Expert Systems With Applications 298 (2026) 129778

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Generating realistic pruning solutions for automated grape vine pruning using graph neural networks

Jaco Fourie [a,*], Jeffrey Hsiao [a,1], Oliver Batchelor [b,1], Kevin Langbroek [b,1], Henry Williams [c,1], Richard Green [b], Armin Werner [a]

[a] Lincoln Agritech Ltd, Engineering Drive, Lincoln, 7674, New Zealand
[b] University of Canterbury, 20 Kirkwood Avenue, Christchurch, 8041, New Zealand
[c] University of Auckland, 34 Princes Street, Auckland, 1010, New Zealand

ARTICLE INFO

Keywords:
Vine pruning
Neural network
Graph neural network
Generative AI
Synthetic vines

ABSTRACT

In our prior work we showed that graph neural networks (GNNs) can be trained to generate pruning solutions that could direct robotic pruning robots to perform automated cane pruning of wine grape vines. That study introduced the feasibility of the technology but also showed that there were many open questions and issues with the research results that needed to be addressed. In this study we address some of these questions. For example, we answer the question of how would a model like this perform on real vine architectures compared with pruning solutions from real experienced pruners. Our most notable contributions include moving away from a per-cane classification model that attempts to define a single *perfect* pruning solution, to a model that ranks multiple good solutions and picks the best one. We addressed a key limitation of the previous training data by moving away from synthetic vine architectures to realistic ones recorded from real vines and using pruning solutions collected by expert pruners as our ground-truth. Our primary goal was to show that learning by example using a GNN-based model was a viable approach to automated pruning, even when compared with experienced pruners. We showed robust performance from our model by training on a dataset of 90 pruning solutions generated by expert pruners in the 2022 season, and testing our performance on 117 pruning solutions from an independent set of pruners from the 2021 season. The model was able to correctly score all the pruning solutions from the 2021 dataset as *good* to *very good* and none of the expert solutions were classified as *poor*.

1. Introduction

Cane pruning is a technique whereby the pruner selects a number of canes (usually 2 or 4) from the season's new growth and trains them to a trellis to serve as the bearer canes for the next season's growth. All other spurs and older canes are pruned off. It is a laborious task that must be completed every season by skilled workers and poor pruning can have a very detrimental impact to the yield and quality of the grape harvest. There is therefore a potential for large economic gains if this task can even be partially automated. It is worth highlighting that we are concerned here with cane pruning and not the easier problem that has already been successfully automated known as spur pruning.

This study is part of a larger project that aims to develop a fully automated pruning robot that is able to measure the vine canopy architecture using various optical sensors and then direct robot arms to prune the correct canes leaving only the canes that should be reserved as spurs for the following season and the new bearer canes that will then be tied to a trellis in a later step. Our work only focuses on the problem of how to teach a robotic system to pick the correct canes in a vine architecture for pruning, based only on the measurements of the vine architecture and the example pruning solutions it was trained on.

Our primary contribution and objective in this paper was to address the main limitations of our prior work. In particular, we replaced relying on synthetic data for training and used realistic vine architectures from physical vine measurements. We also used pruning solutions provided by experienced vine pruners instead of relying on simple rule-based labels. Results from this improved dataset led to the discovery of further limitations in our original approach that was also consequently addressed. For details see Sections 4 and 5.

* Corresponding author.
E-mail addresses: jaco.fourie@lincolnagritech.co.nz (J. Fourie), jeffrey.hsiao@lincolnagritech.co.nz (J. Hsiao), oliver.batchelor@canterbury.ac.nz (O. Batchelor), kla101@uclive.ac.nz (K. Langbroek), henry.williams@auckland.ac.nz (H. Williams), richard.green@canterbury.ac.nz (R. Green), armin.werner@lincolnagritech.co.nz (A. Werner).
[1] Equally contributing authors

https://doi.org/10.1016/j.eswa.2025.129778
Received 17 June 2025; Received in revised form 31 August 2025; Accepted 16 September 2025
Available online 22 September 2025

2. Related work

Because of the large effect automated cane pruning could have on the productivity of the wine grape industry (Archer & Fouche, 2017; Bramley et al., 2011; Greven et al., 2014), many researchers have contributed towards the development of a fully automated system (Botterill et al., 2017; Corbett-Davies et al., 2012; Gittoes et al., 2011; Paulin et al., 2015). This is also true of our colleagues working on the larger project that our study is part of (Epee et al., 2022; Williams et al., 2023). In this study we focus only on how a vine canopy can be described and analysed in a way that descriptive features can be extracted and used to train a model for automated generation of pruning solutions.

