Title

Automated Machine Learning for High-Throughput Image-Based Plant Phenotyping

Authors

Joshua C.O. Koh,[1] German Spangenberg,[2,3] and Surya Kant [1,4*]

Affiliations

[1] Agriculture Victoria, Grains Innovation Park, 110 Natimuk Rd, Horsham, Victoria 3400, Australia

[2] Agriculture Victoria, AgriBio, Centre for AgriBioscience, 5 Ring Road, Bundoora, Victoria 3083, Australia

[3] School of Applied Systems Biology, La Trobe University, Bundoora, Victoria 3083, Australia

[4] Centre for Agricultural Innovation, School of Agriculture and Food, Faculty of Veterinary and Agricultural Sciences, The University of Melbourne, Victoria 3010, Australia

*Corresponding author. Email: surya.kant@agriculture.vic.gov.au

Abstract

Automated machine learning (AutoML) has been heralded as the next wave in artificial intelligence with its promise to deliver high performance end-to-end machine learning pipelines with minimal effort from the user. AutoML with neural architecture search which searches for the best neural network architectures in deep learning has delivered state-of-the-art performance in computer vision tasks such as image classification and object detection. Using wheat lodging assessment with UAV imagery as an example, we compared the performance of an open-source AutoML framework, AutoKeras in image classification and regression tasks to that of a manual approach using transfer learning with a ResNet-50 architecture pretrained on the ImageNet dataset. AutoKeras had the best accuracy (92.0%) compared to the ResNet-50 (90.4%) in image classification which assigned plot images as either lodged or non-lodged. For image regression, lodged plot images were used as inputs to predict lodging scores. AutoKeras also performed better (R^2=0.8273, RMSE=10.65, MAE=8.24, MAPE=13.87%) in this task compared to the ResNet-50 (R^2=0.8079, RMSE=10.81, MAE=7.84, MAPE=14.64%). Results in our study provide a basis for the adoption and application of AutoML systems for high-throughput image-based plant phenotyping.

1. Introduction

High-throughput plant phenotyping (HTP) plays a crucial role in meeting the increasing demand for large-scale plant evaluation in breeding trials and crop management systems [1-3]. Concurrent with the development of various ground-based and aerial (e.g. unmanned aerial vehicle, UAV) HTP systems is the rise in use of imaging sensors for phenotyping purposes. Sensors for color (RGB), thermal, spectral (multi- and hyperspectral) and 3D (e.g. LiDAR) imaging have been applied extensively for phenotyping applications encompassing plant morphology, physiology, development

and postharvest quality [3, 4]. Consequently, the meteoric rise in big image data arising from HTP systems necessitates the development of efficient image processing and analytical pipelines. Conventional image analysis pipelines typically involve computer vision tasks (e.g. wheat head counting using object detection) which are addressed through the development of signal processing and/or machine learning (ML) algorithms. However, these algorithms are sensitive to image quality (e.g. illumination, sharpness, distortion) variations and do not tend to generalize well across datasets with different imaging conditions [4]. Although traditional ML approaches have to some degree improved upon algorithm generalization, most of them still fall short of the current phenotyping demands and require significant expert guidance in designing features that are invariant to imaging environments. To this end, deep learning (DL), a subset of ML has emerged in recent years as the leading answer to meeting these challenges. One key benefit of DL is that features are automatically learned from the input data, thereby negating the need for laborious manual feature extraction and allow well-generalized models to be trained using datasets from diverse imaging environments. A common DL architecture is deep convolutional neural networks (CNNs) which have delivered state-of-the-art (SOTA) performance for computer vision tasks such as image classification/regression, object detection and image segmentation [5-7]. The progress of transfer learning, a technique which allows the use of pre-trained SOTA CNNs as base models in DL and the availability of public DL libraries have contributed to the exponential adoption of DL in plant phenotyping. Deep CNN approaches for image-based plant phenotyping have been applied for plant stress evaluation, plant development characterization and crop postharvest quality assessment [4, 8, 9]. However, not all modern CNN solutions can be readily implemented for plant phenotyping applications and adoption will require extra efforts which may be technically challenging [4]. In addition, the process of building a high-performance DL network for a specific task is time-consuming, resource expensive and relies heavily on human expertise through a trial-and-error process [4, 5].

Following the exponential growth of computing power and availability of cloud computing resources in recent years, automated machine learning (AutoML) has received tremendous attention in both industry and academia. AutoML provides an attractive alternative to the manual ML practice as it promises to deliver high-performance end-to-end ML pipelines covering data preparation (cleaning and preprocessing), feature engineering (extraction, selection and construction), model generation (selection and hyperparameter tuning) and model evaluation requiring minimal effort or intervention from the user [10-12]. AutoML services have become a standard offering in many technology companies, for example Cloud ML by Google and SageMaker by Amazon. Early work by Zoph et al. [13] highlighted the potential of AutoML in which a recurrent network was trained with reinforcement learning to automatically search for the best performing architecture. Since then, research interest in AutoML has exploded, with a primary focus on neural architecture search (NAS) which seeks to construct well-performing neural architectures through the selection and

combination of various basic modules from a predefined search space [10, 14, 15]. NAS algorithms can be categorized based on three dimensions: the search space which defines the type of model that can be designed, the search strategy used to explore the search space and the performance estimation strategy which evaluates the performance of possible models from its design [14, 15]. NAS-generated models have achieved SOTA performance and outperformed manually designed architectures on computer vision tasks such as image classification [16], object detection [17] and semantic segmentation [18].

However, despite AutoML showing great promise for computer vision tasks, to the best of our knowledge, no study has used AutoML for image-based plant phenotyping. To address this gap in knowledge, we report for the first time the application of AutoML for image-based plant phenotyping using wheat lodging assessment with UAV imagery as an example. The performance of an open-source AutoML system, AutoKeras was compared to transfer learning using a pre-trained ResNet-50 network on image classification and image regression tasks. For image classification, plot images were classified as either non-lodged or lodged; for image regression, lodged plot images were used as inputs to predict lodging scores. The merits and drawbacks of AutoML and transfer learning for image-based plant phenotyping are discussed. Results in our study provide a basis for the adoption and application of AutoML systems for high-throughput image-based plant phenotyping.

2. Materials and Methods

2.1 Field Experiment

A wheat breeding experiment was conducted at Agriculture Victoria, Horsham, Australia during the winter-spring cropping season of 2018 (Lat:36°44’35.21”S Lon:142°6’18.01”E) (Figure 1). Seeds were sown to a planting density of 150 plants/m^2 in individual plots measuring 5 m long and 1 m wide (5 m^2), with a total of 1,248 plots. Severe wind events toward the end of the cropping season (30th November – 9th December) resulted in significant lodging of wheat plots across the experiment. Ground truth labels for “lodged” and “non-lodged” were provided by an experienced field technician and a plant scientist.

