgpb.org

RESEARCH ARTICLE

Wheat yield prediction through artificial bee colony-enhanced convolutional neural network

M. Lokeshwari, Girish Kumar Jha*, Achal Lama, A. Praveenkumar, Jyoti Kumari[1], K. J. Yashavantha Kumar[2] and Rajender Parsad

Abstract

Crop improvement programs aim to develop high-yielding varieties, coupled with resistance to biotic and abiotic stresses with nutritional superiority. Grain yield, being a complex trait is governed by genotypes, environment, and their interaction. Growing of large number of genotypes under multiple environments and measuring grain yield and its components are tedious and resource-consuming tasks. Therefore, there is a great need for novel, cost-effective techniques to evaluate the performance of crops at the field scale through indirect selection of easily scorable traits using sound algorithms based on comprehensive data. Convolutional neural networks (CNN) are one of the most promising deep learning methods for dealing with several complex tasks including crop yield prediction, but their performance is affected by manually set hyper-parameters. To address this, we proposed the artificial bee colony optimizer to efficiently search the hyper-parameters of CNN models for predicting the wheat yield on the basis of normalized difference vegetation indices, canopy temperature and plant height. Models are developed on crop yield data using 3350 germplasm of wheat planted in two growing environments as well as two different locations during the winter season of 2020-21. When compared to other popular optimization algorithms, such as genetic algorithms and particle swarm optimizers, the proposed model is proven to be superior for predicting wheat yield in terms of root mean square error (RMSE) (66.44–80.68 g/m^2) and R^2 (0.88–0.91) and at the same time greatly reduced the computational time. In addition, crop yield prediction using the proposed model can support different management decisions, including timing and amount of fertilization and selective breeding.

Keywords: Deep learning, hyperparameters optimization, yield prediction

Introduction

In the realm of modern agriculture, the ability to predict crop yield at a field scale stands as a pivotal factor influencing farming practices, resource utilization, and food security measures. Furthermore, non-destructive prediction of crop yield with high accuracy would allow the identification of high-yielding genotypes rapidly and efficiently from a large number of promising genotypes (Bendig et al. 2015; Elsayed et al. 2017). However, predicting crop yield at a field scale is challenging due to the intricate interplay of diverse factors like weather, soil quality, genetics, and farming practices. This complexity arises from the difficulty in capturing spatial variations, temporal changes, and unforeseen events, compounded by limited data and the need to balance model complexity with practicality. Therefore, innovative crop yield prediction models are essential to address these challenges.

Remote sensing has found extensive application in agriculture with different vegetation indices providing a non-destructive, real-time measure of crop growth. The normalized difference vegetation index (NDVI), is one of the most commonly used vegetation indices based on the reflectance of red and near-infrared lights. It can be used to characterize crop growth stages, evaluate crop density, and predict crop yield (Rutkoski et al. 2016). In crops, such as

ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110 012, India.

[1]ICAR-National Bureau of Plant Genetic Resources, New Delhi 110 012, India.

[2]Agharkar Research Institute, Agarkar Road, Shivajinagar 411 004, Pune, Maharashtra, India.

***Corresponding Author:** Girish Kumar Jha, ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110 012, India, E-Mail: girish.stat@gmail.com

How to cite this article: Lokeshwari M., Jha G.K., Lama A., Praveenkumar A., Kumari J., Kumar K.J.Y. and Parsad R. 2025. Wheat yield prediction through artificial bee colony-enhanced convolutional neural network. Indian J. Genet. Plant Breed., **85**(1): 78-86.

Source of support: ICAR-NBPGR-DBT, Govt. of India (Network Project)

Conflict of interest: None.

Received: Feb. 2024 **Revised:** Nov. 2024 **Accepted:** Dec. 2024

maize, wheat, sorghum, and barley, scientists have identified significant correlations between biomass and NDVI with some correlation coefficients above 0.70. The values of NDVI, especially 2 to 3 weeks before and after the heading stage, are highly correlated with grain yield in wheat (Babar et al. 2006). The NDVI information contains dense data and has a non-linear relationship with the spatial crop yield (Taşan et al. 2022). By utilizing NDVI data during the early crop growth stage, we can enhance yield estimation for crop forecasting and effectively identify high-yielding lines in large breeding populations.

Many studies have used machine learning (ML) techniques such as regression trees, random forest, multivariate regression, support vector machine, and artificial neural networks for predicting the crop yield using NDVI (Liu et al. 2001; Nuarsa et al. 2011; Sultana et al. 2014). ML models treat the output, crop yield, as an implicit function of the input variables such as NDVI, weather components and soil conditions, which could be a very complex and non-linear function. Artificial neural networks have a high ability to model nonlinear complex relationships between dependent and independent variables (Cui et al. 2018). Support vector machines (SVMs) have the potential to solve the overfitting problem when using high-dimensional data such as spectral data (Chlingaryan et al. 2018). Spectral indices have been used in ML methods for yield estimation of potatoes (Wolanin et al. 2020), wheat (Pantazi et al. 2016) and cotton (Prasad et al. 2006). Marques Ramos et al. (2020) used five different ML methods with the vegetation indices to estimate maize yield. The random forest model estimated maize yield more accurately than other models. However, few studies have been conducted using the vegetation indices with ML methods to estimate vegetable yield (Wei et al. 2020).