In our prior research (Fourie et al., 2021) we showed that a graph neural network (GNN) can be trained to classify individual canes within a vine canopy to determine whether they should be pruned or kept as renewal spurs or new bearer canes. This was in contrast to the previous approaches that either relied on a combination of simple pruning rules or the classification of canes in isolation. It has been shown in the study by Corbett-Davies et al. (2012) that human pruners do not evaluate individual canes in isolation and that good pruning solutions require one to take all the canes and their relationship to each other into account. In other words, the cane architecture plays an important role in the pruning solution and any algorithm that aims to generate them has to encode this cane architecture as part of the input.

An algorithm that is based on a collection of pruning rules and attempts to search through a space of possible solutions for one that maximally agrees with the rules is also problematic. This is because expert pruners often cannot explain their decisions in terms of a combination of simple rules and the best pruners have noted that deciding which canes to prune and which to keep comes with experience over several seasons and is specific to a particular vineyard. It is therefore very difficult to construct a realistic objective function based on pruning rules that can be optimised to find a good pruning solution.

In our prior work we took the approach that an accurate algorithm needs to have two key features.

1. Firstly, it has to base its solution on not only the features of individual canes but also needs to **capture the total architecture** of the vine and the relationships between neighbouring canes.
2. Secondly, good solutions should be **based on examples** from real expert pruners and should not be based on pruning rules.

Due to a lack of realistic examples of unpruned vines and expert pruners that can provide good pruning solutions, we were forced to test our ideas on synthetic data and pruning solutions based on a combination of simple pruning rules. However, we showed that vine canopy architecture can effectively be captured by representing the vine as a mathematical graph with each node representing a cane and the relationship between canes encoded in the edges between nodes. We also showed that a graph neural network (GNN) based on the gated attention (GAT) layer can then be trained to interpret the vine graph and generate a realistic pruning solution based on the total architecture of the vine. It also does not rely on pruning rules but learns from example pruning solutions that are ideally provided by expert pruners.

Even though our results were encouraging, we identified several limitations of our research,not least of which was the lack of realistic vine examples and the input from expert pruners to build a database of good pruning solutions that can be used to train our model. In this paper we address the lack of good training data but after analysing the results from training our original model on realistic data we also found further serious limitations that we will address and discuss in this paper.

In Section 3 we will review the VineGraph model and the concept of cane pruning as a classification problem. This will not be a detailed explanation of the model and we refer the interested reader to our prior work for more detail (Fourie et al., 2021). In Section 4 we look at the limitations that were previously identified in our previous paper (and were addressed here) as well as further limitations that were only realised once we started working with a realistic dataset. We then look at the data collection and pre-processing that went into building our new dataset of realistic vines and pruning solutions in Section 7. In Section 8 the results from our current new model will be presented and we will end this paper with our conclusion and plans for future work in Section 9.

3. The VineGraph model

The inspiration of the VineGraph model came from the realisation that vine canopy structures can be effectively and succinctly described as a mathematical graph and that graph neural networks can therefore be used to extract classifications, quality metrics, or other features from them. In our case we represent the individual canes of our vine as nodes in the graph and each cane is described by an associated feature vector. Each feature vector contains seven features that pruners have identified as important to deciding whether a cane should be pruned, kept as a renewal spur, or tied down to the trellis as a new bearer cane. These features are described in Table 1.

We encode the relationship between canes as edges in our graph and found that the best way to describe the strength or intensity of the relationship that neighbouring canes have, is by setting the edge weight proportional to the euclidean distance between cane origins. We can now visualise a vine structure plus a pruning solution in an abstract but succinct way by drawing the graph as a series nodes (represented by coloured balls) on a coordinate frame centred on the vine trunk and lowest trellis wire. The relative distances between cane origins are preserved in the diagram and graph edges can be shown as connecting lines between neighbouring nodes. We give an example of this in Fig. 1.

