



Deposited via The University of York.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/201750/>

Version: Accepted Version

Article:

Sykes, Jamie, Denby, Katherine and Franks, Daniel Wayne (2024) Computer vision for plant pathology: A review with examples from cocoa agriculture. Applications in Plant Sciences. e11559. ISSN: 2168-0450

<https://doi.org/10.1002/aps3.11559>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Computer vision for plant pathology: A review with examples from cocoa agriculture

Jamie R. Sykes^{a,1}, Katherine J. Denby^b, Daniel W. Franks^{a,c}

^a*Department of Computer Science, University of York, Deramore Lane, York, YO10
5GH, Yorkshire, UK*

^b*Centre for Novel Agricultural Products, Department of Biology, University of York, Wentworth
Way, York, YO10 5DD, Yorkshire, UK*

^c*Department of Biology, University of York, Wentworth Way, York, YO10 5DD, Yorkshire, UK*

ABSTRACT

Plant pathogens can decimate crops and render the local cultivation of a species unprofitable. In extreme cases this has caused famine and economic collapse. Timing is vital in treating crop diseases and use of computer vision for precise disease detection and timing of pesticide application is gaining popularity. Computer vision can reduce labour costs, prevent misdiagnosis of disease and prevent misapplication of pesticides. Pesticide misapplication is both financially costly and can exacerbate pesticide resistance and pollution. Here we review the application and development of computer vision and machine learning methods for detection of plant disease. This review goes beyond the scope of previous works to discuss important technical concepts and considerations when applying computer vision to plant pathology. We present new case studies on adapting standard computer vision methods and we review techniques for training data acquisition, use of diag-

Email addresses: jamie.sykes@york.ac.uk (Jamie R. Sykes),
katherine.denby@york.ac.uk (Katherine J. Denby), daniel.franks@york.ac.uk (Daniel W. Franks)

nostic tools from biology and inspection of informative features. In addition to in-depth discussion of convolutional neural networks and transformers, we also highlight the strengths of methods such as support vector machines and evolved neural networks. We discuss the benefits of carefully curating training data and situations where less computationally expensive techniques are advantageous. This includes a comparison of popular model architectures and a guide to their implementation.

Keywords: agronomy; disease detection; machine learning; plant pathology

Manuscript received -----; revision accepted -----.

1 INTRODUCTION

2 Computer vision (CV), typically powered by machine learning (ML), is now used
3 for a variety of tasks in agriculture, botany and ecology. These tasks include plant
4 health assessments (Patrício and Rieder, 2018), identification of weeds (Wu et
5 al., 2021), identification of drought prone areas of land (Ramos-Giraldo et al.,
6 2020), yield prediction (Sarkate et al., 2013) and detection of defects or bruising
7 in fruits and vegetables (Tripathi and Maktedar, 2020). We are seeing substan-
8 tial improvement in the efficiency of CV techniques (He et al., 2016; Howard et al.,
9 2017; Zhang et al., 2018) and, at least for now, computational resources continue
10 to become cheaper (Mack, 2011). As a result, CV is becoming available to whole
11 industries, not just areas of highest commercial value. For example, ML has been
12 used with increasing regularity for cocoa specific tasks such as the exploration and
13 optimisation of aroma profiles (Fuentes et al., 2019), monitoring of cocoa bean fer-
14 mentation (Parra et al., 2018; Oliveira et al., 2021) and bean quality classification

15 (Mite-Baidal et al., 2019). Large research and development budgets for areas like
16 wheat production have allowed for the use of unpiloted aerial vehicle photogra-
17 phy to identify disease outbreaks (Su et al., 2018; Chiu et al., 2020) and the use of
18 multispectral satellite photography to monitor outbreaks of yellow rust from space
19 (Nagarajan et al., 1984). Yet the application of ML to sectors with fewer financial
20 resources has had to take a different form. Onboard GPUs can run large neural
21 networks, analysing image data from farm machinery in real time locally and fast
22 internet connections can be used to run the same large models remotely (Grosch,
23 2018). However, implementation in poorer sectors must rely on older hardware,
24 edge devices and older model smartphones. This means that an emphasis must be
25 placed on ultra low cost implementation and high computational efficiency of al-
26 gorithms. This provides us with an opportunity and motivation to steer the ML
27 field away from brute force computing and towards more nuanced and efficient ap-
28 proaches.

29 The cultivation of cocoa, *Theobroma cacao*, represents a prime example of a sector
30 that could benefit greatly from non-intrusive and highly optimised CV disease de-
31 tection and will be used as an example throughout this review. The International
32 Cocoa Organisation estimates that up to 38% of the global cocoa crop is lost to
33 disease annually, with over 1.4 million tonnes of cocoa lost to just three diseases in
34 2016 (Maddison et al., 1995; Marelli et al., 2019). Additionally, international dis-
35 ease spread has been devastating to this industry in the past and could be again in
36 the future (Phillips-Mora and Wilkinson, 2007; Meinhardt et al., 2008). Following
37 the loss of a cocoa crop to witches' broom disease, a plot of land will typically be

38 cleared of forest and the previous robust agroforestry system will be replaced with
39 a monoculture (Rice and Greenberg, 2000; Meinhardt et al., 2008). This disease
40 is therefore not only capable of devastating the livelihoods of whole communities
41 of cocoa farmers, eliminating 50-90% of their crop (Meinhardt et al., 2008), but it
42 is also destructive to local biodiversity and has significant negative impact on the
43 carbon capture potential of the land (Kuok Ho and Yap, 2020). Such loss of ama-
44 zonian forest is a driver of climate change, causing positive feedback, exacerbating
45 this global crisis (Malhi et al., 2008).

46 A review from 1986 on the use of systemic fungicides to tackle oomycetes, like
47 *Phytophthora spp.*, highlights the concern about damage to the environment and
48 human health by pesticides such as methyl bromide, which are still in use (Cohen
49 and Coffey, 1986). These concerns and those of the pesticide resistance (Depart-
50 ment of Health. Victoria, 2023) are still present 37 years later. However, the use
51 of CV and ML for targeted application and calibration of pesticide dose are begin-
52 ning to have massive beneficial effects in this area across the agriculture industry.
53 It is estimated that from 2016 to 2026 smartphone use will have gone from approx-
54 imately 3.7 billion people to 7.5 billion (Statista, 2022). Therefore, the necessary
55 hardware to run CV models is largely in place and we need now only develop and
56 deploy the CV models to have great potential for impact with little monetary in-
57 put. Here we discuss how best to do that.

58 This review is composed of three main sections. Section one critically reviews a
59 wide variety of relevant techniques in ML and CV model development and test-
60 ing, and section two discusses techniques for data gathering, data labeling and

61 model testing. While section one focuses on ML theory and comparison of model
62 architectures, section two focuses on more practical issues. Finally, section three
63 discusses a brief roadmap to commercial implementation, which includes multiple
64 points that are important to consider prior to choosing an architecture and begin-
65 ning development.

66 There are several review articles published on the topic of computer vision and
67 deep learning that are applicable to plant pathology (Voulodimos et al., 2018; We-
68 instein, 2018; Chouhan et al., 2020; Xu et al., 2021). High quality works such as
69 Weinstein (2018), which reviews the use of CV in animal ecology, are directly ap-
70 plicable to plant pathology owing to the flexibility of the techniques discussed here.
71 What is missing from these works is a critical review and discussion of the latest
72 and/or less conventional techniques in CV and discussion of data acquisition and
73 validation. Each of the aforementioned reviews were published prior or near to the
74 release of DETR (Carion et al., 2020), VIT (Dosovitskiy et al., 2021) and Con-
75 vNeXT (Liu et al., 2022). So naturally these recent landmark methods are not
76 discussed. However despite all being published after the release of Faster-RCNN
77 (Ren et al., 2015), ResNet (He et al., 2016) and YOLO (Redmon et al., 2016), only
78 Xu et al. (2021) mentions any of these popular and high performing architectures.
79 Those being YOLO and region-based fully convolutional networks, an early prede-
80 cessor to Faster-RCNN.

81 A recent survey (Guo et al., 2022), goes into great detail on the various facets of
82 different attention mechanisms, which are integral to transformer architectures.

83 While this work presents the bleeding edge of CV technology, it does not present

84 the holistic, applied, and data-centric perspective provided here. Another paper
85 aimed to develop CV models for the classification of cocoa beans, comparing the
86 use of ResNet18, ResNet50 and SVMs (Lopes et al., 2022), while another recent
87 review gives a high level discussion of a number of CV studies in agriculture, cov-
88 ering topics of hyper-spectra imaging, use of unpiloted areal vehicles and architec-
89 tures as recent as ResNeXt (Xie et al., 2017; Tian et al., 2020). However, while
90 the latter of these two papers presents a broad view of CV for plant pathology,
91 providing strong links to many plant taxa, no mention is made by either Lopes et
92 al. (2022) or Tian et al. (2020) of architectures or techniques released after 2017.
93 As such, the fusion of industry standard and bleeding edge methods in data ac-
94 quisition, verification and analysis presented here make the present review unique
95 among those listed above.

96 This review provides the reader with an in-depth understanding of computer vision
97 for plant pathology and supports the previous aforementioned works. In doing so
98 we focus on how best to adapt current methods to provide practical solutions for
99 farmers, agronomists and botanists without access to high performance computa-
100 tional resources. While cocoa agriculture is used as a consistent example through-
101 out, all methods discussed here are applicable across plant pathology and agricul-
102 ture as well related fields such as plant and animal ecology and forestry.

103 **1. METHODS IN COMPUTER VISION**

104 **1.1. Background**

105 Ever since AlexNet was presented at NeurIPS in 2012, the field of computer vi-
106 sion has been dominated by convolutional neural networks (CNNs) (Krizhevsky
107 et al., 2017). While subsequent updates to CNN architectures have provided dra-
108 matic improvements over AlexNet (Liu et al., 2022), it is important to recognise
109 that CNNs are not the only tools at our disposal. Previous work on cocoa disease
110 has assessed the performance of support vector machines (SVM), random forest re-
111 gression and artificial neural networks to identify common diseases in cocoa from
112 standard colour images, hereafter referred to as RGB (Red, Green, Blue) images
113 (Rodriguez et al., 2021). Here it was shown that artificial neural networks are
114 capable of identifying late stage disease in RGB images of cocoa but that train-
115 ing data set size is a limiting factor. Another study applied a SVM to perform
116 pixel-wise identification of black pod rot in cocoa (Tan et al., 2018). The result-
117 ing algorithm showed an impressive ability to detect human visible disease symp-
118 toms and, given the high computational efficiency of SVMs, it was able to run on
119 low-powered hardware. Additionally, this model was trained on only 50 images,
120 which is an extremely small training set in CV. However, no mention is made of
121 the ability of these models to detect early disease development or non-human visi-
122 ble symptoms, which will be a central focus of this review.

123 **1.2. Vision transformers**

124 In the early 2010's transformers become the default for natural language process-
125 ing (Liu et al., 2022) and they are now rapidly gaining popularity in vision based

126 tasks. Pure transformer based multilayer perceptrons, such as Vision Transformer
127 (Dosovitskiy et al., 2021), do away with the convolutional layers of a CNN. Instead
128 they subdivide and tokenise an image before passing this data to the fully con-
129 nected layers of a network. The main drawbacks of such transformer based models
130 are that they require training datasets on the order of millions of images and they
131 lack the inductive biases of CNNs, like translational equivariance (Dosovitskiy et
132 al., 2021). In addition, the global structure of objects in an image must be learned
133 from scratch, whereas this is maintained throughout a CNN. However, when pre-
134 trained on a large data set and then fine-tuned on a *more modest* dataset of tens
135 of thousands of images, vision transformers can out-compete CNNs (Dosovitskiy
136 et al., 2021).

137 Although the requirement for vast training datasets may preclude the use of trans-
138 formers for many plant pathology projects, there is a middle ground between the
139 popular ResNet architectures and transformer models. Taking inspiration from
140 transformer designs, the highly competitive ResNet architectures have been up-
141 dated to produce a pure CNN that competes well with transformers in many tasks
142 and is reported to outperform the original ResNets by about 3% accuracy on Im-
143 ageNet (Liu et al., 2022). This family of four models is named ConvNeXt and
144 includes models of varying complexity from ConvNeXt tiny to ConvNeXt large.
145 Additionally, ConvNeXt uses layer normalisation in place of batch normalisation.
146 This modification could have important benefits for plant pathology projects, as
147 discussed in section 1.7. However, as the ConvNeXt architectures are relatively
148 large in size (ConvNeXt-tiny: 29 million parameters, ResNet18: 12 million parame-

149 ters, ResNet50: 26 million parameters), these models too require large and/or com-
150 plex training data sets to avoid overfitting and more powerful hardware to run at
151 inference than the smaller ResNets.

152 **1.3. Object detection and semantic segmentation**

153 Bounding box object detection and semantic segmentation are processes by which
154 objects of interest in an image are both classified and located in the image. In
155 these tasks either a box (bounding box object detection) or a polygon or 'mask'
156 (semantic segmentation) is drawn around the object of interest. For an example of
157 semantic segmentation, see Case Study One below.

158 Semantic segmentation and object detection could help in the accurate manual la-
159 belling of disease states in images. In simple image classification with a CNN, a
160 model must learn what features, across the whole image, can be used as true mark-
161 ers of disease. However, annotation of training images with bounding boxes or seg-
162 mentation masks may be used to focus the attention of the model, thus making
163 training more efficient. This beneficial effect might be more pronounced with se-
164 mantic segmentation than bounding boxes because the edges of a bounding box
165 may extend beyond the edges of the leaf, pod or tree in question and thus misla-
166 bel parts of neighbouring healthy plants. However, when comparing the ability of
167 Faster R-CNN and Mask R-CNN to detect human visible signs of insect damage in
168 sweet peppers, Faster R-CNN was shown to have superior accuracy and mean av-
169 erage precision (mAP) (Lin et al., 2020). Here mAP is defined as the mean preci-
170 sion over all classes of the mean per class precision, with a given Intersection Over
171 Union. These disparities in performance were contingent on which backbone model

172 architecture (Inception v2, ResNet50 or ResNet101) was used. When the more
173 complex ResNet101 was used, Faster R-CNN and Mask R-CNN performed more
174 similarly, although, in this task, Faster R-CNN performed best with the simpler ar-
175 chitectures (Lin et al., 2020). Though it should be noted that average precision is
176 not directly comparable between bounding box detection and semantic segmenta-
177 tion models. This is for two reasons: 1) It is easier to achieve a given intersection
178 over union with a bounding box as this task is less precise than segmentation, and
179 2) Mask R-CNN simply adds the ability to predict a mask in a box predicted by
180 Faster R-CNN, so segmentation is additive in this case. As such the results of Lin
181 et al. (2020) should be considered accordingly.

