CHAPTER ONE

The role of artificial intelligence in crop improvement

Karlene L. Negus[a,b], Xianran Li[c], Stephen M. Welch[d], and Jianming Yu[a,b,*]

[a]Department of Agronomy, Iowa State University, Ames, IA, United States
[b]Interdepartmental Genetics and Genomics, Iowa State University, Ames, IA, United States
[c]Wheat Health, Genetics & Quality Research, USDA-ARS, Pullman, WA, United States
[d]Department of Agronomy, Kansas State University, Manhattan, KS, United States
[*]Corresponding author: e-mail address: jmyu@iastate.edu

Contents

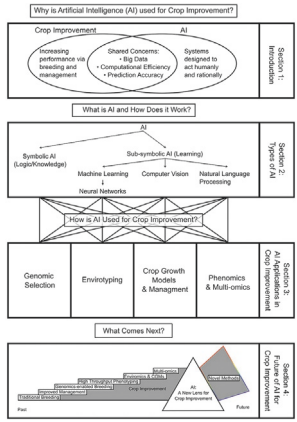
Abstract

The growing global demands for agricultural goods will require accelerated crop improvement. High-throughput genomic, phenomic, enviromic and other multi-omic data collection methods have largely satisfied data acquisition bottlenecks that previously existed within crop breeding and management. Fully capitalizing on large, high-dimensional datasets has now evolved as a new challenge. Artificial intelligence (AI) is currently the foremost solution. Types of AI with the capacity to learn (machine learning), such as neural networks, can better facilitate the translation of data into useful predictions by bypassing the limitations of human expert-driven learning. The potential for applying AI to major crop improvement methods has already been demonstrated

Advances in Agronomy, Volume 184
ISSN 0065-2113
https://doi.org/10.1016/bs.agron.2023.11.001

with preliminary successes shown using deep learning for genomic selection, feature selection for enviromics, ensembles and knowledge-based AI for crop growth modeling, computer vision and convolutional neural networks for phenomics, and unsupervised machine learning for multi-omics. Other types of neural networks including transformer, recurrent, encoding decoding, and generative networks as well as symbolic (non-learning) AI such as robotic process automation, expert systems, and inductive logic programming are also reviewed to contextualize the rapidly changing AI field. Overall, AI has shown strong potential to leverage data for a variety of crop improvement tasks.

Graphical abstract



1. Introduction

1.1 Crop improvement

Producing agricultural goods at rates that meet the demands of growing world populations has been and will continue to be a vital issue. Challenges to be faced in the coming years include not only increasing yields but doing so despite decreases in quantity and quality of agricultural lands and resources while also tolerating diverse and changing environmental conditions worldwide. Equipping food production systems to grow with a global population expected to reach 10 billion by 2050 while retaining the capacity to handle these current and future challenges will require innovative developments within crop improvement approaches.

Innovative methods have been previously adopted to address challenges in crop breeding and management. In the transition away from traditional

crop breeding systems that required labor intensive phenotyping to drive selection in repetitious breeding cycles, plant breeding has taken advantage of growing sources of -omic data including genomic, phenomic, and enviromic data, among others. Next generation sequencing (NGS) and reductions in genotyping costs have driven the integration of genomics into breeding. First-generation genomics-enabled breeding strategies such as marker assisted selection (MAS) which utilized GWAS and QTL study results demonstrated increased gains were possible with genomics-aware breeding. Subsequent expansion and development of additional strategies, like genomic selection, showed further success could be achieved with methods better suited to large-scale genetic data.

Improved strategies have also been concurrently developed in other crop improvement domains. Major advances are being made in the areas of (1) high throughput phenotyping (HTP) to capture data and extract information from plants; (2) envirotyping to obtaine comprehensive meta-data associated with the experiments and production fields with geographic information systems (GIS) and remote sensing; (3) companion -omics technologies that quantifying the gene transcripts, metabolites, proteins, and other molecules; (4) systematic design, application, and data collecting of management practices and condition monitoring; and (5) scalable ecophysiological crop growth models that integrate inputs for genetics, environment, and management to generate crop performance through process-based modeling.

Many prior issues facing crop improvement research revolved around data accumulation. Those bottlenecks have been largely overcome with higher levels of automation achieved through approaches like NGS technologies, HTP, GIS, and remote sensing. In this emerging era of high throughput technologies, terabytes of data can be generated every growing season. Modern crop improvement now has the advantage and challenge of accessing large multi-omic data sets to inform breeding and management decisions. Novel, efficient, and effective analytical methods will be needed to continue bridging the gap between genotypes and phenotypes. Artificial intelligence (AI) has the potential to provide diverse solutions to these ends.

1.2 Artificial intelligence

AI research and applications have evolved since the first instance of a neural network in 1943 (Mcculloch et al., 1943) and the subsequent establishment of the broader AI field in the 1950s. There have been several AI "seasons"

where interest and efforts have risen, known as AI summers, and fallen, known as AI winters (Ilkou and Koutraki, 2020). The first AI summer spanned the years following the field's initiation through the mid-1960s (Kautz, 2022). This era was defined by knowledge representation, formal logic, and heuristics. Interest waned as it became more obvious that making computers learn (machine learning) was still a far-off prospect (Crevier, 1993). A shifting of the dominant perspective from general understanding to expert knowledge underlined the activity of the first winter and success in knowledge representation initiated a second AI summer that extended across the decade of 1980s (Crevier, 1993; Kautz, 2022). During the second AI winter (1988–2011), despite diminished public interest, critical advances were still made. Approaches to efficient probabilistic reasoning were achieved and machine learning was revitalized. These efforts also sought to overcome the knowledge acquisition bottleneck experience in previous seasons (Kautz, 2022). The third AI summer is currently underway and deep learning systems have defined the early years of this era. Reasoning and learning techniques have also become more ubiquitous in this most recent season with new and old types being developed and revisited.

2. Types of AI

AI is concerned with the process of designing computers that can think and act humanly and rationally (Russell and Norvig, 2009). In recent years, AI has been increasingly explored as a means to analyze big data, most popularly through machine learning (ML) approaches. However in addition to ML, AI encompasses a number of diverse sub-fields that can be generally categorized as symbolic or sub-symbolic (Ilkou and Koutraki, 2020; Nilsson, 1998). Symbolic AI (Section 2.1) will herein include robotic process automation (Section 2.1.1), expert and fuzzy systems (Section 2.1.2), and inductive logic programming (Section 2.1.3). While sub-symbolic AI sections will focus on machine learning (Section 2.2 for non-neural network machine learning and Section 2.3 for neural network machine learning), computer vision (Section 2.4), and natural language processing (Section 2.5).

2.1 Symbolic AI

AI approaches that represent knowledge through symbols (Hoehndorf and Queralt-Rosinach, 2017) are described using names like symbolic AI, logical AI, or computationalism (Domingos et al., 2016; Hoehndorf and Queralt-Rosinach, 2017). Symbolic AI applies logical operations to declarative

knowledge represented by symbols. The goal of symbolic AI is to deduce the consequences of the supplied knowledge through symbols and symbol manipulation (Hoehndorf and Queralt-Rosinach, 2017; Nilsson, 1998). For this reason, many symbolic systems are often also categorized as rule-based or logic-based systems. First order logic rules, like "if… then…" statements, are a common form for symbols and highly explainable since this type of computational reasoning mimics human cognition (Hoehndorf and Queralt-Rosinach, 2017; Ilkou and Koutraki, 2020; Nilsson, 1998). However, rules in symbolic systems must be hard coded. Because of this, a functional symbolic system must anticipate all situations and the appropriate corresponding actions (in the form of rules) that lead from input to conclusion (Jordan and Mitchell, 2015). The rigidity of these systems and the extensive prior knowledge needed to define rules are major drawbacks that limit the application of symbolic AI systems in the modern age (Hayes-Roth, 1985; Ilkou and Koutraki, 2020; Nilsson, 1998). But as sub-symbolic systems increase in complexity, symbolic algorithms have been revisited for use as components within a larger sub-symbolic system (d'Avila Garcez et al., 2002). For that reason, revisiting applications of the most popular types of symbolic systems is still meaningful.

2.1.1 Robotic process automation

Robotic process automation (RPA) is the design of an AI agent that is capable of interacting with repetitive processes that have enough variability to prevent the use of standard process automation. Tasks addressed by RPA would otherwise require human interaction through an appropriate interface to complete. An advantage of RPA is that instead of redesigning a system capable of automation, an AI agent is designed to replace the human user in the existing system (van der Aalst et al., 2018).

RPA can mimic actions taken by humans in a point and click interface by deducing the underlying rule/action from input-response pairs. The ways an AI agent in an RPA system comes to know what actions to take can vary. In a system that doesn't learn, intermediate actions must be specified by humans during human interface training. This type of RPA has poor transferability between applications since all steps must be explicitly defined. Some more recent implementations of RPA systems can learn from examples (input-response pairs) through supervised learning (Section 2.2). These systems are more flexible since intermediate actions are learned by the system. RPAs have also begun to use reinforcement learning to eliminate the need for human interfaced training (Chakraborti et al., 2020). These types of

intelligent process automation can be implemented for more complex tasks involving decision making and analysis.

2.1.2 Expert systems/fuzzy systems

Expert systems attempt to replicate the problem-solving capabilities of human domain experts (Jackson, 1998). The components of an expert system include a knowledge base and an inference engine. The knowledge base contains knowledge described by domain experts and represented in the system using formal language. The inference engine then interprets the provided information to solve problems (Luconi et al., 1986). Expert system development can be broken down into four steps: knowledge acquisition, knowledge representation, knowledge utilization, and reasoning explanation (Jackson, 1998).

Where expert systems excel is interpretability. Since knowledge is represented (symbolized) using human-centric syntax rather than syntax optimized for computation, the inference engine also reasons via interpretable syntax. It becomes trivial to then provide a human-interpretable explanation of how the problem was solved. For this reason, there is increased decision acceptance by users despite expert systems sometimes being less accurate than black-box types of AI.

A drawback of knowledge-based expert systems is the knowledge acquisition bottleneck, both in terms of knowledge quantity and, perhaps more importantly, knowledge quality (Kautz, 2022). Expert systems are only as good as the knowledge they contain. For simple problems, limited but accurate knowledge can be sufficient. But for complex problems, encoding sufficient knowledge while maintaining knowledge relevancy becomes less feasible as problems grow more complex. Computational efficiency then also becomes a concern.

Expert systems can vary in terms of the knowledge structure. Rules-based expert systems have knowledge-bases consisting of formalized situation-action rules, known as production rules. Frame-based expert systems, like decision trees (Section 2.2.1), also integrate the relationship between pieces of knowledge as "meta-knowledge" to provide an explicit structure to the knowledge base (Jackson, 1998).

All symbolic systems employ logic to form conclusions. Propositional logic, first-order logic, inductive logic, and some expert systems can utilize fuzzy logic to deal with imprecision of knowledge. Fuzzy logic is an extension of multi-valued logic (Zadeh, 1988). In a two-valued system, rules are true or false. In a multi-valued system, rules can also be partially true or

partially false. Quantifiers for multi-valued logic are *all true* or *some true* (Zadeh, 1988), but degrees of partial true are indistinguishable. Fuzzy logic can deal with degrees of approximated reasoning in ways multi-valued logic cannot. Fuzzy truth-quantifiers like *more or less true*, *rather true*, *not very true*, etc. can distinguish between continuous approximations of partial truth. To be clear, fuzzy logic is not synonymous with probabilities. Fuzzy logic describes an event with imprecision while probabilities describe likelihood of an event occurring. Approximate reasoning can also use imprecise truth tables or approximated inference rules for the same purpose (Zadeh, 1975). In fuzzy systems, the functions that transform input into the appropriate response may overlap creating instances where multiple rules are true. In this situation, fuzzy approximation can be employed to determine the reasoning or function applied to the input.

Let us illustrate this with a very simple example of classifying plant height. Under this expert system, a plant can be considered *tall, normal,* or *short* in height. Rules used to represent these conditions might be "if height = [2 m, 13 m], then plant = *tall*"; "if height = [0 m, 1.2 m], then plant = *short*"; and "if height = [1 m, 3 m], then plant = *normal*". From 2 to 3 m, plants could be considered both *normal* and *tall* in height. But plants at 2.1 and 2.9 m are not equally *somewhat tall*. A fuzzy approximation could be used to classify a 2.1 m plant as *not very tall* and a plant at 2.9 m as *more or less tall*.

Decision rules are often interpreted directly from expert knowledge, but some expert systems also derived rules from declarative knowledge via induction in a manner similar to inductive logic programming (Section 2.1.3).

2.1.3 Inductive logic programming

Induction describes the process of relating knowledge through declarative rules. Many symbolic systems that utilize knowledge rules are drawing from inductions already made by humans. A set of self-synthesized rules derived from background knowledge is the basis for a logic program. Predictions of new observations can be made using the rules within a logic program (Cropper and Dumančić, 2022). The background knowledge in inductive logic programming (ILP) is populated with atoms that classify data terms using symbols. Rules are synthesized from the background knowledge by formulating a declarative statement that maximizes the true atoms and minimizes the false atoms represented in the statement (Cropper and Dumančić, 2022).

For example, we may have a small dataset of plant characteristics (Table 1) from which we want to make some predictions. Because ILP

Table 1 Example of data used to synthesize rules in inductive logic programming.

Replicate	Color	Height	Condition
A	Green	Tall	Healthy
B	Brown	Short	Diseased
C	Brown	Tall	Diseased
D	Green	Short	Diseased

generally uses closed world assumptions (Cropper and Dumančić, 2022), we only need to declare one atom per class, and undeclared atoms are accepted as false. True atoms are considered positive symbols, and false atoms are considered negative symbols. We may pick "green," "tall," and "healthy" as the positive atoms and represent the knowledge base as including "green(A), green(D), tall(A), tall(C), and healthy(A)". An inverse knowledge base defining "brown," "short," and "diseased" could also be used and is essentially equivalent in a two class system. Because in the defined knowledge base, the atom "brown(B)" is equivalent to "not_green(B)" or "green(B) = FALSE". These pieces of knowledge are assumed if "green(B)" is absent from the knowledge base and the same is true for the other undeclared atoms.

Perhaps we want to predict "condition" from the future data. Using the knowledge related to color to predict the healthy condition would yield a rule that represents a false atom since replicate D is both green and diseased. Using knowledge related to height would also yield a rule that represents a false atom. The rule that maximizes true atoms and minimizes false atoms for this example would be: For replicate x, if x is green and tall, then x is healthy (i.e. Green(x) and Tall(x) = Healthy(x)). ILP is used to synthesize this or other rules from the dataset/knowledge base without human inference.

Machine learning (Section 2.2) can be used in conjunction with ILP to synthesize rules via learning. An advantage of ILP is data efficiency, because rules can be induced from small datasets. Expert knowledge (rules formulated by humans outside of the scope of a given dataset) can also be easily integrated alongside the rules synthesized by an ILP system (Cropper and Dumančić, 2022). The syntax of ILP makes transfer learning possible since induced rules use notation identical to the rules in other symbolic systems (Lin et al., 2014). The capacity of ILP for knowledge transfer is an advantage over other ML algorithms that do not reuse knowledge between tasks (Cropper and Dumančić, 2022). The syntax of ILP is also an advantage

in terms of its similarity to natural language and makes induced rules readable by humans.

While ILP can work with small amounts of data, the inclusion of ineffective or mistaken background data can limit the capability of an ILP. Insufficient background knowledge may result in the exclusion of a plausible rule, while too much background knowledge may introduce irrelevant knowledge that adversely affects performance. ILP has historically required user-curated background knowledge that limits ILP in comparison to ML systems (Section 2.2) more robust to data inconsistencies.

Among the types of symbolic AI, ILP is the most likely to be considered a type of AI that learns and according to some definitions may be a primitive type of ML. However, ILP still differs in that it learns relations between knowledge/data and can only induce general hypotheses from specific knowledge, which can be complicated by quantity and quality of background knowledge. Sub-symbolic types of AI, by contrast, learn functions rather than rules (Cropper and Dumančić, 2022).

2.2 Machine learning

Sub-symbolic AI shifts the focus from symbols and symbol manipulation toward less interpretable patterns in the form of mathematical optimizations, statistical classifiers, and neural networks (Nilsson, 1998). These functions are established by the system to link input and response variables (Ilkou and Koutraki, 2020). Sub-symbolic AI, also called connectionist AI, includes all systems that can learn (Ilkou and Koutraki, 2020; Nilsson, 1998). A system that can improve through experience, or learn, yields flexibility lacking in a symbolic system (Liakos et al., 2018; Libbrecht and Noble, 2015; Russell and Norvig, 2009).

At the core of modern sub-symbolic AI is machine learning (ML). ML exists as a broad subfield and can include nearly any method that makes predictions (James et al., 2013). Because of the inclusivity of this definition, many statistical models can also be considered ML. In the extreme case, the difference between statistical and ML modeling is often characterized as inference being the primary focus of statistical modeling while prediction is the primary objective of ML modeling (Bzdok et al., 2018). Realistically in many use cases, both prediction and inference are often relevant (Breiman, 2001a). However, a trend may be noticed in Fig. 1 that most statistical modeling methods are common to ML modeling methods, but ML includes a number of additional methods with low interpretability. This may reflect a

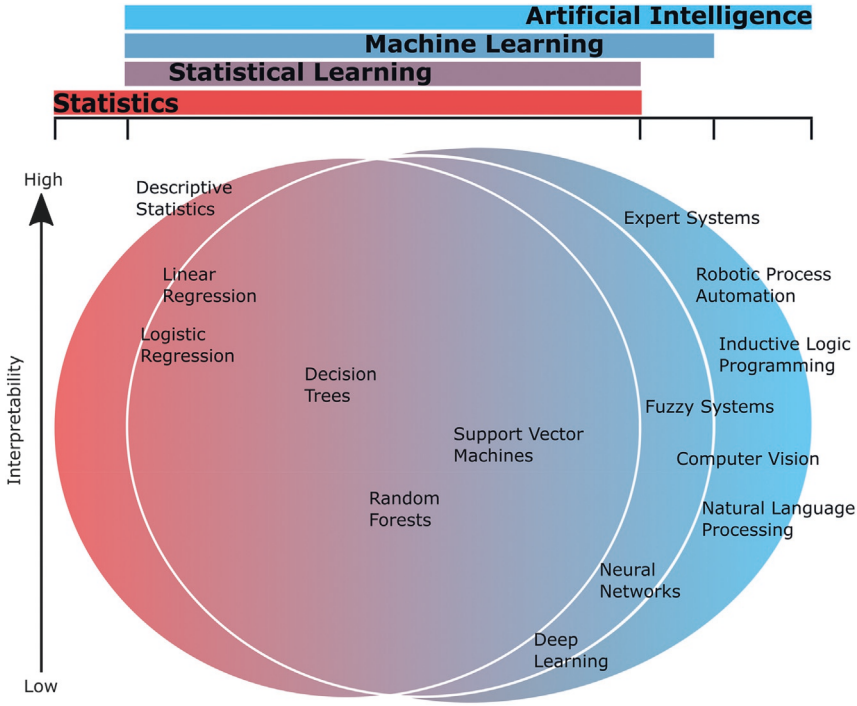


Fig. 1 Schematic diagram of methods and approaches in artificial intelligence, machine learning, and statistics.

more practical difference, namely, that ML modeling is willing to sacrifice inference for prediction, but statistical modeling requires some inference capacity to be maintained.

ML approaches can be further subdivided according to feedback type. ML can be supervised or unsupervised given the system's access to feedback. Supervised learning occurs in two phases, training and testing. During the training phase, predetermined input-response pairs (labeled data) are used as examples, and the learning algorithm attempts to formulate functions that connect input data to respective labels (Liakos et al., 2018; Montesinos-López et al., 2022). During the testing phase of supervised learning, the learned pattern (trained model) is used to generate predictions. The accuracy of the predictions can then be evaluated against user-defined labels. In a supervised learning system, the feedback is considered to be explicit. In unsupervised learning, preassigned labels for the input data are absent. This model can only evaluate patterns within the input and prevents the evaluation of prediction accuracy since the desired answer was not specified.