More recently, a deep learning (DL) framework that takes advantage of state-of-the-art modeling and solution techniques is used to predict crop yield based on spectral vegetation indices, environmental data, and management practices. The convolutional neural network (CNN) model is one of the most important DL models for predicting crop yield (Khaki et al. 2020). You et al. (2017) applied CNNs and recurrent neural networks (RNNs) to predict soybean yield based on a sequence of remotely sensed images. Kim et al. (2019) developed a deep neural network model for crop yield prediction using optimized input variables from satellite products and meteorological datasets between 2006 and 2015. Wang et al. (2018) designed a DL framework to predict soybean crop yields in Argentina and they also achieved satisfactory results with a transfer learning approach to predict soybean harvests with a smaller amount of data in Brazil. Yang et al. (2019) investigated the ability of CNN to estimate rice grain yield using remotely sensed images and found that the CNN model provided a robust yield forecast throughout the ripening stage. Khaki et al. (2019) used deep CNNs to predict corn yield loss across 1,560 locations in the United States and Canada. Due to its great learning and expression ability, more and more CNN models have been studied and proposed in recent years.

However, one obstacle encountered when implementing CNNs is configuring the architecture of the CNN model. Therefore, how to design an efficient network configuration for CNN is challenging yet tough work. For example, how many convolution kernels should be used in the convolutions layer and how large should the kernel be? What kind of activation function is better? And how to set the learning rate to make the CNN learn better? These are determined by the hyper-parameters setting of the neural network. At present, most CNNs usually adopt a fixed CNN structure and then set the hyper-parameters based on the historical experiences of network designers (He et al. 2016; Huang et al. 2017). Network designers manually adjust hyper-parameters through trial-and-error experiments, requiring time and effort to obtain the final model architecture. These procedures, based on historical experience and personal preferences, often result in locally optimal models, that are far from globally best hyper-parameter configurations. Therefore, to obtain better CNN architecture, a more efficient way is to regard finding the best CNN hyper-parameters as an optimization problem and then employ powerful algorithms to solve the optimization problem. However, hyper-parameters are interconnected and have different variable types e.g., integer, real-number, or discrete, making them a black box problem with no explicit objective function presentation. As a result, traditional optimization methods, such as Newton's iterative method and conjugate gradient method, struggle to solve the high complexity CNN hyperparameters optimization problem.

Instead of traditional methods, evolutionary computation (EC) intelligent algorithms, such as genetic algorithm (GA) and particle swarm optimization (PSO), have shown promising search ability in finding optimal solutions for complex problems. These algorithms have become increasingly important in recent years for optimizing network hyper-parameters of CNNs, as they avoid difficulties in manual designs and allow for the automatic generation of promising CNN models. However, EC algorithms are based on population search and iterative evolution, which requires further speeding up to reduce long running time. The fitness evaluation of solutions in EC algorithms consumes significant computational time, making the optimization process often lengthy.GA is better suited for optimizing large and complex parametric spaces, but currently available implementations of genetic algorithms are only for discrete search space (Bellot et al. 2018). On the other side, PSO algorithm enables parallelization and fast convergence but needs proper initialization and might stuck in the local optimum (Zahedi et al. 2021).

This study proposes to adapt the artificial bee colony (ABC), a population-based evolutionary heuristic algorithm, for optimizing CNN hyper-parameters in order to overcome the challenges of traditional EC algorithms. Specifically, the study evaluates the performance of the ABC-optimized CNN model using key indicators such as Normalized Difference Vegetation Index (NDVI), canopy temperature (CT), and plant height for wheat yield prediction across different environmental conditions. Besides, the study aims to compare the ABC-optimized CNN model with CNN models optimized using genetic algorithm (GA) and particle swarm optimization (PSO) in terms of prediction accuracy, optimization speed, and reliability.

Materials and methods

Data collection

This study used real-world data to predict the performance of a diverse set of wheat germplasm under different locations as well as environments. Approximately 3350 wheat accessions were grown in an augmented block design (ABD) during the winter season 2020-2021 under two environments *i.e.,* irrigated and rainfed at ICAR-National Bureau of Plant Genetic Resources, Issapur Farm, New Delhi and at Agharkar Research Institute, Pune. The traits measured included grain yield (GY, g/m^2), plant height (PH, cm), canopy temperature (CT, ℃), and normalized difference vegetation indices (NDVI). Plant height was measured as the length from ground level to the apex of the spike excluding awns. Canopy temperature (CT) and NDVI data were collected during the growing seasons at different growth stages from tillering through senescence (ground cover, heading, anthesis, grain filling and maturity)(Zadoks et al. 1974) using a handheld infrared thermometer(IRT) (Fisher Scientific, UK) and Green Seeker (Trimble Navigation Limited, Sunnyvale, CA, USA) respectively (Table 1). The observations were recorded as three readings per plot during the mid-day from 11 a.m. to 2 p.m. corresponding to solar noon on each day of observation. The cloudy day was avoided so that plant canopy could get a maximum light interception. The IRT readings were taken at a 30º angle from the horizon for measurement and NDVI readings were taken at a distance of 70 cm above the crop canopy (Pask, 2012).