182 Object detection and semantic segmentation are typically performed using either
183 Faster R-CNN (Ren et al., 2015), Mask R-CNN (He et al., 2017) or YOLO (Red-
184 mon et al., 2016). However, these architectures have also been combined with other
185 methods, such as SVMs, to confirm or deny the presence of an object in a pro-
186 posed region (Voulodimos et al., 2018). For example, SVMs have been used in con-
187 junction with Mask R-CNN in automated ML pipelines to identify defects in ma-
188 chined parts (Huang et al., 2019). Additionally, when facing a classification prob-
189 lem with high intraclass variance, low interclass variance and insufficient training
190 examples, the application of SVMs to features learned by a CNN from Imagenet
191 can improve results relative to a CNN alone (Cao and Nevatia, 2016). This may
192 prove useful in projects with few training images or when classifying images of
193 plant disease with similar characteristics such as black pod rot in cocoa caused
194 by *Phytophthora megakarya* or *Phytophthora palmivora*. Furthermore, while *P.*

195 *megakarya* and *P. palmivora* can be distinguished by eye, *Lasiodiplodia* species,
196 of which three are known to infect *T. cacao*, can present with identical morpho-
197 logical characteristics. This means that traditional classification techniques are
198 insufficient and molecular identification techniques must be used in their place
199 (Huda-Shakirah *et al.*, 2022). CV technologies that can make such difficult dis-
200 tinctions would have important implications for all areas of agriculture and botany
201 for two reasons; 1) While *Phytophthora megakarya* and *Phytophthora palmivora*
202 are managed in the same way, different species of *Lasiodiplodia* are not (Khanzada
203 *et al.*, 2005). Thus, failure of a model to distinguish between species of *Phytoph-*
204 *thora* is not critical for effective disease management, but failure to distinguish be-
205 tween species of *Lasiodiplodia* is. 2) Cosmopolitan pathogens such as *Phytophthora*
206 *.spp* and *Lasiodiplodia spp.* have extremely wide host ranges, infecting many com-
207 mercially important crops. *Lasiodiplodia theorumae* alone attacks over 189 plant
208 species across 60 families (Salvatore *et al.*, 2020), while the growing list of *Phy-*
209 *tophthora* (aka "plant destroyer") species described is currently 116 entries long
210 (Kroon *et al.*, 2012).

211 Transformer-based object detection models such as Detection Transformer (DETR)
212 (Carion *et al.*, 2020) are also now available and contend well with Faster R-CNN
213 when trained on the huge COCO dataset. The key benefit of DETR is that it
214 predicts bounding box coordinates directly, negating the need for the region pro-
215 posal network of Faster R-CNN. Faster R-CNN's region proposal network has is-
216 sues trying to identify overlapping objects because of the non-max suppression al-
217 gorithm, which was removed from YOLO in version 3 (Horzyk and Ergün, 2020).

218 However, DETR has problems detecting small objects, and has a very long conver-
219 gence time. These defects are said to be resolved in Deformable DETR (Zhu et al.,
220 2021), though we encountered significant difficulty in retraining Deformable-DETR
221 due to prevailing bugs in the code and so were unable to confirm these benefits.
222 In segmenting instances of nuclei in microscopy images, Mask R-CNN was com-
223 pared with the U-Net architecture, which was designed for medical image segmen-
224 tation. Here the two techniques were shown to give similar mAP, F1 and recall
225 scores (Vuola et al., 2019). However, Mask R-CNN scored 0.812 for precision, while
226 the U-NET scored only 0.68. A subsequent ensemble approach was then described,
227 which shares the outputs of the two independently trained architectures to ex-
228 ploit the U-Net’s purportedly superior F1 scores (+0.057), in tandem with Mask
229 R-CNN’s high mAP, precision and recall. The ensemble model produced compara-
230 ble, if slightly higher, mAP (+0.016), F1 (+0.056) and recall (+0.037) compared
231 to Mask R-CNN, but the precision was 0.087 lower. Although the U-Net was re-
232 ported to produce the best F1 score and the Ensemble model produced the best
233 mAP and recall, these improvements were slight. Additionally, F1 is calculated di-
234 rectly from precision and recall so it seems counterintuitive that the U-net could
235 have the highest F1, yet lowest precision and recall. The most noteworthy result
236 here is the consistently superior precision of Mask R-CNN in this comparison and
237 in another against YOLO (Bharati and Pramanik, 2020; Horzyk and Ergün, 2020).
238 Additionally, in a study comparing the use of U-Net and Mask R-CNN to segment
239 images of pomegranate trees, Mask R-CNN outperformed the U-Net in both preci-
240 sion and recall by wide margins (Zhao et al., 2018).

241 An alternative approach applied an SVM to perform pixel-wise classification to
242 detect black pod rot in cocoa and used a human expert to label diseased pixels
243 in training images (Tan et al., 2018). Like semantic segmentation, this technique
244 achieves the effect of providing the model with additional information on the loca-
245 tion of disease in an image, relative to a simple CNN. However it imposes arbitrary
246 physical boundaries around disease symptoms such as lesions and cankers and
247 the algorithm is unable to define for itself any symptoms that aren't or can't be
248 identified with human vision. By using semantic segmentation with a CNN back-
249 bone, like in Mask R-CNN or DETR, to segment whole trees, these effects could
250 be avoided. *i.e.* the model would be able to detect non-human visible symptoms
251 via feature learning and model the effects of hyphae propagating through the plant
252 or systemic changes to a plant's phenotype away from the site of infection.

Case study one: Semantic segmentation for cocoa disease detection

253 In this case study we applied Mask R-CNN to the task of segmenting images
254 of diseased cocoa trees. The training dataset consisted of 186 images of black
255 pod rot (BPR), 121 images of frosty pod rot (FPR) and 63 images of witches
256 broom disease (WBD). The model was trained, starting with the "mask rcnn R
257 50 FPN 3x" weights, for 1,000 epochs.

258 The preliminary results from this case study were somewhat encouraging. How-
259 ever, although the selected positive results in figure one show that this model
260 has the potential to perform well, these results are not representative of the
261 full testing set. The average precision per class was 4.29, 13.45 and 30 for BPR,
262 FPR and WBD respectively. *i.e.* the model performed acceptably on WBD, de-
263 spite the low number of training images, but poorly on most cases of BPR and
264 FPR.

265 Notwithstanding the potential theoretical benefits discussed above, manual an-
266 notation of a full training dataset with masks is extremely laborious. So without
267 the promise of improved results, relative to a simple CNN, this additional ef-
268 fort may not pay. However, considering the favourable preliminary results in this
269 study and one other (Zhao et al., 2018), with the incorporation of automated
270 annotation tools and/or semi-supervised learning, semantic segmentation shows
271 promise as an avenue of research for CV in plant pathology.

272 **1.4. Variational autoencoders for outlier detection**

273 In addition to discriminative modelling, ML provides several powerful tools for
274 generative modelling. Modelling with generative deep neural networks (DNNs)
275 can aid in gaining an intuitive understanding of the physical laws that led to the
276 creation of the data to be modelled. An example of this is the use of artistic style
277 transfer with generative adversarial networks (Li and Wand, 2016), where specific
278 semantic features in an image can be isolated and utilised. Another popular deep
279 generative model architecture is the variational autoencoder (VAE), which we will

280 focus on here for the task of image dataset filtering.

281 When working with autonomously collected data, for example from camera traps
282 or webscraping bots, the acquisition of vast quantities of data is often the easy
283 part of creating a good training data set. Camera traps tend to produce a lot of
284 uninformative data and the data from naive webscraping bots can be badly con-
285 taminated with miss-classified and irrelevant images. For example, a search for the
286 keyword "Acer" will return many more images of laptops than it will Japanese
287 maple trees and a search for "black pod rot" will include many images frosty pod
288 rot, cherville wilt and insect damage. Therefore some level of human supervision is
289 vital in curating training data and the importance of consulting farmers and re-
290 searchers in data collection and labeling cannot be overstated. However, manual
291 labeling of a full dataset can be extremely costly and a potential method to offset
292 some of this cost is said to be the use of VAEs for outlier detection.

293 A VAE is composed of two neural networks which are trained in parallel. The en-
294 coder network projects the image data to a smaller latent vector space, thus com-
295 pressing it, and the decoder network predicts the original image from this com-
296 pressed data as best it can.

297 Generative models tend to generalise to the real world much better than discrim-
298 inative models, which aim to uncover correlative relationships between data and
299 class labels (Kingma and Welling, 2019). However, deep generative models are typ-
300 ically considered excessive for classification problems, they often have higher bias
301 (Banerjee, 2007) and are computationally expensive.

302 VAEs have been used successfully for text classification (Xu et al., 2017; Xu and

303 Tan, 2020), data clustering (Dilokthanakul et al., 2017; Lim et al., 2020), anomaly
304 detection (An and Cho, 2015), recommender systems (Li and She, 2017), dimen-
305 sionality reduction (Lin et al., 2020) and there are published papers on the use
306 of VAEs for anomaly detection with colour images (Fan et al., 2020), though not
307 many.

308 Here we consider two methods by which a VAE might be used to detect outly-
309 ing data in collections of large colour images. To do so we will use the example
310 of detecting non-plant images in a webscraped collection of plant images for use in
311 building a disease classifier.

312 **Method 1, Distribution of reconstruction loss:** Having trained a VAE on
313 only plant images, use this model to compress and decompress all images in
314 the contaminated dataset and record the reconstruction loss for each image.
315 Plot the distribution of the loss values and record the most extreme high val-
316 ues as outliers. The assumption here is that the model should "fail" to recon-
317 struct non-plant images well as it should be naive to any images that do not
318 show plants.

319 **Method 2, Dimension reduction and clustering:** Using the encoder net-
320 work of a VAE that has been trained on the ImageNet dataset, compress
321 the images in the contaminated dataset and record the values of the latent
322 space for each image. Reduce the dimensions of the latent space further with
323 principal component analysis, t-SNE and/or UMAP. Plot this reduced data.
324 Outliers/contaminant images may then separate from the clean data.

325 Nouveau VAE (NVAE) is the product of an effort to carefully craft the encoding
326 network architecture of a VAE and appears to produce excellent results (Vahdat
327 and Kautz, 2021). After training for just one epoch, this architecture is able to
328 project large colour images onto a latent space and reconstruct them almost per-
329 fectly. However, if the aim of using NVAE is to compress image data, this archi-
330 tecture is not appropriate. This is because, using the recommended settings for the
331 CelebA 64 data set (Liu et al., 2015), the latent space produced for an image with
332 dimensions (3,224,224) is (100,224,224), *i.e.* more than 33 times larger than the
333 original image. Following the authors' provided instructions to constrain the latent
334 space to be as small as possible without excessively modifying the code, the latent
335 space for this same size of image remains the same (100,224,224). This observation
336 is corroborated in another study where the authors explain how NVAE first ex-
337 pands the data dimensions to a large number of latent spaces before pruning those
338 spaces based on KL divergence (Asperti et al., 2021). However, these authors go
339 on to note that, in their use case, NVAE transformed images of size (3,32,32) to a
340 latent space of size (16,16,128) without any subsequent downscaling. It is not sur-
341 prising then that this architecture is able to reconstruct an image so well after just
342 one training epoch, with no pre-trained weights, as the dimensionality of the data
343 is expanded, not compressed. Likewise, NVAE is not appropriate for identifying
344 outliers by the distribution of reconstruction errors as it can reconstruct any image
345 almost perfectly. For example, when we trained NVAE on a dataset of 54,124 plant
346 images, it was able to reconstruct any image in the ImageNet dataset with similar
347 binary cross-entropy loss to that of plant images.

348 As an alternative to NVAE, we attempted to use a custom convolutional VAE with
349 a ResNet-152 (He et al., 2016) backbone to apply the two methods of outlier detec-
350 tion described above. However, we were unable to get this architecture to function
351 well enough to sufficiently compress the data and reconstruct images with high fi-
352 delity.

Case study two: Semi-supervised learning for outlier detection

353 As an alternative to using a variational autoencoder for outlier detection, we
354 trained a semi-supervised binary Outlier-NoneOutlier (in this case, "plant"
355 or "non-plant") classifier, which achieved near perfect results. We used the
356 ResNet18 architecture and initially trained it on a manually curated dataset of
357 57,228 plant images and an equal sized random subset of the ImageNet dataset,
358 which constituted the non-plant images. We then continued training using the
359 below algorithm and the contaminated dataset of 96,692 images.

```
360   while  $nRelabledImages > 0$  do  
361     train model  
362     for image in ContaminatedImages do  
363       classify image  
364       if  $ClassificationConfidence \geq 99\%$  then  
365         label image  
366         add image to training set  
367       end if  
368     end for  
369   end while
```

370 During this process 1,376 none-plant images and 44,212 plant images from the
371 contaminated dataset were correctly labeled by the model. After the first round
372 of semi-supervised training completed, images that this model classified with
373 >99% confidence were manually reviewed. Incorrectly labeled images were man-
374 ually re-labeled and a second round of semi-supervised training was begun. Af-
375 ter the first round of semi-supervised training, classification of images as "plant"
376 with >99% confidence was >99% accurate but classification of images as "non-
377 plant" with >99% confidence was only about 50% accurate. After the second
378 round of semi-supervised training, the model performed with >99% accuracy
379 and F1 score for both classes. Thus showing a clear superiority in this tech-
380 nique's ability to identify contaminant images over the VAE approaches. This
381 is in addition to its ease of implementation, and reduced training time and com-
382 pute requirements. After training, the model was used to classify all 96,692 im-
383 ages in the contaminated dataset.

384 The paucity of papers published on the subject of outlier detection in colour im-
385 ages with VAEs seems to be due to the inherent difficulty of this task. The high
386 dimensions of such data and the large storage and GPU memory requirements that
387 training these models on such data necessitates (Sun et al., 2018) has largely been
388 resolved, though for many GPU memory availability will still preclude this tech-
389 nique. Thus far the inability of the VAE architecture to learn a compression algo-
390 rithm for large colour images suggests a hard physical limitation that might not be
391 overcome. Moreover, while Maalø et al. (2019) contest this argument, Nalisnick et
392 al. (2019) argue comprehensively that generative models are not suitable for outlier

393 detection by the reconstruction loss method described above as these models tend
394 to learn low-level statistics about data rather than high-level semantics. As such
395 they are often unable to differentiate between images that, to the human eye, are
396 obviously different.