Therefore, this type of system has no feedback. Systems that utilize both unsupervised learning to generate labels and then supervised learning to make predictions also exist and are called semi-supervised. Supervised learning includes both regression and classification-based methods which are used with continuous and discreet data, respectively. Linear regression, regression-based decision trees, and support vector machines are all types of regression-based supervised learning. Unsupervised learning includes clustering, dimensionality reduction, and association rules. K-means clustering and hierarchical clustering, are clustering-based methods. Principal component analysis, independent component analysis, linear discriminant analysis, factor analysis, and LASSO (least absolute shrinkage and selection operator) are dimensionality reduction focused methods. Finally, the apriori algorithm, equivalence class transformation, and frequent pattern growth algorithm are association rule methods. Some widely used statistical ML methods referenced in Section 3 will not be covered in this section. Introductions to linear regression (James et al., 2021a; Su et al., 2012), logistic regression (James et al., 2021b; Peng et al., 2002), Bayesian methods (Hoff, 2009), and types of non-parametric methods including K-nearest neighbors (Taunk et al., 2019) are available in the literature.

2.2.1 Decision trees and random forests

Decision trees employ simple tests to classify data using sequential subdivision. These tests are often arranged in a branching structure where the test responses of initial branches determine which tests are subsequently applied. The points from which branches diverge are called nodes. Sub-symbolic learning is currently the dominant approach used for decision trees. However, decision trees had prior success in symbolic AI (Quinlan, 1986). Symbolic decision trees provide a simple introduction to tree-based classification. Symbolic decision trees relied on user-defined logical rules based on existing knowledge or inference related to the data being classified (Section 2.1).

For example, if someone was trying to design a plant species classification system, it would be logical to place the rule "if the plant has bark, then it is a tree" before the rule "if the plant has needles, then it is a conifer tree" or else a cactus may be classified as a conifer. However, often the hierarchy of rules is not so obvious to the user. In these cases, the test rules and their locations in the decision tree can also be learned. Sub-symbolic decision trees, requiring labeled data, attempt to generalize a pattern of tests resulting in accurate classification (Kotsiantis, 2013).

The process of partitioning the data at a given position during tree construction is termed splitting. The process of tree splitting must balance both accuracy and tree complexity. Increased accuracies can be achieved with more complex trees, but compact decision trees are more likely to avoid overfitting and maintain accuracy outside the training set. Tree complexity is gauged by factors such as total number of nodes, total number of leaves, tree depth, and number of attributes used (Kotsiantis, 2013). Tree complexity factors are moderated through user-defined stopping criteria that set maxima or minima for the factors to limit tree size. Branch pruning and feature selection methods can also be used to reduce decision tree complexity.

Decision trees have some flexibility with regard to input variables and predictions. Decision trees can be univariate or multivariate. These trees differ in the number of attributes tested at a single node with univariate testing one and multivariate testing more than one. In addition to the number of attributes tested, the tests used at nodes can also differ. Non-linear model tests can be more suitable for initial nodes where decision complexity is often high, while linear models are more appropriate for less complex splitting such as within lower levels (Yıldız and Alpaydın, 2001). Decision trees, while most often used for general multi-class (exclusive, single output) classification, can also be used for ordinal classification and multi-label (non-exclusive, multi-output) classification/regression (Kotsiantis, 2013).

Random forests are a variant of decision trees that use an ensemble approach to mitigate the instability of recursive partitioning used by decision trees (Breiman, 2001b; Kotsiantis, 2013). A random forest generates a prediction by averaging the predictions from several decision trees. The decision trees within the ensemble are constructed using random subsets of the complete data. The specific ensemble approach used by random forests is a variant of bagging, but other ensemble approaches (Section 2.2.2) can also be applied to decision trees. The bagging variant used here selects the best feature from a subset of features to split a decision tree node into branches. Random forests are a popular type of ML. Their popularity can be attributed to the method being both easy to use due to few tuning parameters and efficient when processing large data sets due to the ease of parallelization (Biau and Scornet, 2016).

2.2.2 Ensembles

Ensemble methods can utilize many types of learning algorithms. The general feature of an ensemble is a collection of base models that are used together for prediction. The use of multiple base models improves generalizability.

If a learning algorithm is most often able to achieve generalizability with a single model, it is termed a "strong" learning algorithm. Likewise if a learning algorithm does not reliably achieve generalizability with a single model, it is called a "weak" learning algorithm. Weak learning algorithms may produce predictions that are little better than random guesses (Schapire, 1990). If the generalizability of the learning algorithm depends largely on successful model tuning, both weak and strong configurations of the learning algorithm may exist. Often ensembles are able to use simpler/weaker configurations for each base model instead of the more complex configuration that would be required in a single model approach. Strong learning algorithms can also be used in an ensemble, but little is gained over using a single model when a single model can achieve stable (though not necessarily highly accurate) predictions. Among the most popular ensemble methods are bagging, boosting, and stacking, which all use collections of base models to make predictions.

Bagging, or Bootstrap AGGregatING, uses random sampling with replacement to create sub-samples. Each base model is then trained separately from other base models using a single data subset. The use of separate base models trained on independently sampled data subsets is advantageous because of the ability to parallelize training yielding increased training efficiency. Predictions from individual base models in the ensemble are then aggregated to produce the ensemble's prediction. The most common aggregation techniques are majority voting and averaging. For the sake of comparison, bagged decision tree ensembles differ from random forests because bagged decision trees use all features for each tree splitting step instead of the single best splitting feature from a subset of all features.

Boosting differs from bagging because base models are trained sequentially with each base model influenced by the training of earlier base models within the ensemble. Boosting supplements the performance of weak learning algorithms through stage-wise additive model fitting (Hastie et al., 2009; Ogutu et al., 2011). The two main boosting approaches are Adaptive Boosting (AdaBoost) and gradient boosting. AdaBoost weights the sampling of data for subsequent base models using prediction accuracies achieved by a prior base model. Data points inaccurately predicted or misclassified in the previous iteration are weighted more strongly. The subsequent base model is then trained on a sub-sample from the new distribution. The focus of subsequent learners is shifted toward data points that are difficult to predict. The weak models are finally combined into a composite model that better explains data points otherwise misclassified in individual base models (Freund and Schapire, 1996). Gradient boosting (Friedman, 2001) is a

generalization of the approach introduced by AdaBoost. AdaBoost and gradient boosting both use loss functions (Section 2.3). However while AdaBoost relies on exponential loss making it sensitive to outliers, gradient boosting is designed to use any loss function in conjunction with the gradient descent to minimize the loss value (Section 2.3) (Mayr et al., 2014). Variants of gradient boosting include GBDT, XGBoost, LightGBM, and CatBoost.

Bagging and boosting both generally use a single type of learning algorithm for all base models. Multiple types of learning algorithms can also be combined in an ensemble. In the simplest cases, the output from each sub-model can be combined into final single prediction using majority voting (classification) or averaging (regression). In other cases, the contributions of each based model may need to be weighted differently than the uniform weighting of averaging. Stacking is an ensemble approach that varies the contribution of each base-model through a subsequent meta-model that receives base-model outputs as inputs. For example, a stacked ensemble may use a linear regression, a support vector regression, and a decision tree model as the base models. The base model predictions may then be aggregated with a final linear regression meta-model.

Ensemble methods can be a good approach for handling big data. Ensembles use a different subset of data to train each base model. This can increase the speed and parallelization achieved by ensemble training compared to single model training. Subset selection depends on sampling parameters such as sub-sample size, number of base models, and sub-sampling method. While individual base models may be simplified compared to an appropriate single model for the same data, the overall ensemble does, however, becomes less interpretable.

2.2.3 Support vector machines and regression

Support vector machines (SVMs) and support vector regression (SVR) are non-parametric machine learning approaches useful for classification and regression analysis (Wang et al., 2018b). SVMs can classify data points in high-dimensional space through hyperplane separation. A hyperplane is the high-dimensional equivalent of a straight line. Hyperplane selection in SVMs is approached by maximizing the margin separating the classifications (Noble, 2006). Hyperplane selection deals with data inconsistencies by using soft margins that accept some anomalous data points to be located on the opposite side of the hyperplane from the correct classification. Users must specify how stringent to be with hyperplane violations (Noble,

2006). When SVMs are used for regression, they are termed SVR. In SVR, data points are regressed to the hyperplane and the margin is used for loss evaluation. Weights are not assigned to data located within the margin. Rather, the distance of outliers from the hyperplane is used for loss optimization. Because of this, SVR can be sensitive to outliers. Other parameters that must be user selected (hyperparameters) include the kernel function. SVMs and SVR can handle both linear and non-linear relationships by changing their kernel function. Common kernel functions include linear kernel, Gaussian radial bias function kernel, and polynomial kernel. Overall, SVM/SVR hyperparameter tuning is simpler than for neural networks (Zhao et al., 2020).

2.3 Neural networks

Neural networks (NNs) are a distinct machine learning approach that can include supervised and unsupervised algorithms. Most can be characterized according to some core architectural components. These core components include learned parameters like weights and biases, as well as hyperparameters that must be chosen during network design. Some such hyperparameters for NNs include number of layers, number of nodes per layer, activation function, loss function, and optimization algorithm (Fig. 2).

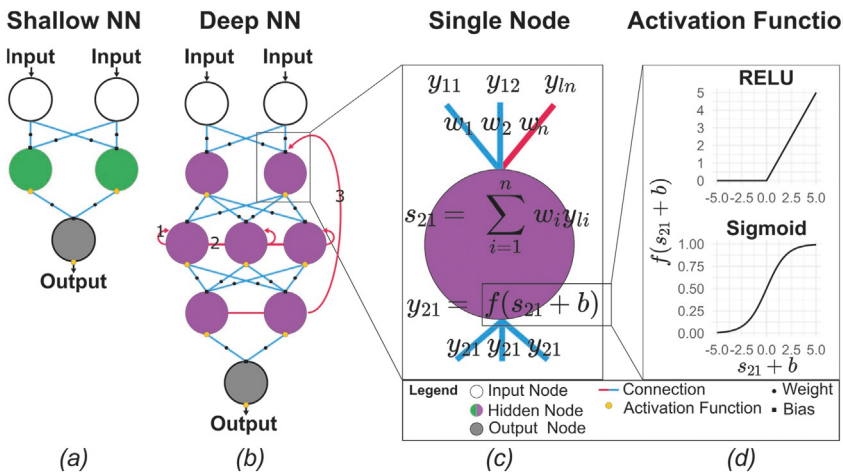


Fig. 2 Essential structures of a shallow, feedforward neural network (NN) with one hidden layer (A) and a deep NN (B) with feedforward, self (B-1), intralayer (B-2) and supralayer (B-3) recurrent/feedback connections as well as components of a single node (C) and activation functions (D).

NNs are made up of nodes that can constitute the parts of an input, output, or hidden layer. Nodes in hidden layers generally receive values from other nodes and not the input data directly. In fully connected layers, each node in the current layer receives the values resulting from the output of every node in the previous layer. Sparse layers have fewer connections than fully connected layers and can have as few as one connection between a node in the current and previous layer. Layers can vary both in number and type of connections. Different connection types are able to pass values in both the feedforward (Fig. 2A) and feedback (Fig. 2B) direction. In networks with feedback connections, information can also be recirculated back to nodes in previous layers, i.e. the output to input direction. Feedforward connections exist between different layers (interlayer), while feedback connection transmit values either back to the node of origin (self; Fig. 2B-1), within a layer (intralayer; Fig. 2B-2), or back to a nonadjacent, previous layer (supralayer; Fig. 2B-3). A deep neural network, also called a deep learning (DL) model, is distinguished from other NN types by the presence of multiple hidden layers (depth). A shallow neural network may contain only 1–2 hidden layers (Fig. 2A), while DL models feature many (more than two) hidden layers (Fig. 2B) (Bengio, 2009; Emmert-Streib et al., 2020). Different hyperparameter configurations are better suited to different problems. One hurdle of implementing a NN is hyperparameter selection. Learned parameters depend on hyperparameter selections. With increasing data dimensionality, finding quality hyperparameter values through user expertise and trial and error testing becomes impossible (Victoria and Maragatham, 2021). Therefore, an additional optimization stage, specifically for hyperparameters is required. Optimization strategies for NN hyperparameters include model-free methods as well as sequential search methods (Bergstra et al., 2011; Feurer and Hutter, 2019). Model-free methods include random search, grid search, and population-based methods like genetic algorithms, evolutionary algorithms, evolutionary strategies, and particle swarm optimization (Feurer and Hutter, 2019). A popular population-based optimization method for NN hyperparameters is the covariance matrix adaption evolutionary strategy (Hansen, 2016). Many model-free approaches can be readily parallelized, but each tested hyperparameter combination is selected independent of the performance of other combinations. Sequential search methods include sequential model-based optimization (SMBO), sequential model-based algorithm configuration (SMAC), and Bayesian optimization. Sequential search methods make subsequent hyperparameter selections

based on the performance of past selections. Other parameters of the system, such as weights and biases, are learned during NN training.

The relative importance of an input and its hidden layer derived values can be scaled using the connection weight. A bias value is also added to the weighted sum of the inputs to further adjust the value before the activation function is applied (Montesinos-López et al., 2021b). Activation functions apply a transformation to the sum of the weighted values and bias (Fig. 2C). Popularly used activation functions include rectified linear activation unit (ReLU), leaky ReLU, sigmoid, hyperbolic tangent, and softmax functions (Fig. 2D). All of which allow non-linearity to be introduced into the network model (Montesinos-López et al., 2021b; Patterson and Gibson, 2017). Linear activation functions can also be used but are generally only used within input layers since they act as an identity function (Patterson and Gibson, 2017).

During model training, prediction accuracy of the current version of a model is evaluated with a loss function, which provides an error metric representing the difference between the predicted output and the observed response provided in the training dataset. Examples of loss functions include squared loss, logistic loss, hinge loss, and negative log likelihood. To improve the model, weights and biases are updated in a manner that reduces the result of the loss function. The direction and magnitude of that change is determined using a search method referred to as an optimization algorithm (Patterson and Gibson, 2017). Optimization algorithms for NN training can be similar to those used for hyperparameter optimization and can include global optimizers such as genetic algorithms, differential evolution, or particle swarm optimization, but are more often based on gradient descent.

Gradient descent attempts to find the minimum error value of all possible weight and bias values. However, gradient descent does not evaluate the entire space of all possible weight and bias configurations. Instead during optimization, a gradient descent algorithm uses the partial derivative of the loss function with respect to an initial weight and bias to determine whether the weight/bias value should be increased or decreased. Gradient descent proceeds in this fashion until it converges on an error minimum or when a specified number of epochs have elapsed. A popular variation called stochastic gradient descent speeds up the process by evaluating the loss function from randomly chosen mini-batches of the training data instead of the complete set.

The step size with which the optimization algorithm searches is termed the learning rate. The success of an optimization algorithm can depend on

the appropriateness of the learning rate. Learning rates can be a fixed value, a value scaled progressively smaller by a fixed amount as the number of steps taken increases, or be adaptive, with step size changing based on the training progress (Daniel et al., 2016; Takase et al., 2018). Learning rates relying on fixed scaling or fixed values can be difficult to tune. Adaptive learning rates are therefore attractive for improved rate identification.

The approach by which the error gradient is calculated across the layers of learned parameters during optimization is termed the learning algorithm. The term learning algorithm has also been used herein to describe different ML approaches. This is because the NN learning algorithm introduced here is simply a specific instance of a learning algorithm that describes how a NN attempts to learn appropriate weights and biases. Backpropagation is currently the dominant type of NN learning algorithm and updates weights first in the layer closest to the output and last in the layer leading to the input. The learning algorithm and optimization algorithm work in tandem but are unique processes. For example, gradient descent determines the weights and biases that are tested next during training and relies on an error gradient to inform that decision. Backpropagation is the approach by which the error gradient is calculated.

The set of "best" weights identified through learning and optimization simply minimizes the error of prediction in the training dataset. A network built on these weights still has the potential to be underfit or overfit. Overfitting can be addressed using regularization. Regularization imposes smoothness constraints on the function approximation of neural network (Girosi et al., 1995), thereby influencing the final weights and biases selected during training. L1 regularization, L2 regularization, dropout, and early stopping are among the dominant methods used to prevent overfitting in neural networks. L1 and L2 regularization are also used by other types of ML. L1 and L2 regularization are particularly common in linear regression, and these implementations are also known, respectively, by the names of LASSO (least absolute shrinkage and selection operator) regression and ridge regression. L2 (ridge) regularization applies a shrinkage penalty to shrink weights toward zero. This reduces weights, but none are set precisely to zero. It also preserves all input as features rather than eliminating unimportant features. L1 (LASSO) regularization places preference on low weight values to reduce model complexity which results in the preference of a zero weight. Unimportant features are then absent from the prediction. Dropout is similar to L1 regularization in the sense that unimportant features are not represented. However, dropout determines feature

importance by testing a series of "thinned" versions of the network where some nodes have been randomly dropped (Srivastava et al., 2014). The dropping of nodes and their associated connections promotes the independent learning of weights and biases. Each thinned network is trained with extensive weight sharing. This means that each thinned model is initialized with the weights learned by the previous thinned model to reduce the computation. An averaging method is then used to produce a final single model from the collection of thinned models (Srivastava et al., 2014). Dropout regularization is conceptually similar to ensemble methods where many "weak"/thinned versions of the network are used to generate one more robust prediction. Early stopping is another approach to regularization. However, early stopping only alters the training process with regards to when training is ended. As the name suggests, early stopping stops training during the approach to the loss minimum when the validation error rate starts to increase (Svozil et al., 1997).

The initiation and completion of these processes associated with minimizing the loss function defines the training stage. Because weights and biases are updated during training, they are considered to be learned parameters. Architectural principles remain largely the same between DL and other NN types. Popular DL topologies can be characterized according to their respective hyperparameters. Given the information provided herein on the application of NNs, it is also important to touch on the underlying theory that has driven the success of NNs. That theory is that NNs are capable of functioning as universal approximators. Given an unspecified (potentially approaching infinite) number of nodes, a feedforward NN with one or more hidden layer(s) is expected to be able to approximate any function (Hornik et al., 1989; Irie and Miyake, 1988. Given this theorem, poor approximations are then attributed to suboptimal hyperparameter selection and model training rather than a limitation of the ability of NNs.