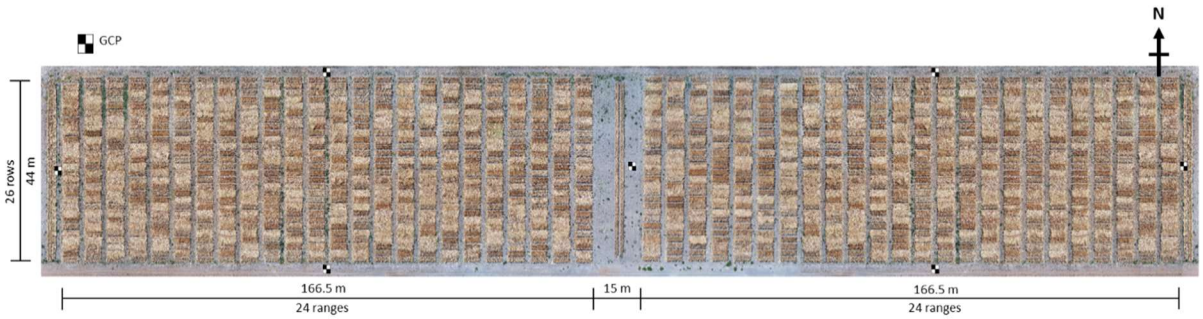

Figure 1 Wheat breeding field experiment. Ground control points (GCPs) indicated in figure.

2.2 Image Acquisition and Processing

High resolution aerial imaging of wheat plots with different lodging grades was performed on 11th December 2018. Aerial imagery was acquired on a DJI Matrice M600 Pro (Shenzhen DJI Sciences and Technologies Ltd., China) UAV with a Sony A7RIII RGB camera (35.9 mm x 24.0 mm sensor size, 42.4 megapixels resolution) mounted on a DJI Ronin MX gimbal. Flight planning and automatic mission control was performed using DJI's iOS application Ground Station Pro (GS Pro). The camera was equipped with a 55 mm fixed focal length lens and set to 1 s interval shooting with JPEG format in shutter priority mode. Images were geotagged using a GeoSnap Express system (Field of View, USA). The flight mission was performed at an altitude of 45 m with front and side overlap of 75% under clear sky conditions. Seven black and white checkered square panels (38 cm x 38 cm) were distributed in the field experiment to serve as ground control points (GCPs) for accurate geo-positioning of images (Figure 1). A real-time kinematic global positioning system (RTK-GPS) receiver, EMLID Reach RS (https://emlid.com) was used to record the centre of each panel with < 1 centimeter accuracy.

Images were imported into Pix4Dmapper version 4.4.12 (Pix4D, Switzerland) to generate an orthomosaic image, with the coordinates of the GCPs used for geo-rectification. The resulting orthomosaic had a ground sampling distance (GSD) of 0.32 cm/pixel. Individual plot images were clipped and saved in TIFF format from the orthomosaic using a field plot map with polygons corresponding to the experimental plot dimension of 5 m x 1 m in ArcGIS Pro version 2.5.1 (Esri, USA).

2.3 Lodging Assessment

A two-stage assessment of lodging was performed in this study and the results were used as the basis for image classification and image regression tasks (see Section 2.4) (Figure 2). The image classification task corresponded to the first stage of assessment in which the lodging status i.e. whether the plot is lodged (yes) or non-lodged (no) was provided by the ground truth data and this could be verified easily from visual inspection of the high-resolution plot images (Figure 3). The image regression task corresponded to the second stage of assessment where plots identified as lodged were evaluated using a modified lodging score based on the method of Fisher and Stapper [19]:

$$Lodging\ score\ = \frac{Lodging\ severity}{3} \times Lodged\ area\ (\%)$$

Lodging severity values of 1 to 3 were assigned to three main grades of lodging based on the inclination angle between the wheat plant and the vertical line as follows: light lodging (severity 1; 10° – 30°), moderate lodging (severity 2; 30° – 60°) and heavy lodging (severity 3; 60° – 90°)(Figure 3)[20]. The lodged area (%) was determined visually from the plot images as the percentage of area lodged in the plot in proportion to the total plot area. The derived lodging score ranged between values of 1 – 100, with a score of 100 indicating that the entire plot was lodged with heavy lodging.

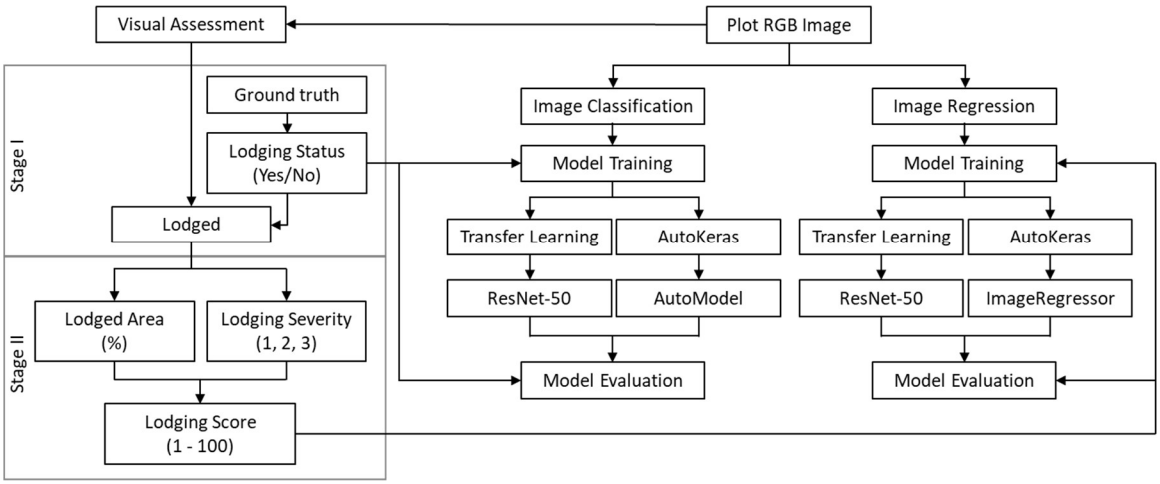

Figure 2 Experimental workflow for wheat lodging assessment.

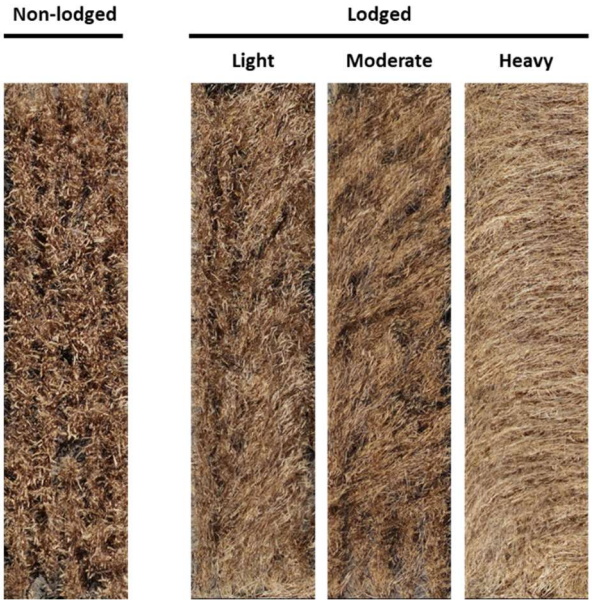

Figure 3 Different wheat lodging severities. Wheat plot images were first classified as non-lodged or lodged using ground truth data. Plots identified as lodged were assessed visually and divided into three lodging severities (light, moderate and heavy) based on lodging angles.