Data preprocessing

The NDVI observations had 6.7% missing values for some germplasms, which we imputed using the mean of the same NDVI data of other germplasms. The CT data had 6.3% missing values for some germplasms, which we imputed using the mean of the same CT data of other germplasms at the same location as well as the environment. The plant height did not have any missing values. We tried other imputation techniques such as median and most frequent and found that the mean approach led to the most accurate results. Table 2 shows the summary statistics of the field-measured parameters. Mean wheat yield data for irrigated and rainfed environments in Delhi and Pune locations is 288.17 g/m^2 and the maximum wheat yield is 1182 g/m^2. The result reveals a robust correlation between wheat yield and NDVI, particularly at the heading stage, with a correlation coefficient of 0.98. This indicates an exceptionally strong positive relationship between these variables. The high NDVI during the heading stage appears closely tied to increased wheat yield, underlining the significance of vegetation health at this growth phase. In this dataset, each and every attribute has its own measurement. In order to obtain accurate prediction, the dataset has been rescaled using Eq. (1).

Table 1. Description of the dataset

Variables	Item	Description
Yield	Irrigated wheat yield (g/m^2) Rainfed wheat yield (g/m^2)	Grain yield per plot after harvesting, threshing and cleaning.
NDVI	Normalized Difference Vegetation Indices (NDVI) at five growth stages	NDVI data were collected during ground cover, heading, anthesis, grain filling and maturity stage.
CT	Canopy Temperature (CT) at four growth stages (℃)	CT data were collected during the heading, anthesis, grain filling and maturity stages.
PH	Plant Height(cm)	Measured in cm

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (1)$$

Where X′ is the rescaled value, X is the attributes value, min(X) is the minimum of the attributes value and max (X) is the maximum of the attributes value.

Convolutional neural networks

Convolutional neural networks (CNNs) are proposed to accommodate situations where input variables are distributed along a space pattern, such as one-dimension (Fig. 1) (e.g., NDVIs or text), and two or three-dimensions (e.g., images). CNNs are a special case of neural networks that uses convolution instead of a full matrix multiplication in the hidden layers (Pérez-Enciso and Zingaretti 2019). A typical CNN is made up of dense, fully connected layers and "convolutional layers".

In each convolutional layer, a convolutional operation is performed along the input of predefined width and strides.

Table 2. Summary statistics of the dataset and correlation coefficient (r) with yield

	Mean	Minimum	Maximum	Standard deviation	r
NDVI_Ground cover	0.34	0.08	0.42	0.12	0.828
NDVI_Heading	0.54	0.19	0.72	0.09	0.980
NDVI_Anthesis	0.61	0.39	0.93	0.08	0.950
NDVI_Grain filling	0.75	0.29	0.80	0.09	0.866
NDVI_Maturity	0.41	0.17	0.65	0.12	0.727
CT_Heading°C	21.12	15.4	36.34	2.51	0.566
CT_Anthesis°C	22.45	18	38.56	1.84	0.520
CT_Grain filling°C	25.56	18.2	39.67	2.39	0.444
CT_Maturity°C	31.67	19.1	39.56	1.77	0.214
Plant height (cm)	98	55	168	21.20	0.190
Grain yield (g/m^2)	288.17	15.5	1182	115.25	1.000

Summary statistics is average of all four datasets under different environment and locations

Each of these convolutional operations is called a "kernel" or a "filter" and is somewhat equivalent to a "neuron" in a multi-layer perceptron. An activation function is applied after each convolution to produce the output. Finally, an operation called "pooling" is usually applied to smooth out the result. It consists of merging the kernel outputs of different successive positions by taking the mean, maximum, or minimum of all values of those positions. One of the main advantages of convolutional networks is their capability to reduce the number of parameters to be estimated. These networks also have sparse interactions and are equivariant to translations.

In this study, the deep neural network model of CNN is used for predicting wheat yield based on NDVI, CT and plant height. It is shown in Figure 1. The input layers of this model are NDVI, CT and PH for five growing stages of wheat. Four convolutional layers and one max-pooling layer are set based on expert knowledge and other hyper-parameters are set according to the results of ABC optimization algorithm in Section 3. To prevent overfitting of the training data a dropout layer is added to the network architecture. All input data need to be normalized before being fed into the model and, finally, back normalized for output. The ratio of training data to test data is set to 8:2, where 80% is training data and 20% is test data.

