An efficient yield prediction model using synthetic inference from L-systems

Chris C. Napier [1], *David M.* Cook[1, 2*], and *Leisa J.* Armstrong[1]

[1]School of Science, Edith Cowan University, 270 Joondalup Drive, Joondalup, Australia
[2]National University of Security Science, Islamabad, Pakistan

Abstract. The ability to predict and estimate the harvest production is of vital significance to the food security and financial value of global agriculture. The use of synthetic inference in determining crop yield estimations cannot be underestimated. L-systems uses high-level estimation dependent upon inference techniques through sources ranging from real to synthetic data. This paper combines intelligent and highly visual features to infer crop characteristics. It provides a method of assembling superior synthetic datasets that reduce the reliance upon time-consuming fully trained neural networks. It demonstrates a mature approach to using synthetic inference that provides cost-effective reliable crop yield estimations. This model allows for scalable and affordable application of synthetic inference as part of an optimised yield-positive farming enterprise in crops such as wheat. By leveraging synthetic inference, digital twins, and visualization, scientists, agronomists, and farmers can gain deeper insights into improved crop management.

1 Introduction

This paper builds upon deep learning work [1], and L-system work [2] to develop a computer-platform independent L-system called L-NAP. This paper introduces a new and refined two-stage [3] synthetic inference process [4][5][6] named L-NAPI. The process of L-NAPI is designed to estimate crop yield [7], by following an algorithmic approach that employs the existing L-NAP synthetic plant dataset. This performs direct image comparisons of synthetic and real plants. The synthetic inference process is open and visible to scrutiny, since the synthetic data is open, and is not a hidden black-box process within a neural network [8]. This research demonstrates how accurate and efficient the L-NAPI synthetic inference process is in measuring plants.

1.1 L-systems define and create synthetic plants.

L-systems follow the standards set by Lindenmayer, Prusinkiewicz, and Hanan [10][11]. One of the functions of L-systems is to facilitate the creation of 3D plants. This paper refers to a novel pure-Python [9] L_NAP.py class demonstrated in Fig. 1 as an exemplar of an extensible and platform-independent L-system [2]. This class has been upgraded to process deterministic, context-sensitive, stochastic, bracketed, and parametric L-system algorithms [10][11]. Its novelty is with the L-system algorithms that are processed as separated L-system production rules and L-system parameters. This allows a clear understanding of the plant that is produced by the algorithm and allows an independent variation of the L-system parameters as required before and during plant growth, when there is a context-sensitive external interaction between the plant and its surroundings, or internally between parts of the plant. In the L_NAP.py class these parameter values are stored and communicated in standard Python dictionaries, for clear data definitions and readable code. This paper addresses the lack of visibility in neural networks [8] and answers how synthetic inference produces visible and accurate crop yield estimates based on synthetic plant models.

2 Review of Literature for Crop Yield Prediction Using Synthetic Inference

Recent attempts to create systems for reliable crop yield prediction have involved the integration of a diverse set of data sources and models. These have centred on the amalgamation of remote sensing data, weather data, soil information, crop growth models and machine learning approaches. The inclusion of synthetic inference assists with the development of a multi-source integrated approach that can be applied in agricultural modelling.

2.1 Exemplar approaches for predicting wheat crop yields using synthetic inference.

There are five exemplars showing value and repeatable reliability in terms of results and accepted practice.

2.1.1 Remote Sensing and Machine Learning.

The first involves combining Remote Sensing and Machine Learning. The combination of remote sensing data such as satellite imagery and Machine Learning models such as Random Forests [12], Support Vector

*Corresponding author: david_cookind@yahoo.com

Machines [13], and Deep Neural Networks [14], have provided enhanced yield prediction by correlating crop health data with vegetation indices such as NDVI and EVI. Past examples include time-series data from Sentinel-2 or Landsat satellites to monitor wheat fields throughout a growing season [15]. With ground truth data this approach allows training of a machine learning model merging vegetation indices, phenological stages and weather conditions.

2.1.2 Data Fusion &APSIM.

The second exemplar has encouraging results using Data Fusion in concert with multiple sources. These models incorporate complex interactions between weather data, soil information, management practices and historical yield data. This kind of fusion uses a crop growth simulation model like APSIM (Agricultural Production Systems sIMulator) [16][17]. This model integrates weather forecasting, soil moisture levels, nitrogen application rates and Machine Learning techniques to develop a highly informed Decision Support System (DSS) [18][19].