With this description of the vine canopy structure as well as individual cane features, we have everything we need to build a model that

Table 1
The features used by the VineGraph model to describe a cane. These seven features form the feature vector associated with each node in our graph.

feature	description
`horizontal_dist`	The horizontal distance in mm from the cane origin to the centre of the vine trunk. Left of the trunk are negative values and right of trunk positive ones.
`vertical_dist`	The vertical distance in mm from the cane origin to the lowest trellis wire. Positive values represent origins below the wire and negative values originate above the wire.
`cane_diam`	The diameter of the cane measured after the second node in mm.
`cane_length`	The length of the cane. This was originally a number in mm but it was later realised that a boolean value of 1 or 0 that describes whether the cane is long enough to be viable is closer to what a real pruner would use.
`node_count`	The number of nodes counted from the origin to the end of the cane.
`healthy`	A boolean value of 1 or 0 to indicate whether a cane looks healthy or whether it has obvious defects.
`orientation`	The overall direction (left or right of the trunk) in which the cane points. This is not the same as vert_dist as a cane may originate on the left side of the trunk and curve towards the right or vice versa.

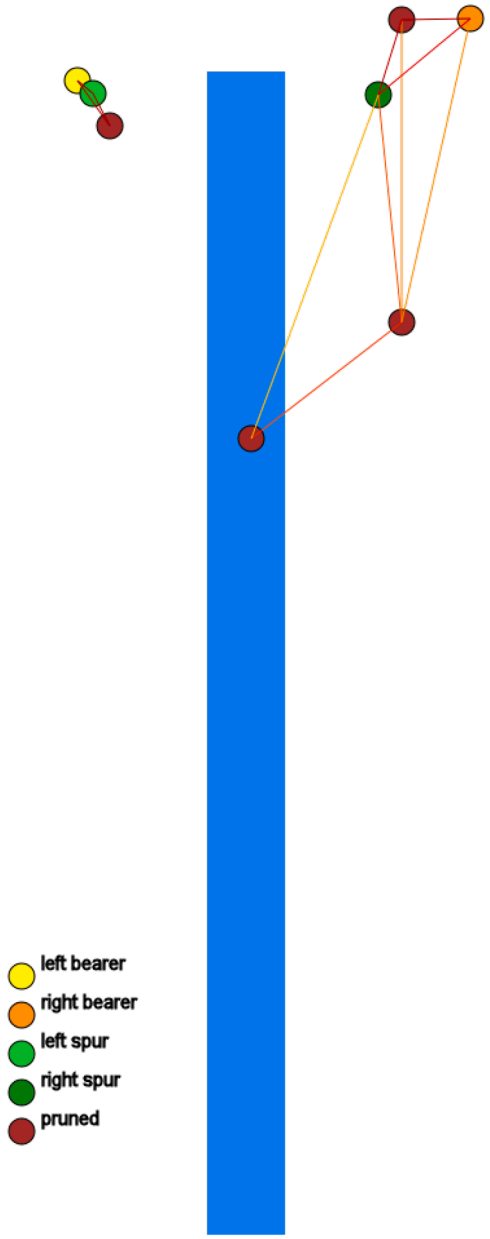

Fig. 1. An example of a vine canopy represented as a graph. The nodes (canes) shown as coloured balls encode a potential pruning solution. This vine has 8 canes with three on the left and five on the right of the trunk. Lines between nodes are coloured to indicate the weight of the edge which in this case is proportional to the distance between canes. To keep the diagram tidy we only draw the strongest edges. The vine trunk is represented by the blue rectangle but is not drawn to scale and only serves to centre the diagram. The thin line above the trunk represents the lowest trellis wire.

classifies canes to describe a pruning solution and is able to take the canopy architecture into account.

3.1. The classification problem

If we start from the assumption that human pruners, once they've analysed the canopy structure as a whole, pick their bearer canes and renewal spurs sequentially based on cane features, then the following strategy seems reasonable. Train a multi-class classifier model that takes cane features and the global graph architecture as input and produces a class probabilities vector that indicates the most likely classification for each cane. This idea has been explored by previous researchers, notably in the work by Botterill et al. (2017) and then later in our own prior work (see Fourie et al. (2021)).