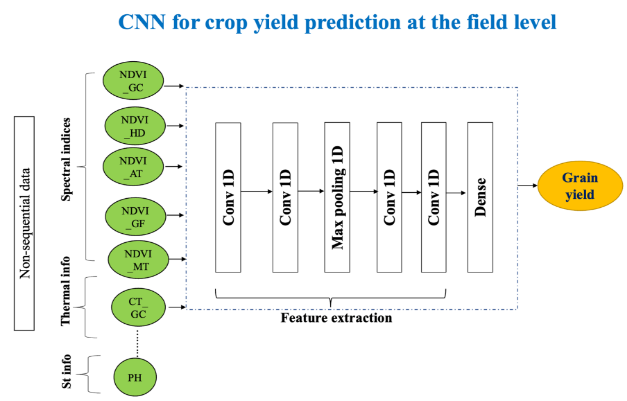

Fig. 1. 1D convolutional neural network for crop yield prediction

Artificial bee colony optimization of CNN hyper-parameters

The process of building CNN network models involves the determination of many hyper-parameters, such as network depth, learning rate, batch size, and so on. The most intuitive way is to find the optimal parameters by manual trial and error, but the manual trial and error method is too inefficient. It lacks a certain exploration process, and the parameters can only be adjusted manually repeatedly for different problems and data. It takes a lot of time, and the final combination of model hyper-parameters may not be optimal, which will affect the prediction of the model, including the degree of network fit and the generalization ability to the test set. Artificial bee colony (ABC) is an optimization algorithm based on honeybees' foraging behavior. Introduced by Dervis Karaboga in 2005, ABC iteratively explores solution spaces, balancing exploration, and exploitation. It's applicable in optimization problems like function optimization and hyper-parameter tuning in ML due to its simplicity and effectiveness in finding global optimal solutions. The main optimized hyper-parameters and the range of values are shown in Table 3.

The number of filters (Depth) defines the depth or number of channels in each convolutional layer. More filters enable the network to learn diverse and hierarchical features, enhancing its representational capacity but increasing computational cost. Filter size (Kernel Size) determines the receptive field size of filters. Larger filters capture broader features, while smaller filters focus on finer details. It directly affects the network's ability to detect different levels of features in the input data. Epoch indicates the number of iterations of the data set during model training. If the number of iterations is set too large, the training time of the model is longer, resulting in overfitting of the model, over-reliance on training data, and poor prediction of unknown data, which makes the generalization ability of the model lower. If the number of iterations is set too small, it will make the model fit poorly and affect the prediction accuracy of the model. The size of the mini-batch to use for each training iteration is indicated, specified as the comma-separated pair consisting of 'Minibatch Size' and a positive integer. A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights. If the number of iterations is set too large, the training time of the model is long, causing the model to be overfitted and overly dependent on the training data. The prediction ability of the unknown data is poor, thus making the generalization ability of the model lower. If the number of iterations is set too small, it will make the model not fit well and affect the prediction accuracy of the model.

The initial learning rate, α, is a relatively important hyperparameter in the CNN model. When the learning rate is too large, it will cause the parameters to be optimized to fluctuate around the minimum value, thus skipping the optimal solution. When the learning rate is set too small, it will affect the convergence speed of the model, resulting in a slow convergence rate. In this paper, α is set to 0.01 based on the empirical value. Dropout means that, during the training process of the model, the network units, are temporarily dropped from the network according to a certain probability. This hyperparameter plays a crucial role in preventing model overfitting and improving the generalization ability of the model. The key hyperparameters searched in this paper are data batch size, number of iterations, discard factor, and number of filters in the convolutional layer. The remaining hyperparameters are based on experience, the optimizer is selected as "adam", Learn rate schedule is set to "piecewise", and the root mean square error is selected as the target loss function.

The optimization of the CNN model hyperparameters using the artificial bee colony algorithm is an eight-step process as follows (Fig. 3, the implementation of the whole process is done in Python):

Step 1: The input data is normalized before being fed into the model, and then the dataset is divided into training, validation, and testing sets, where the training set is used to train the CNN model and optimize its parameters, the validation set is used to evaluate and fine-tune hyperparameters during optimization, and the testing set is reserved for the final stage to assess the model's generalization performance on unseen data.

Step 2: The CNN hyperparameters to be optimized and the range are set; a random set of initialized hyperparameters as the initial hyperparameters of the CNN model are generated. The training set is input for the training of the CNN. The RMSE is used as the objective function for the hyperparameters optimization of the CNN model.

Table 3. CNN parameter setting and range

Hyperparameters	Parameter space (CNN)	Encoding type
Number of filters	[2–128]	Integer
Filter size	[2–20]	Integer
Pooling	MaxPooling 1D	
Dropout rate	[0.1–0.7]	Continuous
Epochs	[100–800]	Integer
MiniBatch size	[8–20]	Integer
Learning rate	0.01	
Objective function	RMSE	

Step 3: Employed bees explore the neighborhood of their current solutions by modifying hyperparameters. It generates a new solution by slightly adjusting one or more hyperparameters while ensuring they remain within predefined bounds.

Step 4: Onlooker bees evaluate solutions based on their RMSE values on the validation set. Bees with lower RMSE attract more onlooker bees, who then explore similar regions of the search space.

Step 5: Scout Bees Phase introduces randomness by replacing solutions that have not improved for a certain number of iterations. It explores entirely new hyperparameter configurations.

Step 6: Repeat the employed, onlooker, and scout bee phases for multiple iterations. Allow the algorithm to explore and exploit the search space, aiming to converge towards optimal or near-optimal hyperparameters.

Step 7: The maximum number of iterations (40) is completed and the minimum objective function and the corresponding trained CNN model hyper-parameters are returned.

Step 8: The testing set, which has remained unseen during training and validation, is now fed into the final trained model to construct a CNN wheat yield prediction model based on the ABC algorithm.

The general framework of this study is shown in Figure 3. Based on the indicators required for wheat yield calculation, the influence of different input variables on wheat yield prediction was analyzed on the ABC-CNN model. Then the performance of other popular optimization algorithms i.e., GA and PSO optimized CNN for wheat yield prediction was compared; finally, the robustness of ABC-CNN is explored. All models used were run in a Python environment.