2.3.1 Feedforward networks and multilayer perceptrons

Feedforward networks can feature a variety of characteristics in addition to feedforward connections. Multilayer perceptrons (MLPs) are distinguished from other feedforward networks by standardly using fully connected layers. Another special case of a feedforward network is a radial bias function network which uses a radial bias function as the activation function. MLPs are among the most basic of NN types and one of the most widely used. The name multilayer perceptron is in reference to the most basic type of neural network: the single layer perceptron, often abbreviated as simply the

perceptron. A perceptron is made up of a single hidden node with associated connections, weights, and activation function (Fig. 2B). The multilayer perceptron is built as a collection of these perceptrons (Fig. 3). Additionally, other NN types are often described based on how the architecture differs

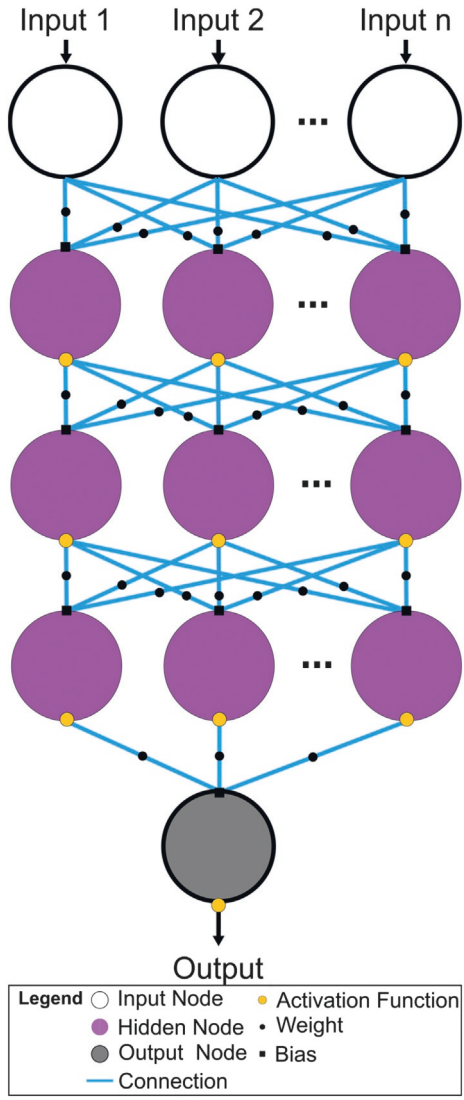


Fig. 3 A deep, feedforward, fully connected neural network. This configuration defines a multilayer perceptron.

from MLPs. Therefore, generalized descriptions of NNs (as in Section 2.3) are often framed in the context appropriate for MLPs.

2.3.2 Convolutional neural networks

Convolutional neural networks (CNNs) are an extension of deep NNs and they are characterized by having three distinct types of layers: convolutional, pooling, and fully connected layers.

Convolutional layers are a type of sparse, locally connected layer that reuses shared weights across each node in the subsequent layer (Fig. 4) (Emmert-Streib et al., 2020). The use of a uniform filter across the input results in all nodes in the subsequent layer having the same number of connections. Convolutional networks, which contain convolutional layers, often receive data matrices, like photos, as input. In this situation the filter would likely also take on a two-dimensional structure and the output of the convolutional layer would be a two-dimensional feature map (also called an activation map). The use of shared weights via a filter, instead of independent connections weights like in a MLP, serves as a spatial feature extractor with different filter values able to extract different features (Emmert-Streib et al., 2020). Some important convolutional layer hyperparameters that differ from the hyperparameters of other neural network types include the size of the filter, stride, and amount of zero-padding (Emmert-Streib et al., 2020). The size of the filter (i.e. the number of connections) determines the dimensions of the region (think number of pixels in the case of an image input) that will be used to generate a new value on the feature map. The stride size determines how many units to move the filter between calculating adjacent values in the feature map. The size of the filter and the stride determine how much overlapping information is used to generate

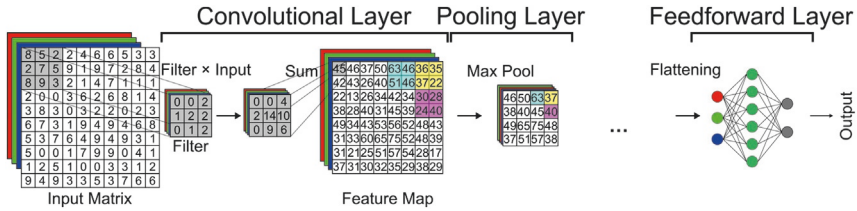


Fig. 4 A convolutional neural network containing representative convolutional, pooling, and feedforward layers. In this example, a 3×3 filter is applied in the convolutional layer. The max pooling method is applied in the pooling layer. Convolutional and pooling layers can be repeated and precede a layer with feedforward connections.

each feature map value. Zero-padding refers to the use of a border of zero values around the border of the convolutional layer input. Without zero-padding, values on the edge of the matrix are considered in the calculation of fewer feature map values and the feature map dimensions will be reduced from input.

Pooling layers are used to further reduce the dimensions of the layer's input, which is usually a feature map as shown in Fig. 4. The mechanics of a pooling layer share some similarities with convolutional layers. The hyperparameters of this layer type include pooling window size, stride, and zero padding. Pooling window size is similar to the filter size. It determines how many units of the feature map are surveyed to generate the pooling layer output. However instead of applying weights, a pooling method is applied to the considered units. As the most common method, max-pooling simply preserves the maximum value from within the window size to the pooling output. Some other pooling methods include averaging-pooling, min-pooling, fractional max-pooling, and stochastic pooling (Emmert-Streib et al., 2020). The stride and zero-padding in the pooling layer are conceptually identical to the convolutional layer.

In CNNs, convolutional layers are generally followed by pooling layers. These two layers can be stacked a number of times to generate a deep network. Fully connected layers are often used as the last hidden layer(s) and output layer in CNNs (Emmert-Streib et al., 2020; Krizhevsky et al., 2017). This layer type is used to help capture the relationships between features extracted in the convolutional and pooling layers and the network output.

2.3.3 Recurrent neural networks

Recurrent neural networks (RNNs) have both feedforward and recurrent (feedback) connections. Recurrent connections function as cyclic connections that preserve values obtained during the process of making initial predictions to inform subsequent predictions (Fig. 5). This process gives RNNs "memory" and makes RNNs well suited for sequential data. RNNs can take on a variety of forms. But even when featuring simple architectures with few nodes, RNNs are considered as DL because learning occurs over many cycles of the RNN in place of the many layers of a deep feedforward NN (Schmidhuber, 2015). Fig. 5 shows a RNN that has been unrolled across time; however not all RNNs can be unrolled in this manner (Emmert-Streib et al., 2020). RNNs include Hopfield networks, Boltzmann machines, and long short-term memory (LSTM) networks.

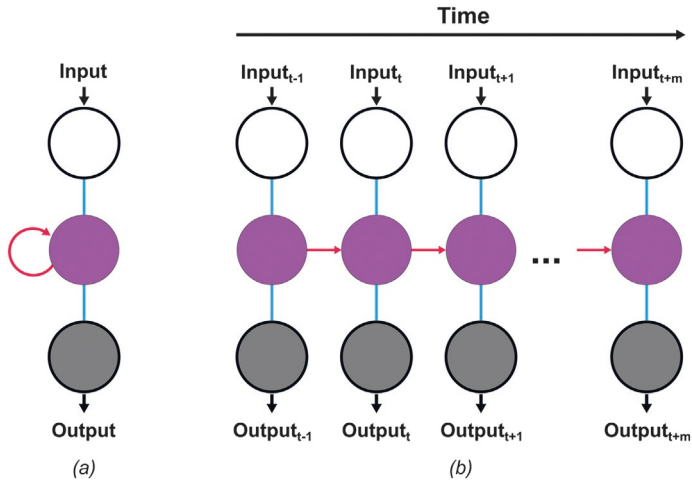


Fig. 5 A recurrent neural network (RNN) with a single hidden node (A) and the same RNN that has had the recurrent connection unrolled across time (B).

Hopfield networks (HNs) are fully connected recurrent networks. HNs differ from MLPs because they are fully connected in the recurrent sense rather than the feedforward sense, meaning that every node is connected to every other node in the network. This also means that nodes cannot be organized into layers. HNs also do not distinguish layer types (input, hidden, and output). Instead, each node serves all three functions in sequence (Emmert-Streib et al., 2020). Boltzmann machines are a variant of HNs that use a special type of probabilistic activation function (see Hinton and Sejnowski (1983) for more details). Boltzmann machines are still fully connected networks, but nodes are split into a visible layer (input/output) and a hidden layer between which values can move bidirectionally (Emmert-Streib et al., 2020).

Long short-term memory (LSTM) networks are an increasingly used type of RNN that were designed to address drawbacks of earlier RNN types. With standard RNNs, memories or patterns learned from prior data points in a sequence are often only preserved in the short term. When attempting to train a standard RNN that stores information over the long term, backpropagation becomes a time-consuming process because loss functions must also be evaluated through time (Hochreiter and Schmidhuber, 1997). LSTM solves this problem by truncating the error gradient calculation through time where possible (Gers et al., 2002; Hochreiter and Schmidhuber, 1997). The capability of a LSTM to store information for

longer periods gave it access to more learned features than prior RNNs. However, the original LSTM configuration led to the indefinite accumulation of information until eventually the network would break. A method of resetting a network's memory called a "forget gate" was developed to address this problem (Gers et al., 2000). The modern LSTM network is composed of memory blocks that are made up of one or more memory cells, an input gate, an output gate, and a forget gate (Gers et al., 2002). The memory blocks can then be chained together in the same manner as traditional NN nodes. The learning algorithm used by LSTM is a variant of backpropagation fused with components of an earlier RNN learning algorithm called real-time recurrent learning (RTRL) (Williams and Zipser, 1995). Variant architectures have also been developed. One of which is called Gated Recurrent Unit (GRU). GRU also relies on the gate structure introduced by LSTM (Dey and Salem, 2017). Where a LSTM has three gates within each memory block, a GRU uses only two gates to control the use of current input and values from prior memory (Dey and Salem, 2017). Further in the RNN succession, there are also GRU variants that use other configurations of gates to generate NNs with memory. Broadly speaking, the memory block used by LSTM and the other single units used by GRU and variants can be called RNN cells.

2.3.4 Encoding decoding networks

Encoding decoding type networks rely on reducing variable length inputs to a fixed length bottleneck (encoding) then reconstructing the output to a similar length as the input (decoding). The bottleneck forces the network to distill a compact feature set that is representative of the input data. In autoencoders, encoding layers feature progressively reduced node numbers toward a bottleneck layer. The bottleneck layer is followed by decoding layers that reconstruct input data points using increasing node numbers (Kramer, 1991). Autoencoders are a type of unsupervised ML that attempt to reconstruct the input data from the features learned through the bottleneck, so labeling of input data is unnecessary and the decoder output is compared to the input for optimization (Hinton and Zemel, 1993). Autoencoder subtypes include denoising autoencoders and variational autoencoders. Autoencoders can also be combined with other NN structures to produce adversarial autoencoders, convolutional autoencoders, and sparse autoencoders. Encoder-decoder recurrent networks (also referred to as sequence-to-sequence learning) utilize RNN cells from LSTM or GRU to form a similar variable input to bottleneck to variable output

structure. However, the bottleneck is accomplished through time or sequence and the important features are those that are preserved through rounds of recurrent connections.

2.3.5 Generative networks

Generative neural networks are a class concerned with generating new examples that resemble the training data. Generative models more broadly are approaches that attempt to learn the probability distributions that underlie the training examples they are provided (Goodfellow et al., 2020). An example of a simple ML generative model is maximum likelihood estimation. Generative models use the learned probability distribution estimate to formulate the model output. This class of models exists in contrast to discriminative models that learn boundaries to separate different classes of labeled data. Examples of discriminative models include logistic regression, support vector machines (SVMs) (Section 2.2.3), and decision trees (Section 2.2.1). Generative adversarial networks (GANs) are currently the most popular type of generative neural network, but variational autoencoders (Section 2.3.4), autoregressive NNs, fully-visible belief networks, and transformers are also types of generative networks.

GANs rely on two models, one of which is a generator while the other is a discriminator. The job of the discriminator is to determine if the values provided to it are real (sampled from the training data) or fake (created by the generator) (Goodfellow et al., 2020). Both the generator and the discriminator are optimized during training. However, these models are at odds with one another. As the generator model improves, it becomes more difficult for the discriminator to determine whether values were observed (real) or created (fake) (Goodfellow et al., 2020). Until eventually, the generator is capable of producing realistic outputs that the discriminator cannot distinguish from observed data points. Variational autoencoders are a type of generative autoencoder that preserves the encoding decoding structure of autoencoders but learns in a probabilistic manner similar to other types of generative networks (Patterson and Gibson, 2017). Autoregressive NNs are a type of feedforward network that also structures the network into encoding and decoding components (described in Section 2.3.4). However, the key feature is that this network uses the output of previous time step as input for subsequent time steps. This allows the network to generate output indefinitely (or until a stopping mechanism deploys) since new input is also continuously available.

2.3.6 Transformers

Transformers are neural networks that differ from other types because of the attention mechanism that they rely on. Attention is the learning of representations of input in a context dependent manner (Vaswani et al., 2017). In other words, attention can learn interactions between values in an input vector (Cheng et al., 2021). The attention mechanism identifies critical context information for encoding each input and decoding each output. This context information is represented in a context vector made up of the sums of the weighted annotation values. Different context vectors can then be used to decode different outputs in conjunction with the encoded feature vector. Attention therefore produces predictions that are more representative of the complete input vector (Bahdanau et al., 2014).

Transformers are also capable of parallelization of sequential/time series data because of positional encoding. Essentially, transformers are provided with an additional input vector that specifies order numerically (positional encoding) separately from the input vector that specifies input values (input embedding). Many types of transformers exist, including variants that use only the encoding (BERT, RoBERT) or decoding (GPT-4) transformer subunit. In addition to the natural language applications of the original transformer (Vaswani et al., 2017), transformer variants have been adapted to audio (Wave2Vec, HuBERT), image (ViT), video (BEiT, Maskformer), multi-modal (VisualBERT, DALL-E 3), and protein sequence (AlphaFold2, ESMFold) applications (Betker et al., 2023; Jumper et al., 2021; Lin et al., 2022b; Lin et al., 2023; OpenAI, 2023).

Even though the attention mechanism is well known for its role in transformers (Vaswani et al., 2017), it was originally developed to improve RNNs. The memory mechanism in RNNs has a recency bias. Information learned from more distant values in the input sequence has more opportunities to be forgotten than recently evaluated values in each hidden state. For example, in a dataset containing a 24 h time-series experiment when evaluating the hidden state for the hour 23 input, an RNN has had many more opportunities to forget important features from hour 1 than from hour 22. Particularly in encoder-decoder RNNs where memory is used to encode the input into a fixed length feature vector, it can be difficult to preserve relevant learning dependencies between distant values (Bahdanau et al., 2014). The attention mechanism grew out of this context.

2.3.7 Multi-modal neural networks

The task of fusing multiple types or modes of data has been approached a number of ways. These approaches can generally be divided into early, late, and intermediate fusion (Ramachandram and Taylor, 2017). Early and late fusion refers to the time when data sources are integrated. Early fusion occurs prior to feeding the data into a model and is accomplished through some method of dimensionality reduction (like PCA), while late fusion instead integrates the information sources after already passing data through separate models in a manner similar to an ensemble using average aggregation (Ramachandram and Taylor, 2017). Multi-modal neural networks represent an intermediate fusion of learned representations (Ramachandram and Taylor, 2017).

Multi-modal neural networks (m-NNs) combine sub-networks where each can accept a different type or mode of data (Fig. 6). Sub-networks in a m-NN can feature architectures identical to other single network architectures, such as GANs, LSTMs, CNNs, and MLPs. Similar to ensembles, m-NNs can have sub-networks of the same type or different network architectures. Sub-networks are connected and interact through a multi-modal layer, similar to the meta-model of stacked ensembles (Section 2.2.2). An advantage of using a m-NN over feeding outputs from single networks to

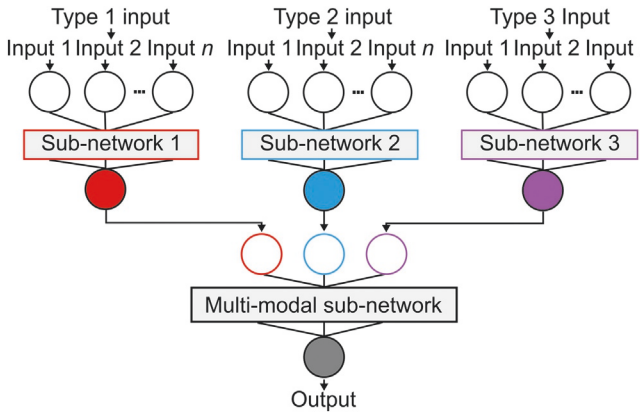


Fig. 6 A multi-modal neural network accepting three input data types. A separate sub-network receives each data type. The intermediate values resulting from the sub-networks are propagated through a joint, multi-modal sub-network to make the final prediction.

another subsequent single NN for the final prediction is that all m-NN sub-models can be trained collectively, instead of independently.

2.4 Computer vision

Computer vision is the field of artificial intelligence associated with deriving meaning from images. A computer vision system comprehends its environment in two stages: image acquisition and image processing (Narendra and Hareesh, 2010; Patrício and Rieder, 2018). Images can be acquired using cameras and other imaging technologies many times in combination with a mobility system. Image acquisition systems are often mono-RGB (red, green, blue) vision systems, but may also include stereo vision, multi/hyperspectral cameras, time-of-flight cameras, LiDAR technology, thermography imaging, fluorescence imaging, and tomography imaging (Perez-Sanz et al., 2017). These various types of sensors may also be accompanied by mobility components such as conveyors, ground vehicles, unmanned aerial vehicles (UAVs), and motorized gantries (Mochida et al., 2019).

Image processing can be further divided into pre-processing operations, segmentation, feature extraction, feature selection, and classification (Fig. 7A) (Jayas et al., 2008; Perez-Sanz et al., 2017). To facilitate other image processing steps, images may be preprocessed to crop the field of view, improve contrast, eliminate noise, reduce dimensionality, and apply filters, among other more task specific goals (Perez-Sanz et al., 2017).

2.4.1 Image segmentation

Image segmentation is important for background and object differentiation and is critical for downstream recognition and classification tasks. Some common image segmentation methods include those that are threshold based, edge based, color-index based, region based, clustering based, or based on deep learning. Threshold based segmentation separates a greyscale image into groups based on pixel intensity (Li et al., 2020). Edge based segmentation also relies on pixel intensity but recognizes large changes in intensities of adjacent pixels as edge boundaries (Kuruvilla et al., 2016). Edges can be further refined through edge thinning and edge linking. Threshold and edge based segmentation methods are simple and efficient, but can be ill suited for complex images. Both segmentation methods rely on difference in intensity or greyscale color space which can fail when an image has little intensity variation. Across an RGB image, color index-based segmentation may be a better approach since it can preserve differences in RBG color space rather than greyscale space. Color indices convert RGB space into

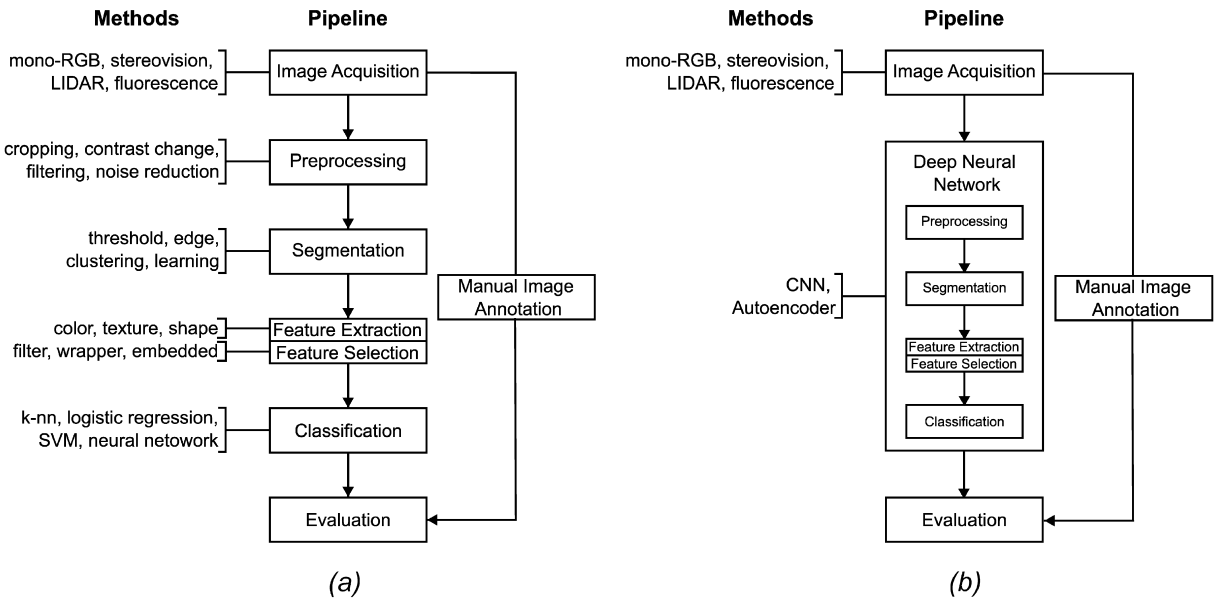


Fig. 7 Steps and different methods used in (A) conventional and (B) neural network-based image processing.

alternative one-dimensional color spaces. For example, a green index would preserve differences in green color (Hamuda et al., 2016). Region based segmentation divides an image into regions of alike pixels. Region based segmentation works well in noisy images where borders may be more difficult to detect (Kuruvilla et al., 2016). Region based segmentation starts with individual pixels or pixel groups and processively adds similar pixels to the initial region or splits dissimilar pixels from the region (Kaur and Kaur, 2014). Clustering based segmentation also groups alike pixels together. However, clustering based segmentation strategies use clustering algorithms, such as hierarchical or partition based methods, to identify regions with pixel similarity. Clustering can be either hard or soft. Hard clustering assigns only one group per pixel. Soft clustering uses fuzzy logic to assign partial membership of a pixel to multiple groups (Kaur and Kaur, 2014). Region based and clustering based methods may more successfully segment complex images but are more complicated algorithms that require additional parameter tuning and heavier computation than edge and threshold based segmentation methods (Li et al., 2020). Deep learning methods can be highly accurate but are also the most computationally intensive and require large amounts of training data compared to the other segmentation methods. A variety of deep neural networks, such as convolutional, encoder-decoder, or recurrent networks, can be used as a backbone for segmentation (Minaee et al., 2021). General details on various neural network types are addressed in Section 2.3. For a more thorough review of the uses of NNs for image segmentation see Minaee et al. (2021).