397 **1.5. Evolutionary algorithms**

398 The field of CV is currently dominated by handcrafted DNNs with fixed topolo-
399 gies. However, the seldom used techniques of evolved neural networks have real
400 potential in the field of plant pathology. Computational efficiency at inference and
401 improved ability to generalise is of paramount importance to models developed
402 for plant pathology in the field. This is because such models must be able to cope
403 with complex and highly variable symptoms and backgrounds, and often must run
404 on low-powered hardware. Growing neural networks take far longer to train/grow
405 than those with fixed topologies but this is of minor concern given efficient par-
406 allelisation and the vast computation resources now available for training. The
407 hardware available to farmers in low income sectors like cocoa, cassava or coffee,
408 however, is restricting. This restriction means that producing a model that is opti-
409 mised for runtime speed at inference is a vital factor and growing neural networks
410 with evolutionary algorithms may be an ideal way to achieve this.

411 Evolving neural networks has been shown to be highly effective in producing neu-
412 ral networks with a high degree of modularity (Amer and Maul, 2019). This in-
413 creased modularity is said to be the result of applying a cost to the number of
414 connections, which both reduces computational cost and promotes evolvability as
415 sharing of modular units between parents is made simpler. It is also said that such

416 modularity helps these models to generalise better as each modular unit is capa-
417 ble of independent generalisation (Schmidt and Bandar, 2001). With evolutionary
418 algorithms, one can also promote diverse populations of networks with techniques
419 like niching (Shir, 2012) and use of non-elitism strategies can allow for the simul-
420 taneous exploration of fitness valleys and local optima without getting stuck there
421 (Dang et al., 2021). While elitism follows the biologically implausible assumption
422 that the fittest individual/network will always survive to reproduce, non-elitism
423 allows weaker individuals to explore fitness valleys, which may lead them to undis-
424 covered maxima.

425 While a direct comparison of evolved neural networks with popular CNN archi-
426 tectures could not be found, table 1 shows an indirect comparison between a re-
427 cent method for evolving neural networks (EVOCNN) and two popular CNNs
428 (ResNet18 and VGG16). EVOCNN appears to perform very well in this compar-
429 ison. However, the error rate for these models was calculated when trained on the
430 Fashion MNIST dataset, while the top 1 and top 5 accuracy was produced using
431 ImageNet. Fashion MNIST, which is composed of 28x28 pixel grey scale images
432 of clothing (Xiao et al., 2017), is not a challenging proposition for modern CNNs
433 and is not reflective of real world plant pathology problems. Additionally, it should
434 be noted that in the EVOCNN paper (Sun et al., 2020), the number of parame-
435 ters of VGG16 is miss-reported as 26 million, rather than 138M (*Torchvision Main*
436 *Documentation* 2023). This suggeststs that VGG16 would have massively overfit
437 to the fashion MNIST data, making this an inappropriate comparison. However
438 EVOCNN does offer a very low error rate on this more simple problem and with

439 a very low number of parameter when compared with other modern architectures
440 (Table 1 & table 2). However, it does not seem that evolved neural networks are
441 not yet ready to tackle the more difficult problems in plant pathology and so more
442 work is required in this area.

443 **1.6. Architecture comparison and recommendations**

444 The field of CV has produced a numerous and diverse set of architectures, each
445 with unique strengths and weaknesses. Here we will compare these architectures,
446 focusing on their application in image classification, object detection, and seman-
447 tic segmentation. Table 2 gives a detailed breakdown of the pros and cons of each
448 of these architectures as well a the number of trainable parameters, which acts as
449 a proxy for model complexity, and the number of giga floating point operations
450 (GFLOPS), which gives a sense of computation cost of running inference with
451 these architectures.

452 *1.6.1. Image Classification Architectures*

453 ResNet introduced the concept of skip connections, enabling the training of much
454 deeper models. Despite its age, ResNet remains a strong competitor and ResNet18
455 is probably still the best choice for most small projects with fewer training exam-
456 ples. EfficientNetV2 is more computationally demanding than ResNet and Con-
457 vNeXT and, while it tends to yield high accuracy on large datasets (Dosovitskiy et
458 al., 2021; Liu et al., 2022), we found that it is prone to overfitting, making it a less
459 favorable choice. The key innovation of EfficientNet was to allow the depth, width
460 and resolution of the model to be scaled by adjusting a single coefficient (Tan and

461 Le, 2020). However, in practice this requires editing the source code, thus render-
462 ing such adjustments less than convenient. ConvNeXT is an updated version of
463 ResNet, incorporating several modern features. Unlike EfffientNet, ConvNeXT is
464 easy to scale, making it a promising choice for medium to large-scale applications
465 for which it has been shown to give superior performance to ResNet and VIT (Liu
466 et al., 2022). As the first transformer to perform favorably against CNNs for image
467 classification, VIT represents a significant milestone. However, image classifica-
468 tion may not be the optimal use case for transformer architectures and at present
469 ConvNext outperforms VIT while requiring less data to train on and being less
470 computationally expensive (Dosovitskiy et al., 2021).

471 *1.6.2. Object Detection and Semantic Segmentation Architectures*

472 Although more complex than YOLO, and arguably DETR, Faster-RCNN delivers
473 excellent results and requires only modest resources for training. For most object
474 detection use cases in plant pathology, Faster-RCNN will be the optimal choice.
475 Mask-RCNN extends Faster-RCNN by adding the ability to predict a mask in
476 a bounding box, enhancing its utility for semantic segmentation tasks. YOLO is
477 most suitable for real-time object detection and offers lower precision than Faster-
478 RCNN. It is not suitable for use in plant pathology unless inference time is of pri-
479 mary concern. DETR and Deformable-DETR present a novel approach to object
480 detection and offer competitive results (Zhu et al., 2021). However, implementing
481 these architectures can be difficult and they require substantial GPU VRAM for
482 training.

483 The choice of CV model architecture for a given project depends on a variety of

484 factors including dataset size, signal to noise ratio, computational resources, mode
485 of deployment and accuracy requirements. However, at present, for most use cases
486 in plant pathology, ResNet18, ConvNeXT_tiny or Faster-RCNN will yield the best
487 results while minimising computational cost, risk of overfitting and financial cost of
488 training.

489 **1.7. Image, batch and layer normalisation**

490 In a comparison of a EVAL-COVID (Gong et al., 2021) with other strong com-
491 petitors like EVOCNN to the detect COVID-19 with evolved CNNs, it was show
492 that the overuse of batch normalisation (BN) can be deleterious to the training of
493 DNNs for disease diagnosis. While BN often improves the training time of CNNs
494 and can negate the need for small learning rates and dropout (Ioffe and Szegedy,
495 2015), its negative effect on the diagnosis of disease was also observed in case study
496 three, below.

Case study three: Disease detection and normalisation

497 Here we conducted an ablation study with ResNet18 and ConvNeXt_tiny (Ta-
498 ble 3) to assess the effects of image normalisation (IN), batch normalisation (BN)
499 and layer normalisation (LN) in disease detection. BN in ResNet18 increased
500 training speed by 2.39 times, while IN slowed training by 1.74 times. IN did not
501 affect training time in ConvNeXt_tiny. We also found that BN improved stabil-
502 ity in training, as assessed by plots of training and validation loss. However, IN
503 decreased the F1 score by 0.76% and 0.34% in ConvNeXt and ResNet18 respec-
504 tively, and increased overfitting. Removal of BN in ResNet18 decreased F1 by
505 1.92% but the ConvNeXt model (in which BN is replaced with LN) had an F1
506 score 2.84% higher than ResNet18 with BN. Therefore simply deactivating the
507 BN layers in ResNet18 led to worse results in every metric. However the use of
508 LN instead of BN in ConvNeXt appears to have had no deleterious effect. The
509 removal of the IN transformation, which occurs prior to data input, improved
510 the performance of both model architectures for the purpose of disease detection
511 in all metrics, including training time and overfitting.

512 Several state-of-the-art generative models now omit BN entirely, while others re-
513 place it with weight normalisation or focus on fine tuning the momentum hyper-
514 parameter of BN layers (Vahdat and Kautz, 2021). As with simply removing the
515 BN layers of a ResNet, reported above, replacing BN in ResNet with the alterna-
516 tive layer normalisation (LN) also results in worse performance (Wu and He, 2018).
517 However, when the authors of ConvNeXt use LN as opposed to BN in their archi-
518 tecture, they observe that the model has no difficulty in training with this substi-

519 tution (Liu et al., 2022). The BN momentum hyperparameter is a fixed weight ap-
520 plied to the running mean and variance calculations that are tracked during train-
521 ing and used during the application of BN at evaluation or inference time. Thus,
522 adjusting the BN momentum will not affect effect training (Vahdat and Kautz,
523 2021). However, BN can cause the output of a layer to be slightly shifted during
524 evaluation and a supposed solution to this is to adjust the momentum hyperpa-
525 rameter (Vahdat and Kautz, 2021).

Case study four: Optimisation of BN momentum and image size for cocoa disease detection

526 While training a cocoa disease detection model we ran a hyperparameter optimi-
527 sation sweep using the Weights and Biases platform (WANDB) (Biewald, 2023),
528 which included the BN momentum hyperparameter and image input size (Fig.
529 3). The model architecture used was ResNet18 (He et al., 2016) and the dataset
530 included the following four classes: black pod rot, frosty pod rot, healthy cocoa
531 and witches broom disease with a 90:10 split and training set size of $n = 271$,
532 266, 436 and 92 respectively. 100 models were trained with these hyperparam-
533 eters randomly sampled from predefined ranges (Image size: 124:1224 pixels,
534 BN mom.: 0, 10^{-5} :0.9). We also used WANDB to run a random forest regression
535 with the validation F1 as the dependent variable and the two hyperparameters
536 as independent variables. From this an importance score was calculated for each
537 hyperparameter on a scale of 0-1. The highest performing model scored vali-
538 dation F1:0.75 and AUC:0.87. Additionally the per class F1 score for healthy
539 cocoa was 0.88, showing a strong ability to detect non-specific disease.

540 While the importance of image size (0.694) is not surprising, the BN momentum
541 score (0.306) is quite low. This casts doubt on the assertion above that opti-
542 misation of BN momentum can have much impact in lessening the deleterious
543 effects of BN. However, this result and that of the optimised BN momentum
544 value (0.001) (Fig. 3 A), suggests that this hyperparameter should be optimised,
545 rather than relying on the default value of 0.1. Training the same model with
546 a BN momentum set at 0.1 yielded an F1 score of 0.737. *i.e.* a 1.3% decrease
547 relative to the optimised value.

548 This study also provides an optimised image input size for mid- to late-stage
549 disease detection, using ResNet18, of 277 pixels² (Fig. 3 B), though this should
550 be optimised for each use case. Previously, image compression has been said to
551 have minor effects on disease detection (Barbedo *et al.*, 2016), while elsewhere
552 it is suggested that image compression should even be avoided completely for
553 small symptoms (Barbedo, 2016) or kept above an arbitrary 1 megapixels (1,000
554 x 1,000 pixels) (Steddom *et al.*, 2005). However, with the present dataset, which
555 contains images of diseases at varying degrees of progression, using a image size
556 greater than 277x277 was deleterious to validation F1 score. This is in addition
557 to the reduced image size providing faster runtime in training and inference and
558 a reduction in overfitting.

559 Above we have listed a host of reasons why unnecessary normalisation of data is to
560 be avoided. While BN will shorten training time for a CNN, it changes the input
561 data in unpredictable ways, thus worsening prediction results. However, at present
562 the best off-the-shelf CNN that is small enough to run on an older model smart-

563 phone is ResNet18. So until a more suitable architecture becomes available, BN is
564 unavoidable. We have shown that optimisation of the BN momentum hyperparam-
565 eter in ResNet18 lead to a slight improvement in the results of our cocoa disease
566 detection model, that IN should not be included in the training pipeline of a model
567 that aims to make predictions from subtle colour features and that excessive image
568 input size should be avoided.

569 **2. DATA ACQUISITION AND MODEL TESTING**

570 In this section we review various interdisciplinary methods available for gathering
571 a training dataset and developing a suitable model. While the previous section was
572 concerned with the theory of ML in CV, this section will focus on practicalities
573 with respect to low cost solutions.

574 **2.1. Obtaining the required training dataset**

575 Training an image classifier to a high accuracy in a controlled laboratory environ-
576 ment is often a trivial task. However such a model may perform poorly when pre-
577 sented with the challenges of the real world (Singh et al., 2020). For example, it
578 was found that after training a leaf-disease classifier on images taken in the field,
579 the model performed with around 68% accuracy when tested against images taken
580 in the lab (Ferentinos, 2018). However, when trained in the lab and tested in the
581 field, the same model architecture performed with about 33% accuracy. This effect
582 is likely due to the plain white background of the lab images causing the model
583 to generalise poorly to real world application. This exemplifies the importance of

584 curating a realistic, high quality training dataset. By naively training and releas-
585 ing models that are trained on publicly available datasets, we risk exacerbating the
586 problems of disease miss-classification. At low frequencies, the effect of mislabeled,
587 misleading or uninformative data will have limited effect on the performance of a
588 neural network. This feature of neural networks is largely an artefact of batch gra-
589 dient descent and the learning rate (Motamedi et al., 2021), which act to greatly
590 buffer the effect of infrequent miss-classifications in the training data. However, at
591 higher frequencies, these sources of error can have more serious consequences. The
592 most obvious solution to this problem is to carefully curate, label and annotate the
593 training data. However, error resulting from misclassification can be challenging to
594 eradicate. For example, frosty pod rot (FPR), black pod rot (BPR) and witches'
595 broom disease (WBD) in cocoa can all present with black or brown lesions on the
596 pod and both FPR and BPR can both coat a pod in white mycelium. This means
597 that without sufficient training in plant pathology or access to diagnostic tests, one
598 could easily mislabel these diseases. This problem can be solved by two means,
599 which should be used in tandem: 1) With careful attention to detail and a detailed
600 knowledge of the pathogen in question, and 2) Using tools and techniques from
601 molecular biology and spectroscopy to better inform model development and sub-
602 sequent disease detection. Such techniques/tools include DNA sequencing, qPCR,
603 LAMP, MultispeQ and hyperspectral imaging.