Model performance evaluation

In this study, the coefficient of determination (R^2), root mean square error (RMSE), and mean absolute percentage error (MAPE) were used as indicators to assess the model performance. The equations are written as follows:

$$R^2 = \left[\frac{\sum_{t=1}^{n}(o_t - \bar{o})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{n}(o_t - \bar{o})^2(y_{i-} - \bar{y})^2}}\right]^2$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(o_t - y_t)^2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_t - o_t}{y_t}\right| * 100$$

Where, n is the number of samples, and y_j and o_i are the measured and predicted values of winter wheat yield, respectively. R^2 is a measure of the strength of the linear relationship between the predicted and measured values of the model, with larger R^2 indicating that the measured and predicted values have similar trends. RMSE is used to assess the deviation between measured and predicted values. MAPE measures error as a percentage of actual values that deviate from predicted ones. The smaller the value, the smaller the deviation is between the measured and predicted values of the model; the higher R^2, the smaller RMSE and MAPE values are and the better the model.

Results and discussion

The proposed model is implemented in Python using the TensorFlow open-source software library. For comparative analysis, we also implemented genetic algorithm (GA) and particle swarm optimization (PSO) as benchmark optimization methods. In the Artificial Bee Colony (ABC) algorithm, the number of employed bees and onlooker bees is set to 10 each, and the number of iterations is fixed at 40. For the genetic algorithm (GA), the population size (NP) is set to 10, the maximum number of generations (Gmax) is 10, the crossover probability (CR) is 0.3, and the mutation rate (MR) is 0.15. For the particle swarm optimization (PSO) algorithm, the number of particles is set to 10, with parameter values defined as $c_1 = 2$, $c_2 = 2$, and inertia weight (ω) = 0.7, while the number of iterations is 10. The number of function evaluations in the ABC algorithm is twice per iteration (once in the employed bee phase and once in the onlooker bee phase). However, in the standard GA and PSO implementations, function evaluations occur only once per iteration. To ensure a fair comparison, we modified the GA and PSO algorithms to perform two function evaluations per iteration, aligning them with the ABC method.

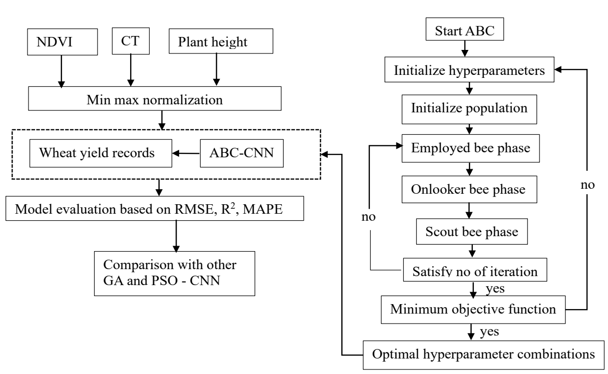

Fig. 3. A research framework of the ABC-CNN model for wheat yield prediction

Table 4 compares the performance of the three models on the basis of time efficiency, "time" and the accuracy, RMSE, of the model after tuning the hyperparameters. These results suggest that our proposed approach outperformed the other models to varying extents. The weak performance of PSO is mainly due to its susceptibility to local optima trapping within complex and high-dimensional search spaces, hindering adequate exploration. GA outperformed PSO since it has the ability to explore diverse solutions within complex search space, leveraging mechanisms like crossover and mutation to introduce variations among solutions. ABC outperformed the other two with minimum RMSE (68.87 g/plot) and maximum R^2 (0.91). ABC showcases efficiency in tuning hyperparameters due to its simple implementation, and effective balance between exploring diverse hyperparameter configurations and exploiting promising ones. Its ability to navigate the hyperparameters space efficiently and robustly avoids local optima, making it a versatile choice across different models and architectures. The hyperparameters optimization was performed for all three models across four datasets, and the best combination was selected for training. The optimized hyperparameters for the Delhi irrigated dataset are illustrated in Table 4.

The best estimates of wheat yield obtained from different optimized CNN models are compared with corresponding observed values using scatterplots (Fig. 4). As expected from the comparable R^2 and RMSE values, the distribution pattern of predicted versus the observed grain yield of different optimization method is very similar. Furthermore, the amount of overfitting for wheat yield prediction using ABC-CNN is much less compared to other optimization methods. Figure 4**,** suggests that the proposed model is more effective in utilizing information and less prone to local optima. We plotted the probability density functions of the ground truth yield and the predicted yield by the ABC-CNN model to see if the proposed model can preserve the distributional properties of the ground truth yield. As shown in Fig. 5, the ABC-CNN model can approximately preserve the distributional properties of the ground truth yield. However, the variance of the predicted yield is less than the variance of the ground truth yield, which indicates the ABC-CNN model's prediction is more centralized around the mean.