As our data is described as a graph, we used a graph neural network (GNN) as the basis of our model. While we achieved some success using this approach (see Fourie et al. (2021)), it's important to note that because canes are classified independently of each other post-processing of the solutions were required to filter out impossible solutions or those that contradicted basic pruning objectives. For example, in a two-cane system a basic objective is that two new bearer canes have to be chosen, one on the left side of the vine head and one on the right. If more than two bearer canes are picked by the model or multiple on the same side of the head, then the post-processing should filter out superfluous canes and keep only the most likely candidates.

Human pruners do not reason like this and if multiple good cane candidates exist in the canopy they will pick the ones that are easiest or least risky (as they could break when bent) to tie down. Individual cane classification is therefore a practical way to generate pruning solutions but it has fundamental limitations that we will elaborate on in Section 4. Instead we briefly describe the way that GNNs were used to build the VineGraph model with further details available in Fourie et al. (2021).

3.2. Graph neural networks to describe vine architecture

As we've already demonstrated in Fig. 1, we parameterise both the features of individual canes and the relationships between them using an undirected graph. Scarselli et al. showed that GNNs can be trained to classify the nodes of such graphs based on the node features and the adjacency matrix that describes the edges between nodes (Scarselli et al., 2009). This is the approach that the VineGraph model uses as we choose our nodes to represent the canes of the vine. Node features are as described in Table 1 with the exception of the last two features (health, orientation) as these two were added in later research to improve results and address limitations. They were not included in the original VineGraph model.

Our choice to use euclidean distance between cane origins to encode the edge strengths between our nodes is more a practical choice rather than based on realistic pruning practice. It is known that the distance between canes play an important role in the decision-making process but the relationship between canes is more complex and nuanced than just distance between them. However, collecting this information from expert pruners for multiple vines in a consistent objective way is very challenging while calculating the distance based on measured vertical and horizontal distances is not. We elaborate on our data collection approach from real vines and human expert pruners in Section 7.

3.3. Graph attention networks (GAT)

The attention mechanism greatly improved the state-of-the-art in the field of natural language processing and was later also adapted to other applications (Devlin et al., 2018; Vaswani et al., 2017). By building on this work (Veličković et al., 2018) developed the graph attention network (GAT) that uses the attention mechanism to build a network that operates on graph-structured data.

The GAT layer can be stacked to form multi-layer networks and can be configured for various graph sizes and learning capacities. This highly-configurable feature and the ability for GAT layers to generalise to unseen nodes in unknown parts of the graph inspired us to use a stacked GAT layer architecture for VineGraph. More details on the GAT layer and how we used it to develop the VineGraph model can be found in Fourie et al. (2021) but the key reasoning was that each GAT layer's attention mechanism is used to encode the effect that features from neighbouring canes have on the classification of a cane.

4. Limitations of VineGraph

Our prior work was pioneering but had several limitations that prevents realistic implementation of a model to generate pruning solutions. Much of this was already identified and discussed in Fourie et al. (2021) but as we addressed some of the previously identified limitations and further developed on more realistic datasets, we discovered a more fundamental limitation that required a redesign of VineGAT and a major change in how we train the model.

4.1. The limitations of cane classification

We previously mentioned that due to multiple canes possibly getting classified as bearers on the same side of the vine head, that post-processing of the pruning solution was required to make valid pruning solutions of the results from the VineGraph model. This highlights a key limitation: even though neighbouring canes are considered when classifying any given cane (by using the attention mechanism), once the classification has been made this choice does not affect the classification of other canes. This leads to multiple conflicting choices and the need for restricting the output by using simple rules determined by the pruning scheme, for example, a two-cane pruning system must have two bearer canes with one on the left of the vine head and one on the right.

Even with a graph that describes the vine architecture, the problem is that classification of each cane constrains the application of global constraints on the pruning solution. In other words, cane classification should be a sequential operation where the classification of one cane has a strong effect on the classification of subsequent canes. This limitation was identified in our prior research and was partially addressed by adding recurrent layers to the network in an attempt to add a sequential component to the decision making, but the core problem remained. See Fourie et al. (2021) for more details on how this was done and the results from adding these recurrent layers.