604 *2.1.1. Tools from molecular biology*

605 DNA sequencing for the identification of cryptic species (Bickford et al., 2007;
606 Ovaskainen et al., 2010) and plant pathogens (O'Donnell et al., 2015) is now a

607 common place and invaluable tool. Once sequenced, reads can be used to search
608 previously categorized sequences with the Basic Local Alignment Search Tool (BLAST)
609 from NCBI (Boratyn et al., 2013) to identify a sample by species or other tax-
610 onomic group. However, if we know which pathogen we aim to detect, sequenc-
611 ing the whole genome is excessive. Rather, we can use loci like the internal tran-
612 scribed spacer (ITS) region of the nuclear ribosomal RNA genes, which are both
613 highly conserved across taxa and highly variable between species. Such regions of
614 the genome can be utilised with amplification techniques like polymerase chain
615 reaction (PCR) or Loop-mediated isothermal amplification (LAMP) to detect a
616 pathogen or identify it with relatively low cost and high accuracy. ITS is often
617 used on its own for near-species level identification or in concert with other loci for
618 better specificity (Horton and Bruns, 2001). Such work with ITS is now ubiquitous
619 in the molecular study of fungal ecology and phylogeny, while previous techniques
620 relied on the morphology of fruiting bodies for identification (Horton and Bruns,
621 2001).

622 Quantitative polymerase chain reaction (qPCR) is used to detect asymptomatic
623 disease across the agricultural industry (Luchi et al., 2020). Traditionally PCR
624 has been unsuitable for portable operations or use in the field (Ray et al., 2017).
625 However, rapid real-time PCR in the field is now possible (Schaad and Frederick,
626 2002). Real-time PCR can also be used to quantify relative levels of a pathogen in
627 plants (Horevaj et al., 2011). Information from such analyses could be extremely
628 informative when fine tuning and assessing the performance of the models dis-
629 cussed here.

630 LAMP can be used in place of qPCR and has four key benefits. 1) It is consider-
631 ably cheaper (£211 for 100 samples) because a thermo cycler is not required, 2)
632 It is fast, 3) Reagents don't need to be refrigerated, and 4) Like Real-time PCR,
633 there is potential for it to be used in the field. Like qPCR, LAMP can be used to
634 quantify the relative amount of DNA present as well as simply for detection. If
635 detection is the only goal, colour- or turbidity-based methods can be used to de-
636 tect DNA presence by visual inspection. A drawback of the use of this method is
637 that any pre-existing PCR primers cannot be used. This is because PCR primers
638 are designed to amplify a specific region of DNA by binding to complementary
639 sequences on opposite strands of the target DNA. LAMP primers, on the other
640 hand, are designed to bind to multiple regions of the target DNA in a way that
641 allows for the simultaneous amplification of multiple regions of the DNA.

642 While universal PCR primers for the ITS region exist, it may be necessary to de-
643 sign LAMP primers or species specific PCR primers for ITS or other regions. For
644 a detail discussion on the use of ITS amplification in fungal ecology and the poten-
645 tial pitfalls of specific ITS primers design, see Horton and Bruns (2001).

646 If novel primers are to be designed, the region of interest must first be sequenced
647 and if we aim to identify an as yet unknown pathogen with BLAST, all of the
648 DNA in a sample must be sequenced. Sequencing with the Oxford Nanopore Tech-
649 nology MinION platform can be a an ideal tool for this purpose, offering multi-
650 ple features: 1) With Oxford Nanopore Technology's field sequencing and library
651 preparation kit, this method allows for sequencing in the field, immediately af-
652 ter tissue samples are gathered. This eliminates the need for cold-chain storage to

653 avoid sample degradation (ONT, 2023). 2) It allows for high quality sequencing in
654 countries where Illumina sequencing is not available. 3) It is slightly cheaper than
655 using the Illumina platform. 4) The long read length eliminates amplification bias
656 (Goodwin et al., 2015). The avoidance of amplification bias is important for gene
657 expression quantification, which is relevant to the discussion in section 2.2.2. How-
658 ever, the MinION 1B requires a high spec computer and, at £98/sample exclud-
659 ing library preparation, use of this platform also remains too expensive for many
660 projects.

661 *2.1.2. Spectroscopy and hyperspectral imaging*

662 Although not capable of specific disease diagnosis, the MultispeQ is an important,
663 low cost tool to consider in the context of disease detection in the absence of vis-
664 ible symptoms. This handheld plant phenotyping device can be used to indicate
665 the non-specific presence of plant disease at an extremely low cost (Kuhlgert et al.,
666 2022). The MultispeQ operates similarly to photo spectroscopy and measures envi-
667 ronmental conditions such as light intensity, temperature and humidity. It can also
668 be used to measure photosystem II quantum yield, which is an indicator of plant
669 health and to detect non-photochemical exciton quenching, which has been shown
670 to have a significant negative correlation with disease index (Kuhlgert et al., 2022).

671 A highly informative technique that we can utilise in the prediction of plant dis-
672 ease with CV is to sample more continuously from the electromagnetic spectrum
673 with hyperspectral imaging (HSI). As with the MultispeQ, HSI enables us to de-
674 tect changes in the chemical composition of biological tissue as conditions such as

675 ripeness or disease status change (Bock et al., 2010). The term ‘spectral signature’
676 is used to describe the pattern of electromagnetic radiation reflected by a subject.
677 However, particularly in the case of biology, the term signature is misleading as bi-
678 ological samples often have highly heterogeneous reflectance spectra (Bock et al.,
679 2010). All of the above mentioned CV studies applied ML techniques to RGB im-
680 ages. RGB images capture three discrete bands of the visible spectrum from 400-
681 700 nm. Black and white digital images have two spatial dimensions and a single
682 dimension that describes the darkness of each pixel on a scale of 0-255, whereas
683 RGB images have three colour dimensions represented by values between 0-255,
684 each describing the intensity of red, green or blue light. Hyperspectral images how-
685 ever store a more complete reflectance spectrum for each pixel while also main-
686 taining spatial relationships. The spectral range of these images can be as wide as
687 400-2500 nm (Goetz et al., 1985).

688 Although the applications of hyperspectral photography have long been explored
689 by NASA, this technology is only now becoming cheap enough to be used in in-
690 dustries like agriculture. However, commercially available cameras capable of cap-
691 turing data from the 400-2500 nm range remain expensive and more typically used
692 cameras only sample 400-1000 nm (Table 4). Despite the reduced spectral range of
693 the cheaper cameras, they still provide orders of magnitude more data than RGB
694 cameras, though a lot of this data is highly correlated.

695 Uptake of HSI has recently exploded in a host of fields including archaeology, art
696 conservation, food safety, medicine and crime scene investigation (Lu and Fei,
697 2014). Typical applications of HSI in agriculture include the estimation of yield

698 (Gutiérrez et al., 2019; Li et al., 2020), assessment of vigour (Feng et al., 2018),
699 remote weed identification (Okamoto et al., 2007), nutrient status (Nguyen et al.,
700 2020) and disease monitoring (Pan et al., 2019).

701 The analysis of HSI data presents problems which are familiar to ML engineers
702 and nowadays are solved routinely. These problems include the large size of HSI
703 hypercubes, high dimensionality, high intra-class variability, and high correlation
704 between spectral bands. Many approaches have been taken to analyse this data
705 and, for a long time, SVMs were the most widely used (Yue et al., 2015). DNNs
706 are now commonly used to analyse this data as they are particularly well suited to
707 the task of classification with HSI data. DNNs have the ability to isolate hidden
708 and complex data structures, they can utilise a great variety of data types, they
709 are flexible in their architectures and the complexity of the functions they can ap-
710 ply, and they are ideally suited to distributed computing (Paoletti et al., 2019). As
711 such, with the addition of dimension reduction techniques such as principal com-
712 ponent analysis (Yue et al., 2015), the analysis of HSI data with DNNs, although
713 more computationally demanding, becomes little more complex than such analyses
714 of RGB image data.

715 While the field of CV is advancing at a rapid pace, so too are the fields of molecu-
716 lar biology and spectroscopy. The use of tools and knowledge from these fields will
717 allow projects of various budgets to go beyond the simple application of CNNs to
718 RGB images and, in doing so, model disease in greater detail with tangible biologi-
719 cal explications of model behaviour.

720 **2.2. Model testing**

721 *2.2.1. The black box of DNNs*

722 It is well known how poorly current CV models deal with unexpected edge cases
723 and shifts in test data distribution (Schölkopf et al., 2021). However, in apply-
724 ing CV to plant pathology and agriculture we encounter more cases than most
725 where the test data does not align well with the training data. These problems
726 routinely arise in CV from effects of camera blur, image quality or shifting camera
727 angle. However, in plant pathology we must also contend with the perturbations
728 of weather, climate, plant growth stage, crop variety, a plant’s developmental re-
729 sponse to growing conditions and so on. While it is contentious how robust of a fix
730 techniques like data augmentations or inductive biases may be to solve the former
731 list of issues (Schölkopf et al., 2021), the latter issues will only be solved by truly
732 understanding how our models are making predictions.

733 Although DNNs are still considered black box optimisers, much work has been
734 done to understand their various facets and potential foibles. For example, the role
735 of each dense layer of a CNN has been shown to have distinct roles in feature level
736 extraction and generalisability (Yosinski et al., 2014), and the output of convolu-
737 tion layers have been visualised to show which physical features in an image were
738 more exaggerated (Zeiler and Fergus, 2014). In a similar study, a host of prede-
739 fined layer-wise and neuron-wise visualisation techniques were applied to a CNN
740 that had been trained on images of plant disease (Toda and Okura, 2019). This
741 work showed that the CNN in question was indeed using visible symptoms of the
742 disease that were similar to those used by human experts. Others have sought to

743 learn how best to actively deceive or manipulate a deep neural network into mis-
744 classification. Working within the remit of cyber security, it was shown that image
745 classifiers based on SVMs and DNNs could easily be deceived with a simple eva-
746 sion algorithm (Biggio *et al.*, 2013). This shows how brittle these classifiers can be
747 and highlights the importance of adopting techniques that rely more heavily on
748 causal inference, such as semi-supervised learning, (Peters *et al.*, 2017) or semantic
749 segmentation. It also highlights the importance of rigorous and conciliatory inter-
750 rogation of models prior to deployment. At present our methods of model evalua-
751 tion are widely considered insufficient and much more work is needed in this area.

752 *2.2.2. Inspecting informative features*

753 A key benefit to the use of CNNs is feature learning. This is the process by which
754 a model will define for itself which features of a dataset it considers informative
755 (Voulodimos *et al.*, 2018). In other CV algorithms, an engineer must handcraft
756 descriptive features of a subject manually, using their expertise and/or diagnostic
757 tools to guide them. In this latter case, pre-processed data are used rather than
758 raw data, as in a CNN. In the convolution layers of a CNN however, kernels and
759 attention weights are applied to raw or augmented image data which emphasise in-
760 formative physical features, and apply inductive biases and self attention, before
761 this data is passed to the dense layer(s) of the network (O'Mahony *et al.*, 2020).
762 We might assume that these physical features would include those that humans
763 consider to be the obvious visible markers for plant disease, such as the presence of
764 lesions on a leaf. However, it is likely that these networks will also identify mark-
765 ers that humans do not notice or cannot perceive and may ignore some features

766 that plant pathologists have long considered important. This provides us with the
767 opportunity to learn more about how to identify disease early with human vision,
768 CV, and molecular biology. Using time-series qPCR, transcriptome or metabolome
769 data to identify the biological markers used by CNNs at the earliest moments of
770 detection would allow for the validation of the image features used by the model.
771 Such a biological explanation of the models informative features would tell us if
772 the model is making correct inferences for, what we consider, correct reasons or if
773 it is correct for spurious reasons, suggesting a poor ability to generalise stemming
774 from naive inductive reasoning. Such work may also highlight new ways to identify
775 disease with and without ML or new ways of combating disease spread through
776 phytosanitation, agro-chemistry or plant breeding.

777 In recent years the combination of CNNs and transcriptomics in medical research
778 has seen a surge in popularity. Such studies involve spatial transcriptomics (Chelebian
779 et al., 2021; Yang and McCord, 2021), the identification of Non-small cell lung
780 cancer subtypes (Yu et al., 2020) and the elucidation of the various functions of
781 drugs (Meyer et al., 2019). CNNs have also been applied alongside transcriptomics
782 in plant science in the investigation of gene regulation in *Arabidopsis* (MacLean,
783 2019). However, the investigation of the black box nature of CNNs by means of
784 omics appears to be completely absent from the literature.

785 Attention maps produced by software like GRAD-CAM (Selvaraju et al., 2017;
786 Wang et al., 2018) are another way to inspect informative features of image data.
787 GRAD-CAM produces an explanation for the decision that a model makes about a
788 given image by visually highlighting the informative features of that image. GRAD-

789 CAM is described as ‘gradient-based’ as it uses the gradient data that is fed into
790 the last convolution layer of a CNN. This allows us to make assessments before the
791 spatial relationships in the data are lost in the fully connected layers (Selvaraju et
792 al., 2017). Alternative ‘reference based’ systems, such as DeepLIFT, rely on back-
793 propagation (Shrikumar et al., 2017) or forward propagation (Explanation map)
794 (Ghosal et al., 2018), using a reference image that does not contain the feature
795 of interest. Applying these methods to miss-classified images can highlight why
796 a model is performing suboptimally (Toda and Okura, 2019) as results produced
797 with these methods have been shown to be highly correlated with assessments of
798 plant disease made by human experts (Ghosal et al., 2018).

799 **3. A roadmap to commercial implementation**

800 Once you have developed, trained and evaluated your model, it is time to begin
801 the process of implementation. However, it is best to have considered and planned
802 this step well ahead of time. There are several decisions made during development
803 that may depend on the intended mode of implementation. For example, if the
804 model is to be run on an edge device or smartphone, computation cost must be
805 kept to a minimum. Likewise, if the model is to be made available via a rented
806 server, reducing computational cost will reduce financial cost. Prior to training,
807 choosing to use architectures such as ResNet18 and MobileNetV3 (Howard et al.,
808 2019) will help to keep computational cost down and, after training, methods such
809 as pruning and quantisation may reduce this cost further. While Google Colab
810 offers free limited access to GPUs for model training, the rental cost of a 16GB

811 NVIDIA V100 GPU, which would be the minimum needed to train a transformer
812 model or large CNN, is \$2.48/hour. As such, developing and training such large
813 models for days, or even weeks, can soon become expensive.