Table 4. Results of optimized CNN hyperparameters on the Delhi irrigated dataset

Model	ABC-CNN	GA-CNN	PSO-CNN
Number of filters	7	9	12
Filter size	2	3	2
Epochs	350	150	410
Minibatch	15	13	9
Tim/min	867	1001	1098
RMSE (g/plot)	68.87	88.97	91.47
R^2	0.91	0.86	0.77
MAPE (%)	121.78	143.87	167.43

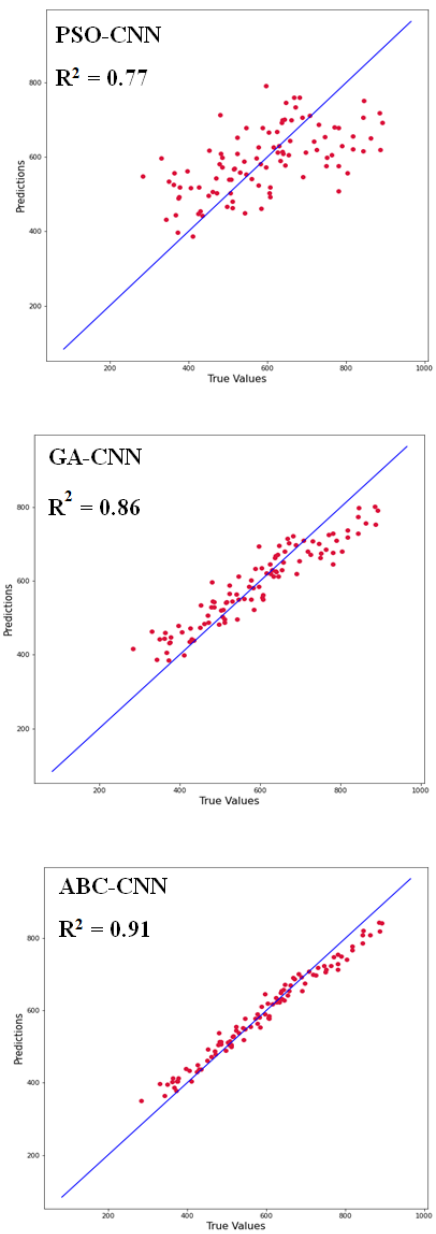

Fig. 4. Scatterplots of the observed yield and predicted yield

Generalization power of ABC-CNN model

To examine the power of the model, we trained the ABC – CNN model on four different datasets which includes grain yield data using 3350 germplasm of wheat planted in two growing environments (irrigated and rainfed) as well as two different locations (Delhi and Pune). Table 5 shows the performance of the proposed model for wheat yield predictions at different environments. As shown in Table 5**,** the prediction accuracy of the ABC-CNN model emphasizes the significance of achieving consistent and high performance across untested datasets. A model's consistent performance across diverse datasets indicates its robustness against data variations, noises, and biases. It demonstrates effective generalization, adaptability, and accuracy in real-world scenarios, reducing concerns about overfitting and ensuring reliability with new, unseen data. Fig. 6 represents a comparison of R^2 and RMSE values for different plant traits (NDVI, CT and Plant Height) on the Delhi irrigated dataset. NDVI exhibits the highest R^2, followed by CT, and plant height, suggesting that the predictive model performs best for NDVI and least effectively for plant height. A lower RMSE corresponds to better model accuracy. While NDVI and CT have moderate RMSE values, Plant height exhibits the highest, indicating substantial prediction error. The results highlight that the model performs well for NDVI and CT, whereas the prediction of plant height remains a challenge, requiring further model optimization or additional explanatory variables.

Important comparison between input variables

To compare the importance of NDVI, CT and plant height individually in the yield prediction, we employed the ABC-CNN model to capture the linear and nonlinear effects of individual components. Fig. 6 shows the yield prediction performance of the ABC-CNN model with three different input variables. We found that yield estimates using NDVI ($R^2 = 0.77$, RMSE = 75.67 g/m^2) alone are more accurate than those using CT ($R^2 = 0.56$, RMSE = 98.06 g/m^2) alone. Because NDVI helps track the density and health of the developing plants, ensuring they progress uniformly and detecting stress factors that might hinder growth or yield potential (Nuarsa et al. 2011).

This study proposed the ABC-CNN model for wheat yield prediction using NDVI, CT, and plant height, outperforming

Table 5. Performance of ABC-CNN model on different validation dataset

Datasets	RMSE	R^2	MAPE
Delhi irrigated	68.87	0.91	121.78
Delhi rainfed	69.76	0.89	128.71
Pune irrigated	70.23	0.87	136.98
Pune rainfed	68.90	0.88	130.05