2.4.2 Feature extraction and selection

Feature extraction and feature selection are other mid-level image processing steps that generally follow segmentation and can be used together or individually. A segmented image may contain points of redundant data. Feature extraction summarizes the segmented image into a new feature set of the smallest possible size while preserving the characteristics of the complete image. Feature extraction methods differ by feature type with separate approaches existing for color, texture, and shape features. A few examples include color histograms, color moments, color correlograms, spatial texture methods, and spectral texture methods (Tian, 2013). Feature selection can then be used to identify the features that are interesting and the features that are irrelevant to the computer vision task (Zebari et al., 2020). Proceeding with the subset of relevant features identified by either feature extraction or feature selection rather than the segmented image further reduces

dimensionality, which can reduce training time and improve accuracy of classification (Zebari et al., 2020).

The dominant feature selection methods include filter, embedded, wrapper, hybrid, or ensemble. Filtering is one of the earliest developed methods and operates by comparing features to the characteristics inherent to the computer vision task. With filtering, feature selection and any subsequent learning tasks, which would include classification in computer vision, remain separate processes. Filtering generally has good performance and is highly efficient computationally, which allows scaling to large datasets (Zebari et al., 2020). The wrapper method wraps feature selection around the classification task so that features minimizing estimation error are selected (Zebari et al., 2020). The wrapper methods can achieve better performance than filter methods, but at the risk of overfitting and increased computational complexity. The embedded method also integrates feature selection with the classification task but avoids repeating classification (once for feature selection and once for classification) by embedding feature selection within the classification task (Zebari et al., 2020). This design results in a method with similar accuracy, but improved efficiency and reduced complexity compared to the wrapper method (Zebari et al., 2020). Hybrid methods combine two feature selection methods. One of the most common combinations is a hybrid filter-wrapper method. Hybrid methods seek to combine complementary suitabilities of two methods by using evaluation criteria of different methods at different search stages (Liu and Yu, 2005). Ensemble methods for feature selection operate similarly to ensemble methods described in Section 2.2.2. Generally, ensemble feature selection builds feature sets on a variety of subsamples and then aggregates the subsample feature sets into a single feature set (Zebari et al., 2020). Ensemble feature selection preserves the advantages of the ensemble approach, reducing over fitting and maintaining robustness to unstable data.

2.4.3 Image classification

Image classification, also termed image recognition, is tasked with comprehending the objects contained within an image or, more specifically, comprehending the features contained within the previously identified feature set. Image classification is most often accomplished using supervised learning algorithms that are also common to other classification tasks and relies on manually labeled images for training. Image classification can be accomplished using decision trees or random forests (Section 2.2.1), logistic regression, support vector machines (Section 2.2.3), Bayesian networks,

K-nearest neighbor classifiers, and various types of neural networks (Section 2.3).

Recent improvements in computer vision have almost exclusively relied on neural networks. The performance of deep-learning based computer vision pipelines has been shown to far exceed other approaches that dominated the field prior to the renewed interest in neural networks, which has defined the third AI summer (2012–present). This success in the field of computer vision can be attributed to neural networks being capable of jointly processing feature selection/extraction and image recognition steps (Fig. 7B) (Deng, 2014; Guo et al., 2016). Successful neural network-based computer vision methods have been largely derived from a few major types of neural networks including convolutional neural networks, restricted Boltzmann machines, autoencoders, and other sparse coding networks (Guo et al., 2016). Among those, CNN-based computer vision has been the most widely deployed for a variety of computer vision tasks (Guo et al., 2016).

As the scale of training data available for tasks like computer vision has increased, the use of foundation models has become appealing in place of separately training every task-specific computer vision model. Foundation models are models that are trained on broad data usually without a very specific classification or prediction objective. Foundation models can then be adapted to a variety of more specific tasks through additional training, usually fine-tuning (Bommasani et al., 2021). Current foundation models in computer vision include CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and Florence (Lu et al., 2021) for visual-language representation learning; the Segment Anything Model (SAM) (Kirillov et al., 2023) for image segmentation; and DALL·E (Ramesh et al., 2021) for image generation.

2.5 Natural language processing

Natural language processing (NLP) is a type of AI concerned with learning from language similar to the human task of reading. To solve this problem, NLP can employ other types of AI within processing steps. Modern NLP generally takes the form of statistical NLP or neural NLP that use statistical learning and neural network learning methods respectively. Previous, less successful iterations of NLP did, however, use symbolic AI. NLP is primarily concerned with text and speech processing, morphological analysis, syntactic analysis, lexical semantics, relational semantics, and discourse. These NLP

processes can be used to accomplish tasks like machine translation, speech recognition, and speech synthesis.

Much like computer vision, modern NLP is largely accomplished with NN-based approaches. These NN types have been largely covered in previous sections and, due to the low implementation frequency of NLP within various crop improvement methods, we will not revisit them here in NLP specific context. However, the NLP field has yielded many interesting developments in NNs. For example, ChatGPT (OpenAI, 2022), a recently popularized feat in NLP, is derived from an innovative lineage of NNs called generative pre-trained transformer (GPT) networks (Brown et al., 2020; Radford et al., 2018, 2019) that are themselves variants of transformer networks (Section 2.3.6) (Vaswani et al., 2017). GPT networks are NLP foundation models that combine generative pre-training and discriminative fine-tuning of a transformer model. Outside of NLP, transformers have been explored for computer vision (Khan et al., 2022; Yuan et al., 2021), time series forecasting (Li et al., 2019; Lim et al., 2021), and protein structure prediction (Jumper et al., 2021; Lin et al., 2023). AlphaFold1 (Senior et al., 2020) is an example of an AI accomplishment that is widely recognized within the scientific community, but AlphaFold2 underwent a complete redesign based around the transformer (Fig. 8) to achieve even greater protein structure accuracy (Jumper et al., 2021). The NLP field has and will continue to be an area to watch for novel AI and NN approaches that may be suited to alternative uses.

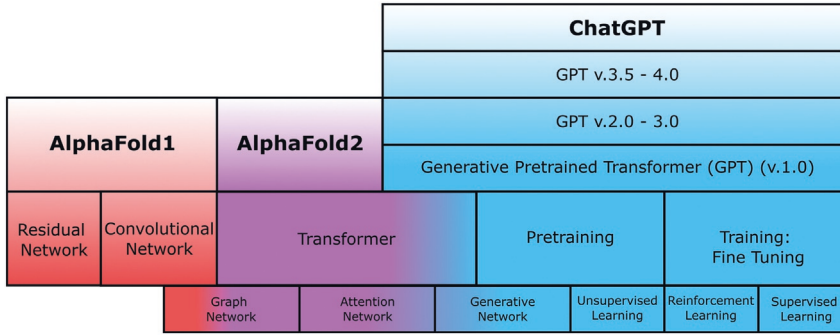


Fig. 8 Transformers connect natural language processing with protein structure prediction through the shared use of this neural network architecture by Chat-GTP and AlphaFold2.

3. AI applications in crop improvement

3.1 Genomic selection

Across the last several decades increasing focus has been placed on genetics as a means to accelerate crop breeding. First generation genomics-enhanced breeding methods like marker-assisted selection, relied on identifying individual loci that affected a phenotype. Given that many agronomically important traits, including flowering time, yield, and disease tolerance, are polygenic in nature, a more comprehensive method of selection based on genetics was needed (Buckler et al., 2009; de los Campos et al., 2013; Varshney et al., 2017). Genomic selection (GS) was designed to address this issue (Bernardo and Yu, 2007; Heffner et al., 2009; Meuwissen et al., 2001). Genomic selection uses genome-wide DNA markers and observed phenotypes in a training population to produce a model, which predict phenotypes of an untested population from its DNA marker data for selection (Montesinos-López et al., 2021b; Varshney et al., 2017). While both GS and genomic prediction (GP) were often used interchangeably, GS is a general term that stresses the entire process with an actual selection step within a breeding program and GP stresses the model development and prediction as objectives.

Many conventional GS approaches generally employ linear parametric regression. Such conventional GS approaches include best linear unbiased prediction (BLUP) based methods like genomic BLUP (GBLUP) and ridge regression BLUP (rrBLUP) as well as Bayesian models such as BayesA, BayesB, BayesC, BayesR, and Bayesian LASSO. Currently, conventional parametric regression-based approaches remain popular as they are more computationally efficient than many non-linear, non-regression, and/or non-parametric-based approaches and also often perform as well as or better than other AI-based methods (Ogutu et al., 2011). However, linear parametric models have some limitations such as capturing only additive effects and assuming phenotypes are normally distributed and continuous (González-Camacho et al., 2018). Despite the pervasiveness of conventional methods, any prediction method that expresses the relationship between the training set's input (genotypes) and output (phenotypes) can be used for GS (Montesinos-López et al., 2021b). Alternative GS approaches that can maintain or improve prediction accuracies, capture complex genetic interactions, and balance computational efficiency continue to be investigated, and many types of AI-based GS methods remain promising. Conventional GS

approaches have been thoroughly reviewed elsewhere (Crossa et al., 2017; Jannink et al., 2010; Meuwissen et al., 2016) and so here we will focus on the other types of statistical ML; including ensemble learning such as random forests (RF), support vector machines (SVM), and neural networks (NNs) for GS.

3.1.1 Ensembles

Some ensemble ML methods include RF, bagging, Adaboost, and gradient boosting. RFs overcome some of the limitations of other types of regression-based methods. RFs can handle instances where the number of markers exceeds the number of observations and when highly correlated and interacting markers affect the model fit. RFs are non-parametric and so do not make assumptions regarding predictor variable distribution (Breiman, 2001b; Ogutu et al., 2011). GS using RF regression on simulated data has found that while RFs are capable of capturing epistatic effects, the sub-sampling approach used can cause under-sampling of SNPs near QTL (Ogutu et al., 2011). RF GS was used for disease count prediction in spring wheat and outperformed a generalized Poisson ridge regression model (Montesinos-López et al., 2021a). Adaboost, because of the way it increases the weights of previously incorrectly classified data points, may not be robust to outliers, missing data, and correlated data which is a disadvantage for its use in GS (Freund and Schapire, 1996). In contrast, gradient boosted GS models can handle interactions, outliers, missing data, correlated variables, and irrelevant variables while also automating variable selection. The variable weighting for a gradient boosted model is accomplished in the same manner as RF (Hastie et al., 2009; Ogutu et al., 2011). Boosting also employs sub-sampling and can therefore exhibit some of the same limitations as RFs when it comes to undersampling of SNPs near QTL. However, boosting has been shown in some instances to perform slightly better than RFs (Ogutu et al., 2011).

3.1.2 Support vector machines

Support vector machines (SVMs) are well suited for GS on categorical phenotypes (Noble, 2006; Zhao et al., 2020). However, performance varies by the suitability of kernel function for a given data set and so kernel testing is a required step in developing a SVM GS model (Zhao et al., 2020). This also means that SVMs are highly adaptable to different data types.

Similar to linear models, linear kernels have an advantage in speed over non-linear kernels. Using a radial bias function (non-linear) kernel has been

shown to increase prediction accuracy in datasets with linear inseparability over linear kernels (Zhao et al., 2020). The use of SVMs for GS does require the testing of suitable kernels and hyperparameters which may be a drawback for these methods. Additionally in practice, SVMs have been observed to perform similarly to linear GS methods (Montesinos-López et al., 2019; Zhao et al., 2020).

3.1.3 Deep neural networks

Deep learning (DL) methods are nonparametric and can capture patterns with unknown structure, which is an advantage in terms of flexibility over parametric models like GBLUP, rrBLUP, and Bayesian methods. Feedforward NNs like MLPs have been more widely implemented than other deep learning networks for GS. MLPs are the simplest deep NN to train but can suffer from overfitting. Feedforward NNs often perform similarly (Azodi et al., 2019; González-Camacho et al., 2012; Montesinos-López et al., 2018a) or better than conventional methods (Gianola et al., 2011; Khaki and Wang, 2019). There are additional instances where model choice has depended on modeled effects (Mcdowell, 2016; Montesinos-López et al., 2018b; Sandhu et al., 2021; Zingaretti et al., 2020). MLPs do not necessarily outperform other network architectures. CNNs have also been experimentally tested for yield prediction in soybeans (Liu and Wang, 2017), stem height in loblolly pine (Liu and Wang, 2017), grain traits in wheat (Ma et al., 2018), and several traits including height, flowering time, yield in six species (maize, rice, sorghum, soy, spruce, switchgrass (Azodi et al., 2019). The biological relevance of network architectures may need to be considered. Pook et al. (2020) raised concerns regarding the appropriateness applying CNN filters, which search for structural features, to SNP data since adjacent SNP markers are not expected to have direct functional relations. CNNs with local filters were proposed as an alternative and did outperform traditional CNNs and MLPs using Arabidopsis and simulated maize data (Pook et al., 2020). Overall, the uses of RNNs and other NN architectures for GS have been limited.

High dimensionality is a major concern for predictive breeding (Crossa et al., 2017; Ramstein et al., 2019). One that also extends to DL GS methods (Washburn et al., 2020). Appropriate dimensionality reduction therefore remains a concern for improved DL GS. Another reason that deep neural networks may fail in practice is that, given the focus on loss optimization, systems may learn unintended shortcuts that exploit superficial correlation to arrive at simpler solutions (Geirhos et al., 2020; Wen et al., 2022).

This leads to good predictions in the training set but poor predictions when the superficial correlations are absent in the novel data (Geirhos et al., 2020). Within GS, one way NNs have been observed to take shortcuts is by making predictions based on relatedness of individuals, or the correlation of molecular markers, and ignoring the epistatic interactions that neural networks appear well suited to incorporate into the prediction (Ubbens et al., 2021). Often similar prediction accuracies between NNs and linear regression GS is hypothesized to be explained by a phenotype being largely additively controlled. A failure of NNs to estimate epistatic interactions may provide an alternative explanation as to why NN-based GP fails to outperform linear approaches. Incorporating logic-based programming, the same approach used in many symbolic AI systems, is an approach currently being investigated to allow NNs to be right for the right reasons. Examples include PrimeNet which exploits domain knowledge to prime a NN by explicitly identifying task relevant information to the NN (Wen et al., 2022).

3.2 Enviromics

Envirotyping has emerged as the environmental complement of other "typing" techniques like genotyping and phenotyping (Xu, 2016). It encompasses the process of collecting, processing, and associating environmental data (Costa-Neto and Fritsche-Neto, 2021). Similar to genomics and phenomics, enviromics differs from conventional environmental characterization largely in its scope. The aim for enviromics is to capture all environmental factors, both major and minor, at site, plot, or even plant-specific levels (Xu, 2016). The collection of these envirotypes is what makes up an enviromic dataset. While environment has always been considered in crop breeding, the rise of predictive breeding has driven interest in characterizing genotype-by-environment interactions (GEI). Developing crop cultivars with performance less affected by GEI is a major breeding objective because it is desirable to have a stable crop performance across a target population of environments (TPEs). Achieving this objective often requires multi-environment trials to capture performance across a TPE. However, the entirety of a TPE cannot be tested. Crop performance prediction for unobserved locations/seasons within a TPE may be improved by characterizing the environmental factors driving GEIs and integrating these predictors into a genomic prediction model. Historically, most environmental data was available only at low resolutions. More recently, data with high levels of

granularity has become accessible due to GIS, remote sensing, and wireless sensing networks becoming more prevalent and has allowed the observation of environmental differences within a TPE. Environmental factors which may be envirotyped include climate factors like light, temperature, or precipitation; soil factors like soil type, soil fertility, or soil pH; biotic factors like the presence of insects, viruses, or weeds; and cropping system factors like intercropping or crop rotation (Xu, 2016). Enviromics may be assisted in a number of ways by AI. Data collection methods for GIS and other remote sensing systems are largely image-based and subsequent image processing is fundamentally similar to the use of AI for image-based phenotyping. More detailed overviews of these types of enviromic data pre-processing are described elsewhere (Costa-Neto and Fritsche-Neto, 2021; Xu, 2016). Once pre-processed, enviromic data has many applications which may also benefit from AI. Those covered here will include environment classification, and genomic prediction integration.

Environmental classification is the identification of mega- or micro-environments within the TPE. Mega-environments are contiguous or dis-contiguous areas of broadly similar environmental factors, consumer preferences, and scales of production (Rajaram et al., 1994). Within the scope of plant breeding, it is generally not necessary to use all environmental factors for environmental classification, but rather focusing on environmental factors that limit or are otherwise particularly relevant to the GEI response for trait(s) being selected upon. When a TPE consists of multiple mega- or micro-environments, a range of GEI responses are possible. METs attempt to capture the possible phenotypic responses of cultivars to the wide range of environmental variation that may occur in actual crop production. Environment classification can be informative in the MET design stage for site selection (Bänziger et al., 2006; Crespo-Herrera et al., 2022; Xu, 2016). Environment classification has been approached using a number of different methods such as factor analysis (Rogers et al., 2021), hierarchical cluster analysis (Crespo-Herrera et al., 2022), principal component analysis, enviromic assemblies (Costa-Neto et al., 2021). Other unsupervised ML algorithms for cluster analysis, pattern discovery, or data reduction may also be relevant for environmental classification.

In the enviromic assembly approaches to environmental classification of Costa-Neto et al. (2021), environmental similarity was assessed using enviromic markers. Costa-Neto et al. (2021) constructed discrete markers that assessed stress and non-stress conditions directly from raw envirotyping data based on prior knowledge of the environment factor thresholds for plant

stress. In this case enviromic markers accomplished the dimensionality reduction task that is often necessary for -omic data.