814 ONNX Runtime from Microsoft (Microsoft, 2023) offers a huge array of tools to
815 help accelerate, quantise and deploy trained DNNs. Such models can be incor-
816 porated into Android or IOS apps, using the phones builtin camera, they can be
817 deployed via the web, on edge devices like a Raspberry Pi or in embedded sys-
818 tems for drone mapping or smart irrigation. However the operator schemas sup-
819 ported by ONNX runtime must be considered here. For example, ConvNeXT,
820 which uses GELU and stochastic depth, may cause problems as these operators
821 are not yet supported. TensorFlow also offers a pipeline for model deployment
822 and the Pytorch toolkit for techniques like quantisation aware training and model
823 compression is maturing, but presented difficulties when we attempted to use it.

824 In contrast, the ONNX Runtime pipeline is extremely easy to use and supports
825 all popular model formats like Pytorch, TensorFlow and SciKit Learn. While the
826 latest methods of pruning are reported to achieve a 30% reduction in the size of
827 ResNet18 with only a 2% loss in accuracy on ImageNet (Solodskikh et al., 2023),
828 this remains an active area of research, producing inconstant results. There is no
829 guarantee that pruning will lessen computational cost. Techniques such as training
830 aware pruning show promise but require further research.

831 For implementation of object detection or segmentation models, we recommend the
832 Detectron2 library from Facebook (facebookresearch, 2023). This library incorpo-
833 rates Faster-RCNN, Mask-RCNN and some new transformers models like ViTDet,

834 and offers a host of tutorials on the whole process from training to implementa-
835 tion.

836 **Conclusion**

837 Described here are all of the tools necessary to develop highly optimised and ro-
838 bust ML models that use minimal computational power and provide real benefit to
839 sectors that have more modest budgets. The application of these tools will allow
840 us to break from the common trend in the ML industry, where expensive hard-
841 ware is employed to develop complex and computationally expensive models to the
842 detriment of improving training data quality.

843 With the application of off-the-shelf architectures to stock datasets, such as the
844 plant village dataset (Geetharamani and Pandian, 2019), we can easily achieve pre-
845 diction accuracy scores in the high 90% range (Thapa et al., 2020). However, such
846 models have little value because they will not generalise to complex real-world en-
847 vironments due to the simplicity of the training data.

848 We offer the following recommendations for the development of efficient, inexpen-
849 sive and robust CV models for plant pathology.

850 **Garbage in - garbage out:** The thoughtless application of advanced models to
851 poorly labeled, simplistic, contaminated or maltransformed data will yield models
852 that have little value in the field, with slow inference time, poor accuracy and an
853 inability to generalise. To avoid this fate we should; (A) where possible, consult
854 with specialists and utilise the invaluable tools from biology, chemistry and spec-
855 troscopy to label data, (B) use the minimum appropriate image input size to im-

856 prove runtime speed and help avoid overfitting, and (C) avoid needless data trans-
857 formations like normalisation, which can alter data in unreliable ways.

858 **The potential in training procedures:** Techniques like semantic segmentation
859 and semi-supervised learning have potential to lessen both bias and variance in
860 a models predictions by promoting deductive reasoning over inductive reasoning.

861 While appropriately scaled CNNs and evolved neural networks offer the potential
862 to produce models with optimised runtime speed and improved generalisation abil-
863 ity.

864 **Robust and conciliatory interrogation of models:** While simpler model-
865 ing methods, such as SVMs, still have a role to play in modern computer vision,
866 most of the models we employ for this purpose are exceedingly complicated and
867 are prone to failing in equally complicated ways. Failure of a disease detection
868 model resulting in an outbreak of disease could have very serious consequences.
869 It is vital therefore that we test the models we develop rigorously to ensure that
870 they are not prone to miss-classification born of overfitting and naive generalisa-
871 tions. While metrics such as accuracy, F1, AUC, recall and precision are valuable,
872 DNNs are often capable of learning to optimise these summary statistics indirectly,
873 rather than learning to produce reliable predictions. Tools such as confusion matri-
874 ces and explanation maps go much further in understanding the behaviour of CV
875 models. However, it is important that we invest in the development of new and
876 tailored means of understanding these models, such as the application of omics, as
877 discussed in section 2.2.2.

878 If we apply our wealth of knowledge and proven techniques from botany and agron-

omy to the acquisition of training data, the development of data processing pipelines, and the interrogation of trained models, we can produce applications with game changing potential. We are now only 27 years away from a predicted global population of 9.7 billion people (UN, 2022). Thus, with the devastating effects of the climate crisis already very much apparent, it is vital that we act now to build robust international infrastructure targeted at securing food supplies and eliminating extreme poverty. The techniques discussed here, such as semi-supervised learning, evolving neural networks and incorporation of omics to model development may enable us, as a community of growers, botanists and ML developers, to help reduce poverty, improve the relationship between growers and the natural environment, and increase stability in the agriculture industry from the foundation up.

AUTHOR CONTRIBUTIONS

J.R.S. conceived of this review, read and summarized the literature, and wrote the first draft of the manuscript. K.J.D. and D.W.F. continually reviewed and edited the manuscript and approved the final manuscript before submission and publication.

Acknowledgments

This work was made possible by funding from the Doctoral Centre for Safe, Ethical and Secure Computing at the University of York.

898 **Data availability**

899 **Case study one: Semantic segmentation for cocoa disease detection** The
900 image data, annotations and link to the accompanying Github repository can
901 be found at:

902 `osf.io/79kx3/?view_only=4a2c1dccee1a4baeb85de5002c702f10`

903 **Case study two: Semi-supervised learning for outlier detection** The data to
904 train the initial supervised model, the .csv search terms file for the below web
905 scraper and the final semi-supervised model weights can be found at: `osf.io/`
906 `h5gj7/?view_only=dbf9f245e21a41e185f5b73e718b4cad` The 'contaminated'
907 data used to train the semi-supervised model was generated using the code at:
908 `github.com/jrsykes/Google-Image-Scraper`

909 The custom code used to train both the initial model and the final semi-supervised
910 model can be found at:

911 `github.com/jrsykes/CocoaReader/blob/main/PlantNotPlant`

912 **Case study three: Disease detection and normalisation** The custom code used to
913 conduct this study can be found in the following Github repository, with accompa-
914 nying Readme.md: `github.com/jrsykes/CocoaReader`

915 The data for this study was scraped from the internet using the code in to follow-
916 ing github repository: `github.com/jrsykes/Google-Image-Scraper`

917 The location of the accompanying ".csv search terms file" is described below.

918 **Case study four: Optimisation of BN momentum and image size** The custom
919 code to run this sweep can be found at the following Github repository: `github.`

920 `com/jrsykes/CocoaReader/tree/main/CocoaNet`

921 The main script is titled `CocoNetsweep_min.sh` and the wandb config file is titled
922 `CocoaNetSweepConfig_min.yml`. The data used to generate these results and the
923 full wandb report can be found at:

924 `osf.io/2fw6g/?view_only=adc66ba66f83465a9e7b111515a60bf2`

925 REFERENCES

926 Amer, M. and T. Maul, (2019). “A Review of Modularization Techniques in Artifi-
927 cial Neural Networks”. In: *Artificial Intelligence Review* 52.1, pp. 527–561.

928 An, J. and S. Cho, (2015). “Variational Autoencoder Based Anomaly Detection
929 Using Reconstruction Probability”. In: *Special lecture on IE*.

930 Asperti, A., D. Evangelista, and E. Loli Piccolomini, (2021). “A Survey on Varia-
931 tional Autoencoders from a Green AI Perspective”. In: *SN Computer Science*
932 2.4, p. 301.

933 Banerjee, A., (2007). “An Analysis of Logistic Models: Exponential Family Con-
934 nections and Online Performance”. In: *Proceedings of the 2007 SIAM Interna-*
935 *tional Conference on Data Mining*. Proceedings of the 2007 SIAM International
936 Conference on Data Mining. Society for Industrial and Applied Mathematics,
937 pp. 204–215.

938 Barbedo, J. G. A., (2016). “A Review on the Main Challenges in Automatic Plant
939 Disease Identification Based on Visible Range Images”. In: *Biosystems Engi-*
940 *neering* 144, pp. 52–60.

- 941 Barbedo, J. G. A., L. V. Koenigkan, and T. T. Santos, (2016). “Identifying Multi-
942 ple Plant Diseases Using Digital Image Processing”. In: *Biosystems Engineering*
943 147, pp. 104–116.
- 944 Bharati, P. and A. Pramanik, (2020). “Deep Learning Techniques—R-CNN to
945 Mask R-CNN: A Survey”. In: *Computational Intelligence in Pattern Recogni-
946 tion*. Ed. by A. K. Das, J. Nayak, B. Naik, S. K. Pati, and D. Pelusi. Advances
947 in Intelligent Systems and Computing. Singapore: Springer, pp. 657–668.
- 948 Bickford, D., D. J. Lohman, N. S. Sodhi, P. K. L. Ng, R. Meier, K. Winker, K. K.
949 Ingram, and I. Das, (2007). “Cryptic Species as a Window on Diversity and
950 Conservation”. In: *Trends in Ecology & Evolution* 22.3, pp. 148–155.
- 951 Biewald, L., (2023). *Experiment Tracking with Weights and Biases*.
- 952 Biggio, B., I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto,
953 and F. Roli, (2013). “Evasion Attacks against Machine Learning at Test Time”.
954 In: *Advanced Information Systems Engineering*. Ed. by C. Salinesi, M. C. Nor-
955 rie, and Ó. Pastor. Red. by D. Hutchison, T. Kanade, J. Kittler, J. M. Klein-
956 berg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan,
957 B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, and G. Weikum.
958 Vol. 7908. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 387–402.
- 959 Bock, C. H., G. H. Poole, P. E. Parker, and T. R. Gottwald, (2010). “Plant Dis-
960 ease Severity Estimated Visually, by Digital Photography and Image Analysis,
961 and by Hyperspectral Imaging”. In: *Critical Reviews in Plant Sciences* 29.2,
962 pp. 59–107.

- 963 Boratyn, G. M., C. Camacho, P. S. Cooper, G. Coulouris, A. Fong, N. Ma, T. L.
964 Madden, W. T. Matten, S. D. McGinnis, Y. Merezhuk, Y. Raytselis, E. W.
965 Sayers, T. Tao, J. Ye, and I. Zaretskaya, (2013). “BLAST: A More Efficient Re-
966 port with Usability Improvements”. In: *Nucleic Acids Research* 41.W1, W29–
967 W33.
- 968 Cao, S. and R. Nevatia, (2016). “Exploring Deep Learning Based Solutions in Fine
969 Grained Activity Recognition in the Wild”. In: *2016 23rd International Con-
970 ference on Pattern Recognition (ICPR)*. 2016 23rd International Conference on
971 Pattern Recognition (ICPR), pp. 384–389.
- 972 Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko,
973 (2020). “End-to-End Object Detection with Transformers”. In: *Computer Vi-
974 sion – ECCV 2020*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm.
975 Lecture Notes in Computer Science. Cham: Springer International Publishing,
976 pp. 213–229.
- 977 Chelebian, E., C. Avenel, K. Kartasalo, M. Marklund, A. Tanoglidi, T. Mirtti, R.
978 Colling, A. Erickson, A. D. Lamb, J. Lundeberg, and C. Wählby, (2021). “Mor-
979 phological Features Extracted by AI Associated with Spatial Transcriptomics in
980 Prostate Cancer”. In: *Cancers* 13.19 (19), p. 4837.
- 981 Chiu, M. T., X. Xu, Y. Wei, Z. Huang, A. G. Schwing, R. Brunner, H. Khacha-
982 trian, H. Karapetyan, I. Dozier, G. Rose, D. Wilson, A. Tudor, N. Hovakimyan,
983 T. S. Huang, and H. Shi, (2020). “Agriculture-Vision: A Large Aerial Image
984 Database for Agricultural Pattern Analysis”. In: *Proceedings of the IEEE/CVF
985 Conference on Computer Vision and Pattern Recognition*, pp. 2828–2838.

- 986 Chouhan, S. S., U. P. Singh, and S. Jain, (2020). “Applications of Computer Vi-
987 sion in Plant Pathology: A Survey”. In: *Archives of Computational Methods in*
988 *Engineering* 27.2, pp. 611–632.
- 989 Cohen, Y. and M. D. Coffey, (1986). “Systemic Fungicides and the Control of Oomycetes”.
990 In: *Annual Review of Phytopathology* 24.1, pp. 311–338.
- 991 Dang, D.-C., A. Ereemeev, and P. K. Lehre, (2021). “Escaping Local Optima with
992 Non-Elitist Evolutionary Algorithms”. In: *Proceedings of the AAAI Conference*
993 *on Artificial Intelligence* 35.14 (14), pp. 12275–12283.
- 994 Department of Health. Victoria, A., (2023). *Methyl Bromide Use in Victoria Com-*
995 *munity Factsheet*. URL: [https://www.health.vic.gov.au/publications/](https://www.health.vic.gov.au/publications/methyl-bromide-use-in-victoria-community-factsheet)
996 [methyl-bromide-use-in-victoria-community-factsheet](https://www.health.vic.gov.au/publications/methyl-bromide-use-in-victoria-community-factsheet) (visited on
997 06/26/2023).
- 998 Dilokthanakul, N., P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K.
999 Arulkumaran, and M. Shanahan, (2017). “Deep Unsupervised Clustering with
1000 Gaussian Mixture Variational Autoencoders”.
- 1001 Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner,
1002 M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby,
1003 (2021). *An Image Is Worth 16x16 Words: Transformers for Image Recognition*
1004 *at Scale*. URL: <http://arxiv.org/abs/2010.11929> (visited on 11/14/2022).
1005 preprint.
- 1006 facebookresearch, (2023). *Facebookresearch*. Meta Research.
- 1007 Fan, Y., G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, (2020). “Video Anomaly
1008 Detection and Localization via Gaussian Mixture Fully Convolutional Vari-

- 1009 ational Autoencoder”. In: *Computer Vision and Image Understanding* 195,
1010 p. 102920.
- 1011 Feng, L., S. Zhu, C. Zhang, Y. Bao, X. Feng, and Y. He, (2018). “Identification of
1012 Maize Kernel Vigor under Different Accelerated Aging Times Using Hyperspec-
1013 tral Imaging”. In: *Molecules* 23.12 (12), p. 3078.
- 1014 Ferentinos, K. P., (2018). “Deep Learning Models for Plant Disease Detection and
1015 Diagnosis”. In: *Computers and Electronics in Agriculture* 145, pp. 311–318.
- 1016 Fuentes, S., G. Chacon, D. D. Torrico, A. Zarate, and C. Gonzalez Viejo, (2019).
1017 “Spatial Variability of Aroma Profiles of Cocoa Trees Obtained through Com-
1018 puter Vision and Machine Learning Modelling: A Cover Photography and High
1019 Spatial Remote Sensing Application”. In: *Sensors* 19.14 (14), p. 3054.
- 1020 Geetharamani, G. and A. Pandian, (2019). “Identification of Plant Leaf Diseases
1021 Using a Nine-Layer Deep Convolutional Neural Network”. In: *Computers &
1022 Electrical Engineering* 76, pp. 323–338.
- 1023 Ghosal, S., D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and
1024 S. Sarkar, (2018). “An Explainable Deep Machine Vision Framework for Plant
1025 Stress Phenotyping”. In: *Proceedings of the National Academy of Sciences* 115.18,
1026 pp. 4613–4618.
- 1027 Goetz, A. F. H., G. Vane, J. E. Solomon, and B. N. Rock, (1985). “Imaging Spec-
1028 trometry for Earth Remote Sensing”. In: *Science* 228.4704, pp. 1147–1153.
- 1029 Gong, Y., Y. Sun, D. Peng, P. Chen, Z. Yan, and K. Yang, (2021). “Analyze COVID-
1030 19 CT Images Based on Evolutionary Algorithm with Dynamic Searching Space”.
1031 In: *Complex & Intelligent Systems* 7.6, pp. 3195–3209.