2.1.3 Hybrid Models combining Process-based & Machine Learning approaches.

The third exemplar draws from Hybrid Models that combine Process-Based and Machine Learning approaches. In these combinations there is still a reliance upon a process-based growth model such as APSIM or WOFOST (WOrld FOod STudies) with the integration of machine learning methods to achieve an inclusion of synthetic inference [20], [21]. This type of approach simulates the biophysical processes of crop growth with a mixed grouping to incorporate machine learning models to correct biases and improve predictions as additional data is added. In this type of model there is benefit in including the output of APSIM (for example using biomass and a leaf area index) and to feed this into a neural network to develop a final yield prediction [22][23].

2.1.4 Deep Learning with Synthetic Datasets

The fourth exemplar combines Deep Learning with Synthetic Datasets. These examples use learning models such as Convolutional Neural Networks (CNNs) that are trained on synthetic datasets. That have been generated from high-fidelity crop simulation models together with real world observational data [24][25]. These models can learn spatial and temporal patterns that influence wheat yields and are trusted for the provision of predictions where there are complex environmental interactions. This shows the benefit of generating synthetic data and having a process-based model that allows for a wide range of different weather scenarios and management practices. This is a useful model that allows for aggregated practices across different regions and climate conditions.

2.1.5 Bayesian Networks & Ensemble Methods

The fifth exemplar is based upon Bayesian Networks and Ensemble Methods. These methods include Random Forests to integrate different sources of uncertainty and variability in wheat yield predictions [26][27][28]. In Bayesian networks there is a commitment to expert knowledge, probabilistic reasoning, and observational data to infer yield outcomes. By using an ensemble model, it is possible to combine multiple sources of data including crop growth models, weather-driven data models, and remotely sensed vegetation indices. This approach can produce a consensus yield prediction that incorporates uncertainty.

2.2 Increased inference and reduced uncertainty from L-systems

These inference models offer variations in predictive modelling that apply across different regions conditions and prevailing levels of uncertainty. The two-stage synthetic inference process can be integrated into any of the five exemplars reviewed. This offers the additional benefits of L-systems in terms of a reinforced combinatory approach using mathematical modelling, data integration and simulation to enhance a higher level of understanding and prediction of crop yields (Table I).

Table 1. Benefits of Synthetic Inference Using L-systems in predictive Crop Yield Estimates

	Synthetic Inference Integration Benefits
1.	Detailed Representation of Plant Architecture and Growth
2	Integration of Multiscale Biological Processes
3.	Modeling of Environmental Interactions
4.	Incorporation of Genetic and Phenotopic Variability
5.	Dynamic and Adaptive Modeling Capability
6.	Improved Light Inception and Photosynthesis Modeling
7.	Enhanced Predictive Accuracy through Data Integration
8.	Facilitation of Precision Agriculture Practices
9.	Support for Decision-Making in Crop Management

Using L-systems, synthetic inference provides high accuracy in predictive output without the reliance on heavily trained advanced machine learning and deep learning techniques. Synthetic Inference is a highly sought after component for crop yield estimation.

3 An Extensible Method L-NAP

The extensible feature of the L-NAP Python class file allows a user to add or replace functionality in L-NAP to suit their needs. This section describes the implementation of an extensible method combining the benefits of a program that draws and illuminates a synthetic plant part in the form of a wheat head. For example, a new command type named “Multi” was

recently added as an extension of "Draw", to allow drawing commands which have multiple values to implement extended context-sensitive functionality. Earlier a "Curve" data type was added for general use, to define a sequence of drawing commands, each preceded by a pitch and a turn.

Fig. 1. Comparison of a synthetic 3D model (L-NAP) and a real wheat head plant

3.1 Synthetic 3D plants and real plant measurement

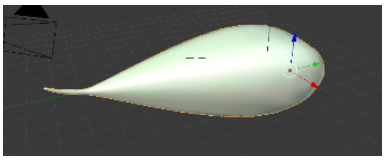

Fig. 2. An individual synthetic 3D grain object (from Fig.1.)