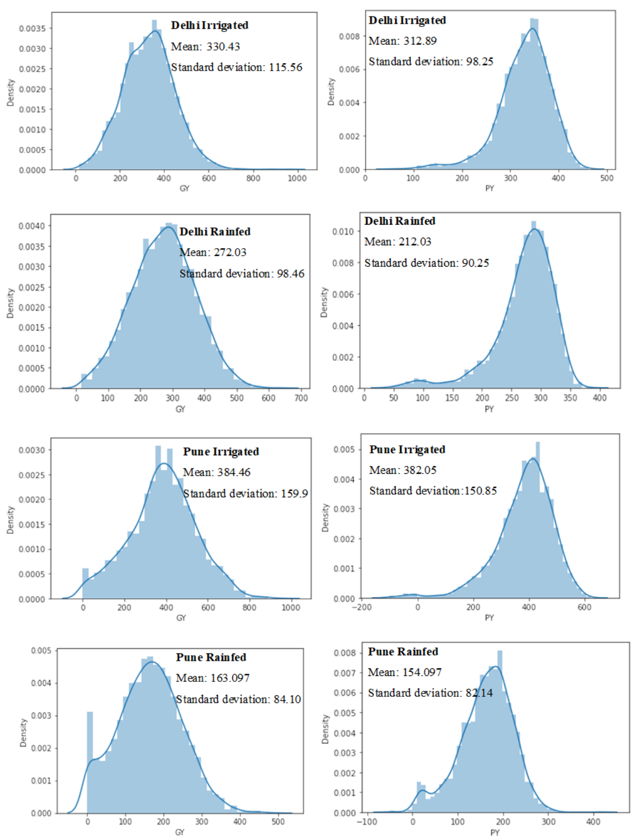

Fig. 5. Probability density function of observed grain yield and predicted yieldby ABC-CNN model

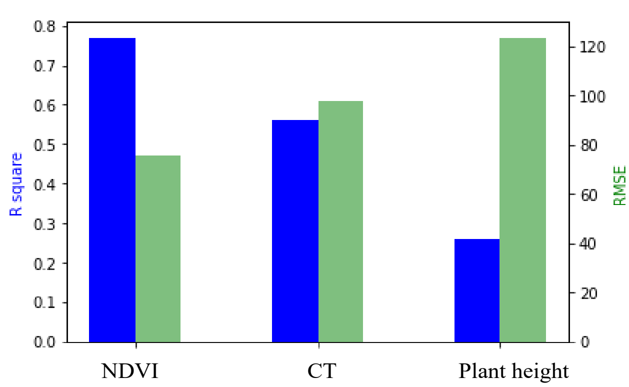

Fig. 6. Yield prediction performance of ABC-CNN for individual input variable of Delhi irrigated dataset

GA-CNN and PSO-CNN in accuracy. NDVI had the most significant impact on yield prediction, making it a crucial trait for large-scale phenotyping. Future work will explore parameter sensitivity and develop a package or web interface for ABC-optimized CNN in crop yield prediction.

Authors' contribution

Conceptualization of research (ML, GKJ); Designing of the experiments (ML, GKJ, RP); Contribution of experimental materials (JK, ML); Execution of field/lab experiments and data collection (JK, KJYK, ML); Analysis of data and interpretation (ML, GKJ, AL, AP); Preparation of the manuscript (ML, GKJ, AP).

Acknowledgments

The first author acknowledges the Post-Graduate School, ICAR-Indian Agricultural Research Institute, New Delhi for providing the necessary facilities for carrying out this research. The paper uses data generated under ICAR-NBPGR-DBT Network Project.

References

Babar M. A., Reynolds M. P., van Ginkel M., Klatt A. R., Raun W. R. and Stone M. L. 2006. Spectral Reflectance Indices as a Potential Indirect Selection Criteria for Wheat Yield under Irrigation. Crop Sci., **46**(2): 578–588. https://doi.org/10.2135/cropsci2005.0059

Bellot P., de los Campos G. and Pérez-Enciso M. 2018. Can Deep Learning Improve Genomic Prediction of Complex Human Traits? Genet., **210**(3): 809–819. https://doi.org/10.1534/genetics.118.301298

Bendig J., Yu K., Aasen H., Bolten A., Bennertz S., Broscheit J., Gnyp M. L. and Bareth G. 2015. Combining UAV-based plant height from crop surface models, visible, and near infrared vegetation indices for biomass monitoring in barley. Int, J. Appl. Earth Obs. Geoinf., **93**: 79–87. https://doi.org/10.1016/j.jag.2015.02.012

Chlingaryan A., Sukkarieh S. and Whelan B. 2018. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. ComputElectronAgric., **151**: 61–69. https://doi.org/10.1016/j.compag.2018.05.012

Cui Z., Zhang H., Chen X., Zhang C., Ma W., Huang C., Zhang W., Mi G., Miao Y., Li X., Gao Q., Yang J., Wang Z., Ye Y., Guo S., Lu J., Huang J., Lv S., Sun Y. and Dou Z. 2018. Pursuing sustainable productivity with millions of smallholder farmers. Nat., **555**(7696): 363–366. https://doi.org/10.1038/nature25785

Elsayed S., Elhoweity M., Ibrahim H. H., Dewir Y. H., Migdadi H. M. and Schmidhalter U. 2017. Thermal imaging and passive reflectance sensing to estimate the water status and grain yield of wheat under different irrigation regimes. Agric. Water Manag., **189**: 98–110. https://doi.org/10.1016/j.agwat.2017.05.001

He K., Zhang X., Ren S. and Sun J. 2016. Deep Residual Learning for Image Recognition. IEEE conf. CVPR, 770–778. https://doi.org/10.1109/CVPR.2016.90

Huang G., Liu Z., Van Der Maaten L. and Weinberger K. Q. 2017. Densely Connected Convolutional Networks. IEEE conf. CVPR, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Liu J., Goering C. E. and Tian L. 2001. A neural network for setting target corn yields. TransASAE., **44**(3). https://doi.org/10.13031/2013.6097