A better way to address this limitations would be to describe a pruning solution without relying on each individual cane classification but rather a strategy that only classifies the non-pruned canes and ignores the remaining canes once the bearers and renewal spurs have been identified. This is closer to what human pruners would do as they would also focus on the choice of which canes to keep and then ignore the remaining canes that will be pruned.

We experimented with a model that was similar to VineGraph but instead of generating a classification for each cane it generated a much smaller output that only specified which cane from the vine was chosen for each of the non-pruned classes. For example, in a pruning system where two bearer canes and two renewal spurs are required the output vector would only contain 4 values with each value being a cane index associated with a cane in the vine. This greatly simplified the model and eliminated the need for enforcing a pruning scheme with post-processing of the result, but we were not able to train an accurate model that was able to consistently generate good solutions. Instead, this model led to a further insight that required a more fundamental redesign of the model.

4.2. No "correct" pruning solutions

In our adapted VineGraph model (see previous section) we noticed that our model was able to generate good pruning solutions that seemed to follow the unstated guidelines of what constitutes a good pruning solution. However, the model uses a categorical cross-entropy loss function that compares the model output to the label and penalises any solution that is not equal to the label. The model is therefore trained to find the exact same pruning solution as was provided by the expert human pruner. All other solutions are considered to be equally incorrect and the accuracy of a trained model was scored by counting how many vines it could prune to be exactly the same as the human label. However, this is not realistic.

In our conversation with human expert pruners we realised that when senior pruners teach junior ones, multiple good pruning solutions can be produced by the junior pruners even though they are not precisely the same as the senior's example. Not only are there multiple good ways to prune a single vine, but pruning solutions are scored on a continuum rather that just being good or bad. There are subtleties in pruning so that certain canes should always be pruned and keeping one immediately makes the pruning solution a bad one, but there are also many alternate canes that could be substituted as the bearer cane without affecting the quality of the pruning much.

There is therefore no *correct* pruning solution, but rather only good and bad ones with multiple variations of *good* and *bad*. Our model should therefore be capable of using the *good* pruning examples from our expert human pruners to train itself to generate, similarly good, but not necessarily identical pruning solutions. It has to take into account both the cane features and the relationship that canes have with their neighbours and avoid merely classifying each cane independently without considering that pruning is a sequential operation.

An example pruning situation that illustrates these concepts is shown in Fig. 2. In that example we suppose that the pruner has identified two canes on the right side of the vine head that could be renewal spurs or new bearer canes and three similar candidates on the left (indicated with red and blue arrows respectively). All the other canes should be pruned off and are no longer considered as part of the decision making process. If any of these canes are not pruned the pruning may be considered poor but the key insight is that, for example, any of the left potential bearers could become the left bearer cane and the pruning would still be considered good for any of those options. In a two-cane pruning scheme two canes will be picked and 1–2 renewal spurs. This means at most four of the five remaining canes will be selected, two on the left and two on the right. The pruner may decide to pick the best two candidates as renewal spurs and then pick the remaining right cane to be the right bearer cane. This leaves two remaining options to become the left bearer cane and either one would lead to a good pruning but the remaining candidate has to be pruned off. This example illustrates the sequential nature of the decision-making process. Choosing a bearer cane on the left eliminates the possibility of any of the other good options becoming the bearer. Independently classifying a cane without considering previous decisions will clearly lead to poor choices.

The development of a model that can correctly deal with the previous example make it a very ambitious goal and, depending on how one chooses to implement the model features, may lead to mutually exclusive objectives. In Section 5 we introduce a new model that attempts to address the previously mentioned limitations by first focusing on the simpler problem of how we can evaluate the quality of a pruning solution.

5. The PruningGraph model

As we previously mentioned, it is unhelpful to think of pruning solutions as *correct* or *incorrect*. Rather, the multiple good solutions should be taken into account and the aim of our model should be to discover the features that make a pruning solution better than random alternatives and maximise these to generate a good solution. Conversely, those features that define an unacceptable or *bad* solution should also be identified and should be minimised when generating a solution.

A simpler alternative strategy would be to focus on the problem of scoring a given pruning solution on a continuum that can be thresholded to classify solutions as good enough or not acceptable. Given that we only have pruning solution data that defines good solutions as those provided by our expert human pruners, a *pruning scoring model* will also be easier to train. If we could then generate a collection of potential pruning solutions based on simple rules we can use our scoring model to rank the solutions and keep the best one as the *optimal* model output.