The 3D plant on the left in Figure 1, is a synthetic wheat head, created as drawing commands by L_NAP.py that are drawn, illuminated, and visualized by Blender [29]. The resultant synthetic plant is created by the algorithm described in part IV. Its structure is defined by L-system rules, and its size and 3D shape are defined by L-system parameters that will produce a plant like the real plant on the right. To create this plant, multiple individual synthetic grain objects, matching the one shown in Figure 2, are scaled, and positioned following parameter guidance, with small random grain size and orientation variations, with awn grain extensions along the plant.

4 Proposed Approach: A Refined Two-Stage Synthetic Inference Process

This research applies 2D views of synthetic plants in direct comparisons with real plant images, to recognize and measure real plants using Synthetic Inference [3]. A refined two-stage process named L-NAPI has been developed based on previous findings. It improves the accuracy of direct comparisons and significantly reduces the comparison times, compared with the previous single-stage process [1][2]. The improvement in accuracy and processing time is related to the smaller size of the top-edge of Figure 3 that is sensitive to edge variations (compared to a whole synthetic plant image), and to locating edges since edges exist in all plants.

4.1 First Stage – Top Edge Recognition

The first stage of the refined two-stage synthetic inference process recognizes the top-edges of real plants. This builds upon the use of edge detection [30] [31] and involves matching the edge plus part of a partial plant. Its advantage is that a single partial top-edge image can efficiently locate the top-edges of varying real plant images of a suitable size, as shown in Figure 4, where red bounding boxes indicate the actual matching area. This first stage process locates separate plants significantly faster than using full plant images. At present it is a one-size-fits-all approach, which needs the addition of smaller top-edge images, and edge images of other objects such as leaves to improve the overall recognition score.

Fig. 3. A synthetic top-edge partial image with matched partial images shown by bounding boxes.

The single synthetic wheat head top-edge shown above is the locator in the first stage of the refined real wheat head location process. The red bounding boxes show the located top-edges. The coloured lattice squares show the extent of located objects in and around each top-edge. Lattices are delimited by low image intensity pixels.

4.2 Second Stage – Whole Object Location by Lattice Squares

The second stage of the refined two-stage synthetic inference process L-NAPI calculates the extent of the whole visible object, at and around the located top-edge, as a sequence of adjacent lattice squares [32]. This stage relies on real image intensity variations to locate object edges and is coded to separate overlapping objects. This is shown below by colored lattice squares in Figure 4.

Fig. 4. Object location indicated by lattice squares.

Smaller plant objects are not yet located, and the pink central double plant (two plants side by side) was not

separated into two plant objects, since there was an insignificant top-edge variation. A modified process which located top and bottom edges, found more pronounced bottom-edges, but included the challenge of choosing between bottom and top edged.

The calculated lattices play an important part in plant measurement and define the extent of the plant image as an area of a particular shape. These values represent the ground truth of the visible plant and are directly employed in later Intersection-over-Union [33] image matching. These characteristics are given to the synthetic dataset to request plant views, of similar size, area, and orientation. Lattices can be extended into the third dimension to represent solid 3D objects, and the synthetic inference process can also be extended to locate and measure objects such as the branches of a tree. This would require stereo imaging of real plants.

Image scaling is considered prior to the two-stage process; however scaling effects are visible in Figure 4, on the top-left plant that has separate green and blue lattice parts, caused by the implementation of limited size lattices. This size limit ensures that adjacent white-yellow lattices correctly identify separate plants. To overcome these limits, further development is required.

4.3 Dataset Matching and Measurement

The size and shape of each located lattice object is given to the L-NAP synthetic dataset, which returns information on possible matching images. Since the dataset contains ordered and rotated 2D images and 3D models, a look-up is performed to find matches. For the 2D image case, Intersection-over-Union scores are compared to find the best synthetic plant image which matches each located real plant image.

Fig. 5. A global wheat head field image manually annotated with bounding boxes by GWHD

These best matches are shown in Figure 6, in translucent brown on each located real image. Figure 6 shows that automatic annotation is performed with white captions, and synthetic bitmasks, on the same field image, using only synthetic wheat head data. The synthetic method locates and recognises 20 of 23 wheat heads plus 1 false positive with an accuracy of 79.17 %. The separately captioned individual accuracy percentages are Intersection-over-Union area-matching scores.

Fig. 6. Automatic annotation shown by white captions, and synthetic bitmasks.

The Intersection-over-Union scores of the best match of synthetic image to real image is given as a caption above each real image, together with the grain-row count and orientation of the best matching synthetic image. The grain-row count is a known value for each synthetic image, allowing both matching and measurement.