2.4 Deep Learning Experiments

Deep learning experiments were conducted in the Python 3.7 environment. Performance of the open-source AutoML framework, AutoKeras [21] version 1.0.1 in image classification and image regression was compared to a manual approach using transfer learning with a pre-trained ResNet-50 neural network [22] implemented in Keras, Tensorflow-GPU version 2.1. For the image classification task, a binary classification scheme assigning individual plot images to either non-lodged (class 0) or lodged (class 1) was performed as the images had a relatively balanced distribution between the two

classes. For the image regression task, plot images were used as inputs to predict the lodging score. Training and evaluation of the models were performed on a NVIDIA Titan RTX GPU (24 GB of GPU memory) at SmartSense iHub, Agriculture Victoria.

2.4.1 Training, Validation and Test Datasets

For image classification, the image dataset consisted of 1,248 plot images with 528 plots identified as non-lodged (class 0) and 720 plots identified as lodged (class 1). Images were first resized to the dimensions of 128 width x 128 height x 3 channels (Section 2.4.2) and these were split 80:20 (seed number=123) into training (998 images) and test (250 images) datasets. For image regression, the 720 resized plot images identified as lodged were split 80:20 (seed number=123) into training (576 images) and test (144 images) datasets. Images were fed directly into AutoKeras without preprocessing as this was done automatically by AutoKeras. In contrast, images were preprocessed to the ResNet-50 format using the provided preprocess_input function in Keras. For model training on both image classification and regression, the training dataset was split further 80:20 (seed number=456) into training and validation datasets. The validation dataset is used to evaluate training efficacy, with lower validation loss (as defined by the loss function, Section 2.4.2) indicating a better trained model. Performance of trained models was evaluated on the test dataset (section 2.2.4).

2.4.2 AutoML with AutoKeras

AutoKeras is an open-source AutoML framework built using Keras which implements state-of-the-art NAS algorithms for computer vision and machine learning tasks [21]. It is also the only open-source NAS framework to offer both image classification and regression abilities at the time of this study. In our study, we experienced great difficulty in getting AutoKeras to stably complete experiments in default settings due to errors relating to GPU memory usage and model tuning. This is not entirely unexpected as beginning version 1.0, AutoKeras has undergone significant application programming interface (API) and system architecture redesign to incorporate KerasTuner ver. 1.0 and Tensorflow ver. 2.0. This change was necessary for AutoKeras to capitalize on recent developments in NAS and the DL framework, Tensorflow, in addition to providing support for the latest graphics processing unit (GPU) hardware. To partly circumvent the existing issues, we had to implement two approaches for the DL experiments to stably complete up to 100 trials, which is the number of models evaluated by AutoKeras (i.e. 100 trials = 100 models). Firstly, all images were resized to the dimensions of 128 x 128 x 3 and secondly, for image classification, we had to implement a custom image classifier using the provided AutoModel class in AutoKeras. We were not successful in completing experiments beyond 100 trials, as such only results up to 100 trials were presented in this study.

For image classification, a custom image classifier was defined using the AutoModel class which allows the user to define a custom model by connecting modules/blocks in AutoKeras (Figure 4). In

most cases, the user only needs to define the input node(s) and output head(s) of the AutoModel, as the rest is inferred by AutoModel itself. In our case, the input nodes were first an ImageInput class accepting image inputs (128 x 128 x 3), which in turn was connected to an ImageBlock class which selects iteratively from either a ResNet [22], ResNext [23], Xception [24] or simple CNN building blocks to construct neural networks of varying complexity and depth. The input nodes were joined to a single output head, the ClassificationHead class which performed the binary classification (Figure 4). The AutoModel was fitted to the training dataset with the tuner set as "bayesian", loss function as "binary_crossentropy", evaluation metrics as "accuracy" and 200 training epochs (rounds) for 10, 25, 50 and 100 trials with a seed number of 10. For image regression, the default Autokeras image regression class, ImageRegressor was fitted to the training dataset with the loss function as mean squared error (MSE), evaluation metrics as mean absolute error (MAE) and mean absolute percentage error (MAPE), and 200 training epochs for 10, 25, 50 and 100 trials with a seed number of 45 (Figure 4).

The performance of the best models from 10, 25, 50 and 100 trials were evaluated on their respective test datasets (Section 2.4.4) and exported as Keras models to allow neural network visualization using the open-source tool, Netron [25]. Model inference times were measured using the built-in Python function, timeit and presented as mean ± standard deviation in milliseconds (ms).

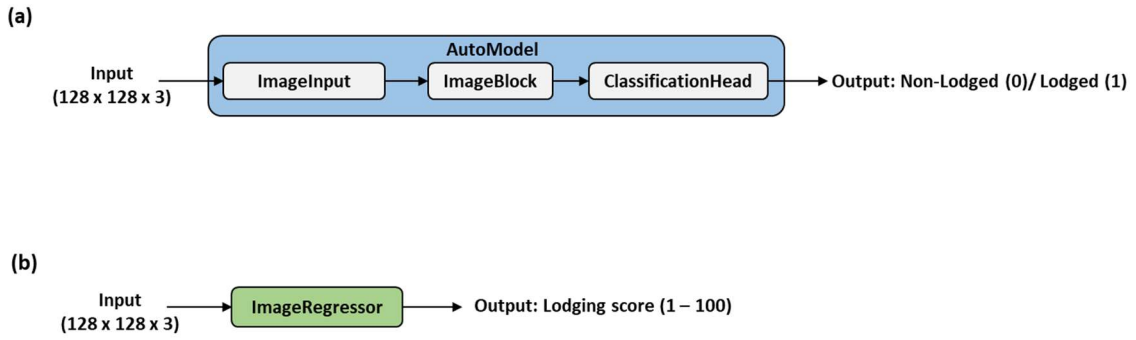

Figure 4 Automated machine learning with AutoKeras. (a) AutoModel for image classification. (b) ImageRegressor for image regression.