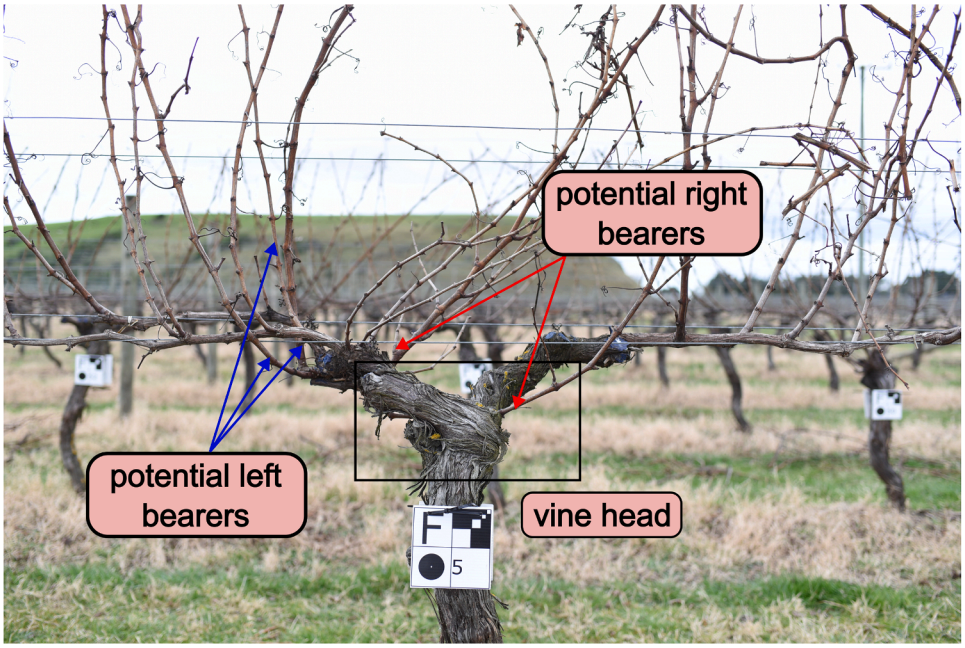

Fig. 2. An example pruning scenario that highlights the necessity of considering previous decisions when choosing canes and also illustrates that multiple good options may exist leading to multiple good pruning solutions.

5.1. The solution scoring network

In order to develop our solution scoring model a reasonable starting point would be to use the VineGraph model as a feature extractor and then add a *regression head* which is just a fully connected dense layer with a single output (also known as a perceptron). However, through experimentation we found that a much simpler model can still produce good results and has the advantage of faster inference. It is also not as prone to overfit during training on our limited dataset of expert pruning solutions.

Our experience with the VineGraph model showed that GNNs were very useful for capturing the vine architecture which we know will be critical to scoring the quality of a solution. We therefore based our new model on the same GAT layers that formed the core of the VineGraph model. As with VineGraph we define the vine using a graph with each cane in the vine represented as a node in the graph with an associated node feature vector. The seven features we use to describe a cane are those described in Table 1, which means that our feature matrix X is a $N \times 7$ matrix where N is the number of canes in the vine. Similarly, the adjacency matrix which describes the relationship between every node pair in the graph is a $N \times N$ matrix. Each row in the feature matrix describes the features of the cane associated with that row and each row in the adjacency matrix describes the strengths (or weights) of the relationship that each other node in the graph has with the node associated with that row.

These two matrices, the feature matrix and the adjacency matrix, serve as the input to our model which is fed directly to our first layer, a GAT layer. The output is then fed to two more GAT layers forming a 3-layer GAT feature extractor similar to that found in VineGraph (see Fourie et al. (2021) for further details). Instead of using the LSTM layers found in VineGraph we then feed the output of the GAT layers directly into a three-layer multi-layer-perceptron (MLP) which is much simpler than the LSTM layers and led to better results in our experimentation. Our final output is a single value which we send through a sigmoid[1] activation function to turn it into a value between 0 and 1 that represents the score of the pruning solution.