5 The L-NAP Plants: Rules and Parameters Files

The short L-NAP algorithm, in a file named Plants.py, is listed below. It defines the L-system rules to create Synthetic Wheat head models. To create these models, it uses a complimentary L-system parameters file. This short algorithm is written in the L-NAP representation of an L-system. It employs a readable Python code with clear Python syntax [34] and it is run by L-NAP to generate drawing commands. From these commands, the L-NAP framework [2] creates a sequence of synthetic wheat head 3D models, and multiple rotated and illuminated 2D views, which populates a synthetic dataset. This dataset is employed in the synthetic inference process of synthetic to real image comparisons, described herein.

```
#
# Plants.py, L-system production rules in L-NAP
format
#
from L_NAP import *    # import L_NAP.py, and
from Plants_p import * # import Plants_p.py
                       # with parameters and
                       # related functions.
#
# Initialise a Plants object with its parameters.
#
Plants = L_NAP("Plants_p")
#
# Define Plants L-system production rules:
# Axiom, Stalk, GrainSpace, Grain and Awn.
# Note the (square) bracketed sub-rules Pitch,
Object,
# and Awn, in the main Grain rule, which indicate
# save and restore of positional and orientation
# information, before and after the Grain rule.
#
def Axiom():         Plants(Stalk, GrainSpace)
def Stalk():         Plants(Curve)
def GrainSpace():        Plants(Pitch, Turn, Roll,
                         Draw, Grain, GrainSpace)
def Grain():         Plants([Pitch, Object, Awn])
def Awn():      Plants(Curve)

#
```

```
# Grow the Plants starting each at Axiom,
# following the above rules and applying the
# parameters, and functions defined in
Plants_p.py.
#
while Plants.next_plant():
    while Plants.next_stage(Axiom):
        Plants.grow()
# L-NAP Parameters file Plants_p.py, defining and
processing L-system parameters
#

# L-system parameter definitions (truncated)
p = {
  'Stalk_length':    0.02,
  'Stalk_width':          0.1625,
  'Cutoff_factor':   0.5,
  'Grain_factor':    0.5,
  'Awn_length':      4.0,
  'Grain_scale':          0.11,
  'Grain_pitch':          63.0,

# Special L-system parameters
  'stage':      0,
  'stages':          110,
  'run':        "LN",
  'cmds_file':       0,

  'Plants':          4,
  'Groll': (+4,-1,-1,-1,-1),
  'Grolln':          0,
  'PlantNr':         0,

  # Parameter values for plant variations.
  'deltas': ( ('Awn_length',   1,   2,   4,    8),
),
}
"""
  Define a parameter function and rule functions

"""
def Parameters_p():  return p

def Axiom_p(): p['stage']  =  0;  return  {'rule':
"Axiom"}
```

and so on…

5.1 Creating Plants

The resulting annotated commands are performed to create plants. The command file is executed by a python program in Blender to create and visualize the plants. This command file has 4,500 lines, summarised as:

```
plantnr(1)# Start
# Plant: 1: has parameter Awn_length: 1
# stage: 0, stages:110
# Grain: random normals: 0.1785 0.6527 0.2382
0.0994 0.2432 0.7223 0.8557 0.8302
     draw(length=0.8928, width=0.1625)
#'F' : Stalk Draw
     pitch_up(angle=0.0165)
#'&' : GrainSpace Pitch
     turn_left(angle=0.7759)
#'+' : GrainSpace Turn_left
     roll_left(angle=31.0058)
#'\' : GrainSpace Roll
     draw(length=0.0144, width=0.2438)
#'F' : GrainSpace Draw
     save()
#'[' : save
     pitch_up(angle=63.0000)
#'&' : Grain-1(110) Pitch
     draw_obj("Seed",
scale=(0.0640,0.0628,0.0639))        #'~' : Grain-
1(110) Object
     pitch_up(angle=-50.0000)
#'.' : Awn Pitch
     draw(length=1.0000, width=0.0250)
#'F' : Awn Draw
     restore()
#']' : restore
```

Note *that above shows 1 grain of 110 grains of a plant.*

6 Results: The Refined Crop Yield Estimation Using Synthetic Inference

There is a two-stage process for estimating crop yield. It shows the level of each visible wheat head from real image pixel values, and then shows how the spun synthetic wheat heads are related to each lattice area.