Environmental classification is not essential for enviromic-integrated genomic prediction (GP). Following METs, characterization of environments into continuous environmental indices can assist in performance prediction with GP under GEI. Critical environmental regressor through informed search (CERIS) was developed to generate an environmental index to quantitatively connect all tested environments (Guo et al., 2020; Li et al., 2018; Li et al., 2021; Li et al., 2022; Mu et al., 2022). The environmental index was chosen from combinations of environmental variables and growth periods to be statistically correlated with the overall performance of tested genotypes across environments, biologically relevant based on physiology, and estimable for new environments. With the environmental index from CERIS, joint genomic regression analysis (JGRA, so CERIS-JGRA) has been shown to be promising for performance prediction in multiple traits and multiple crops (Guo et al., 2020; Li et al., 2018; Li et al., 2021; Mu et al., 2022). Within the CERIS-JGRA frame, different GP models can be used. Resende et al. (2021) used a random forest procedure to interpolate yield prediction across an environmental gradient of observed test sites and neighboring unobserved TPE regions. It was proposed that any type of powerful kriging approach could be appropriate for this method of environmental indexing (Resende et al., 2021). Kriging is common method for spatiotemporal interpolating of GIS data, but other ML and NN approaches have also been demonstrated to be successful for broad spatiotemporal interpolation tasks (Amato et al., 2020; Wu et al., 2021).

The advantage of building or selecting an environmental index is the simplicity with which genetic effects and environmental factors can be integrated into a genomic prediction model. This approach focuses on identifying major patterns across environments and between environmental variables during a critical crop growth stage and modeling the associated effect on crop performance. This approach is different from both other approaches where either all values of many environmental variable throughout the season were used in constructing an environmental relationship matrix and approaches where crop growth models were used to identify the significant environmental variables to be fitted as a covariate in the genomic prediction model (Cooper et al., 2016; Heslot et al., 2014; Jarquín et al., 2014) An environmental index can be used as the explanatory variable in a linear regression to generate reaction norms predicting trait performance across the TPE (Li et al., 2018; Li et al., 2021; Resende et al., 2021).

This format for integrating the environmental dimension separates feature selection from prediction. Alternative AI approaches that integrate feature selection into model training are increasing in relevance as the amount of envirotyping data increases and may complement NN based GP approaches. It is expected that AI approaches can be designed to outperform current methods such as CERIS-JGRA when a single environmental index does not capture an adequate amount of variation in data or when feature selection and model building should be integrated in the combined space of genotype and environment for performance prediction. Due to the inclusivity of these approaches for integrating other types of data into a GP model, these approaches are covered in Section 3.4.2.

3.3 Crop growth modeling and management

Crop growth models (CGMs) use quantitative descriptions of ecophysiological processes to model plant growth and development as influenced by environmental conditions and crop management, which are specified for the model as input data (Hodson and White, 2010). Another term used interchangeably with CGMs is Ecophysiological Crop Models (ECMs). The basic notion underlying CGMs is to use systems of differential equations to represent the temporal dynamics of plant physiological processes, morphological variables, and selected environmental variables to model plant growth, dry matter production, and grain yield. Crop modeling has significantly expanded in recent years and CGMs have been established for all major crops, many minor crops, and a number of weed species. More importantly, CGMs have become an important decision-making tool for crop management and is now used in breeding programs to connect with molecular biology (Cooper et al., 2014). More importantly, crop modeling has become a major tool for tackling challenges in climate change, global food security, and bioenergy (Jägermeyr et al., 2021; Lobell et al., 2013).

Like many other crop model platforms (e.g., DSSAT, EPIC, STICS, WOFOST, ORYZA, CROPSYST, RZWQM, TOA, IMPACT, SWAP, and GTAP), the Agricultural Production Systems sIMulator (APSIM) contains a suite of modules that enable the simulation of systems that cover a range of plant, soil, climate and management interactions (Holzworth et al., 2014). APSIM is an open-source, advanced simulator of farming systems with many crop models for different species, soil water, nitrogen, temperature and environmental models together with advanced management capabilities in a modular design. Realizing the needs to

advance CGM research to address challenges from climate change, the Agricultural Model Intercomparison and Improvement Project (AgMIP) was established (Rosenzweig et al., 2013; Rötter et al., 2011) and has over 1000 members (https://agmip.org).

Within the CGM paradigm, different crop species and varieties within a species are represented by differing numeric values within vectors of constants embedded within the differential equations. Some example parameters are radiation use efficiency; flowering time when expressed in photothermal units; maximum leaf area; phyllochron intervals measured in degree-days, etc. Because such constants often encode environmental sensitivities, the premise is that they will have higher heritabilities than the more complex traits encoded by interwoven processes within the bulk of the CGM. A prediction scheme for interrelating crop phenotypes and genotypes assumes that these constants are genetically determined and, therefore, predictable from allelic and/or marker data via any of the genomic methods described above (Hammer et al., 2006; Reymond et al., 2003; Technow et al., 2015; White and Hoogenboom, 1996; Yin et al., 1999). To predict the trait behaviors of novel genotypes in untested environments, one would (1) use genomic prediction to estimate the constants for the line and then (2) use the CGM, as driven by time series of environmental data characterizing any location-year of interest to predict the resulting phenotype.

While individual studies have been conducted to integrate AI and CGM, the overall fusing of knowledge-driven CGM with data-driven AI was termed as Knowledge- and Data-Driven Modeling (KDDM) (Zhang et al., 2023b). There are many points where AI can enhance CGM, including (1) relating easily taken sensor data to needed but hard-to-measure CGM inputs, (2) phenotyping for model calibration or in-season state variable correction, and (3) improving computational efficiency by AI. For example, a random forest model was used to compute a developmental stage dependent harvest index of wheat in APSIM (Feng et al., 2019), and radial bias function network was leveraged within a CGM to study tomato growth (Fan et al., 2015). More interestingly, when ML methods such as random forest, XGBoost and other ensembles were coupled with CGM for yield prediction in maize, accuracy was much increased (Shahhosseini et al., 2021).

In addition, ML methods can replace the traditional genomic prediction approach within the CGM framework. In a recent study, CNN was found to perform similarly or better than standard genomic prediction methods when sufficient genetic, environmental, and management data were provided (Washburn et al., 2021). In this case, CGM outputs were used as

additional inputs to the ML prediction models, highlighting the value of integration in KDDM. In a follow-up study, different optimization strategies (consecutive optimization and simultaneous optimization) for creating a multi-modal deep neural network were investigated and these deep learning models were compared against a set of classic ML and statistical methods (Kick et al., 2023). The consecutively optimized deep CNN model was found to have a slightly higher average error than the best BLUP model, but to be more consistent in its performance across model replicates.

3.4 Phenomics and other multi-omics

3.4.1 Image-based phenotyping

3.4.1.1 Image acquisition

Phenotyping has been revolutionized by high-throughput methods, particularly image-based phenotyping. With traditional plant phenotyping, we were often limited to characteristics identifiable to the human eye. Today, images can be acquired for studying crop phenotypes with a diversity of sensors, external conditions, and traits. These image-based methods are fast and non-destructive and can be used to capture complex trait information.

The methods selected for image-based phenotyping will depend on the crop, trait, developmental stages, and resources available (Chawade et al., 2019). However, the most common type of sensors used to collect phenotype data include red, green, blue (RGB) cameras, RGB-depth cameras, hyperspectral cameras, thermal infrared cameras, near-infrared cameras, light detection and ranging (LiDAR) devices, and computed tomography (CT) scanners (Li et al., 2020).

RGB cameras image the visible spectral range (380–800 nm) and are capable of capturing many phenotypes that are otherwise visually phenotyped. Two dimensional RGB images have been used for measuring biomass, leaf area, root architecture, leaf diseases and yield (Li et al., 2014). RGB cameras can also aid in more precisely quantifying traits that are difficult or time-consuming to score by the human eye. Stereo vision systems employ two mono-vision cameras like an RGB camera to generate 3D images. Stereo vision has an advantage over mono vision when it comes to structural features. This can be especially important in field conditions where plants cannot be staged in trait relevant positions.

Hyperspectral imaging captures one dimension of spectral information within visible plus non-visible range (380–2500 nm) and two dimensions of spatial information via point spectroscopy to generate a 3D matrix

(Liu et al., 2020). Hyperspectral imaging has had more limited use in UAV and ground-based vehicle systems compared to the more RGB cameras, largely due to the increased cost. Hyperspectral UAV systems have been used to measure chlorophyll in barley (Aasen et al., 2015) and to estimate biomass in wheat (Honkavaara et al., 2013). Hyperspectral images have been more widely collected using satellites or high-altitude aircraft. Satellite hyperspectral images have been used to predict yield and biomass in wheat (Tattaris et al., 2016). Hyperspectral imaging is relevant for measuring changes in growth dynamics through vegetative indices, water contents, and pigment composition (Li et al., 2014). The near-infrared (NIR) spectrum (800–2500 nm) is another commonly imaged range. NIR imaging is often done in conjunction with RGB or as a part of hyperspectral imaging. The NIR range is important for the calculation of vegetative indices like NDVI (normalized difference vegetative index).

Thermal infrared (TIR) imaging is able to detect radiation in the thermal range or far infrared (15–1000 μm) range. TIR image pixels represent temperature values and after correcting for environmental factors, these images can be indicative of plant stress (Pineda et al., 2020). TIR is useful for evaluating differences in stomatal conductance which is related to water status response and abiotic stress transpiration rate adaptation (Li et al., 2014).

LiDAR measures the time by taken by a light pulse to be reflected off an object of interest and back to the sensor (Perez-Sanz et al., 2017). LiDAR is another sensor type that generates a 3D image structure and is effective at large and small distances from several centimeters to thousands of kilometers. LiDAR has been used for phenotyping plant height, biomass, and leaf traits, among others (Panjvani et al., 2019; Sun et al., 2018).

Tomographic imaging methods like CT, magnetic resonance imaging (MRI), positron emission tomography (PET) or ultrawideband radar scanning (URS) are other types of sensors that can be used for the take of plant phenotyping via computer vision. These approaches have lagged behind other imaging systems for phenotyping tasks as they have not increased in throughput capabilities as rapidly. Examples of phenotyping with tomographic imaging methods include using MRI for water diffusion and transport evaluation (Windt et al., 2006), CT for root system architecture evaluation (Hargreaves et al., 2009; Windt et al., 2006), and URS for examining the interiors of dense canopies closed to direct vision (Gomez-Garcia et al., 2022).

In addition to the sensors used, the image acquisition system can vary depending on the environmental conditions. Plants may be located in a field,

greenhouse, or laboratory depending on the trait and tissue being studied. In many situations for large-scale field imaging, a sensor is often mounted on remotely controlled equipment, like a satellite, UAV, or ground-based vehicle, while imaging in a greenhouse or laboratory, may also include stationary or handheld systems. Satellite imaging generally relies on images captured by existing satellite systems that are either freely or commercially distributed. Satellite imaging has an advantage over other systems for large plot sizes and dispersed multi-environment trials. Given the scale at which satellite imagery is taken, environmental variability and necessary travel can be minimized with satellite image acquisition (Chawade et al., 2019). However, satellite imaging is limited by revisit frequency, image precision, and cloud cover interfering with image capture (Chawade et al., 2019). UAVs also need to contend with environmental considerations when collecting images but have increased flexibility of the timing and repetition of flights. UAVs include both rotocopter systems like octocopters and hexacopters, and also parachutes, blimps, and fixed wing systems (Sankaran et al., 2015). Rotocopters are flown at altitudes between 10 and 200 m, verse 700 km for satellites. This means UAVs collect images at a higher spatial resolution than satellites. Quality orthomosaics are necessary for downstream research tasks. However, generating high-quality orthomosaics can be a hurdle for use of UAVs. Inaccuracies can be induced by lens distortion, ground sample distance (physical distance represented by the span of one image pixel), degree of image overlap, and position estimating equipment (Chawade et al., 2019). Such inaccuracies would need to be corrected during orthomosaic building.

Proximal phenotyping through ground-based sensing approaches can be stationary mounted, handheld, and ground-based vehicle mounted systems (Deery et al., 2014). Stationary and handheld systems can reduce costs compared to vehicle systems, but time and labor associated with these methods increase as the number of plants increase. Ground-vehicle systems can reduce labor and time inputs for a large number of plots and also provide further increased spatial resolution over UAV systems. However, ground vehicles still generally require more time over UAV systems for imaging an equivalently sized plot which can increase the amount of temporal variation (Chawade et al., 2019).

3.4.1.2 Data processing

In addition to the imaging system used, image processing and data processing are critical steps for image-based phenotyping. The computer

vision approaches discussed in the previous computer vision section (Section 2.4) remain true for plant phenotype. Additionally, prior reviews specific to several topics within computer vision for plant phenotyping such as ML and DL approaches for plant stress phenotyping (Gill et al., 2022; Singh et al., 2016; Singh et al., 2018); DL approaches for leaf classification, disease detection, plant recognition, and fruit counting (Kamilaris and Prenafeta-Boldú, 2018); and CNNs for plant phenotyping (Jiang and Li, 2020) are also available. Rather, recent applications developed specifically for phenotyping will be detailed here with particular consideration for NN-based applications.

Several categories of traits represent the most studied crop phenotypes, those include root morphology, leaf characteristics, biomass, yield-related traits, photosynthetic efficiency, and biotic and abiotic stress response (Jiang and Li, 2020; Li et al., 2014; Yang et al., 2013).

Root phenotyping has remained among phenotyping methods with high labor costs. In the field, roots cannot be imaged non-destructively with a UAV or ground-based vehicle fitted with an RGB camera as many above ground traits can be. Approaches that are root phenotyping specific have needed to be developed. One approach is to grow seedlings in the lab using containers that can be opened for imaging such as agar plates or germination paper. Falk et al. (2020) introduced a root phenotyping approach that utilized a fixed position RGB camera to image plants growing on germination paper. Growth pouch units were manually transferred to and from the imaging platform yielding a throughput of approximately 60–100 seedlings per hour. Image processing relied on the self-developed software ARIA 2.0 that was built on top of the original ARIA (Automatic Root Image Analysis) tool (Pace et al., 2014). ARIA 2.0 relied on a convolutional autoencoder NN architecture that eliminated the need for separate preprocessing and feature extraction steps.

Lube et al. (2022) developed a system that images up to 18 Petri dish plates simultaneously and processed up to 100 images per hour using an RGB camera fitted with a visible bandwidth pass filter. Petri dishes were mounted in a carousel that allowed for automated image acquisition making the system suited to time-lapse captures. This system, called MultipleXLab, also used an image analysis pipeline built on self-developed NNs called SeedNet and RootNet for seed and root pixel identification. Both networks are based on the U-Net architecture, a fully convolutional encoder-decoder network, that was developed for image segmentation (Ronneberger et al., 2015).

ChronoRoot, as described by Gaggion et al. (2021), utilized time series RGB images of agar plated seedlings and developed an automated image processing pipeline that used NN-based segmentation. Two image processing models were developed. The first was a modified ResUNet model. ResUNet is a derivative of the U-Net architecture that uses residual blocks in place of convolutional layers (He et al., 2015). Residual blocks are characterized by skip connections (feedforward, supra-connections) that provide memory without the use of gates like those used by RNNs. The residual block is made up of both skip connections and the conventional feedforward type connections that are bypassed by the skip connections. ChronoRoot used ResUNet as the base of a DSResUNet model. The DSResUNet model uses the ResUNet to generate intermediate feature maps that are then further processed with two convolutional layers. This introduces an additional loss term for both the initial segmentation map generated by the ResUNet framework and for final output succeeding the convolutional layers. The second segmentation method used by the ChronoRoot system was an ensemble method that combined five popular models: U-Net, ResUNet, DSResUNet, SegNet, and DeepLab v3. SegNet is another type of fully convolutional encoder-decoder neural network (Badrinarayanan et al., 2017). SegNet differs from U-Net in the information that is transferred between encoding and decoding sections of the network. U-Net transfers the whole feature map, while SegNet only transfers pooled indices. DeepLab v3 uses atrous (dilated) convolutions for feature extraction (Chen et al., 2018). Traditional convolutions apply filters to neighboring values in the input. Atrous convolutions skip some values per a defined spacing to obtain a wider field of view with the same size kernel. For example, a 2×2 filter might be used on a 3×3 field of view where only the corners and center values are applied to the filter. The NN output was considered a soft-segmentation and was further refined in post-processing stages to skeletonize the root images and accommodate temporal tracking.

Thesma and Mohammadpour Velni (2022) used the same root image dataset generated by Gaggion et al. (2021) but approached segmentation through a conditional generative adversarial network (cGAN). One can think of cGANs as a GAN where the generator and discriminator are conditioned on supplementary information (Mirza and Osindero, 2014). Since GANs are a type of unsupervised learning, the use of supplementary information (like class labels or numerical values) also allows cGANs to be considered a type of multi-modal NN. In the case of this root imaging approach, a Pix2PixHD (Wang et al., 2018a) cGAN architecture derivative

(Isola et al., 2017) and the root annotations, used as labels for supervised methods, became the supplementary, conditional information. The cGAN approach was used to generate additional root images that were segmented using SegNet. The accuracy of the cGAN + SegNet system used here was similar, but slightly lower than the UNet and DSResUNet models. However, mean IOU (intersection over union) and dice score were higher for the SegNet approach. Mean IOU represents the ratio of area of intersection of the predicted and observed bounding boxes over their union. Dice score represents a similar metric as mean IOU but uses the formula of 2 times the area of overlap over the area of the predicted bounding box plus the area of the observed bounding box.

Root phenotyping has successfully adopted approaches from broader computer vision processing techniques but is still bottlenecked by the need for increased staging of the roots. Automated field root imaging remains an inaccessible task. As aboveground plant components, leaf, shoot and growth traits (i.e. height, biomass, leaf diseases, and leaf angle) have been more likely to achieve the throughput expected to justify more complicated imaging systems.

Nguyen et al. (2023) used a swarm of UAVs with each fitted with different sensors. The first UAV used a hyperspectral sensor (capturing the visible to NIR spectra), thermal sensor, and a GPS (global positioning system). The second UAV included a LiDAR sensor and an RGB camera. The third UAV consisted of a thermal camera, an RGB camera, and a multispectral camera. Images taken by these UAVs were used to estimate cob biomass, dry grain yield, dry stalk biomass, harvest index, grain density, grain nitrogen content, grain nitrogen utilization efficiency, and plant nitrogen content. Image processing included generating orthomosaics from thermal and hyperspectral imagery and generating density maps from the LiDAR data. After image processing, three approaches were used to process data for phenotype prediction: normalized difference spectral index (NDSI), ML, and DL approaches. The NDSI approach combined all data modalities from all the different sensors and used the combined values to investigate correlations with the trait data. The ML approach used a handcrafted feature set representing a fusion of vegetation, structural, and thermal indices. The feature set was used with the ML approaches (support vector regression and RF). In the third, DL was used for single-modality and multi-modality prediction. The DL models were CNNs designed for 2D and 3D data. For the multi-modal CNN, the three CNNs, one for each of the hyperspectral, LiDAR, and thermal single-modalities, were joined with a block of

feedforward prediction layers. There was variable contribution of different data modalities to different traits with hyperspectral data being nearer to a primary contributor than others. Automated feature extraction via DL proved feasible with similar prediction success compared to handcrafted feature sets. Multi-modal DL was shown to provide a computational advantage over single modal approach with less time and resources required for generating predictions.

Xiang et al. (2023) used a ground-based vehicle (PhenoBot3.0) equipped with multiple tiers of stereo RGB cameras to investigate leaf angle architecture. These authors developed an image processing pipeline, AngleNet, that was built on a CNN for key point detection followed by bounding box detection. A disparity map and the 2D color images were used to generate the 3D leaf model. Keypoints could be mapped onto the 3D leaf model and the leaf angle evaluated. The generation of a 3D model made the AngleNet system robust to diverse leaf orientations that are possible for a camera to capture in field conditions. This automated imaging pipeline demonstrates the increase in throughput possible with computer vision techniques with a 20 plant plot captured in 3–4 s verse 8–10 min required for manually measuring the same quantity.