[1] See https://en.wikipedia.org/wiki/Sigmoid_function

In order to encode the candidate pruning solution as part of the input we add three extra feature values to the feature vector of each cane namely, `is_bearer`, `is_left`, and `is_pruned`. All three these values are boolean and classify each cane as either pruned off, chosen either on the left or to the right of the vine head, and chosen as either a new bearer cane or a new renewal spur. For example if all three these values are 0 (false) then the corresponding cane is chosen as a renewal spur on the right side of the head. With these three values added to each feature vector our feature matrix X becomes a $N \times 10$ matrix. We call this model the PruningGraph model and its architecture is illustrated in Fig. 3.

5.2. Training the model

As the output of our model is a single value we have at least two options as an appropriate loss function. We can use the mean-square-error loss between the label (which is always 1 for human expert pruning solutions) and the model output or, alternatively, we can use the binary cross-entropy loss. We experimented with both approaches and found the binary cross-entropy loss to produce slightly better results.

However, our primary challenge in training the model was in preparing the dateset. In Section 7 we will further discuss how our dataset was gathered, including how the vines were measured and the pruning solutions were collected, but here we focus on the pre-processing of the dataset. Unlike the synthetic dataset that we trained on in our prior work (see VineGraph Fourie et al. (2021)), our realistic dataset consists solely of human pruned vines which are all considered to be optimally pruned. All our labels are therefore equal to 1 (100 %) which means that any model trained on this dataset will be hopelessly biased towards good prunings. Our initial approach to resolving this was to add an equal number of bad prunings to the dataset to balance the labels between 0 and 1 values. We constructed the bad prunings by randomly substituting the canes kept by the human pruners with pruned canes far from head, knowing that these should never be chosen as bearer canes or renewal spurs.

Unfortunately, this approach failed to produce good results and upon further investigation and experimentation we realised that the reason for this is due to the model only having access to extreme values as labels. Since all the pruning examples were labelled as either 1 or 0 the

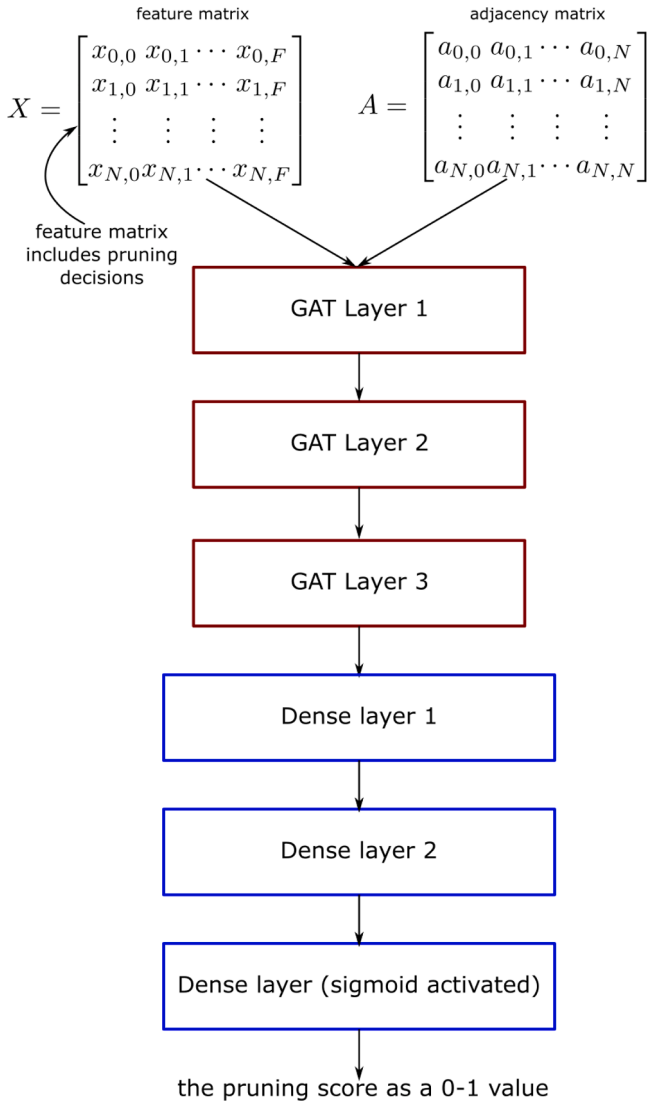

Fig. 3. The PruningGraph model architecture. The model input is shown at the top of the figure with the output from each layer feeding into the input of the next layer until the output value is produced at the bottom of the figure.