In step one the estimate of the Crop Yield per unit area of GWHD [35] Domain-1 is calculated following the two-stage location and recognition processes. In the first step, the extent of each visible wheat head is calculated from real image pixel values and is shown as coloured lattice squares in Figures 7, 8, and 9. The lattice area is labelled in the caption by A, and the area value is scaled to grain row units to compare with the 2nd stage R value.

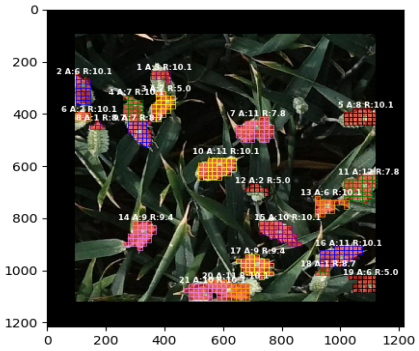

Fig. 7. Automatically annotated Crop Yield estimates of Area and Grain row count for each located real wheat head.

In the second step, rotated synthetic wheat heads are compared to each lattice area by the Intersection-over-Union method to find the best area match.

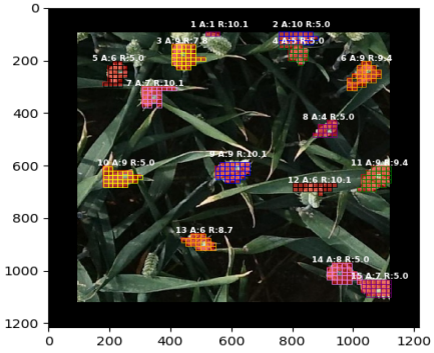

Fig. 8. Automatically annotated Crop Yield estimates of Area and Grain row count for located real wheat head in GWHD.

The synthetic grain row count of the best area match is shown by "R' in the caption. The domain total grain row count is calculated by the sum of the individual grain row counts shown as 'R', over all located wheat heads and all domain images. The grain row counts are summed over all located wheat heads and all Domain field images to give a Domain total grain row count. For example, the caption "10 A: 11 R: 10.1" above the central object in Figure 7, annotates the located wheat head number 10, and gives the visible area estimate of 11, and the grain row count estimate of 10.1. The left bottom wheat head of Figure 6 with caption "14 A:9 R: 9.4", shows for wheat head number 14, the similarity of an estimated area of 9 and estimated grain row count of 9.4. The Figures 7, 8 and 9 show the resulting GWHD real field images [35] with automatic annotations of the area and grain row count of each located real wheat head. The 'A' value is the scaled wheat head Area based on the calculated (coloured) lattice, and the 'R' value is the grain row count of the best match of the real image lattice to rotated synthetic images. The 'R' values are summed over the Domain for the total grain row count.

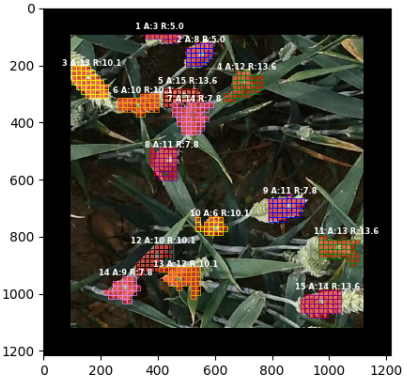

Fig. 9. Annotated Crop Yield estimates of Area and Grain row count for located & numbered real GWHD wheat head.

6.1 Post-process Crop Yield Estimate

The final estimate of the Crop Yield of Domain-1, represented by 58 field images, depends on four conditions. Firstly, the sum of the estimated grain row counts over all recognized wheat heads, over all field images, are taken as the "Domain_Grain_Row_Count". Secondly, the number of grains in a grain row, taken as "Grains_per_row", which would be 6 for Domain-1, but may differ for other domains. Thirdly, the weight of a wheat head grain and connected plant parts, is taken as the "Single_Grain_Weight". Fourthly, the calibration of the field image areas give the total recorded ground area of the Domain, taken as the "Visible_Domain_Area".