- 1032 Goodwin, S., J. Gurtowski, S. Ethe-Sayers, P. Deshpande, M. C. Schatz, and W. R.
1033 McCombie, (2015). “Oxford Nanopore Sequencing, Hybrid Error Correction,
1034 and de Novo Assembly of a Eukaryotic Genome”. In: *Genome Research* 25.11,
1035 pp. 1750–1756.
- 1036 Grosch, K., (2018). *John Deere – Bringing AI to Agriculture*. Technology and Op-
1037 erations Management. URL: [https://digital.hbs.edu/platform-rctom/
1038 submission/john-deere-bringing-ai-to-agriculture/](https://digital.hbs.edu/platform-rctom/submission/john-deere-bringing-ai-to-agriculture/) (visited on
1039 05/16/2022).
- 1040 Guo, M.-H., T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang,
1041 R. R. Martin, M.-M. Cheng, and S.-M. Hu, (2022). “Attention Mechanisms in
1042 Computer Vision: A Survey”. In: *Computational Visual Media* 8.3, pp. 331–368.
- 1043 Gutiérrez, S., A. Wendel, and J. Underwood, (2019). “Ground Based Hyperspec-
1044 tral Imaging for Extensive Mango Yield Estimation”. In: *Computers and Elec-
1045 tronics in Agriculture* 157, pp. 126–135.
- 1046 He, K., G. Gkioxari, P. Dollar, and R. Girshick, (2017). “Mask R-CNN”. In: Pro-
1047 ceedings of the IEEE International Conference on Computer Vision, pp. 2961–
1048 2969.
- 1049 He, K., X. Zhang, S. Ren, and J. Sun, (2016). “Deep Residual Learning for Image
1050 Recognition”. In: Proceedings of the IEEE Conference on Computer Vision and
1051 Pattern Recognition, pp. 770–778.
- 1052 Horevaji, P., E. Milus, and B. Bluhm, (2011). “A Real-Time qPCR Assay to Quan-
1053 tify *Fusarium Graminearum* Biomass in Wheat Kernels”. In: *Journal of Applied
1054 Microbiology* 111.2, pp. 396–406.

- 1055 Horton, T. R. and T. D. Bruns, (2001). “The Molecular Revolution in Ectomy-
1056 corrhizal Ecology: Peeking into the Black-Box”. In: *Molecular Ecology* 10.8,
1057 pp. 1855–1871.
- 1058 Horzyk, A. and E. Ergün, (2020). “YOLOv3 Precision Improvement by the Weighted
1059 Centers of Confidence Selection”. In: *2020 International Joint Conference on*
1060 *Neural Networks (IJCNN)*. 2020 International Joint Conference on Neural Net-
1061 works (IJCNN), pp. 1–8.
- 1062 Howard, A., M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu,
1063 R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, (2019). *Searching for Mo-*
1064 *bileNetV3*. URL: <http://arxiv.org/abs/1905.02244> (visited on 06/21/2023).
1065 preprint.
- 1066 Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. An-
1067 dreetto, and H. Adam, (2017). *MobileNets: Efficient Convolutional Neural Net-*
1068 *works for Mobile Vision Applications*. URL: [http://arxiv.org/abs/1704.](http://arxiv.org/abs/1704.04861)
1069 [04861](http://arxiv.org/abs/1704.04861) (visited on 10/14/2022). preprint.
- 1070 Huang, H., Z. Wei, and L. Yao, (2019). “A Novel Approach to Component Assem-
1071 bly Inspection Based on Mask R-CNN and Support Vector Machines”. In: *In-*
1072 *formation* 10.9 (9), p. 282.
- 1073 Huda-Shakirah, A. R., N. M. I. Mohamed Nor, L. Zakaria, Y.-H. Leong, and M. H.
1074 Mohd, (2022). “Lasiodiplodia Theobromae as a Causal Pathogen of Leaf Blight,
1075 Stem Canker, and Pod Rot of Theobroma Cacao in Malaysia”. In: *Scientific*
1076 *Reports* 12.1 (1), p. 8966.

- 1077 Ioffe, S. and C. Szegedy, (2015). “Batch Normalization: Accelerating Deep Network
1078 Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd In-*
1079 *ternational Conference on Machine Learning*. International Conference on Ma-
1080 chine Learning. PMLR, pp. 448–456.
- 1081 Khanzada, M., A. Lodhi, and S. Shahzad, (2005). “Chemical Control of Lasiodiplo-
1082 dia Theobromae, the Causal Agent of Mango Decline in Sindh”. In: *Pakistan*
1083 *Journal of Botany* 37, pp. 1023–1030.
- 1084 Kingma, D. P. and M. Welling, (2019). “An Introduction to Variational Autoen-
1085 coders”. In: *Foundations and Trends[®] in Machine Learning* 12.4, pp. 307–392.
- 1086 Krizhevsky, A., I. Sutskever, and G. E. Hinton, (2017). “ImageNet Classification
1087 with Deep Convolutional Neural Networks”. In: *Communications of the ACM*
1088 60.6, pp. 84–90.
- 1089 Kroon, L. P. N. M., H. Brouwer, A. W. A. M. de Cock, and F. Govers, (2012).
1090 “The Genus *Phytophthora* Anno 2012”. In: *Phytopathology[®]* 102.4, pp. 348–
1091 364.
- 1092 Kuhlert, S., G. Austic, R. Zegarac, I. Osei-Bonsu, D. Hoh, M. I. Chilvers, M. G.
1093 Roth, K. Bi, D. TerAvest, P. Weebadde, and D. M. Kramer, (2022). “Multi-
1094 speQ Beta: A Tool for Large-Scale Plant Phenotyping Connected to the Open
1095 PhotosynQ Network”. In: *Royal Society Open Science* 3.10 (), p. 160592.
- 1096 Kuok Ho, D. T. and P. S. Yap, (2020). *A Systematic Review of Slash-and-Burn*
1097 *Agriculture as an Obstacle to Future-Proofing Climate Change*, p. 19. 01 p.
- 1098 Li, B., X. Xu, L. Zhang, J. Han, C. Bian, G. Li, J. Liu, and L. Jin, (2020). “Above-
1099 Ground Biomass Estimation and Yield Prediction in Potato by Using UAV-

- 1100 based RGB and Hyperspectral Imaging”. In: *ISPRS Journal of Photogramme-*
1101 *try and Remote Sensing* 162, pp. 161–172.
- 1102 Li, C. and M. Wand, (2016). “Precomputed Real-Time Texture Synthesis with
1103 Markovian Generative Adversarial Networks”. In: *Computer Vision – ECCV*
1104 *2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Lecture Notes in
1105 Computer Science. Cham: Springer International Publishing, pp. 702–716.
- 1106 Li, X. and J. She, (2017). “Collaborative Variational Autoencoder for Recom-
1107 mender Systems”. In: *Proceedings of the 23rd ACM SIGKDD International*
1108 *Conference on Knowledge Discovery and Data Mining*. KDD ’17. New York,
1109 NY, USA: Association for Computing Machinery, pp. 305–314.
- 1110 Lim, K.-L., X. Jiang, and C. Yi, (2020). “Deep Clustering With Variational Au-
1111 toencoder”. In: *IEEE Signal Processing Letters* 27, pp. 231–235.
- 1112 Lin, E., S. Mukherjee, and S. Kannan, (2020). “A Deep Adversarial Variational
1113 Autoencoder Model for Dimensionality Reduction in Single-Cell RNA Sequenc-
1114 ing Analysis”. In: *BMC Bioinformatics* 21.1, p. 64.
- 1115 Liu, Z., H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, (2022). “A
1116 ConvNet for the 2020s”.
- 1117 Liu, Z., P. Luo, X. Wang, and X. Tang, (2015). “Deep Learning Face Attributes in
1118 the Wild”. In: *Proceedings of the IEEE International Conference on Computer*
1119 *Vision*, pp. 3730–3738.
- 1120 Lopes, J. F., V. G. T. da Costa, D. F. Barbin, L. J. P. Cruz-Tirado, V. Baeten,
1121 and S. Barbon Junior, (2022). “Deep Computer Vision System for Cocoa Clas-
1122 sification”. In: *Multimedia Tools and Applications* 81.28, pp. 41059–41077.

- 1123 Lu, G. and B. Fei, (2014). “Medical Hyperspectral Imaging: A Review”. In: *Jour-*
1124 *nal of Biomedical Optics* 19.1, p. 010901.
- 1125 Luchi, N., R. Ioos, and A. Santini, (2020). “Fast and Reliable Molecular Methods
1126 to Detect Fungal Pathogens in Woody Plants”. In: *Applied Microbiology and*
1127 *Biotechnology* 104.6, pp. 2453–2468.
- 1128 Maalø, L., M. Fraccaro, V. Liévin, and O. Winther, (2019). “BIVA: A Very Deep
1129 Hierarchy of Latent Variables for Generative Modeling”. In: *Advances in Neural*
1130 *Information Processing Systems*. Vol. 32. Curran Associates, Inc.
- 1131 Mack, C. A., (2011). “Fifty Years of Moore’s Law”. In: *IEEE Transactions on*
1132 *Semiconductor Manufacturing* 24.2, pp. 202–207.
- 1133 MacLean, D., (2019). “A Convolutional Neural Network for Predicting Transcrip-
1134 tional Regulators of Genes in Arabidopsis Transcriptome Data Reveals Classifi-
1135 cation Based on Positive Regulatory Interactions”. In: p. 618926.
- 1136 Maddison, A. C., G. Macias, C. Moreira, R. Arias, and R. Neira, (1995). “Cocoa
1137 Production in Ecuador in Relation to Dry-Season Escape from Pod Rot Caused
1138 by *Crinipellis Perniciosa* and *Moniliophthora Roreri*”. In: *Plant Pathology* 44.6,
1139 pp. 982–998.
- 1140 Malhi, Y., J. T. Roberts, R. A. Betts, T. J. Killeen, W. Li, and C. A. Nobre, (2008).
1141 “Climate Change, Deforestation, and the Fate of the Amazon”. In: *Science*
1142 319.5860, pp. 169–172.
- 1143 Marelli, J.-P., D. I. Guest, B. A. Bailey, H. C. Evans, J. K. Brown, M. Junaid,
1144 R. W. Barreto, D. O. Lisboa, and A. S. Puig, (2019). “Chocolate Under Threat
1145 from Old and New Cacao Diseases”. In: *Phytopathology*® 109.8, pp. 1331–1343.

- 1146 Meinhardt, L. W., J. Rincones, B. A. Bailey, M. C. Aime, G. W. Griffith, D. Zhang,
1147 and G. a. G. Pereira, (2008). “Moniliophthora Perniciosa, the Causal Agent of
1148 Witches’ Broom Disease of Cacao: What’s New from This Old Foe?” In: *Molec-*
1149 *ular Plant Pathology* 9.5, pp. 577–588.
- 1150 Meyer, J. G., S. Liu, I. J. Miller, J. J. Coon, and A. Gitter, (2019). “Learning
1151 Drug Functions from Chemical Structures with Convolutional Neural Networks
1152 and Random Forests”. In: *Journal of Chemical Information and Modeling* 59.10,
1153 pp. 4438–4449.
- 1154 Microsoft, (2023). *ONNX Runtime*. URL: <https://onnxruntime.ai/> (visited on
1155 06/21/2023).
- 1156 Mite-Baidal, K., E. Solís-Avilés, T. Martínez-Carriel, A. Marcillo-Plaza, E. Cruz-
1157 Ibarra, and W. Baque-Bustamante, (2019). “Analysis of Computer Vision Al-
1158 gorithms to Determine the Quality of Fermented Cocoa (Theobroma Cacao):
1159 Systematic Literature Review”. In: *ICT for Agriculture and Environment*. Ed.
1160 by R. Valencia-García, G. Alcaraz-Mármol, J. del Cioppo-Morstadt, N. Vera-
1161 Lucio, and M. Bucaram-Leverone. *Advances in Intelligent Systems and Com-*
1162 *puting*. Cham: Springer International Publishing, pp. 79–87.
- 1163 Motamedi, M., N. Sakharnykh, and T. Kaldewey, (2021). “A Data-Centric Ap-
1164 proach for Training Deep Neural Networks with Less Data”.
- 1165 Nagarajan, S., G. Seibold, J. Kranza, E. E. Saari, and L. M. Joshi, (1984). “Moni-
1166 toring Wheat Rust Epidemics With the Landsat-2 Satellite”. In: *Phytopathology*
1167 74.5, p. 585.