3.4.2 Multi-omic characterization

Multi-omic studies may better represent biological processes than single-omic analysis and can bridge the gap between sequenced genotypes and observed phenotypes by capturing additional genome-level variation. Compared to genomic data, intermediate -omic data can be resource and cost intensive, which makes it difficult to capture on a large scale. Westhues et al. (2019) has shown that -omics data can be successfully imputed from a subset of inbred lines and the result can be used for trait predictions. In terms of imputation, DL is a very capable method because it can utilize correlations among different -omic datasets for multi-omic imputation (Song et al., 2020). Autoencoders have been successfully used on single-omic imputation (Chen and Shi, 2019). DL has also been investigated as a part of a multi-modal structure for integrating multi-omic data with other heterogeneous sets of features (Li et al., 2016). Multi-modal structures have the advantage allowing different NN architectures to be selected which best fit a given data modality within an integrative system (Song et al., 2020). DL and other types of ML also have other more specific multi-omic applications in addition to the integration of multi-omic data into a GS model.

3.4.2.1 Single cell RNAseq

Pluripotent stem cells in meristems give rise to all the above- and below-ground organs. Profile transcriptomes of a large population of single cells is a state-of-the-art approach to study how the stem-cell niches are maintained and organized through the data-driven discovery approach (Liu et al., 2021; Xu et al., 2021; Zhang et al., 2021). Unsupervised ML methods, t-Stochastic Neighborhood Embedding (t-SNE) (Van Der Maaten and Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018), are widely applied to extract critical information from single-cell transcriptomes (Xu et al., 2021; Zhang et al., 2021). These unsupervised learning algorithms represent state-of-the-art dimensionality reduction algorithms to reveal prominent patterns from high dimension data. In combination with unique experimental designs, these algorithms are powerful to unveil biological insights. Sequencing transcriptomes of 327 cells sampled from maize shoot apical meristem showed that cells in the tip tend to have high activities in maintaining the genome integrity and that the divergent molecular mechanisms of stem cell regulation exist between Arabidopsis and maize (Satterlee et al., 2020). The regulatory programs associated with cell types at the root tip are highly conserved in rice cultivars, while some programs are divergent between rice and Arabidopsis (Zhang et al., 2021). By analyzing allele specific transcriptomes from single pollen precursors from a maize hybrid along the developmental stages, novel discoveries of diploid-derived transcripts persist long into the haploid phase and a rapid transition to monoallelic expression around pollen mitosis I were reported (Nelms and Walbot, 2022).

3.4.2.2 Regulatory genomics

Selection of gene expression pattern change is a major mechanism in domesticating and improving crops (Chen et al., 2021). However, at the genome-wide level, understanding of how gene expression is determined by the linear (and non-linear) combinations of nucleotides remains limited. Sequencing technologies have the capacity to assemble gapless genomes, and accurately and efficiently measure expressions for tens of thousands of genes across many conditions of the same organism. Therefore, compiling large training datasets (labeled with gene expressions) is feasible. This opens the possibility of systemically investigating learning from data by designing and training ML and DL algorithms (Avsec et al., 2021; Novakovsky et al., 2023). Using frequencies of mononucleotides and dinucleotides from

six regions of each gene as features, random forests were trained to predict the transcriptional response to cold stress across plants (Meng et al., 2021). Given the frequent genome duplication events in crops, it was recommended that evolutionary relatedness be incorporated into CNNs designed for transcription activities (Washburn et al., 2019). Similarly, deep CNNs have also been expanded into other critical genomic features, or molecular phenotypes, such as DNA accessibility and chromatin, chromosome interaction, and 3D organization (Kelley et al., 2016).

3.4.2.3 Population genomics

A major frontier of applying ML and DL for genomics is identifying and cataloging polymorphisms beyond SNPs. Deep NN algorithms have been trained to call small variants from short resequencing reads aligned to a single reference genome (Luo et al., 2019; Poplin et al., 2018; Torracinta and Campagne, 2016). Multiple high-quality genome assemblies build the foundation to unveil the full spectrum of DNA polymorphisms of small and large size, but the complexity of DNA polymorphisms across pan-genomes requires innovative analytic strategies. Unsupervised ML algorithms have been tested to streamline the identification of large structural variations for genes of interest across assemblies (Zhang et al., 2023a). To leverage state-of-the-art computer vision DL frameworks, strategies such as SVision via the sequence-to-image schema have been tested to resolve complex structure (Lin et al., 2022a). ML algorithms also demonstrated promising potential for inferring evolutionary signatures from DNA polymorphisms (Schrider and Kern, 2018). With available pan-genomes from accessions representing the genetic global diversity in different crops, novel ML and DL algorithms may be used to combine assemblies and short reads to gain a comprehensive understanding of population genomics.

4. Future of AI for crop improvement

Artificial intelligence is a diverse field, as are the tasks associated with crop improvement. AI applications can appear in individual components of a system or as the overall system structure in different aspects of crop improvement. Together, these joint applications have been revolutionizing agricultural practices in the field, greenhouse, and laboratory. Just as genomic selection has been enhancing our capacity beyond the traditional breeding and marker-assisted breeding approaches, so may neural network-based

genomic selection as we continue to generate big data and novel high-throughput approaches. With increased focus on the multiple dimensions affecting phenotypes, new methods must be capable of considering diverse genomic, enviromic, and multi-omic datasets that are collected with rapidly changing technologies. A challenge that those involved in crop improvement are familiar with is keeping up. For the last century that challenge has been framed around keeping up with the demand of a growing world population, and this continues to be the major concern. Now, however, that challenge is further complicated by climate change. Going forward into an era that may become dominated by AI approaches toward all types of problems, the need to educate both scientists and the public about AI is becoming clear.

For individual researchers in crop improvement to adopt AI methods, it is desirable to know the potential advantages of AI methods over traditional methods when data size is relatively small and how to assess the signal-noise ratio before integrating diverse data sources. It is also desirable to compare the efficiency of applying new AI methods to observational studies to the efficiency of designed cropping experiments that have and will continue to be a major data source for many aspects of crop improvement. For producers, it is desirable to leverage AI to achieve sustainable production via enhanced on-farm production management. The improved capacity to monitor and forecast crop growth and health under different genetic × environment × management combinations would greatly facilitate the decision-making process. At a larger space and a longer time scale, how to leverage AI to generate guidelines and action plans for crop improvement should be examined.

J.B.S. Haldane spent his life operating at the intersection of genetics, evolutionary biology, physiology, and mathematical statistics. Indeed, in 1929 he advanced foundational notions relevant to later theories on the chemical origin of life. He also once famously said (as frequently paraphrased by others), "The universe is not only stranger than we imagine; it is stranger than we can imagine." Given the twin advancing/adverse drumbeats of population growth and climate change, it might well be that our ability to imagine what is possible at the intersection of breeding, genetics, crop management, and AI/ML is the existential challenge of our time. However, if the prior success achieved leveraging innovative technologies for crop improvement is any indication, the future of AI-assisted crop improvement is bright.

Acknowledgments

This work was supported by the National Science Foundation (IOS-2210259 and OIA-1826820), the Agriculture and Food Research Initiative competitive grant (2021-67013-33833) and the Hatch project (1021013) from the USDA National Institute of Food and Agriculture, the In-House Project 2090-21000-033-00D of the USDA Agricultural Research Service, the Iowa State University Raymond F. Baker Center for Plant Breeding, and the Iowa State University Plant Sciences Institute.

References

Aasen, H., Burkart, A., Bolten, A., Bareth, G., 2015. Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: from camera calibration to quality assurance. ISPRS J. Photogramm. Remote Sens. 108, 245–259. https://doi.org/10.1016/j.isprsjprs.2015.08.002.

Amato, F., Guignard, F., Robert, S., Kanevski, M., 2020. A novel framework for spatio-temporal prediction of environmental data using deep learning. Sci. Rep. 10, 22243. https://doi.org/10.1038/s41598-020-79148-7.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J.R., Grabska-Barwinska, A., Taylor, K.R., Assael, Y., Jumper, J., Kohli, P., Kelley, D.R., 2021. Effective gene expression prediction from sequence by integrating long-range interactions. Nat. Methods 18, 1196–1203. https://doi.org/10.1038/s41592-021-01252-x.

Azodi, C.B., Bolger, E., McCarren, A., Roantree, M., de los Campos, G., Shiu, S.-H., 2019. Benchmarking parametric and machine learning models for genomic prediction of complex traits. G3 Genes Genomes Genetics 9, 3691–3702. https://doi.org/10.1534/g3.119.400498.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, pp. 1–15.

Bänziger, M., Setimela, P.S., Hodson, D., Vivek, B., 2006. Breeding for improved abiotic stress tolerance in maize adapted to southern Africa. Agric. Water Manag. 80, 212–224. https://doi.org/10.1016/j.agwat.2005.07.014.

Bengio, Y., 2009. Learning Deep Architectures for AI. vol. 2 Foundations and Trends in Machine Learning, pp. 1–127, https://doi.org/10.1561/2200000006.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 1–9.

Bernardo, R., Yu, J., 2007. Prospects for genomewide selection for quantitative traits in maize. Crop Sci. 47, 1082–1090. https://doi.org/10.2135/cropsci2006.11.0690.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Improving image generation with better captions. Preprint. https://openai.com/dall-e-3.

Biau, G., Scornet, E., 2016. A random forest guided tour. TEST 25, 197–227. https://doi.org/10.1007/s11749-016-0481-7.

Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L.,

Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., Liang, P., 2021. On the opportunities and risks of foundation models. arXiv.2108.07258v3.

Breiman, L., 2001a. Statistical modeling: the two cultures. Stat. Sci. 16, 199–231.

Breiman, L., 2001b. Random forests. Machine learning 45, 5–32.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Buckler, E.S., Holland, J.B., Bradbury, P.J., Acharya, C.B., Brown, P.J., Browne, C., Ersoz, E., Flint-Garcia, S., Garcia, A., Glaubitz, J.C., Goodman, M.M., Harjes, C., Guill, K., Kroon, D.E., Larsson, S., Lepak, N.K., Li, H., Mitchell, S.E., Pressoir, G., Peiffer, J.A., Rosas, M.O., Rocheford, T.R., Romay, M.C., Romero, S., Salvo, S., Villeda, H.S., Sofia da Silva, H., Sun, Q., Tian, F., Upadyayula, N., Ware, D., Yates, H., Yu, J., Zhang, Z., Kresovich, S., McMullen, M.D., 2009. The genetic architecture of maize flowering time. Science 325, 714–718. https://doi.org/10.1126/science.1174276.

Bzdok, D., Altman, N., Krzywinski, M., 2018. Statistics versus machine learning. Nat. Methods 15, 233–234. https://doi.org/10.1038/nmeth.4642.

Chakraborti, T., Isahagian, V., Khalaf, R., Khazaeni, Y., Muthusamy, V., Rizk, Y., Unuvar, M., 2020. From robotic process automation to intelligent process automation: emerging trends. In: Lecture Notes in Business Information Processing 393 LNBIP, pp. 215–228.

Chawade, A., van Ham, J., Blomquist, H., Bagge, O., Alexandersson, E., Ortiz, R., 2019. High-throughput field-phenotyping tools for plant breeding and precision agriculture. Agronomy 9, 258. https://doi.org/10.3390/agronomy9050258.

Chen, J., Shi, X., 2019. Sparse convolutional denoising autoencoders for genotype imputation. Gene 10, 652. https://doi.org/10.3390/genes10090652.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018. DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40, 834–848. https://doi.org/10.1109/TPAMI.2017.2699184.

Chen, Q., Li, W., Tan, L., Tian, F., 2021. Harnessing knowledge from maize and rice domestication for new crop breeding. Mol. Plant 14, 9–26. https://doi.org/10.1016/j.molp.2020.12.006.

Cheng, J., Bendjama, K., Rittner, K., Malone, B., 2021. BERTMHC: improved MHC–peptide class II interaction prediction with transformer and multiple instance learning. Bioinformatics 37, 4172–4179. https://doi.org/10.1093/bioinformatics/btab422.

Cooper, M., Messina, C.D., Podlich, D., Totir, L.R., Baumgarten, A., Hausmann, N.J., Wright, D., Graham, G., 2014. Predicting the future of plant breeding: complementing empirical evaluation with genetic prediction. Crop Past. Sci. 65, 311. https://doi.org/10.1071/CP14007.

Cooper, M., Technow, F., Messina, C., Gho, C., Totir, L.R., 2016. Use of crop growth models with whole-genome prediction: application to a maize multienvironment trial. Crop Sci. 56, 2141–2156. https://doi.org/10.2135/cropsci2015.08.0512.

Costa-Neto, G., Fritsche-Neto, R., 2021. Enviromics: bridging different sources of data, building one framework. Crop Breed. Appl. Biotechnol. 21, 393521–393533. https://doi.org/10.1590/1984-70332021V21SA25.

Costa-Neto, G., Crossa, J., Fritsche-Neto, R., 2021. Enviromic assembly increases accuracy and reduces costs of the genomic prediction for yield plasticity in maize. Front. Plant Sci. 12. https://doi.org/10.3389/fpls.2021.717552.

Crespo-Herrera, L.A., Crossa, J., Vargas, M., Braun, H.-J., 2022. Defining target wheat breeding environments. In: Reynolds, M.P., Braun, H.-J. (Eds.), Wheat Improvement. Springer International Publishing, Cham, pp. 31–45, https://doi.org/10.1007/978-3-030-90673-3_3.

Crevier, D., 1993. AI: The Tumultuous History of the Search for Artificial Intelligence. Basic Books, Inc.

Cropper, A., Dumančić, S., 2022. Inductive logic programming at 30: a new introduction. J. Artif. Intell. Res. 74, 765–850. https://doi.org/10.1613/jair.1.13507.

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., Burgueño, J., González-Camacho, J.M., Pérez-Elizalde, S., Beyene, Y., Dreisigacker, S., Singh, R., Zhang, X., Gowda, M., Roorkiwal, M., Rutkoski, J., Varshney, R.K., 2017. Genomic selection in plant breeding: methods, models, and perspectives. Trends Plant Sci. 22, 961–975. https://doi.org/10.1016/j.tplants.2017.08.011.

Daniel, C., Taylor, J., Nowozin, S., 2016. Learning step size controllers for robust neural network training. In: Proceedings of the AAAI Conference on Artificial Intelligence. 30, pp. 1519–1525, https://doi.org/10.1609/aaai.v30i1.10187.

d'Avila Garcez, A.S., Broda, K.B., Gabbay, D.M., 2002. Introduction and overview. In: Neural-Symbolic Learning Systems: Foundations and Applications. Springer, London, pp. 1–12, https://doi.org/10.1007/978-1-4471-0211-3_1.

de los Campos, G., Hickey, J.M., Pong-Wong, R., Daetwyler, H.D., Calus, M.P.L.L., 2013. Whole-genome regression and prediction methods applied to plant and animal breeding. Genetics 193, 327–345. https://doi.org/10.1534/genetics.112.143313.

Deery, D., Jimenez-Berni, J., Jones, H., Sirault, X., Furbank, R., 2014. Proximal remote sensing buggies and potential applications for field-based phenotyping. Agronomy 4, 349–379. https://doi.org/10.3390/agronomy4030349.

Deng, L., 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. In: APSIPA Transactions on Signal and Information Processing. 3, pp. 1–29, https://doi.org/10.1017/atsip.2013.9.

Dey, R., Salem, F.M., 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, pp. 1597–1600, https://doi.org/10.1109/MWSCAS.2017.8053243.

Domingos, P., Lowd, D., Kok, S., Nath, A., Poon, H., Richardson, M., Singla, P., 2016. Unifying logical and statistical AI. In: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science. ACM, New York, NY, USA, pp. 1–11, https://doi.org/10.1145/2933575.2935321.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., Dehmer, M., 2020. An introductory review of deep learning for prediction models with big data. Front. Artif. Intell. 3. https://doi.org/10.3389/frai.2020.00004.

Falk, K.G., Jubery, T.Z., Mirnezami, S.V., Parmley, K.A., Sarkar, S., Singh, A.A.K., Ganapathysubramanian, B., Singh, A.A.K., 2020. Computer vision and machine learning enabled soybean root phenotyping pipeline. Plant Methods 16, 1–19. https://doi.org/10.1186/s13007-019-0550-5.

Fan, X.-R., Kang, M.-Z., Heuvelink, E., de Reffye, P., Hu, B.-G., 2015. A knowledge-and-data-driven modeling approach for simulating plant growth: a case study on tomato growth. Ecol. Model. 312, 363–373. https://doi.org/10.1016/j.ecolmodel.2015.06.006.

Feng, P., Wang, B., Liu, D.L., Waters, C., Yu, Q., 2019. Incorporating machine learning with biophysical model can improve the evaluation of climate extremes impacts on wheat yield in South-Eastern Australia. Agric. For. Meteorol. 275, 100–113. https://doi.org/10.1016/j.agrformet.2019.05.018.

Feurer, M., Hutter, F., 2019. Hyperparameter optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (Eds.), Automated Machine Learning: Methods, Systems, Challenges. Springer International Publishing, Cham, pp. 3–33, https://doi.org/10.1007/978-3-030-05318-5_1.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, pp. 148–156. http://dl.acm.org/citation.cfm?id=3091696.3091715.

Friedman, J., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29, 1189–1232.

Gaggion, N., Ariel, F., Daric, V., Lambert, É., Legendre, S., Roulé, T., Camoirano, A., Milone, D.H., Crespi, M., Blein, T., Ferrante, E., 2021. ChronoRoot: high-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. GigaScience 10, 1–15. https://doi.org/10.1093/gigascience/giab052.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A., 2020. Shortcut learning in deep neural networks. Nat. Mach. Intell. 2, 665–673. https://doi.org/10.1038/s42256-020-00257-z.

Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Comput. 12, 2451–2471. https://doi.org/10.1162/089976600300015015.

Gers, F.A., Schraudolph, N.N., Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. J. Mach. Learn. Res. 3, 115–143.

Gianola, D., Okut, H., Weigel, K.A., Rosa, G.J.M., 2011. Predicting complex quantitative traits with Bayesian neural networks: a case study with Jersey cows and wheat. BMC Genet. 12, 87. https://doi.org/10.1186/1471-2156-12-87.

Gill, T., Gill, S.K., Saini, D.K., Chopra, Y., de Koff, J.P., Sandhu, K.S., 2022. A comprehensive review of high throughput phenotyping and machine learning for plant stress phenotyping. Phenomics 2, 156–183. https://doi.org/10.1007/s43657-022-00048-z.

Girosi, F., Jones, M., Poggio, T., 1995. Regularization theory and neural networks architectures. Neural Comput. 7, 219–269. https://doi.org/10.1162/neco.1995.7.2.219.

Gomez-Garcia, D., Rodriguez-Morales, F., Welch, S., Leuschen, C., 2022. High-throughput phenotyping of wheat canopy height using Ultrawideband radar: first results. IEEE Geosci. Remote Sens. Lett. 19, 1–5. https://doi.org/10.1109/LGRS.2020.3039179.

González-Camacho, J.M., de los Campos, G., Pérez, P., Gianola, D., Cairns, J.E., Mahuku, G., Babu, R., Crossa, J., 2012. Genome-enabled prediction of genetic values using radial basis function neural networks. Theor. Appl. Genet. 125, 759–771. https://doi.org/10.1007/s00122-012-1868-9.

González-Camacho, J.M., Ornella, L., Pérez-Rodríguez, P., Gianola, D., Dreisigacker, S., Crossa, J., 2018. Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. Plant Genome 11, 170104. https://doi.org/10.3835/plantgenome2017.11.0104.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial networks. Commun. ACM 63, 139–144. https://doi.org/10.1145/3422622.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S., 2016. Deep learning for visual understanding: a review. Neurocomputing 187, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116.