Khaki S., Khalilzadeh Z. and Wang L. 2019. Classification of Crop Tolerance to Heat and Drought-A Deep Convolutional Neural Networks Approach. Agron., **9**(12): 833. https://doi.org/10.3390/agronomy9120833

Khaki S., Wang L. and Archontoulis S. V. 2020. A CNN-RNN Framework for Crop Yield Prediction. Front. Plant Sci., **10**: 1–14. https://doi.org/10.3389/fpls.2019.01750

Kim N., Ha K.J., Park N.W., Cho J., Hong S. and Lee Y.W. 2019. A Comparison Between Major Artificial Intelligence Models for Crop Yield Prediction: Case Study of the Midwestern United States, 2006–2015. ISPRS Int. J. Geoinf., **8**(5): 240. https://doi.org/10.3390/ijgi8050240

Marques Ramos A. P., Prado Osco L., Elis Garcia Furuya D., Nunes Gonçalves W., Cordeiro Santana D., Pereira Ribeiro Teodoro L., Antonio da Silva Junior C., Fernando Capristo-Silva G., Li J., Henrique Rojo Baio F., Marcato Junior J., Eduardo Teodoro P. and Pistori H. 2020. A random forest ranking approach to predict yield in maize with uav-based vegetation spectral indices. Comput Electron Agric., **178**: https://doi.org/10.1016/j.compag.2020.105791

Nuarsa I. W., Nishio F., Nishio F., Hongo C. and Hongo C. 2011. Relationship between Rice Spectral and Rice Yield Using Modis Data. J. Agric. Sci., **3**(2): https://doi.org/10.5539/jas.v3n2p80

Pantazi X. E., Moshou D., Alexandridis T., Whetton R. L. and Mouazen A. M. 2016. Wheat yield prediction using machine learning and advanced sensing techniques. Comput.Electron. Agric., **121**: 57–65. https://doi.org/10.1016/j.compag.2015.11.018

Pask A. 2012. Physiological breeding. II, A field guide to wheat phenotyping.

Pérez-Enciso and Zingaretti. 2019. A Guide for Using Deep Learning for Complex Trait Genomic Prediction. Genes, **10**(7): 553. https://doi.org/10.3390/genes10070553

Prasad A. K., Chai L., Singh R. P. and Kafatos M. 2006. Crop yield estimation model for Iowa using remote sensing and surface parameters. Int. J. Appl. Earth Obs. Geoinf., **8**(1): 26–33. https://doi.org/10.1016/j.jag.2005.06.002

Rutkoski J., Poland J., Mondal S., Autrique E., Pérez L. G., Crossa J., Reynolds M. and Singh R. 2016. Canopy Temperature and Vegetation Indices from High-Throughput Phenotyping Improve Accuracy of Pedigree and Genomic Selection for Grain Yield in Wheat. G3 Genes|Genomes|Genetics, **6**(9): 2799–2808. https://doi.org/10.1534/g3.116.032888

Sultana S. R., Ali A., Ahmad A., Mubeen M., Zia-Ul-Haq M., Ahmad S., Ercisli S. and Jaafar H. Z. E. 2014. Normalized Difference Vegetation Index as a Tool for Wheat Yield Estimation: A Case Study from Faisalabad, Pakistan. Sci. World J., 1–8. https://doi.org/10.1155/2014/725326

Taşan S., Cemek B., Taşan M. and Cantürk A. 2022. Estimation of eggplant yield with machine learning methods using spectral vegetation indices Comput. Electron Agric., **202**: 107367. https://doi.org/10.1016/J.COMPAG.2022.107367

Wang A. X., Tran C., Desai N., Lobell D. and Ermon S. 2018. Deep Transfer Learning for Crop Yield Prediction with Remote Sensing Data. Proc. Conf. Comput. Sustain. Soc., 1–5. https://doi.org/10.1145/3209811.3212707

Wei, M. C. F., Maldaner, L. F., Ottoni, P. M. N., and Molin, J. P. 2020. Carrot Yield Mapping: A Precision Agriculture Approach Based on Machine Learning. Artif Intell., 1(2): 229–241. https://doi.org/10.3390/ai1020015

Wolanin A., Mateo-García G., Camps-Valls G., Gómez-Chova L., Meroni M., Duveiller G., Liangzhi Y. and Guanter L. 2020. Estimating and understanding crop yields with explainable deep learning in the Indian Wheat Belt. Environ. Res. Lett., **15**(2): 024019. https://doi.org/10.1088/1748-9326/ab68ac

Yang Q., Shi L., Han J., Zha Y. and Zhu P. 2019. Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images. Field Crops Res., **235**: 142–153. https://doi.org/10.1016/j.fcr.2019.02.022

You, J., Li, X., Low, M., Lobell, D., and Ermon, S. 2017. Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data. AAAI ConfArtif Intell., 30(1).

Zadoks J. C., Chang T. T. and Konzak C. F. 1974. A decimal code for the growth stages of cereals. Weed Res., **14**(6): 415–421. https://doi.org/10.1111/j.1365-3180.1974.tb01084.x

Zahedi L., Mohammadi F. G. and Amini M. H. 2021. HyP-ABC: A Novel Automated Hyper-Parameter Tuning Algorithm Using Evolutionary Optimization. JIEEE Trans Artif Intell.http://arxiv.org/abs/2109.05319.