The Domain_Grain_Row_Count was calculated as 11,616, giving the estimate Domain_Grain_Count = Grains_per_row * 11,616, as a value of 69,696 for the total grains located in the 58 field images. (Assuming 6 grains per wheat head row. Equation 1 estimates the total weight of visible grains is 69,696 times "Single_Grain_Weight", and the Estimated Crop Yield per unit area, is this product divided by the "Visible_Domain_Area", which gives the GWHD Domain-1, with units such as Tons/hectare:

$$\textit{Estd. Crop Yield tons/hect} = 69,969 * \frac{\textit{Single Grain Weight}}{\textit{Visible Domain Area}}$$

Equation 1. Estimated Crop Yield (units of weight over area)

7 The L-NAP Coding Examples

The following section provides the coding examples for the sourced process. Example 1 is based on AB from (Prusinkiewicz & Lindenmayer, 1990) and is the derivation process in a deterministic and context-free DOL-system, using two letters A and B, with associated rewriting rules, and an axiom rule to begin the process. An object named 'plant' is created by the L_NAP class from user parameters *run* and *stages*. The plant object implements the L-system rewriting rules 'Axiom', 'A', and 'B'. The class functions 'next_stage' and 'grow' implement the DOL-system process. Working Example 1: shows a listing of L-system Algorithm Ex1.py:

```
# Ex1.py
from L_NAP import *       # Import the L_NAP.py
class file and related functions

# A Local dictionary of L-system parameter values
parameters = {'run': "Ex1", 'stages': 6}

# Create a plant object based on the local L-
system parameters
plant = L_NAP(parameters)

# Define L-system rewriting production rules
def Axiom():    plant(B)        # Axiom -> B
def A():        plant(A, B)     # A -> AB
def B():        plant(A)        # B -> A

# Derive the plant in stages
while plant.next_stage(Axiom): # For all stages
starting at
     plant.grow()         # Derive (grow) the
plant to
```

The code (Ex1) provides a working of the resulting axiom and derivation stages written to file Ex1.cmds. The 'plant' is derived from the axiom rule B, in six stages as L-strings: A, AB, ABA, ABAAB, ABAABABA, and ABAABABAABAAB.

```
# Axiom == B
# 1     A
# 2     AB
# 3     ABA
# 4     ABAAB
# 5     ABAABABA
# 6     ABAABABAABAAB
```

Example 2 provides a Context-Sensitive L-system Algorithm Cx2.py (Right-to-Left, 2-step) example of the derivation process in a context-sensitive 1L-system, using two letters 'a' and 'b', with associated rewriting rules, and an axiom rule to begin the process. An object named 'module' is created from user parameters 'run' and stages. The 'module' object implements the L-system rewriting rules 'Axiom', 'a', and 'b', to represent message flow from right to left by 2 steps at each stage. Rule 'a' is sensitive to its context, and is implemented by L_NAP class function followed by () given two arguments a and b, showing two steps:

def a(): module(b) if module.followed_by(a, b) else module(a)
a becomes b if it is followed by a, b else no change.

The following code provides a working example (Example 2) part 1: Listing of Algorithm Cx2.py

```
# Cx2.py
from L_NAP import *
parameters = {'run': "Cx2", 'stages': 5}
module = L_NAP (parameters)

# Define the Axiom, the initial L-string,
def Axiom():    module(a,a,a,a,a,a,a,a,b)

# Define that letter b becomes a at the next stage
def b():        module(a)

# define that letter a becomes b if it is followed
by a, b in that order, otherwise a is unchanged.
def a():        module(b) if module.followed_by(a,
b) else module(a)

while module.next_stage(Axiom):
     module.grow()        # grow the module
```

The following code provides a Working Example (Ex2), of a Context-sensitive Result file. The effect is that letter b moves from right to left by two positions in the L-string at each stage and reaches the left in 4 stages.

```
#
# Axiom == aaaaaaaab
# 1      aaaaaabaa
# 2      aaaabaaaa
# 3      aabaaaaaa
# 4      baaaaaaaa
# 5      aaaaaaaaa
```

8 Discussion and Conclusions

The L-NAP framework creates realistic plants that visually match real plants. With automatic annotation real plant images are matched using synthetic plants. In a large-scale processing of 1,522 wheat head plants over 58 field images of one GWHD Domain [35], a recognition accuracy score of 69.65% was achieved. This score was manually confirmed to include false and true positives (FP, TP), and false and true negatives (FN, TN) according to the formula:

$$\textit{Wheat Head Recognition Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Matching synthetic plants are taken as a measurement of real plants described as synthetic inference whereby synthetic plants infer the characteristics of real plants. In this case one whole GWHD domain of 1,522 wheat heads, was estimated as:

$$\textit{Estd. Crop Yield tons/hect} = 69,696 * \frac{\textit{Single Grain Weight}}{\textit{Visible Domain Area}}$$

8.1 Future Work

This research extends L-system parameter adjustments for effective domain transfer [36], and automated backpropagation [37]. Creating datasets can be developed for affordable crop yield forecasting. The application of an L-systems approach using synthetic inference offers inexpensive yet precise yield prediction in forecasting crop yields. It delivers reliable predictions benefitting both small and large-scale agriculture. The method allows scalable, affordable, crop yield analytics, resource-efficient yield monitoring, and enhanced yield prediction.