2.4.3 Transfer Learning with ResNet-50

Transfer learning is a popular approach in DL where a pre-trained model is reused as the starting point for a model on a second task [4]. This allows the user to rapidly deploy complex neural networks, including state-of-the-art DL architectures without incurring time and computing costs. One popular deep neural network often used in transfer learning is the residual network-50 (ResNet-50) [22]. In our study, transfer learning (i.e. the manual approach) using a ResNet-50 pre-trained on the ImageNet dataset was compared to automated machine learning (AutoKeras) in image

classification and image regression tasks. The ResNet-50 was implemented in Keras as a base model using the provided Keras API with the following parameters: weights="imagenet", include_top=False and input_shape=(128, 128, 3) (Figure 5). Outputs from the ResNet-50 base model was joined to a global average pooling 2D layer and connected to a final dense layer, with the activation function set as either "sigmoid" for image classification or "linear" for image regression. The model was compiled with the batch size as 32, optimizer as 'Adam' and corresponding loss functions and evaluation metrics as described in section 2.4.2. Model training occurred in two stages for both image classification and regression tasks: in the first stage (100 epochs), weights of the pre-trained layers were frozen and the Adam optimizer had a higher learning rate (lr=0.1, decay=0.01) to allow faster training of the top layers; in the second stage (200 epochs), weights of the pre-trained layers were unfrozen and the Adam optimizer had a smaller learning rate (lr=0.01, decay=0.001) to allow fine-tuning of the model. Performance of the trained models was evaluated on their respective test datasets (Section 2.4.4).

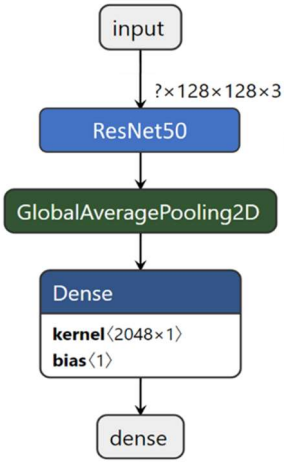

Figure 5 Transfer learning with the ResNet-50 architecture. Outputs from the ResNet-50 was joined to a global average pooling 2D layer and connected to a final dense layer, with the activation function set as either "sigmoid" for image classification or "linear" for image regression.

2.4.4 Model Evaluation Metrics

For image classification, model performance on the test dataset was evaluated using classification accuracy. In addition, classification results using ResNet-50 (transfer learning) and the best AutoKeras model were visualized using confusion matrices. For image regression, in addition to the mean absolute error (MAE) and the mean absolute percentage error (MAPE) provided by AutoKeras and Keras, the coefficient of determination (R^2) and the root mean-squared error (RMSE) were also calculated to determine model performance on the test dataset. Results from ResNet-50 and the best AutoKeras model were visualized using regression plots which plotted predicted lodging scores (y_predict) against actual scores (y_test).

3. Results

3.1 Image Classification

Both transfer learning (ResNet-50) and AutoKeras performed strongly in the image classification task, with the best AutoKeras model (from 100 trials) achieving an accuracy of 92.0%, which is higher compared to the accuracy of 90.4% obtained by the ResNet-50 model (Figure 6). A closer inspection at the confusion matrices of both models showed that they had very similar classification performance, with the better performance of the AutoKeras model explained by having a higher number of true negatives (100) and lower false positives (2) when compared to the ResNet-50 model (94 true negatives, 7 false positives)(Figure 6). Impressively, AutoKeras was able to achieve these results using a simple 2-layers CNN (43,859 parameters) consisting only of a single 2D convolutional layer (Figure 7) as opposed to the 50-layers deep ResNet-50 architecture (~25 million parameters). Inference time on the test dataset for the 2-layers CNN model (8.46 ms ± 46.6 µs) was ~10-fold faster compared to the ResNet-50 model (87.3 ms ± 18.9 ms). AutoKeras performance improved as more models (trials) were evaluated, as reflected in the increase in model classification accuracy from 10 to 100 trials (Figure 8). Examination of the model architectures returned by AutoKeras revealed that the best model architecture resulting from the 10 and 25 trials was a deep CNN model comparable in depth and complexity to the ResNet-50 (Figure S1), highlighting the ability of AutoKeras to explore deep CNN architectures even in a small model search space. Subsequently, when the model search space was extended to 50 and 100 trials, the best model architecture discovered by AutoKeras was the 2-layers CNN model (Figure 8).

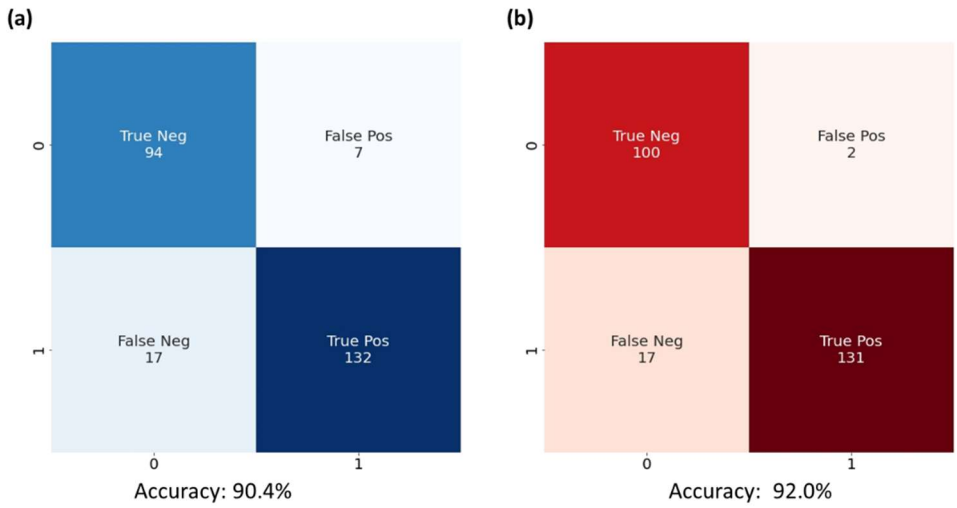

Figure 6 Confusion matrix of the classification results using (a) ResNet-50 and (b) AutoKeras. Results for AutoKeras were produced using the model discovered from 100 trials.

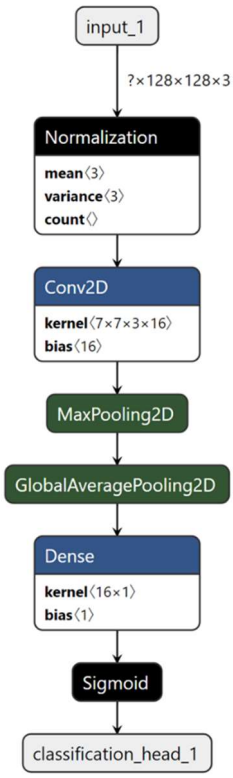

Figure 7 Best AutoKeras model architecture for image classification. A simple 2-layers CNN model with 43,895 parameters was the best architecture discovered in 50 and 100 trials. The best classification performance was provided by the model from 100 trials.