Guo, T., Mu, Q., Wang, J., Vanous, A.E., Onogi, A., Iwata, H., Li, X., Yu, J., 2020. Dynamic effects of interacting genes underlying rice flowering-time phenotypic plasticity and global adaptation. Genome Res. 30, 673–683. https://doi.org/10.1101/gr.255703.119.

Hammer, G., Cooper, M., Tardieu, F., Welch, S., Walsh, B., van Eeuwijk, F., Chapman, S., Podlich, D., 2006. Models for navigating biological complexity in breeding improved crop plants. Trends Plant Sci. 11, 587–593. https://doi.org/10.1016/j.tplants.2006.10.006.

Hamuda, E., Glavin, M., Jones, E., 2016. A survey of image processing techniques for plant extraction and segmentation in the field. Comput. Electron. Agric. 125, 184–199. https://doi.org/10.1016/j.compag.2016.04.024.

Hansen, N., 2016. The CMA Evolution Strategy: A Tutorial. arXiv:1604.00772v2.

Hargreaves, C.E., Gregory, P.J., Bengough, A.G., 2009. Measuring root traits in barley (*Hordeum vulgare* ssp. *vulgare* and ssp. *spontaneum*) seedlings using gel chambers, soil sacs and X-ray microtomography. Plant Soil 316, 285–297. https://doi.org/10.1007/s11104-008-9780-4.

Hastie, T., Tibshirani, R., Friedman, J., 2009. Boosting and additive trees. In: The Elements of Statistical Learning, second ed. Springer Series in Statistics, Springer, New York, New York, NY, https://doi.org/10.1007/978-0-387-84858-7_10.

Hayes-Roth, F., 1985. Rule-based systems. Commun. ACM 28, 921–932.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385v1.

Heffner, E.L., Sorrells, M.E., Jannink, J.-L., 2009. Genomic selection for crop improvement. Crop Sci. 49, 1–12. https://doi.org/10.2135/cropsci2008.08.0512.

Heslot, N., Akdemir, D., Sorrells, M.E., Jannink, J.-L., 2014. Integrating environmental covariates and crop modeling into the genomic selection framework to predict genotype by environment interactions. Theor. Appl. Genet. 127, 463–480. https://doi.org/10.1007/s00122-013-2231-5.

Hinton, G.E., Sejnowski, T.J., 1983. Optimal perceptual inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Citeseer, pp. 448–453.

Hinton, G.E., Zemel, R.S., 1993. Autoencoders, minimum description length and Helmholtz free energy. Adv. Neural Inf. Process. Syst. 6, 3–10.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hodson, D., White, J., 2010. GIS and crop simulation modelling applications in climate change research. In: Reynolds, M.P. (Ed.), Climate Change and Crop Production. CABI, Wallingford, pp. 245–262, https://doi.org/10.1079/9781845936334.0245.

Hoehndorf, R., Queralt-Rosinach, N., 2017. Data science and symbolic AI: synergies, challenges and opportunities. Data Sci. 1, 27–38. https://doi.org/10.3233/DS-170004.

Hoff, P.D., 2009. A First Course in Bayesian Statistical Methods. Springer Texts in Statistics, Springer, New York, NY, https://doi.org/10.1007/978-0-387-92407-6.

Holzworth, D.P., Huth, N.I., DeVoil, P.G., Zurcher, E.J., Herrmann, N.I., McLean, G., Chenu, K., van Oosterom, E.J., Snow, V., Murphy, C., Moore, A.D., Brown, H., Whish, J.P.M., Verrall, S., Fainges, J., Bell, L.W., Peake, A.S., Poulton, P.L., Hochman, Z., Thorburn, P.J., Gaydon, D.S., Dalgliesh, N.P., Rodriguez, D., Cox, H., Chapman, S., Doherty, A., Teixeira, E., Sharp, J., Cichota, R., Vogeler, I., Li, F.Y., Wang, E., Hammer, G.L., Robertson, M.J., Dimes, J.P., Whitbread, A.M., Hunt, J., van Rees, H., McClelland, T., Carberry, P.S., Hargreaves, J.N.G., MacLeod, N., McDonald, C., Harsdorf, J., Wedgwood, S., Keating, B.A., 2014. APSIM—evolution towards a new generation of agricultural systems simulation. Environ. Model. Softw. 62, 327–350. https://doi.org/10.1016/j.envsoft.2014.07.009.

Honkavaara, E., Saari, H., Kaivosoja, J., Pölönen, I., Hakala, T., Litkey, P., Mäkynen, J., Pesonen, L., 2013. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. Remote Sens. 5, 5006–5039. https://doi.org/10.3390/rs5105006.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8.

Ilkou, E., Koutraki, M., 2020. Symbolic vs sub-symbolic AI methods: friends or enemies? In: CIKM (Workshops).

Irie, B., Miyake, S., 1988. Capabilities of three-layered perceptrons. In: IEEE International Conference on Neural Networks. IEEE, pp. 641–648, https://doi.org/10.1109/ICNN.1988.23901.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 5967–5976, https://doi.org/10.1109/CVPR.2017.632.

Jackson, P., 1998. Introduction to Expert Systems, third ed. Addison-Wesley Longman Publishing Co., Inc, USA.

Jägermeyr, J., Müller, C., Ruane, A.C., Elliott, J., Balkovic, J., Castillo, O., Faye, B., Foster, I., Folberth, C., Franke, J.A., Fuchs, K., Guarin, J.R., Heinke, J., Hoogenboom, G., Iizumi, T., Jain, A.K., Kelly, D., Khabarov, N., Lange, S., Lin, T.-S., Liu, W., Mialyk, O., Minoli, S., Moyer, E.J., Okada, M., Phillips, M., Porter, C., Rabin, S.S., Scheer, C., Schneider, J.M., Schyns, J.F., Skalsky, R., Smerald, A., Stella, T., Stephens, H., Webber, H., Zabel, F., Rosenzweig, C., 2021. Climate impacts on global agriculture emerge earlier in new generation of climate and crop models. Nat. Food 2, 873–885. https://doi.org/10.1038/s43016-021-00400-y.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning. Springer Texts in Statistics, Springer, New York, New York, NY, https://doi.org/10.1007/978-1-4614-7138-7.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2021a. Linear regression. In: An Introduction to Statistical Learning: With Applications in R. Springer US, New York, NY, pp. 59–128, https://doi.org/10.1007/978-1-0716-1418-1_3.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2021b. Classification. In: An Introduction to Statistical Learning: With Applications in R. Springer US, New York, NY, pp. 129–195, https://doi.org/10.1007/978-1-0716-1418-1_4.

Jannink, J.-L., Lorenz, A.J., Iwata, H., 2010. Genomic selection in plant breeding: from theory to practice. Brief. Funct. Genomics 9, 166–177. https://doi.org/10.1093/bfgp/elq001.

Jarquín, D., Crossa, J., Lacaze, X., Du Cheyron, P., Daucourt, J., Lorgeou, J., Piraux, F., Guerreiro, L., Pérez, P., Calus, M., Burgueño, J., de los Campos, G., 2014. A reaction norm model for genomic selection using high-dimensional genomic and environmental data. Theor. Appl. Genet. 127, 595–607. https://doi.org/10.1007/s00122-013-2243-1.

Jayas, D.S., Ghosh, P.K., Paliwal, J., Karunakaran, C., 2008. Quality evaluation of wheat. In: Sun, D.-W. (Ed.), Computer Vision Technology for Food Quality Evaluation. Elsevier, pp. 351–376, https://doi.org/10.1016/B978-012373642-0.50018-1.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T., 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning. PMLR, 139, pp. 4904–4916. https://proceedings.mlr.press/v139/jia21b.html.

Jiang, Y., Li, C., 2020. Convolutional neural networks for image-based high-throughput plant phenotyping: a review. Plant Phenomics 2020, 1–22. https://doi.org/10.34133/2020/4152816.

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: trends, perspectives, and prospects. Science 349, 255–260. https://doi.org/10.1126/science.aaa8415.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D., 2021. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589. https://doi.org/10.1038/s41586-021-03819-2.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

Kaur, D., Kaur, Y., 2014. Various image segmentation techniques: a review. Int. J. Comput. Sci. Mob. Comput. 3, 809–814.

Kautz, H.A., 2022. The third AI summer: AAAI Robert S. Engelmore memorial lecture. AI Mag. 43, 105–125. https://doi.org/10.1002/aaai.12036.

Kelley, D.R., Snoek, J., Rinn, J.L., 2016. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 26, 990–999. https://doi.org/10.1101/gr.200535.115.

Khaki, S., Wang, L., 2019. Crop yield prediction using deep neural networks. Front. Plant Sci. 10, 621. https://doi.org/10.3389/fpls.2019.00621.

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M., 2022. Transformers in vision: a survey. ACM Comput. Surv. 54, 1–41. https://doi.org/10.1145/3505244.

Kick, D.R., Wallace, J.G., Schnable, J.C., Kolkman, J.M., Alaca, B., Beissinger, T.M., Edwards, J., Ertl, D., Flint-Garcia, S., Gage, J.L., Hirsch, C.N., Knoll, J.E., de Leon, N., Lima, D.C., Moreta, D.E., Singh, M.P., Thompson, A., Weldekidan, T., Washburn, J.D., 2023. Yield prediction through integration of genetic, environment, and management data through deep learning. G3 Genes Genomes Genetics 13, 6. https://doi.org/10.1093/g3journal/jkad006.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment anything. arXiv:2304.02643v1.

Kotsiantis, S.B., 2013. Decision trees: a recent overview. Artif. Intell. Rev. 39, 261–283. https://doi.org/10.1007/s10462-011-9272-4.

Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. AIChE J. 37, 233–243. https://doi.org/10.1002/aic.690370209.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 84–90. https://doi.org/10.1145/3065386.

Kuruvilla, J., Sukumaran, D., Sankar, A., Joy, S.P., 2016. A review on image processing and image segmentation. In: 2016 International Conference on Data Mining and Advanced

Computing (SAPIENCE), pp. 198–203, https://doi.org/10.1109/SAPIENCE.2016.7684170.
Li, L., Zhang, Q., Huang, D., 2014. A review of imaging techniques for plant phenotyping. Sensors 14, 20078–20111. https://doi.org/10.3390/s141120078.
Li, Y., Wu, F.-X., Ngom, A., 2016. A review on machine learning principles for multi-view biological data integration. Brief. Bioinform. 19, 113. https://doi.org/10.1093/bib/bbw113.
Li, X., Guo, T., Mu, Q., Li, X., Yu, J., 2018. Genomic and environmental determinants and their interplay underlying phenotypic plasticity. Proc. Natl. Acad. Sci. 115, 6679–6684. https://doi.org/10.1073/pnas.1718326115.
Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., Yan, X., 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, ISBN: 9781713807933, p. 32.
Li, Z., Guo, R., Li, M., Chen, Y., Li, G., 2020. A review of computer vision technologies for plant phenotyping. Comput. Electron. Agric. 176, 105672. https://doi.org/10.1016/j.compag.2020.105672.
Li, X., Guo, T., Wang, J., Bekele, W.A., Sukumaran, S., Vanous, A.E., McNellie, J.P., Tibbs-Cortes, L.E., Lopes, M.S., Lamkey, K.R., Westgate, M.E., McKay, J.K., Archontoulis, S.V., Reynolds, M.P., Tinker, N.A., Schnable, P.S., Yu, J., 2021. An integrated framework reinstating the environmental dimension for GWAS and genomic selection in crops. Mol. Plant 14, 874–887. https://doi.org/10.1016/j.molp.2021.03.010.
Li, X., Guo, T., Bai, G., Zhang, Z., See, D., Marshall, J., Garland-Campbell, K.A., Yu, J., 2022. Genetics-inspired data-driven approaches explain and predict crop performance fluctuations attributed to changing climatic conditions. Mol. Plant 15, 203–206. https://doi.org/10.1016/j.molp.2022.01.001.
Liakos, K., Busato, P., Moshou, D., Pearson, S., Bochtis, D., 2018. Machine learning in agriculture: a review. Sensors 18, 2674. https://doi.org/10.3390/s18082674.
Libbrecht, M.W., Noble, W.S., 2015. Machine learning applications in genetics and genomics. Nat. Rev. Genet. 16, 321–332. https://doi.org/10.1038/nrg3920.
Lim, B., Arık, S., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int. J. Forecast. 37, 1748–1764. https://doi.org/10.1016/J.IJFORECAST.2021.03.012.
Lin, D., Dechter, E., Ellis, K., Tenenbaum, J., Muggleton, S., 2014. Bias reformulation for one-shot function induction. Front. Artif. Intell. Appl., 525–530. https://doi.org/10.3233/978-1-61499-419-0-525.
Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., Rives, A., 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science 379, 1123–1130. https://doi.org/10.1126/science.ade2574.
Liu, Y., Wang, D., 2017. Application of deep learning in genomic selection. In: 2017 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, p. 2280, https://doi.org/10.1109/BIBM.2017.8218025.
Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. 17, 491–502.
Lin, J., Wang, S., Audano, P.A., Meng, D., Flores, J.I., Kosters, W., Yang, X., Jia, P., Marschall, T., Beck, C.R., Ye, K., 2022a. SVision: a deep learning approach to resolve complex structural variants. Nat. Methods 19, 1230–1233. https://doi.org/10.1038/s41592-022-01609-w.
Lin, T., Wang, Y., Liu, X., Qui, X., 2022b. A survey of transformers. AI Open, https://doi.org/10.1016/j.aiopen.2022.10.001.

Liu, H., Bruning, B., Garnett, T., Berger, B., 2020. Hyperspectral imaging and 3D technologies for plant phenotyping: from satellite to close-range sensing. Comput. Electron. Agric. 175, 105621. https://doi.org/10.1016/j.compag.2020.105621.

Liu, Q., Liang, Z., Feng, D., Jiang, S., Wang, Y., Du, Z., Li, R., Hu, G., Zhang, P., Ma, Y., Lohmann, J.U., Gu, X., 2021. Transcriptional landscape of rice roots at the single-cell resolution. Mol. Plant 14, 384–394. https://doi.org/10.1016/j.molp.2020.12.014.

Lobell, D.B., Hammer, G.L., McLean, G., Messina, C., Roberts, M.J., Schlenker, W., 2013. The critical role of extreme heat for maize production in the United States. Nat. Clim. Chang. 3, 497–501. https://doi.org/10.1038/nclimate1832.

Lu, Y., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., Zhang, P., 2021. Florence: a new foundation model for computer vision. arXiv:2111.11432v1.

Lube, V., Noyan, M.A., Przybysz, A., Salama, K., Blilou, I., 2022. MultipleXLab: a high-throughput portable live-imaging root phenotyping platform using deep learning and computer vision. Plant Methods 18, 38. https://doi.org/10.1186/s13007-022-00864-4.

Luconi, F.L., Malone, T.W., Morton, M.S.S., 1986. Expert systems: the next challenge for managers. Sloan Manag. Rev. 1986-1998 (27), 3–14.

Luo, R., Sedlazeck, F.J., Lam, T.-W., Schatz, M.C., 2019. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. Nat. Commun. 10, 998. https://doi.org/10.1038/s41467-019-09025-z.

Ma, W., Qiu, Z., Song, J., Li, J., Cheng, Q., Zhai, J., Ma, C., 2018. A deep convolutional neural network approach for predicting phenotypes from genotypes. Planta 248, 1307–1318. https://doi.org/10.1007/s00425-018-2976-9.

Mayr, A., Binder, H., Gefeller, O., Schmid, M., 2014. The evolution of boosting algorithms. Methods Inf. Med. 53, 419–427. https://doi.org/10.3414/ME13-01-0122.

Mcculloch, W.S., Pitts, W., Lrerr, W., Pitts, H., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133.

Mcdowell, R.M., 2016. Genomic Selection with Deep Neural Networks. (PhD thesis). Iowa State University, Digital Repository, Ames, https://doi.org/10.31274/etd-180810-5600.

McInnes, L., Healy, J., Melville, J., 2018. UMAP: uniform manifold approximation and projection for dimension reduction. J. Open Source Softw. 3, 861. https://doi.org/10.21105/joss.00861.

Meng, X., Liang, Z., Dai, X., Zhang, Y., Mahboub, S., Ngu, D.W., Roston, R.L., Schnable, J.C., 2021. Predicting transcriptional responses to cold stress across plant species. Proc. Natl. Acad. Sci. 118, e2026330118. https://doi.org/10.1073/pnas.2026330118.

Meuwissen, T.H.E., Hayes, B.J., Goddard, M.E., 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157, 1819–1829. https://doi.org/10.1093/genetics/157.4.1819.

Meuwissen, T., Hayes, B., Goddard, M., 2016. Genomic selection: a paradigm shift in animal breeding. Anim. Front. 6, 6–14. https://doi.org/10.2527/af.2016-0002.

Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D., 2021. Image segmentation using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 44, 3523–3542. https://doi.org/10.1109/TPAMI.2021.3059968.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv:1411.1784v1.

Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., Melgani, F., 2019. Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. GigaScience 8, 1–12. https://doi.org/10.1093/gigascience/giy153.

Montesinos-López, O.A., Martín-Vallejo, J., Crossa, J., Gianola, D., Hernández-Suárez, C.M., Montesinos-López, A., Juliana, P., Singh, R., 2019. A benchmarking between deep learning, support vector machine and Bayesian threshold best linear unbiased prediction for predicting ordinal traits in plant breeding. G3 Genes Genomes Genetics 9, 601–618. https://doi.org/10.1534/g3.118.200998.

Montesinos-López, O.A., Montesinos-López, A., Crossa, J., 2022. Multivariate Statistical Machine Learning Methods for Genomic Prediction. Springer International Publishing, Cham, https://doi.org/10.1007/978-3-030-89010-0.

Montesinos-López, O.A., Montesinos-López, A., Crossa, J., Gianola, D., Hernández-Suárez, C.M., Martín-Vallejo, J., 2018a. Multi-trait, multi-environment deep learning modeling for genomic-enabled prediction of plant traits. G3 (Bethesda) 8, 3829–3840. https://doi.org/10.1534/g3.118.200728.

Montesinos-López, A., Montesinos-López, O.A., Gianola, D., Crossa, J., Hernández-Suárez, C.M., 2018b. Multi-environment genomic prediction of plant traits using deep learners with dense architecture. G3 (Bethesda) 8, 3813–3828. https://doi.org/10.1534/g3.118.200740.

Montesinos-López, O.A., Montesinos-López, A., Mosqueda-Gonzalez, B.A., Montesinos-López, J.C., Crossa, J., Ramirez, N.L., Singh, P., Valladares-Anguiano, F.A., 2021a. A zero altered Poisson random forest model for genomic-enabled prediction. G3 Genes Genomes Genetics 11. https://doi.org/10.1093/g3journal/jkaa057.

Montesinos-López, O.A., Montesinos-López, A., Pérez-Rodríguez, P., Barrón-López, J.A., Martini, J.W.R., Fajardo-Flores, S.B., Gaytan-Lugo, L.S., Santana-Mancilla, P.C., Crossa, J., 2021b. A review of deep learning applications for genomic selection. BMC Genomics 22, 19. https://doi.org/10.1186/s12864-020-07319-x.

Mu, Q., Guo, T., Li, X., Yu, J., 2022. Phenotypic plasticity in plant height shaped by interaction between genetic loci and diurnal temperature range. New Phytol. 233, 1768–1779. https://doi.org/10.1111/nph.17904.

Narendra, V.G., Hareesh, K.S., 2010. Prospects of computer vision automated grading and sorting systems in agricultural and food products for quality evaluation. Int. J. Comput. Appl. 1, 1–12. https://doi.org/10.5120/111-226.

Nelms, B., Walbot, V., 2022. Gametophyte genome activation occurs at pollen mitosis I in maize. Science 375, 424–429. https://doi.org/10.1126/science.abl7392.