- 1168 Nalisnick, E., A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan,
1169 (2019). *Do Deep Generative Models Know What They Don't Know?* URL: <http://arxiv.org/abs/1810.09136> (visited on 11/10/2022). preprint.
- 1170
1171 Nguyen, H. D. D., V. Pan, C. Pham, R. Valdez, K. Doan, and C. Nansen, (2020).
1172 “Night-Based Hyperspectral Imaging to Study Association of Horticultural
1173 Crop Leaf Reflectance and Nutrient Status”. In: *Computers and Electronics
1174 in Agriculture* 173, p. 105458.
- 1175 O’Donnell, K., T. J. Ward, V. A. R. G. Robert, P. W. Crous, D. M. Geiser, and
1176 S. Kang, (2015). “DNA Sequence-Based Identification of Fusarium: Current
1177 Status and Future Directions”. In: *Phytoparasitica* 43.5, pp. 583–595.
- 1178 O’Mahony, N., S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L.
1179 Krpalkova, D. Riordan, and J. Walsh, (2020). “Deep Learning vs. Traditional
1180 Computer Vision”. In: *Advances in Computer Vision*. Ed. by K. Arai and S.
1181 Kapoor. *Advances in Intelligent Systems and Computing*. Cham: Springer In-
1182 ternational Publishing, pp. 128–144.
- 1183 Okamoto, H., T. Murata, T. Kataoka, and S.-I. Hata, (2007). “Plant Classification
1184 for Weed Detection Using Hyperspectral Imaging with Wavelet Analysis”. In:
1185 *Weed Biology and Management* 7.1, pp. 31–37.
- 1186 Oliveira, M. M., B. V. Cerqueira, S. Barbon, and D. F. Barbin, (2021). “Classi-
1187 fication of Fermented Cocoa Beans (Cut Test) Using Computer Vision”. In:
1188 *Journal of Food Composition and Analysis* 97, p. 103771.
- 1189 ONT, (2023). *Field Sequencing Kit*. URL: [https://store.nanoporetech.com/
1190 field-sequencing-kit.html](https://store.nanoporetech.com/field-sequencing-kit.html) (visited on 02/01/2023).

- 1191 Ovaskainen, O., J. Nokso-Koivisto, J. Hottola, T. Rajala, T. Pennanen, H. Ali-
1192 Kovero, O. Miettinen, P. Oinonen, P. Auvinen, L. Paulin, K.-H. Larsson, and
1193 R. Mäkipää, (2010). “Identifying Wood-Inhabiting Fungi with 454 Sequencing
1194 – What Is the Probability That BLAST Gives the Correct Species?” In: *Fungal*
1195 *Ecology* 3.4, pp. 274–283.
- 1196 Pan, T.-t., E. Chyngyz, D.-W. Sun, J. Paliwal, and H. Pu, (2019). “Pathogenetic
1197 Process Monitoring and Early Detection of Pear Black Spot Disease Caused by
1198 *Alternaria Alternata* Using Hyperspectral Imaging”. In: *Postharvest Biology*
1199 *and Technology* 154, pp. 96–104.
- 1200 Paoletti, M. E., J. M. Haut, J. Plaza, and A. Plaza, (2019). “Deep Learning Clas-
1201 sifiers for Hyperspectral Imaging: A Review”. In: *ISPRS Journal of Photogram-*
1202 *metry and Remote Sensing* 158, pp. 279–317.
- 1203 Parra, P., T. Negrete, J. Llaguno, and N. Vega, (2018). “Computer Vision Tech-
1204 niques Applied in the Estimation of the Cocoa Beans Fermentation Grade”. In:
1205 *2018 IEEE ANDESCON*. 2018 IEEE ANDESCON, pp. 1–10.
- 1206 Patrício, D. I. and R. Rieder, (2018). “Computer Vision and Artificial Intelligence
1207 in Precision Agriculture for Grain Crops: A Systematic Review”. In: *Computers*
1208 *and Electronics in Agriculture* 153, pp. 69–81.
- 1209 Peters, J., D. Janzing, and B. Schölkopf, (2017). *Elements of Causal Inference:*
1210 *Foundations and Learning Algorithms*. The MIT Press.
- 1211 Phillips-Mora, W. and M. J. Wilkinson, (2007). “Frosty Pod of Cacao: A Disease
1212 with a Limited Geographic Range but Unlimited Potential for Damage”. In:
1213 *Phytopathology*® 97.12, pp. 1644–1647.

- 1214 Ramos-Giraldo, P., C. Reberg-Horton, A. M. Locke, S. Mirsky, and E. Lobaton,
1215 (2020). “Drought Stress Detection Using Low-Cost Computer Vision Systems
1216 and Machine Learning Techniques”. In: *IT Professional* 22.3, pp. 27–29.
- 1217 Ray, M., A. Ray, S. Dash, A. Mishra, K. G. Achary, S. Nayak, and S. Singh, (2017).
1218 “Fungal Disease Detection in Plants: Traditional Assays, Novel Diagnostic Tech-
1219 niques and Biosensors”. In: *Biosensors and Bioelectronics* 87, pp. 708–723.
- 1220 Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, (2016). “You Only Look
1221 Once: Unified, Real-Time Object Detection”. In: Proceedings of the IEEE Con-
1222 ference on Computer Vision and Pattern Recognition, pp. 779–788.
- 1223 Ren, S., K. He, R. Girshick, and J. Sun, (2015). “Faster R-CNN: Towards Real-
1224 Time Object Detection with Region Proposal Networks”. In: *Advances in Neu-
1225 ral Information Processing Systems*. Vol. 28. Curran Associates, Inc.
- 1226 Rice, R. A. and R. Greenberg, (2000). “Cacao Cultivation and the Conservation of
1227 Biological Diversity”. In: *AMBIO: A Journal of the Human Environment* 29.3,
1228 pp. 167–173.
- 1229 Rodriguez, C., O. Alfaro, P. Paredes, D. Esenarro, and F. Hilario, (2021). “Ma-
1230 chine Learning Techniques in the Detection of Cocoa (*Theobroma Cacao* L.)
1231 Diseases”. In: *Annals of the Romanian Society for Cell Biology*, pp. 7732–7741.
- 1232 Salvatore, M. M., A. Andolfi, and R. Nicoletti, (2020). “The Thin Line between
1233 Pathogenicity and Endophytism: The Case of *Lasiodiplodia Theobromae*”. In:
1234 *Agriculture* 10.10 (10), p. 488.
- 1235 Sarkate, R. S., N. V. Kalyankar, and P. B. Khanale, (2013). “Application of Com-
1236 puter Vision and Color Image Segmentation for Yield Prediction Precision”.

- 1237 In: *2013 International Conference on Information Systems and Computer Net-*
1238 *works*. 2013 International Conference on Information Systems and Computer
1239 Networks, pp. 9–13.
- 1240 Schaad, N. W. and R. D. Frederick, (2002). “Real-Time PCR and Its Application
1241 for Rapid Plant Disease Diagnostics”. In: *Canadian Journal of Plant Pathology*
1242 24.3, pp. 250–258.
- 1243 Schmidt, A. and Z. Bandar, (2001). “Modularity - A Concept For New Neural Net-
1244 work Architectures”. In.
- 1245 Schölkopf, B., F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y.
1246 Bengio, (2021). “Toward Causal Representation Learning”. In: *Proceedings of*
1247 *the IEEE* 109.5, pp. 612–634.
- 1248 Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra,
1249 (2017). “Grad-CAM: Visual Explanations From Deep Networks via Gradient-
1250 Based Localization”. In: Proceedings of the IEEE International Conference on
1251 Computer Vision, pp. 618–626.
- 1252 Shir, O. M., (2012). “Niching in Evolutionary Algorithms”. In: *Handbook of Natu-*
1253 *ral Computing*. Ed. by G. Rozenberg, T. Bäck, and J. N. Kok. Berlin, Heidel-
1254 berg: Springer, pp. 1035–1069.
- 1255 Shrikumar, A., P. Greenside, and A. Kundaje, (2017). “Learning Important Fea-
1256 tures Through Propagating Activation Differences”. In: *Proceedings of the 34th*
1257 *International Conference on Machine Learning*. International Conference on
1258 Machine Learning. PMLR, pp. 3145–3153.

- 1259 Singh, D., N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, (2020). “Plant-
1260 Doc: A Dataset for Visual Plant Disease Detection”. In: *Proceedings of the 7th*
1261 *ACM IKDD CoDS and 25th COMAD*. CoDS COMAD 2020. New York, NY,
1262 USA: Association for Computing Machinery, pp. 249–253.
- 1263 Solodskikh, K., A. Kurbanov, R. Aydarkhanov, I. Zhelavskaya, Y. Parfenov, D.
1264 Song, and S. Lefkimmiatis, (2023). “Integral Neural Networks”. In: Proceedings
1265 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
1266 pp. 16113–16122.
- 1267 Statista, (2022). *Smartphone Users 2026*. Statista. URL: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> (visited
1268 on 05/16/2022).
- 1269 Steddom, K., M. McMullen, B. Schatz, and C. M. Rush, (2005). “Comparing Im-
1270 age Format and Resolution for Assessment of Foliar Diseases of Wheat”. In:
1271 *Plant Health Progress* 6.1, p. 11.
- 1272 Su, J., C. Liu, M. Coombes, X. Hu, C. Wang, X. Xu, Q. Li, L. Guo, and W.-H.
1273 Chen, (2018). “Wheat Yellow Rust Monitoring by Learning from Multispec-
1274 tral UAV Aerial Imagery”. In: *Computers and Electronics in Agriculture* 155,
1275 pp. 157–166.
- 1276 Sun, J., X. Wang, N. Xiong, and J. Shao, (2018). “Learning Sparse Representation
1277 With Variational Auto-Encoder for Anomaly Detection”. In: *IEEE Access* 6,
1278 pp. 33353–33361.
- 1279

- 1280 Sun, Y., B. Xue, M. Zhang, and G. G. Yen, (2020). “Evolving Deep Convolutional
1281 Neural Networks for Image Classification”. In: *IEEE Transactions on Evolu-*
1282 *tionary Computation* 24.2, pp. 394–407.
- 1283 Tan, D. S., R. N. Leong, A. F. Laguna, C. A. Ngo, A. Lao, D. M. Amalin, and
1284 D. G. Alvindia, (2018). “AuToDiDAC: Automated Tool for Disease Detection
1285 and Assessment for Cacao Black Pod Rot”. In: *Crop Protection* 103, pp. 98–
1286 102.
- 1287 Tan, M. and Q. V. Le, (2020). *EfficientNet: Rethinking Model Scaling for Convolu-*
1288 *tional Neural Networks*. URL: <http://arxiv.org/abs/1905.11946> (visited on
1289 06/21/2023). preprint.
- 1290 Thapa, R., K. Zhang, N. Snavely, S. Belongie, and A. Khan, (2020). “The Plant
1291 Pathology Challenge 2020 Data Set to Classify Foliar Disease of Apples”. In:
1292 *Applications in Plant Sciences* 8.9, e11390.
- 1293 Tian, H., T. Wang, Y. Liu, X. Qiao, and Y. Li, (2020). “Computer Vision Tech-
1294 nology in Agricultural Automation —A Review”. In: *Information Processing in*
1295 *Agriculture* 7.1, pp. 1–19.
- 1296 Toda, Y. and F. Okura, (2019). “How Convolutional Neural Networks Diagnose
1297 Plant Disease”. In: *Plant Phenomics* 2019.
- 1298 *Torchvision Main Documentation* (2023). URL: [https://pytorch.org/vision/](https://pytorch.org/vision/main/models)
1299 [main/models](https://pytorch.org/vision/main/models) (visited on 06/21/2023).
- 1300 Tripathi, M. K. and D. D. Maktedar, (2020). “A Role of Computer Vision in Fruits
1301 and Vegetables among Various Horticulture Products of Agriculture Fields: A
1302 Survey”. In: *Information Processing in Agriculture* 7.2, pp. 183–203.

- 1303 UN, (2022). *United Nations*. World Population Prospects 2022. URL: [https://](https://population.un.org/wpp/Graphs/Probabilistic/POP/TOT/900)
1304 population.un.org/wpp/Graphs/Probabilistic/POP/TOT/900 (visited on
1305 05/18/2022).
- 1306 Vahdat, A. and J. Kautz, (2021). *NVAE: A Deep Hierarchical Variational Autoen-*
1307 *coder*. URL: <http://arxiv.org/abs/2007.03898> (visited on 07/12/2022).
1308 preprint.
- 1309 Vouloudimos, A., N. Doulamis, A. Doulamis, and E. Protopapadakis, (2018). “Deep
1310 Learning for Computer Vision: A Brief Review”. In: *Computational Intelligence*
1311 *and Neuroscience* 2018, e7068349.
- 1312 Vuola, A. O., S. U. Akram, and J. Kannala, (2019). “Mask-RCNN and U-Net En-
1313 sembled for Nuclei Segmentation”. In: *2019 IEEE 16th International Sympo-*
1314 *sium on Biomedical Imaging (ISBI 2019)*. 2019 IEEE 16th International Sym-
1315 posium on Biomedical Imaging (ISBI 2019), pp. 208–212.
- 1316 Wang, L., Z. Wu, S. Karanam, K.-C. Peng, and R. Vikram Singh, (2018). *Reducing*
1317 *Visual Confusion with Discriminative Attention*.
- 1318 Weinstein, B. G., (2018). “A Computer Vision for Animal Ecology”. In: *Journal of*
1319 *Animal Ecology* 87.3, pp. 533–545.
- 1320 Wu, Y. and K. He, (2018). “Group Normalization”. In: p. 17.
- 1321 Wu, Z., Y. Chen, B. Zhao, X. Kang, and Y. Ding, (2021). “Review of Weed Detec-
1322 tion Methods Based on Computer Vision”. In: *Sensors* 21.11 (11), p. 3647.
- 1323 Xiao, H., K. Rasul, and R. Vollgraf, (2017). *Fashion-MNIST: A Novel Image Dataset*
1324 *for Benchmarking Machine Learning Algorithms*. URL: [http://arxiv.org/](http://arxiv.org/abs/1708.07747)
1325 [abs/1708.07747](http://arxiv.org/abs/1708.07747) (visited on 06/23/2023). preprint.

- 1326 Xie, S., R. Girshick, P. Dollár, Z. Tu, and K. He, (2017). *Aggregated Residual Trans-*
1327 *formations for Deep Neural Networks*. URL: [http://arxiv.org/abs/1611.](http://arxiv.org/abs/1611.05431)
1328 [05431](http://arxiv.org/abs/1611.05431) (visited on 06/23/2023). preprint.
- 1329 Xu, S., J. Wang, W. Shou, T. Ngo, A.-M. Sadick, and X. Wang, (2021). “Com-
1330 puter Vision Techniques in Construction: A Critical Review”. In: *Archives of*
1331 *Computational Methods in Engineering* 28.5, pp. 3383–3397.
- 1332 Xu, W., H. Sun, C. Deng, and Y. Tan, (2017). “Variational Autoencoder for Semi-
1333 Supervised Text Classification”. In: *Thirty-First AAAI Conference on Artificial*
1334 *Intelligence*. Thirty-First AAAI Conference on Artificial Intelligence.
- 1335 Xu, W. and Y. Tan, (2020). “Semisupervised Text Classification by Variational
1336 Autoencoder”. In: *IEEE Transactions on Neural Networks and Learning Sys-*
1337 *tems* 31.1, pp. 295–308.
- 1338 Yang, X. and R. P. McCord, (2021). “CoSTA: Unsupervised Convolutional Neural
1339 Network Learning for Spatial Transcriptomics Analysis”. In: *bioRxiv*, p. 26.
- 1340 Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, (2014). “How Transferable Are
1341 Features in Deep Neural Networks?” In: *Advances in Neural Information Pro-*
1342 *cessing Systems*. Vol. 27. Curran Associates, Inc.
- 1343 Yu, K.-H., F. Wang, G. J. Berry, C. Ré, R. B. Altman, M. Snyder, and I. S. Ko-
1344 hane, (2020). “Classifying Non-Small Cell Lung Cancer Types and Transcrip-
1345 tomic Subtypes Using Convolutional Neural Networks”. In: *Journal of the Amer-*
1346 *ican Medical Informatics Association* 27.5, pp. 757–769.