This synthetic inference can be improved by adding synthetic top-edge images to locate small real plants and by adding synthetic leaf images [38] to locate real leaves to reduce the false positives score. The natural extension of applied synthetic inference is visualization [39] to allow crop yield prediction through Digital Twin applications. By leveraging synthetic inference, digital twins, and visualization, agronomists and farmers can increase harvest yield and sustainability.

References

1. S. Kar, J. Adinarayana, Deep Learning and Reinforcement Learning Methods for Advancing Sustainable Agricultural and Natural Resource Management, in Harnessing Data Science for Sustainable Agriculture and Natural Resource Management, pp. 201-223 (Springer Nature Singapore, Singapore, 2024).
2. C. C. Napier, D. M. Cook, L. Armstrong, D. Diepeveen, A synthetic wheat L-system to accurately detect and visualise wheat head anomalies, in Proceedings of the 3rd International Conference on Smart and Innovative Agriculture (ICoSIA), pp. 379-391 (Springer Nature, 2022).
3. L. Du, R. Zhang, X. Wang, Overview of two-stage object detection algorithms, Journal of Physics: Conference Series 1544, 012033 (2020).
4. S. Marin, D. Miller, E. Pimentel, M. Volpe, From axioms to synthetic inference rules via focusing, Annals of Pure and Applied Logic **173**(5), 103091 (2022).
5. S. Morandage, E. Laloy, A. Schnepf, H. Vereecken, J. Vanderborght, Bayesian inference of root architectural model parameters from synthetic field data, Plant and Soil **467**, 67-89 (2021).