Nguyen, C., Sagan, V., Bhadra, S., Moose, S., 2023. UAV multisensory data fusion and multi-task deep learning for high-throughput maize phenotyping. Sensors 23, 1827. https://doi.org/10.3390/s23041827.

Nilsson, N.J., 1998. Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publishers, San Francisco, CA.

Noble, W.S., 2006. What is a support vector machine? Nat. Biotechnol. 24, 1565–1567. https://doi.org/10.1038/nbt1206-1565.

Novakovsky, G., Dexter, N., Libbrecht, M.W., Wasserman, W.W., Mostafavi, S., 2023. Obtaining genetics insights from deep learning via explainable artificial intelligence. Nat. Rev. Genet. 24, 125–137. https://doi.org/10.1038/s41576-022-00532-2.

Ogutu, J.O., Piepho, H.-P., Schulz-Streeck, T., 2011. A comparison of random forests, boosting and support vector machines for genomic selection. BMC Proc. 5, S11. https://doi.org/10.1186/1753-6561-5-S3-S11.

OpenAI, 2022. Introducing ChatGPT. Preprint. https://openai.com/blog/chatgpt.

OpenAI, 2023. GPT-4 technical report. arXiv:2303.08774v3.

Pace, J., Lee, N., Naik, H.S., Ganapathysubramanian, B., Lübberstedt, T., 2014. Analysis of maize (*Zea mays* L.) seedling roots with the high-throughput image analysis tool ARIA (automatic root image analysis). PLoS One 9, e108255. https://doi.org/10.1371/journal.pone.0108255.

Panjvani, K., Dinh, A.V., Wahid, K.A., 2019. LiDARPheno—a low-cost LiDAR-based 3D scanning system for leaf morphological trait extraction. Front. Plant Sci. 10, 147. https://doi.org/10.3389/fpls.2019.00147.

Patrício, D.I., Rieder, R., 2018. Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. Comput. Electron. Agric. 153, 69–81. https://doi.org/10.1016/j.compag.2018.08.001.

Patterson, J., Gibson, A., 2017. Deep Learning: A practitioner's Approach, first ed. O'Reilly Media, Inc, Sebastopol.

Peng, C.-Y.J., Lee, K.L., Ingersoll, G.M., 2002. An introduction to logistic regression analysis and reporting. J. Educ. Res. 96, 3–14. https://doi.org/10.1080/00220670209598786.

Perez-Sanz, F., Navarro, P.J., Egea-Cortines, M., 2017. Plant phenomics: an overview of image acquisition technologies and image data analysis algorithms. GigaScience 6, 1–18. https://doi.org/10.1093/gigascience/gix092.

Pineda, M., Barón, M., Pérez-Bueno, M.-L., 2020. Thermal imaging for plant stress detection and phenotyping. Remote Sens. 13, 68. https://doi.org/10.3390/rs13010068.

Pook, T., Freudenthal, J., Korte, A., Simianer, H., 2020. Using local convolutional neural networks for genomic prediction. Front. Genet. 11, 561497. https://doi.org/10.3389/fgene.2020.561497.

Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P.T., Gross, S.S., Dorfman, L., McLean, C.Y., DePristo, M.A., 2018. A universal SNP and small-indel variant caller using deep neural networks. Nat. Biotechnol. 36, 983–987. https://doi.org/10.1038/nbt.4235.

Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1, 81–106. https://doi.org/10.1007/BF00116251.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving Language Understanding by Generative Pre-Training. Preprint https://openai.com/research/language-unsupervised.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (Eds.), Proceedings of Machine Learning Research. Proceedings of the 38th International Conference on Machine Learning. PMLR., 139, pp. 8747–8763. https://proceedings.mlr.press/v139/radford21a.html.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language Models Are Unsupervised Multitask Learners. Preprint. https://openai.com/research/better-language-models.

Rajaram, S.M., van Ginkel, M., Fischer, R.A., 1994. CIMMYT's wheat breeding mega-environments. In: Li, Z.S., Xin, Z. (Eds.), Proceedings of the 8th International Wheat Genetics Symposium. Beijing, pp. 1101–1106.

Ramachandram, D., Taylor, G.W., 2017. Deep multimodal learning: a survey on recent advances and trends. IEEE Signal Process. Mag. 34, 96–108. https://doi.org/10.1109/MSP.2017.2738401.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I., 2021. Zero-shot text-to-image generation. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning. PMLR, 139, pp. 8821–8831. https://proceedings.mlr.press/v139/ramesh21a.html.

Ramstein, G.P., Jensen, S.E., Buckler, E.S., 2019. Breaking the curse of dimensionality to identify causal variants in breeding 4. Theor. Appl. Genet. 132, 559–567. https://doi.org/10.1007/s00122-018-3267-3.

Resende, R.T., Piepho, H.-P.P., Rosa, G.J.M.M., Silva-Junior, O.B., Silva, F.F.E., de Resende, M.D.V., Grattapaglia, D., 2021. Enviromics in breeding: applications and perspectives on envirotypic-assisted selection. Theor. Appl. Genet. 134, 95–112. https://doi.org/10.1007/s00122-020-03684-z.

Reymond, M., Muller, B., Leonardi, A., Charcosset, A., Tardieu, F., 2003. Combining quantitative trait loci analysis and an ecophysiological model to analyze the genetic variability of the responses of maize leaf growth to temperature and water deficit. Plant Physiol. 131, 664–675. https://doi.org/10.1104/pp.013839.

Rogers, A.R., Dunne, J.C., Romay, C., Bohn, M., Buckler, E.S., Ciampitti, I.A., Edwards, J., Ertl, D., Flint-Garcia, S., Gore, M.A., Graham, C., Hirsch, C.N., Hood, E., Hooker, D.C., Knoll, J., Lee, E.C., Lorenz, A., Lynch, J.P., McKay, J., Moose, S.P., Murray, S.C., Nelson, R., Rocheford, T., Schnable, J.C., Schnable, P.S., Sekhon, R., Singh, M., Smith, M., Springer, N., Thelen, K., Thomison, P., Thompson, A., Tuinstra, M., Wallace, J., Wisser, R.J., Xu, W., Gilmour, A.R., Kaeppler, S.M., De Leon, N., Holland, J.B., 2021. The importance of dominance and genotype-by--environment interactions on grain yield variation in a large-scale public cooperative maize experiment. G3 Genes Genomes Genetics 11. https://doi.org/10.1093/g3journal/jkaa050.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention - - MICCAI 2015. Springer, Cham, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

Rosenzweig, C., Jones, J.W., Hatfield, J.L., Ruane, A.C., Boote, K.J., Thorburn, P., Antle, J.M., Nelson, G.C., Porter, C., Janssen, S., Asseng, S., Basso, B., Ewert, F., Wallach, D., Baigorria, G., Winter, J.M., 2013. The agricultural model Intercomparison and improvement project (AgMIP): protocols and pilot studies. Agric. For. Meteorol. 170, 166–182. https://doi.org/10.1016/j.agrformet.2012.09.011.

Rötter, R.P., Carter, T.R., Olesen, J.E., Porter, J.R., 2011. Crop–climate models need an overhaul. Nat. Clim. Chang. 1, 175–177. https://doi.org/10.1038/nclimate1152.

Russell, S., Norvig, P., 2009. Artificial Intelligence: A Modern Approach, third ed. Prentice Hall Press, USA.

Sandhu, K., Patil, S.S., Pumphrey, M., Carter, A., 2021. Multitrait machine- and deep-learning models for genomic selection using spectral information in a wheat breeding program. Plant Genome 14, e20119. https://doi.org/10.1002/tpg2.20119.

Sankaran, S., Khot, L.R., Espinoza, C.Z., Jarolmasjed, S., Sathuvalli, V.R., Vandemark, G.J., Miklas, P.N., Carter, A.H., Pumphrey, M.O., Knowles, N.R., Pavek, M.J., 2015. Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: a review. Eur. J. Agron. 70, 112–123. https://doi.org/10.1016/j.eja.2015.07.004.

Satterlee, J.W., Strable, J., Scanlon, M.J., 2020. Plant stem-cell organization and differentiation at single-cell resolution. Proc. Natl. Acad. Sci. 117, 33689–33699. https://doi.org/10.1073/pnas.2018788117.

Schapire, R.E., 1990. The strength of weak learnability. Mach. Learn. 5, 197–227.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Networks 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Schrider, D.R., Kern, A.D., 2018. Supervised machine learning for population genetics: a new paradigm. Trends Genet. 34, 301–312. https://doi.org/10.1016/j.tig.2017.12.005.

Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A.W.R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D.T., Silver, D., Kavukcuoglu, K., Hassabis, D., 2020. Improved protein structure prediction using potentials from deep learning. Nature 577, 706–710. https://doi.org/10.1038/s41586-019-1923-7.

Shahhosseini, M., Hu, G., Huber, I., Archontoulis, S.V., 2021. Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt. Sci. Rep. 11, 1606. https://doi.org/10.1038/s41598-020-80820-1.

Singh, A., Ganapathysubramanian, B., Singh, A.K., Sarkar, S., 2016. Machine learning for high-throughput stress phenotyping in plants. Trends Plant Sci. 21, 110–124. https://doi.org/10.1016/j.tplants.2015.10.015.

Singh, A.K., Ganapathysubramanian, B., Sarkar, S., Singh, A., 2018. Deep learning for plant stress phenotyping: trends and future perspectives. Trends Plant Sci. 23, 883–898. https://doi.org/10.1016/j.tplants.2018.07.004.

Song, M., Greenbaum, J., Luttrell, J., Zhou, W., Wu, C., Shen, H., Gong, P., Zhang, C., Deng, H.-W., 2020. A review of integrative imputation for multi-omics datasets. Front. Genet. 11. https://doi.org/10.3389/fgene.2020.570255.

Srivastava, N., Hinton, G., Krizhevsky, A., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Su, X., Yan, X., Tsai, C.-L., 2012. Linear regression. Wiley Interdiscip. Rev. Comput. Stat. 4, 275–294. https://doi.org/10.1002/wics.1198.

Sun, S., Li, C., Paterson, A.H., Jiang, Y., Xu, R., Robertson, J.S., Snider, J.L., Chee, P.W., 2018. In-field high throughput phenotyping and cotton plant growth analysis using LiDAR. Front. Plant Sci. 9, 16. https://doi.org/10.3389/fpls.2018.00016.

Svozil, D., Kvasnicka, V., Pospichal, J., 1997. Introduction to multi-layer feed-forward neural networks. Chemom. Intell. Lab. Syst. 39, 43–62. https://doi.org/10.1016/S0169-7439(97)00061-0.

Takase, T., Oyama, S., Kurihara, M., 2018. Effective neural network training with adaptive learning rate based on training loss. Neural Netw. 101, 68–78. https://doi.org/10.1016/j.neunet.2018.01.016.

Tattaris, M., Reynolds, M.P., Chapman, S.C., 2016. A direct comparison of remote sensing approaches for high-throughput phenotyping in plant breeding. Front. Plant Sci. 7. https://doi.org/10.3389/fpls.2016.01131.

Taunk, K., De, S., Verma, S., Swetapadma, A., 2019. A brief review of nearest neighbor algorithm for learning and classification. In: 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, pp. 1255–1260, https://doi.org/10.1109/ICCS45141.2019.9065747.

Technow, F., Messina, C.D., Totir, L.R., Cooper, M., 2015. Integrating crop growth models with whole genome prediction through approximate Bayesian computation. PLoS One 10, e0130855. https://doi.org/10.1371/journal.pone.0130855.

Thesma, V., Mohammadpour Velni, J., 2022. Plant root phenotyping using deep conditional GANs and binary semantic segmentation. Sensors 23, 309. https://doi.org/10.3390/s23010309.

Tian, D.P., 2013. A review on image feature extraction and representation techniques. Int. J. Multimedia Ubiquitous Eng. 8, 385–396.

Torracinta, R., Campagne, F., 2016. Training genotype callers with neural networks. bioRxiv, 097469. https://doi.org/10.1101/097469.

Ubbens, J., Parkin, I., Eynck, C., Stavness, I., Sharpe, A.G., 2021. Deep neural networks for genomic prediction do not estimate marker effects. Plant Genome 14, 1–10. https://doi.org/10.1002/tpg2.20147.

van der Aalst, W.M.P., Bichler, M., Heinzl, A., 2018. Robotic process automation. Bus. Inf. Syst. Eng. 60, 269–272. https://doi.org/10.1007/s12599-018-0542-4.

Van Der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.

Varshney, R.K., Roorkiwal, M., Sorrells, M.E., 2017. Genomic selection for crop improvement: an introduction. In: Varshney, R.K., Roorkiwal, M., Sorrells, M.E. (Eds.), Genomic Selection for Crop Improvement. Springer International Publishing, Cham, pp. 1–6, https://doi.org/10.1007/978-3-319-63170-7_1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 5999–6009.

Victoria, A.H., Maragatham, G., 2021. Automatic tuning of hyperparameters using Bayesian optimization. Evol. Syst. 12, 217–223. https://doi.org/10.1007/s12530-020-09345-2.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., 2018a. High-resolution image synthesis and semantic manipulation with conditional GANs. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, pp. 8798–8807, https://doi.org/10.1109/CVPR.2018.00917.

Wang, X., Xu, Y., Hu, Z., Xu, C., 2018b. Genomic selection methods for crop improvement: current status and prospects. Crop J. 6, 330–340. https://doi.org/10.1016/j.cj.2018.03.001.

Washburn, J.D., Mejia-Guerra, M.K., Ramstein, G., Kremling, K.A., Valluru, R., Buckler, E.S., Wang, H., 2019. Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence. Proc. Natl. Acad. Sci. 116, 5542–5549. https://doi.org/10.1073/pnas.1814551116.

Washburn, J.D., Burch, M.B., Franco, J.A.V., 2020. Predictive breeding for maize: making use of molecular phenotypes, machine learning, and physiological crop models. Crop Sci. 60, 622–638. https://doi.org/10.1002/csc2.20052.

Washburn, J.D., Cimen, E., Ramstein, G., Reeves, T., O'Briant, P., McLean, G., Cooper, M., Hammer, G., Buckler, E.S., 2021. Predicting phenotypes from genetic, environment, management, and historical data using CNNs. Theor. Appl. Genet. 134, 3997–4011. https://doi.org/10.1007/s00122-021-03943-7.

Wen, C., Qian, J., Lin, J., Teng, J., Jayaraman, D., Gao, Y., 2022. Fighting fire with fire: avoiding DNN shortcuts through priming. Int. Conf. Mach. Learn. PMLR, 23723–23750.

Westhues, M., Heuer, C., Thaller, G., Fernando, R., Melchinger, A.E., 2019. Efficient genetic value prediction using incomplete omics data. Theor. Appl. Genet. 132, 1211–1222. https://doi.org/10.1007/s00122-018-03273-1.

White, J.W., Hoogenboom, G., 1996. Simulating effects of genes for physiological traits in a process-oriented crop model. Agron. J. 88, 416–422. https://doi.org/10.2134/agronj1996.00021962008800030009x.

Williams, R.J., Zipser, D., 1995. Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin, Y., Rumelhart, D.E. (Eds.), Backpropagation: Theory, Architectures, and Applications, Developments in Connectionist Theory. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US, pp. 433–486.

Windt, C.W., Vergeldt, F.J., de Jager, P.A., van As, H., 2006. MRI of long-distance water transport: a comparison of the phloem and xylem flow characteristics and dynamics in poplar, castor bean, tomato and tobacco. Plant Cell Environ. 29, 1715–1729. https://doi.org/10.1111/j.1365-3040.2006.01544.x.

Wu, Y., Zhuang, D., Labbe, A., Sun, L., 2021. Inductive graph neural networks for spatio-temporal kriging. Proc. AAAI Conf. Artif. Intell. 35, 4478–4485. https://doi.org/10.1609/aaai.v35i5.16575.

Xiang, L., Gai, J., Bao, Y., Yu, J., Schnable, P.S., Tang, L., 2023. Field-based robotic leaf angle detection and characterization of maize plants using stereo vision and deep convolutional neural networks. J. Field Robot. https://doi.org/10.1002/rob.22166.

Xu, Y., 2016. Envirotyping for deciphering environmental impacts on crop plants. Theor. Appl. Genet. 129, 653–673. https://doi.org/10.1007/s00122-016-2691-5.

Xu, X., Crow, M., Rice, B.R., Li, F., Harris, B., Liu, L., Demesa-Arevalo, E., Lu, Z., Wang, L., Fox, N., Wang, X., Drenkow, J., Luo, A., Char, S.N., Yang, B., Sylvester, A.W., Gingeras, T.R., Schmitz, R.J., Ware, D., Lipka, A.E., Gillis, J., Jackson, D., 2021. Single-cell RNA sequencing of developing maize ears facilitates functional analysis and trait

candidate gene discovery. Dev. Cell 56, 557–568.e6. https://doi.org/10.1016/j.devcel.2020.12.015.

Yang, W., Duan, L., Chen, G., Xiong, L., Liu, Q., 2013. Plant phenomics and high-throughput phenotyping: accelerating rice functional genomics using multidisciplinary technologies. Curr. Opin. Plant Biol. 16, 180–187. https://doi.org/10.1016/j.pbi.2013.03.005.

Yıldız, O.T., Alpaydın, E., 2001. Omnivariate decision trees. IEEE Trans. Neural Netw. 12, 1539–1546.

Yin, X., Stam, P., Dourleijn, C.J., Kropff, M.J., 1999. AFLP mapping of quantitative trait loci for yield-determining physiological characters in spring barley. Theor. Appl. Genet. 99, 244–253. https://doi.org/10.1007/s001220051230.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F.E.H.H., Feng, J., Yan, S., 2021. Tokens-to-Token ViT: training vision transformers from scratch on ImageNet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 558–567.

Zadeh, L.A., 1975. Fuzzy logic and approximate reasoning. Synthese 30, 407–428. https://doi.org/10.1007/BF00485052.

Zadeh, L.A., 1988. Fuzzy logic. Computer 21, 83–93. https://doi.org/10.1109/2.53.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J., 2020. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. J. Appl. Sci. Technol. Trends 1, 56–70. https://doi.org/10.38094/jastt1224.

Zhang, T.-Q., Chen, Y., Liu, Y., Lin, W.-H., Wang, J.-W., 2021. Single-cell transcriptome atlas and chromatin accessibility landscape reveal differentiation trajectories in the rice root. Nat. Commun. 12, 2053. https://doi.org/10.1038/s41467-021-22352-4.

Zhang, B., Huang, H., Tibbs-Cortes, L.E., Vanous, A., Zhang, Z., Sanguinet, K., Garland-Campbell, K.A., Yu, J., Li, X., 2023a. Streamline unsupervised machine learning to survey and graph indel-based haplotypes from pan-genomes. Mol. Plant. https://doi.org/10.1101/2023.02.11.527743. https://doi.org/10.1016/j.molp.2023.05.005.

Zhang, N., Zhou, X., Kang, M., Hu, B.-G., Heuvelink, E., Marcelis, L.F.M., 2023b. Machine learning versus crop growth models: an ally, not a rival. AoB PLANTS 15, 1–7. https://doi.org/10.1093/aobpla/plac061.

Zhao, W., Lai, X., Liu, D., Zhang, Z., Ma, P., Wang, Q., Zhang, Z., Pan, Y., 2020. Applications of support vector machine in genomic prediction in pig and maize populations. Front. Genet. 11, 1537. https://doi.org/10.3389/fgene.2020.598318.

Zingaretti, L.M., Gezan, S.A., Ferrão, L.F.V., Osorio, L.F., Monfort, A., Muñoz, P.R., Whitaker, V.M., Pérez-Enciso, M., 2020. Exploring deep learning for complex trait genomic prediction in polyploid outcrossing species. Front. Plant Sci. 11, 1–14. https://doi.org/10.3389/fpls.2020.00025.