- 1347 Yue, J., W. Zhao, S. Mao, and H. Liu, (2015). “Spectral–Spatial Classification of
1348 Hyperspectral Images Using Deep Convolutional Neural Networks”. In: *Remote*
1349 *Sensing Letters* 6.6, pp. 468–477.
- 1350 Zeiler, M. D. and R. Fergus, (2014). “Visualizing and Understanding Convolutional
1351 Networks”. In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B.
1352 Schiele, and T. Tuytelaars. Lecture Notes in Computer Science. Cham: Springer
1353 International Publishing, pp. 818–833.
- 1354 Zhang, X., X. Zhou, M. Lin, and J. Sun, (2018). “ShuffleNet: An Extremely Effi-
1355 cient Convolutional Neural Network for Mobile Devices”. In: Proceedings of the
1356 IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856.
- 1357 Zhao, T., Y. Yang, H. Niu, D. Wang, and Y. Chen, (2018). “Comparing U-Net
1358 Convolutional Network with Mask R-CNN in the Performances of Pomegranate
1359 Tree Canopy Segmentation”. In: *Multispectral, Hyperspectral, and Ultraspectral*
1360 *Remote Sensing Technology, Techniques and Applications VII*. Multispectral,
1361 Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and
1362 Applications VII. Vol. 10780. SPIE, pp. 210–218.
- 1363 Zhu, X., W. Su, L. Lu, B. Li, X. Wang, and J. Dai, (2021). *Deformable DETR:*
1364 *Deformable Transformers for End-to-End Object Detection*. URL: [http://](http://arxiv.org/abs/2010.04159)
1365 arxiv.org/abs/2010.04159 (visited on 01/09/2023). preprint.

1366 **TABLES AND FIGURES**

Table 1: Test results of three architectures trained on two datasets to show an indirect comparison. ResNet18 was trained only on ImageNet with the top one and top five classification accuracy's shown. EVOCNN was trained only on Fashion MNIST with the % error shown. VGG16 was trained on both datasets. Results were taken from Sun et al. (2020) and *Torchvision Main Documentation* (2023). *Number of parameters for VGG16 was miss-reported by Sun et al. (2020) as 26 million.

Architecture	Top 1 acc.	Top 5 acc.	Error (%)	n parameters
ResNet18	69.758	89.078	-	11.7M
VGG16	71.59	90.38	13.78	138M*
EVOCNN	-	-	7.28	6.52M

Table 2: Pros and cons of popular model architectures for image classification, object detection and semantic segmentation.

(a) Image classification architectures. Ranges of values represent the smallest and largest off-the-shelf versions available. Values for number of trainable parameters (displayed in millions (M)) and giga floating point operations (GFLOPS) were obtained from the pytorch documentation (*Torchvision Main Documentation 2023*)

Image classification			
Arch.	n param.	GFLOPS	Pros & Cons
ResNet (2015)	12M-60M	1.8-11.5	<p>Pros</p> <p>ResNet18 is the smallest and most computationally efficient model here ResNet18 is ideal for modestly sized datasets ResNet152 performs comparably with transformers like VIT Widely used and tested</p> <p>Cons</p> <p>Uses batch normalisation which can introduce instability and inconsistent results</p>
EfficientNet (V2) (2019)	22M-119M	8.4-56.1	<p>Pros</p> <p>Allows the depth, width and resolution of the model to be scaled with a single coefficient</p> <p>Cons</p> <p>Scaling requires editing the source code Evaluation using GradCam showed much overfitting, despite high test scores</p>
ConvNeXT (2022)	29M-198M	4.5-34.3	<p>Pros</p> <p>Reported to outperform any architecture here and requires much less data than VIT Is scaled easily by editing the convolutional block settings Incorporates several modern features like GELU, stochastic depth and layer norm</p> <p>Cons</p> <p>The smallest off-the-shelf configurations are too large for many projects and may overfit Potential compatibility issues with conversion to ONNX format</p>
VIT (2021)	87M-634M	17.6-1,016	<p>Pros</p> <p>If trained on millions of images, VIT may slightly outperform ResNet152</p> <p>Cons</p> <p>Requires huge datasets to outperform CNNs Computationally expensive to train and run at inference</p>

Table 2: Pros and cons of popular model architectures for image classification, object detection and semantic segmentation. (continued)

(b) Object detection and semantic segmentation architectures. Ranges of values represent the smallest and largest off-the-shelf versions available. Values Faster-RCNN and Mask-RCNN were obtained from the pytorch documentation (*Torchvision Main Documentation* 2023), but values for number of trainable parameters (displayed in millions (M)) and giga floating point operations (GFLOPS) for YOLO and DETR were calculated for this comparison with an image size of 224x224 pixels.

Object detection & Semantic segmentation			
Arch.	n param.	GFLOPS	Pros & Cons
Faster-RCNN (2015)	44M	280.4	<p>Pros</p> <p>Generally gives higher ‘Mean Average Precision’ than YOLO Performs better than YOLO on small objects Does poorly when objects overlap</p> <p>Cons</p> <p>More computationally expensive than YOLO</p>
Mask-RCNN (2017)	46M	333.6	
YOLO (2016)	7M	1.01	<p>Pros</p> <p>Extremely fast at inference time Fast to train Very easy to implement</p> <p>Cons</p> <p>Performs poorly on small objects Gives least accurate results of the three architectures listed here</p>
DETR (2021)	40M	11.2	<p>Pros</p> <p>Negates the need for region proposal and non-max suppression Performs better than Faster-RCNN and YOLO for overlapping objects As opposed to classification, transformers like DETR show promise in object detection Faster at inference than Faster-RCNN</p> <p>Cons</p> <p>Very computationally expensive to train Slow to converge in training Requires huge amount of training data Can be challenging to implement Requires a large batch size to achieve stable training</p>

Table 3: Results of an ablation study to assess the effects of image normalisation and batch normalisation on a model’s ability to detect plant disease

Image Norm	Batch Norm	Layer Norm	Train time (m)	Loss (%)	Acc (%)	Recall (%)	Precision (%)	F1 (%)
ConvNeXt Tiny								
No	-	Yes	1,344	0.290	88.25	88.25	88.82	88.14
Yes	-	Yes	1,368	0.322	84.51	87.51	88.14	87.38
ResNet18								
No	Yes	-	739	0.361	85.41	85.41	86.17	85.3
Yes	Yes	-	1,088	0.380	85.14	85.18	85.68	84.96
No	No	-	1,764	0.412	83.49	83.49	84.05	83.38

Table 4: Specifications and use cases for the hyperspectral cameras used in the following studies (Okamoto *et al.*, 2007; Feng *et al.*, 2018; Gutiérrez *et al.*, 2019; Pan *et al.*, 2019; Li *et al.*, 2020; Nguyen *et al.*, 2020).

Make/model	Task	Spectral range (nm)	Spectral bands	Spectral resolution (nm)
Resonon Pika II Vis-NIR	Mango tree yield estimation	390-890	244	2
Headwall Nano-Hyperspec w/ pushbroom	Potato yield estimation	400-1000	272	6
ImSpector N17E	Maize kernel vigour assessment	874-1734	NA	5
ImSpector V10	Weed identification	400-1000	240	10
OCI-UAV-1000 w/ pushbroom	Nutrient assessment in rice	460-983	116	5
ImSpector V10E	Disease monitoring in pears	328-1115	1002	2.8

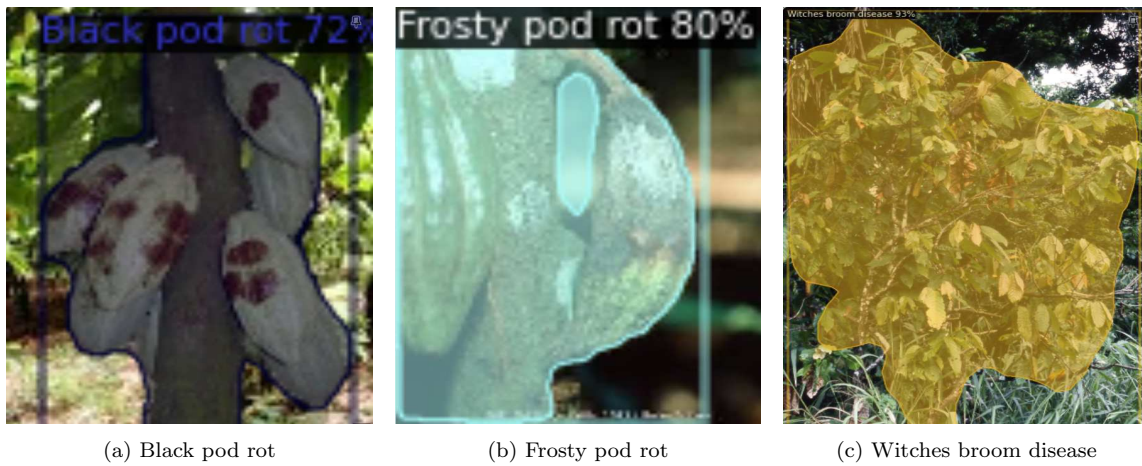


Figure 1: Application of semantic segmentation with Mask-RCNN to highlight whole trees infected with (a) black pod rot, (b) frosty pod rot and (c) witches broom disease. The percentage scores show the degree of confidence in the model's diagnosis.



Figure 2: (a) Original and (b) normalised images of a cocoa pods showing various stages of disease development. Note the affect of normalisation on ones ability to see disease symptoms. Normalisation of pixel values was carried out with the following means and variance values: mean: (0.485, 0.456, 0.406) variance (0.229, 0.224, 0.225)

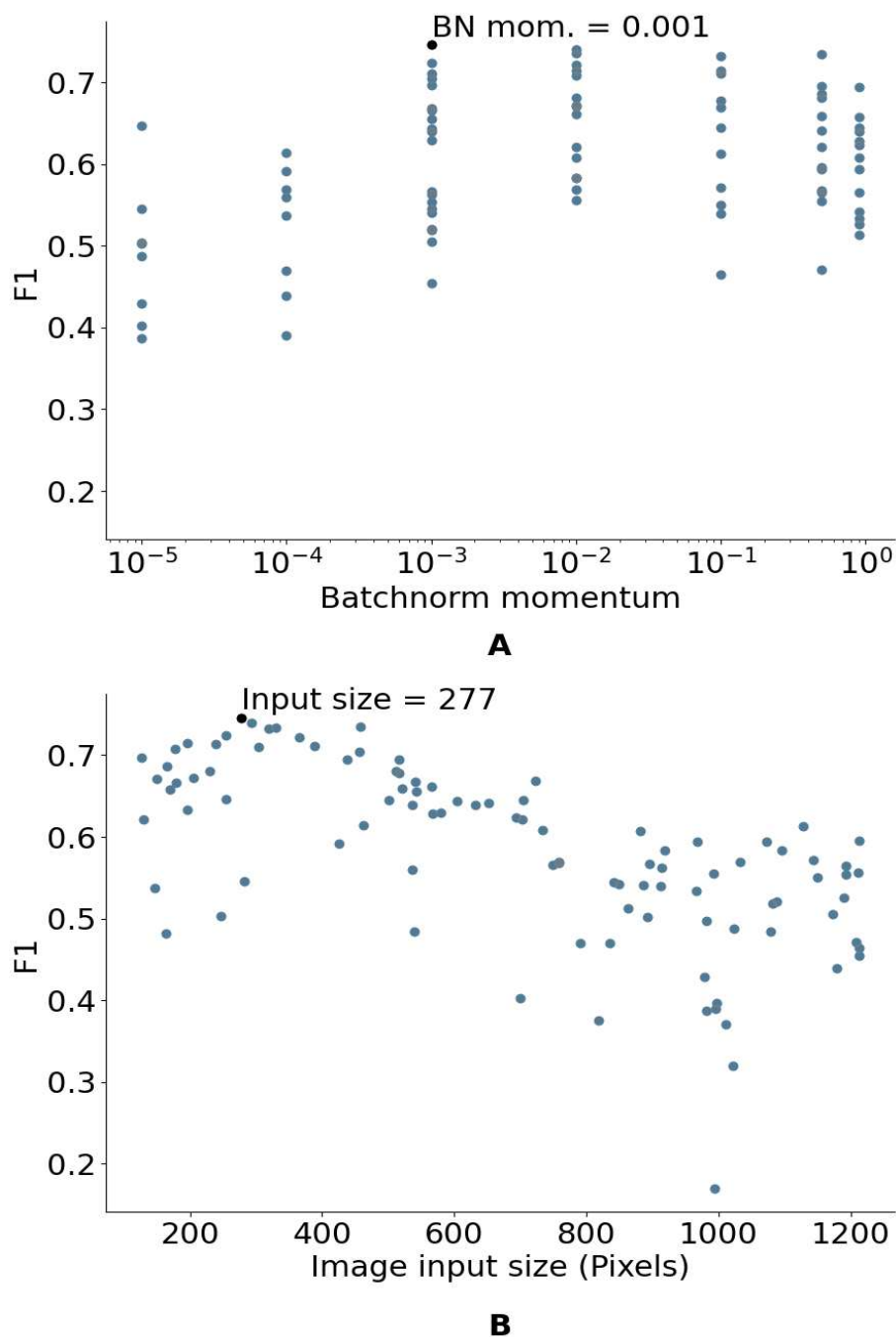


Figure 3: Results of a hyperparameter optimisation sweep training 100 ResNet18 models for disease detection in cocoa trees with variable Batchnorm momentum (A) and square image input size (B). The optimisation sweep randomly sampled from distributions of the two variables concurrently. Beginning with the ImageNet1KV2 weights, the models were trained on a dataset of 1065 images of the following four classes. Black pod rot [271], Frosty pod rot [266], Witches broom disease [92] and Healthy cocoa [436]. The optimised validation F1 score was 0.75.