6. F. Seiler, V. Eichinger, I. Effenberger, Synthetic Data Generation for AI-based Machine Vision Applications, Electronic Imaging **36**, 1-5 (2024).
7. B. Basso, L. Liu, Seasonal crop yield forecast: Methods, applications, and accuracies, Advances in Agronomy **154**, 201-255 (2019).
8. J. M. Benítez, J. L. Castro, I. Requena, Are artificial neural networks black boxes?, IEEE Transactions on Neural Networks **8**, 1156-1164 (1997).
9. L. G. Silvestri, L. J. Stanek, G. Dharuman, Y. Choi, M. S. Murillo, Sarkas: A fast pure-python molecular dynamics suite for plasma physics, Computer Physics Communications **272**, 108245 (2022).
10. P. Prusinkiewicz, A. Lindenmayer, The Algorithmic Beauty of Plants (Springer, New York, NY, 1990).
11. P. Prusinkiewicz, J. Hanan, Lindenmayer Systems, Fractals, and Plants 79 (Springer Science & Business Media, 2013).
12. R. Montorio, F. Pérez-Cabello, D. B. Alves, A. García-Martín, Unitemporal approach to fire severity mapping using multispectral synthetic databases and Random Forests, Remote Sensing of Environment **249**, 112025 (2020).
13. G. Mountrakis, J. Im, C. Ogole, Support vector machines in remote sensing: A review, ISPRS Journal of Photogrammetry and Remote Sensing **66**, 247-259 (2011).
14. R. M. Sheykhmousa, M. Mahdianpari, Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **13**, 6308-6325 (2020).
15. Z. Jelínek, J. Kumhálová, J. Chyba, M. Wohlmuthová, M. Madaras, F. Kumhála, Landsat and Sentinel-2 images as a tool for the effective estimation of winter and spring cultivar growth and yield prediction in the Czech Republic, International Agrophysics **34**(3), 391-406 (2020).
16. P. J. Mitchell, F. Waldner, H. Horan, J. N. Brown, Z. Hochman, Data fusion using climatology and seasonal climate forecasts improves estimates of Australian national wheat yields, Agricultural and Forest Meteorology **320**, 108932 (2022).
17. A. Samourkasidis, I. N. Athanasiadis, A semantic approach for timeseries data fusion, Computers and Electronics in Agriculture **169**, 105171 (2020).
18. S. Fei, M. A. Hassan, Y. Xiao, X. Su, Z. Chen, Q. Cheng, F. Duan, R. Chen, Y. Ma, UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat, Precision Agriculture **24**(1), 187-212 (2023).
19. Z. Zhai, J. Fernan Martinez, V. Beltran, N. Martinez, Decision support systems for agriculture 4.0: Survey and challenges, Computers and Electronics in Agriculture **170**, (2020).
20. Y. Wu, W. Xu, H. Huang, J. Huang, Bayesian posterior-based winter wheat yield estimation at the field scale through assimilation of Sentinel-2 data into WOFOST model, Remote Sensing **14**(15), 3727 (2022).
21. X. Xu, S. Shen, F. Gao, J. Wang, X. Ma, S. Xiong, Z. Fan, Considering different water supplies can improve the accuracy of the WOFOST crop model and remote sensing assimilation in predicting wheat yield, International Agrophysics **36**(4), 337-349 (2022).
22. H. Dokoohaki, T. Rai, M. Kivi, P. Lewis, J. L. Gómez-Dans, F. Yin, Linking remote sensing with APSIM through emulation and Bayesian optimization to improve yield prediction, Remote Sensing **14**(21), 5389 (2022).
23. D. A. D. Grubert, A synergistic approach to sugarcane yield forecasting using machine learning, remote sensing, and process-based modeling, Doctoral dissertation, Universidade de São Paulo (2023).
24. S. Stiller, K. Grahmann, G. Ghazaryan, M. Ryo, Improving spatial transferability of deep learning models for small-field crop yield prediction, ISPRS Open Journal of Photogrammetry and Remote Sensing 12, 100064 (2024).
25. W. Zhao, W. Yamada, T. Li, M. Digman, T. Runge, Augmenting crop detection for precision agriculture with deep visual transfer learning—a case study of bale detection, Remote Sensing **13**(1), 23 (2020).
26. S. Fei, Z. Chen, L. Li, Y. Ma, Y. Xiao, Bayesian model averaging to improve the yield prediction in wheat breeding trials, Agricultural and Forest Meteorology **328**, 109237 (2023).
27. C. M. Utz, Learning ensembles of Bayesian network structures using random forest techniques, Doctoral dissertation, University of Oklahoma (2010).
28. D. M. Cook, E-Governance of the Forest in Common Property Resource Management, in Amir Khan and Mousumi Majumdar (Eds.), (Academic Foundation, New Delhi, India, 2011).
29. Blender, The Stichting Blender Foundation, Amsterdam (2018).
30. E. A. Sekehravani, E. Babulak, M. Masoodi, Implementing canny edge detection algorithm for noisy image, Bulletin of Electrical Engineering and Informatics **9**(4), 1404-1410 (2020).
31. H. Moreno, A. Gómez, S. Altares-López, A. Ribeiro, D. Andújar, Analysis of Stable Diffusion-derived fake weeds performance for training Convolutional Neural Networks, Computers and Electronics in Agriculture **214**, 108324 (2023).
32. V. V. Moca, I. Ţincaş, L. Melloni, R. C. Mureşan, Visual exploration and object recognition by

lattice deformation, PLoS One **6**(7), e22831 (2011).

33. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658-666 (2019).
34. G. Van Rossum, An Introduction to Python (Network Theory Ltd, Bristol, 2003).
35. E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, M. A. Badhon, C. Pozniak, W. Guo, Global wheat head detection (GWHD) dataset: A large and diverse dataset of high-resolution RGB-labelled images to develop and benchmark wheat head detection methods, Plant Phenomics (2020).
36. Z. K. Hartley, A. P. French, Domain adaptation of synthetic images for wheat head detection, Plants **10**(12), 2633 (2021).
37. S. I. Amari, Backpropagation and stochastic gradient descent method, Neurocomputing 5, 185-196 (1993).
38. D. Ward, P. Moghadam, N. Hudson, Deep leaf segmentation using synthetic data, arXiv preprint arXiv:1807.10931 (2018).
39. C. Mitsanis, W. Hurst, B. Tekinerdogan, A 3D functional plant modelling framework for agricultural digital twins, Computers and Electronics in Agriculture 218, 108733 (2024).