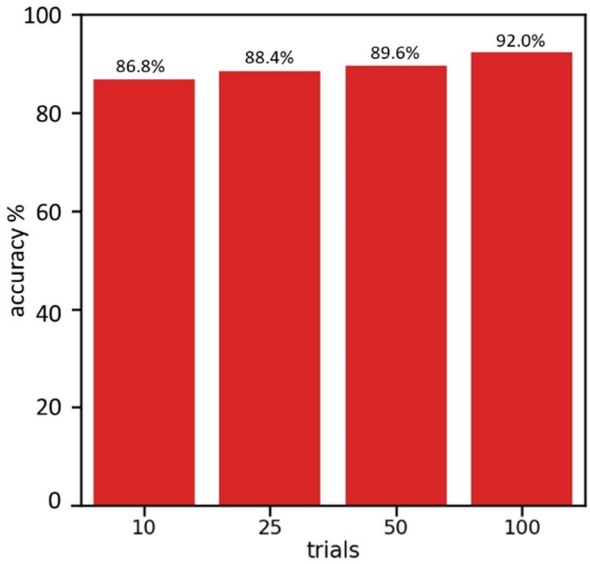

Figure 8 Classification accuracies for AutoKeras models from 10, 25, 50 and 100 trials.

3.2 Image Regression

For the image regression task, the best AutoKeras model (from 100 trials) also performed better (R^2=0.8273, RMSE=10.65, MAE=8.24, MAPE=13.87%) compared to transfer learning using ResNet-50 (R^2=0.8079, RMSE=10.81, MAE=7.84, MAPE=14.64%) (Table 1). AutoKeras was able to achieve this

feat using an 8-layers CNN resembling a truncated mini Xception network with 207,560 total parameters (Figure 9). Two prominent features of the original 71-layers deep Xception network, namely the use of depthwise separable convolution layers and skip connections were evident in the AutoKeras model (Figure 9). Inference time on the test dataset for the mini Xception model (57.5 ms ± 17.8 ms) was comparable to the ResNet-50 (55.1 ms ± 10.7 ms). Results from the AutoKeras experiments showed continued performance gains with more models being explored from 10 trials to 100 trials, as reflected in lower error scores and an increase in R^2 from 0.7568 to 0.8273 (Table 1). Examination of the model architectures returned by AutoKeras revealed that the best model architecture resulting from the 10 and 25 trials was a deep CNN model (Figure S2) whereas the best model architecture discovered from 50 and 100 trials was the 8-layers mini Xception model (Figure 9).

Table 1 Performance metrics of modals in image regression.

Model	R^2	RMSE	MAE	MAPE
ResNet-50	0.8079	10.81	**7.84**	14.64%
AutoKeras (10 trials)	0.7568	12.43	9.54	14.55%
AutoKeras (25 trials)	0.7772	12.28	8.62	14.38%
AutoKeras (50 trials)	0.8133	10.71	8.31	13.92%
AutoKeras (100 trials)	**0.8273**	**10.65**	8.24	**13.87%**

Best scores for each category are indicated in bold.

The best AutoKeras model had better performance scores across the board compared to the ResNet-50 model, with the exception that the ResNet-50 had a lower MAE of 7.84 compared to a score of 8.24 by the AutoKeras model (Table 1). A closer inspection at the regression plots for both models showed that the AutoKeras model had a much higher number of predictions exceeding the maximum value of 100 (n=39), with the largest predicted value being 118 whereas the ResNet-50 model only have one prediction exceeding 100 with a value of 100.29 (Figure 10). This would likely explain why the MAE score, which is the average of the absolute difference between predicted scores and actual scores for the ResNet-50 model was lower compared to the AutoKeras model. The occurrence of predictions exceeding the maximum limit of 100 was possible in our study since no limit was defined for the regression models. To address this issue, predicted values exceeding 100 were simply assigned or binned to a value of 100. As expected, this led to a noticeable improvement in performance scores for the AutoKeras model (R^2=0.8351, RMSE=10.10, MAE=7.09, MAPE=12.70%) with no changes observed for the ResNet-50 model.

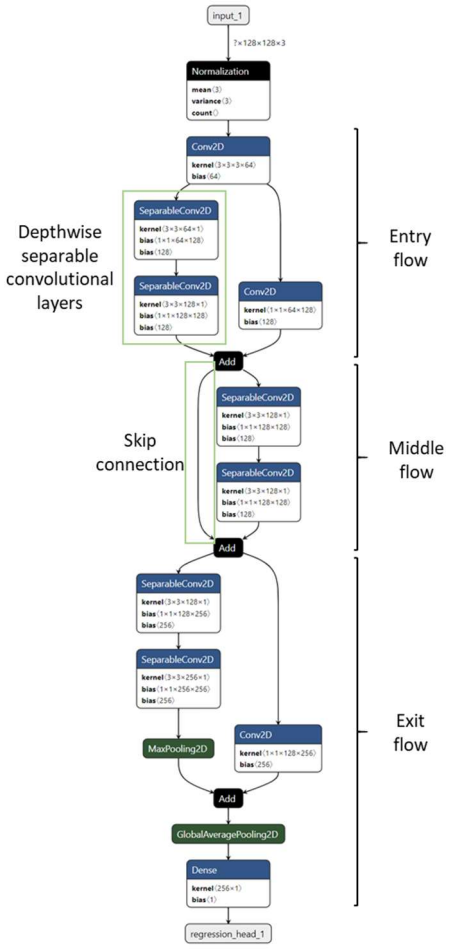

Figure 9 Best AutoKeras model architecture for image regression. An 8-layers mini Xception model with 207,560 parameters was the best architecture discovered in 50 and 100 trials. Three main parts (entry, middle and exit flows) and two key features (examples indicated by green boxes), namely the depthwise separable convolutional layers and skip connections from the original Xception network [24] can be discerned from the architecture. The best regression performance was provided by the model from 100 trials.

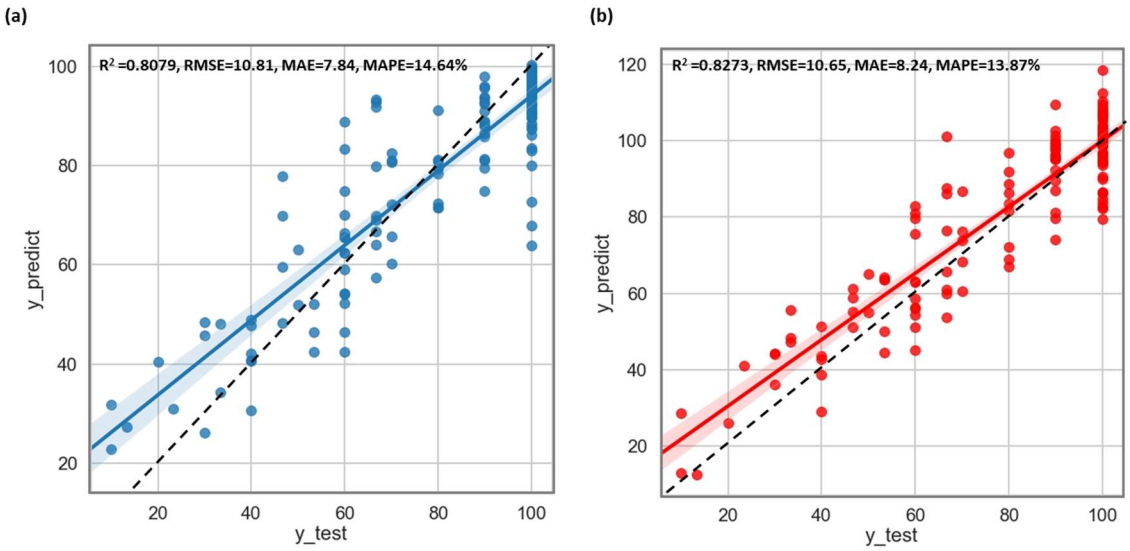

Figure 10 Regression plot for predicted outputs using (a) ResNet-50 and (b) AutoKeras. Predicted lodging scores (y_predict) were plotted against actual scores (y_test). Regression line (blue or red) is shown with a 95% confidence interval and dashed black line indicates the 1:1 line.

4. Discussion

Using wheat lodging assessment as an example, we compared the performance of an open-source AutoML framework, AutoKeras to a manual approach using transfer learning with a pretrained ResNet-50 model for image classification and image regression. As a testament to the power and efficacy of modern DL architectures for computer vision tasks, both AutoKeras and transfer learning approaches performed well in this study, with AutoKeras exhibiting a slight performance advantage over the ResNet-50 model.

For the image classification task, plot images were classified as either non-lodged or lodged. The best AutoKeras model (from 100 trials) achieved an accuracy of 92.0%, which is higher compared to the accuracy of 90.4% obtained by the ResNet-50 model. These results are comparable to those reported in a smaller study which classified 465 UAV-acquired wheat plot multispectral images (red, green, blue channels used in models) as either non-lodged/lodged with a hand-crafted deep neural network, LodgedNet (97.7% accuracy) and other modern DL networks (97.7 – 100.0% accuracies) via transfer learning [26]. The higher classification accuracies reported in that study may be due to the use of image data augmentation (i.e. increasing training dataset via image transformations) which generally improves model performance and that images were acquired pre-maturity (plants were still green) allowing variations in color to potentially contribute more informative features for modelling [20, 26]. Results in our study suggest that NAS generated models (AutoML) can provide very competitive performance compared to modern human designed CNN architectures. A recent survey comparing the performance of manually designed models against those generated by NAS algorithms on the CIFAR-10 dataset, a public dataset commonly used for benchmarking in image classification, found that the top two best-performing models were both manually designed models [10]. However, the gap between the manual and AutoML models was very small (<1% accuracy difference). In our study, AutoKeras was able to achieve results comparable to those of the 50-layers deep ResNet-50 model (~25 million parameters) using only a simple 2-layers CNN model (43,859 parameters) with ~10-fold faster inference speed (8.46 ms), making it suitable for real-time inferencing. Research in the area of network pruning which compresses deep neural network through the removal/pruning of redundant parameters showed that it is possible to have equally performant models with up to 97% of the parameters pruned [27]. Although dissimilar to network pruning, as evidenced in our study, AutoKeras can discover efficient and compact model architectures through the NAS process. However, NAS generated models are typically limited to variants or combinations of modules derived

from existing human designed CNN architectures [10, 14, 15]; although recent innovations in NAS have uncovered novel CNN architectures such as SpineNet for object detection [28].

For the image regression task, lodged plot images were used as inputs to predict the lodging score. The best AutoKeras model (from 100 trials) also performed better in this task (R^2=0.8273, RMSE=10.65, MAE=8.24, MAPE=13.87%) compared to the ResNet-50 model (R^2=0.8079, RMSE=10.81, MAE=7.84, MAPE=14.64%) used in transfer learning. The AutoKeras model resembled a mini 8-layers Xception model (207,560 parameters). In its original form, the Xception network is 71-layers deep (~23 million parameters) and consists of three parts: the entry flow, middle flow and exit flow [24]. These three parts and two key features of the Xception network, namely the depthwise separable convolutions and skip connections originally proposed in ResNet [22] were discernable from the mini Xception model [24]. The Xception network outperformed other modern DL architectures such as Inception-V3, ResNet and VGGNet in image classification on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [24]. This may partly explain the strong image regression performance offered by the mini Xception model. In our study, the mini Xception model and to a lesser extent, the ResNet-50 model produced predictions exceeding the maximum value of 100 for the lodging score. One possible approach to limit the model output values is to implement a sigmoid activation function in the final output layer which will generate values between 0 and 1, and these can be inverse transformed to lodging scores. However, this also meant re-training the models using lodging scores normalized to values between 0 and 1. Although this process is relatively straight-forward for the ResNet-50 model, re-training and modification of the AutoKeras model would likely entail further manual optimization and fine-tuning which may not be ideal compared to the automatic model tuning and hyperparameter optimization performed in AutoKeras, in addition to diminishing the attractiveness of an automated machine learning approach. As such, predictions exceeding the value of 100 were simply binned to the value of 100. This resulted in an appreciable improvement in the performance scores for the AutoKeras model (R^2=0.8351, RMSE=10.10, MAE=7.09, MAPE=12.70%) but no change was observed for the ResNet-50 model.

The lodging score originally proposed by Fisher and Stapper [19] is calculated from the lodging angle and lodged area (lodging score = angle of lodging from vertical position/90 x % lodged area). In our study, the angle of lodging is replaced by lodging severity, which is a score of 1 to 3 assigned to light, moderate and heavy lodging grades as determined by visual assessment of lodged plot images. Consequently, the use of lodging severity introduced a fractional binning (1/3, 2/3, 3/3) of lodging scores, leading to a slightly staggered distribution as evidenced in the regression plots (Figure 10). In addition, under- or over-estimation of lodging scores may happen for plots within the same lodging severity grade. For example, plots with 100% lodged area and lodging angle of 65^0(lodging severity 3) and 90^0 (lodging severity 3) will have the same lodging score of 100 in this study as opposed to

scores of 72 and 100 according to the original method. This may partly account for the model prediction errors in the image regression task. Nonetheless, the modified lodging score allowed a rapid evaluation of wheat lodging based on visual assessments of UAV imagery and was useful as a target for image regression. For detailed assessment of lodging and model performance in lodging score prediction, future studies should incorporate manual ground-truthing of the lodging angle and lodged area to enable a more accurate calculation of lodging scores.

One of the challenges in this study was getting AutoKeras to perform stably and complete the DL experiments. Initial attempts were often met with an out of memory (OOM) error message, arising from AutoKeras trying to load models too large to fit in the GPU memory. Prior to version 1.0, AutoKeras had a GPU memory adaptation function which limits the size of neural networks according to the GPU memory [21]. However, beginning version 1.0, this function is no longer implemented in AutoKeras (personal communication with Haifeng Jin, author of AutoKeras) and may partially account for the OOM errors. To partly circumvent this issue, we had to resize all input images (1544 x 371) to a smaller size of 128 x 128, which allowed AutoKeras to complete experiments up to 100 trials. The impact of the downsized images on AutoKeras model performance would need to be ascertained in future studies, although the 128 x 128 image size is within common ranges observed in DL models, for example, LodgedNet (64 x 128)[26] and established modern CNN architectures (224 x 224)[22, 29, 30]. As AutoKeras is undergoing active development, we are hopeful that issues encountered in our study will be resolved in future releases. It will be interesting to explore AutoKeras' performance with larger model search space (>100 trials) using higher resolution input images coupled with image data augmentation where appropriate in future studies.

Although transfer learning using existing modern CNN architectures provides a good starting point in computer vision tasks, a major limitation is that these CNN models have been trained using 3-channels RGB images and this prevents direct application of these models for image sources beyond 3-channels, such as multispectral and hyperspectral images [26]. In addition, existing CNN architectures may not always provide the best performance compared to custom-designed models. However, modification of existing CNN architectures or manually designing CNN models are time-consuming and technically challenging. In this regard, AutoML provides a compelling alternative as it can deliver high-performance CNN models with minimal effort from the user as evidenced in our study. Furthermore, one of the added benefits of AutoML is its potential through NAS to discover high-performance compact model architectures which are ideal for real-time inferencing. Although not demonstrated in this study, AutoKeras can easily accept inputs of varying sizes and dimensions, making it ideal for use on diverse sensor-derived data including multispectral and hyperspectral imagery. Perhaps the primary downside of AutoML, inherent in DL practices, is the requirement for modern and sometimes costly GPU hardware for model training. This is expected to be offset in time

by the rapid growth in GPU computing power. Another concern relating to AutoML is the reproducibility of results owing to the stochastic nature of NAS [31]. This concern can largely be addressed through making available all datasets, source codes (including exact seeds used) and best models reported in the NAS study – a practice embraced in this study (see Data Availability).

Results in our study demonstrate that AutoML is a powerful and flexible tool for image-based plant phenotyping and should be in the toolbox of both novice and expert users alike. Moving forward, with the exponential rise in GPU computing power, AutoML systems are expected to become more ubiquitous. In tandem with existing DL practices, they can contribute significantly towards streamlined development of image analytical pipelines for HTP systems integral to improving breeding program and crop management efficiencies. Consequently, a comprehensive AutoML system tailored for agricultural research and application is being developed as one of the AgTech SmartSense products at SmartSense iHub, Agriculture Victoria.

Acknowledgments: We thank Dennis Ward and Emily Thoday-Kennedy for technical support in conducting the field experiment.

Author contributions: JCOK designed the deep learning experiments and analyzed the data. JCOK, GS and SK conceived and designed the study. SK managed the field experiment. JCOK, GS and SK wrote and edited the paper. All authors read and approved the final manuscript.

Funding: This study is funded by Agriculture Victoria.

Competing interests: The authors declare that they have no competing interests.

Data Availability: Dataset containing 1,248 digital images (1,544 x 371 pixels) of individual wheat plots with varying grades of lodging and corresponding lodging data are available on Zenodo [32]. Ground truth data for lodging status and calculated lodging scores are provided as a CSV file. Source codes required to replicate the analyses in this article and the best performing models reported for AutoKeras are provided in a GitHub repository [33].

Figures and Tables

Figure 1: Wheat breeding field experiment. Ground control points (GCPs) indicated in figure.

Figure 2: Experimental workflow for wheat lodging assessment.

Figure 3: Different wheat lodging severities. Wheat plot images were first classified as non-lodged or lodged using ground truth data. Plots identified as lodged were assessed visually and divided into three lodging severities (light, moderate and heavy) based on lodging angles.

Figure 4: Automated machine learning with AutoKeras. (a) AutoModel for image classification. (b) ImageRegressor for image regression.

Figure 5: Transfer learning with the ResNet-50 architecture. Outputs from the ResNet-50 was joined to a global average pooling 2D layer and connected to a final dense layer, with the activation function set as either "sigmoid" for image classification or "linear" for image regression.

Figure 6: Confusion matrix of the classification results using (a) ResNet-50 and (b) AutoKeras. Results for AutoKeras were produced using the model discovered from 100 trials.

Figure 7: Best AutoKeras model architecture for image classification. A simple 2-layers CNN model with 43,895 parameters was the best architecture discovered in 50 and 100 trials. The best classification performance was provided by the model from 100 trials.

Figure 8: Classification accuracies for AutoKeras models from 10, 25, 50 and 100 trials.

Figure 9: Best AutoKeras model architecture for image regression. An 8-layers mini Xception model with 207,560 parameters was the best architecture discovered in 50 and 100 trials. Three main parts (entry, middle and exit flows) and two key features (examples indicated by green boxes), namely the depthwise separable convolutional layers and skip connections from the original Xception network [24] can be discerned from the architecture. The best regression performance was provided by the model from 100 trials.

Figure 10: Regression plot for predicted outputs using (a) ResNet-50 and (b) AutoKeras. Predicted lodging scores (y_predict) were plotted against actual scores (y_test). Regression line (blue or red) is shown with a 95% confidence interval and dashed black line indicates the 1:1 line.

Table 1: Performance metrics of modals in image regression.

Supplementary Materials

Figure S1: Best AutoKeras model architecture from 10 and 25 trials for image classification.

Figure S2: Best AutoKeras model architecture from 10 and 25 trials for image regression.

References

1. Ninomiya, S., F. Baret, and Z.-M. Cheng, *Plant Phenomics: Emerging Transdisciplinary Science.* Plant Phenomics, 2019. **2019**: p. 2765120.
2. Tardieu, F., et al., *Plant Phenomics, From Sensors to Knowledge.* Current Biology, 2017. **27**(15): p. R770-R783.
3. Mir, R.R., et al., *High-throughput phenotyping for crop improvement in the genomics era.* Plant Science, 2019. **282**: p. 60-72.
4. Jiang, Y. and C. Li, *Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review.* Plant Phenomics, 2020. **2020**: p. 4152816.
5. Khan, A., et al., *A survey of the recent architectures of deep convolutional neural networks.* Artificial Intelligence Review, 2020: p. 1 - 62.

6. Minaee, S., et al., *Image Segmentation Using Deep Learning: A Survey.* ArXiv, 2020. **abs/2001.05566**.
7. Wu, X., D. Sahoo, and S.C.H. Hoi, *Recent advances in deep learning for object detection.* Neurocomputing, 2020. **396**: p. 39-64.
8. Chandra, A., et al., *Computer Vision with Deep Learning for Plant Phenotyping in Agriculture: A Survey.* ArXiv, 2020. **abs/2006.11391**.
9. Watt, M., et al., *Phenotyping: New Windows into the Plant for Breeders.* Annual Review of Plant Biology, 2020. **71**(1): p. 689-712.
10. He, X., K. Zhao, and X. Chu, *AutoML: A Survey of the State-of-the-Art.* ArXiv, 2019. **abs/1908.00709**.
11. Truong, A., et al. *Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools*. in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. 2019.
12. Zoller, M.-A. and M.F. Huber, *Benchmark and Survey of Automated Machine Learning Frameworks.* arXiv: Learning, 2019.
13. Zoph, B. and Q.V. Le, *Neural Architecture Search with Reinforcement Learning.* ArXiv, 2016. **abs/1611.01578**.
14. Elsken, T., J.H. Metzen, and F. Hutter, *Neural Architecture Search: A Survey.* ArXiv, 2019. **abs/1808.05377**.
15. Wistuba, M., A. Rawat, and T. Pedapati, *A Survey on Neural Architecture Search.* ArXiv, 2019. **abs/1905.01392**.
16. Tan, M. and Q.V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* ArXiv, 2019. **abs/1905.11946**.
17. Wang, N., et al., *NAS-FCOS: Fast Neural Architecture Search for Object Detection.* ArXiv, 2019. **abs/1906.04423**.
18. Weng, Y., et al., *NAS-Unet: Neural Architecture Search for Medical Image Segmentation.* IEEE Access, 2019. **7**: p. 44247-44257.
19. Fischer, R.A. and M. Stapper, *Lodging effects on high-yielding crops of irrigated semidwarf wheat.* Field Crops Research, 1987. **17**(3): p. 245-258.
20. Sun, Q., et al., *Monitoring Maize Lodging Grades via Unmanned Aerial Vehicle Multispectral Image.* Plant Phenomics, 2019. **2019**: p. 5704154.
21. Jin, H., Q. Song, and X. Hu, *Auto-Keras: An Efficient Neural Architecture Search System.* Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.
22. He, K., et al., *Deep Residual Learning for Image Recognition.* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: p. 770-778.
23. Xie, S., et al., *Aggregated Residual Transformations for Deep Neural Networks.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 5987-5995.
24. Chollet, F., *Xception: Deep Learning with Depthwise Separable Convolutions.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 1800-1807.

25. Roeder, L. *Netron: visualizer for neural network, deep learning and machine learning models.* [cited 2020 8th July]; Available from: https://github.com/lutzroeder/netron.
26. Mardanisamani, S., et al., *Crop Lodging Prediction From UAV-Acquired Images of Wheat and Canola Using a DCNN Augmented With Handcrafted Texture Features.* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019: p. 2657-2664.
27. Salama, A.I., et al., *Pruning at a Glance: Global Neural Pruning for Model Compression.* ArXiv, 2019. **abs/1912.00200**.
28. Du, X., et al., *SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization.* ArXiv, 2019. **abs/1912.05027**.
29. Huang, G., Z. Liu, and K.Q. Weinberger, *Densely Connected Convolutional Networks.* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 2261-2269.
30. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition.* CoRR, 2015. **abs/1409.1556**.
31. Li, L. and A. Talwalkar. *Random Search and Reproducibility for Neural Architecture Search*. in *UAI*. 2019.
32. Koh, J., G. Spangenberg, and S. Kant, *Dataset*. 2020, Zenodo. DOI: 10.5281/zenodo.3952422
33. Koh, J., G. Spangenberg, and S. Kant. *Source Codes for AutoML Manuscript*. 2020 [cited 2020 August 19th]; Available from: https://github.com/AVR-SmartSense/automl_paper.

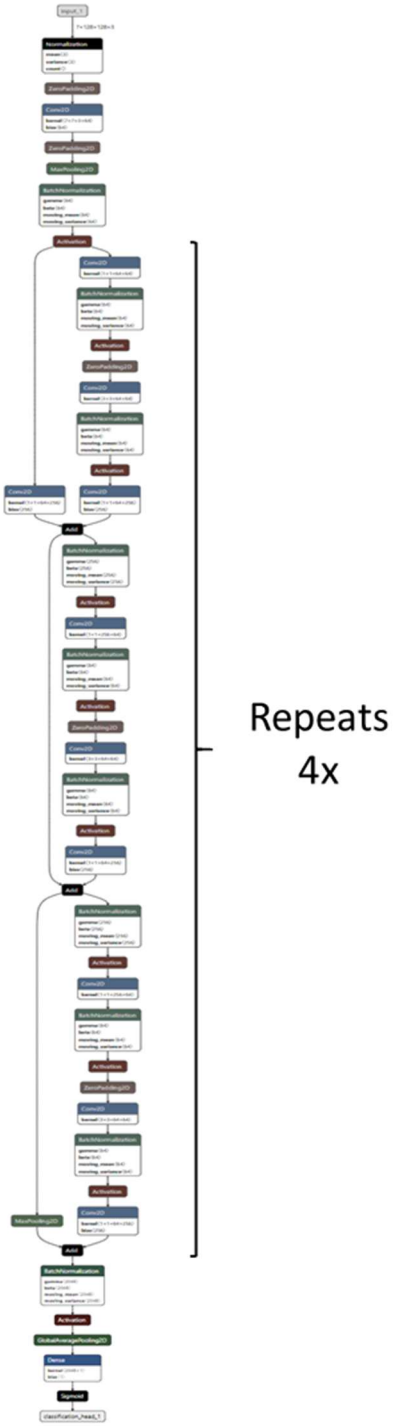

Figure S1 Best AutoKeras model architecture from 10 and 25 trials for image classification.

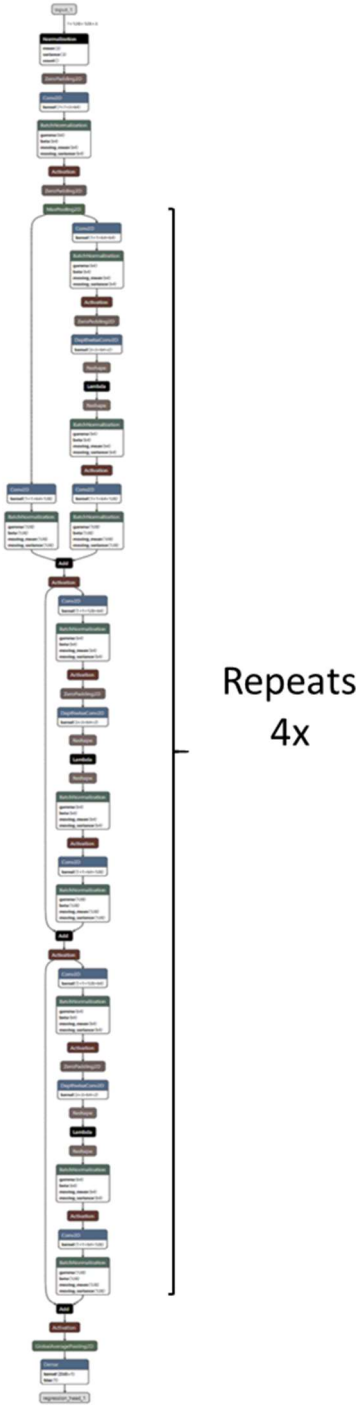

Figure S2 Best AutoKeras model architecture from 10 and 25 trials for image regression.