model had no access to examples that were partially correct, or poor prunings where some of the decisions were acceptable. With this dataset the model was never able to learn how to move from a random poor solution to a better one, or which features make the pruning solution better. The ideal solution to this problem is clearly to arrange for human experts to rate the quality of pruning solutions on a continuous scale so we can construct a balanced dataset with label values uniformly spread between 0 and 1. However this was not possible and we instead had to come up with a data augmentation strategy that would add the missing values to the dataset.

The challenge in augmenting this dataset with synthetic pruning solutions is how to define the quality levels between a completely random poor solution and the optimal good pruning from the human experts. Good pruners report that there are multiple good solutions to a pruning problem but that some choices are more optimal than others. The reason for this is often specific to the vineyard and is highly dependant on individual pruner experience with that vineyard. As our goal is to train a generic model, getting the singular optimal pruning solution specific to a particular vineyard is not helpful. Rather, we aim to find generically good solutions and avoid clearly poor ones. We therefore need to augment our dataset with generically sub-optimal solutions and combine it with a scoring scheme that measures how far a solution is from the optimal expert solution.

We experimented with several synthetic scoring schemes and found that a key consideration in generating this kind of augmentation on the dataset is that it is very important for pruning features (the pruning choices that define the quality of the solution) and labels to be consistent. For example, if changing a specific feature causes a optimal pruning solution to drop in quality from 1 to 0.75 that same feature change should cause the same drop in quality in other vines and pruning solutions. However, this is very difficult to keep consistent across multiple vine architectures and is sometimes not possible due to equal numbers of canes on a particular side of a vine not being available on all other vines. At the same time we need to ensure that our augmented solutions do not bias the model towards one end of the scoring spectrum and keep the jumps between score values as smooth as possible to avoid discontinuities in the search space which could cause the training to fail.

It was very challenging to maintain these considerations across our dataset and we eventually opted to use a simple strategy that just counts the number of *correct* canes to determine the pruning score. For example, in a two-cane pruning scheme many experts will choose four canes to be either bearer canes or renewal spurs, all other canes are pruned. We would then augment this solution by adding three extra pruning solutions.

1. The first is constructed by randomly picking one of the four chosen canes and replacing it with a previously pruned cane. Since only one cane is different from the optimal solution we set the quality label to 0.75.
2. Then we pick another random cane from the remaining three originally chosen canes and also replace this cane with one that was originally chosen to be pruned. Since half the canes are now different from the optimal solution we set the quality score to 0.5.
3. Similarly, our third augmented pruning solution has three of the canes replaced with previously pruned ones and the quality score is set to 0.25.
4. For our final augmented solution we replace all the chosen canes with previously pruned canes and set the quality score to 0.

This data augmentation scheme has the advantage of multiplying our dataset size by four and adding substantial diversity to the possible labels, but we found that models trained on this data often fail to converge and perform poorly in scoring pruning solutions. This is probably because the five distinct quality scores (0, 0.25, 0.5, 0.75, 1) do not represent a smooth transition in quality from the completely random solution to the optimal human solution and we found that adding small amounts of random noise to the augmented scores helped to train better models. A distribution of pruning scores for our augmented dataset is shown in Fig. 4. We clearly see five distinct bands of scores but with the random noise added the values are more evenly distributed so the bands are thicker and are close to merging into one another to form a continuum of scores.

By using this strategy we can generate a dataset that's four times larger than our original dataset and has enough diversity in labels to prevent bias in the model. In Section 8 we will provide an overview with performance metrics of the results of this model but at this point it is important to remember that this model only scores given pruning solutions and our original goal was to generate new pruning solutions from recorded vine features.

6. Generating pruning solutions

As we demonstrated in our prior research it is possible to train a model on a dataset similar to the one described in the previous section that can classify each cane in a vine as either pruned, bearer cane, or renewal spur, but this approach has serious limitations. However, if we use the PruningGraph model as our starting point there are multiple ways to generate candidate pruning solutions.

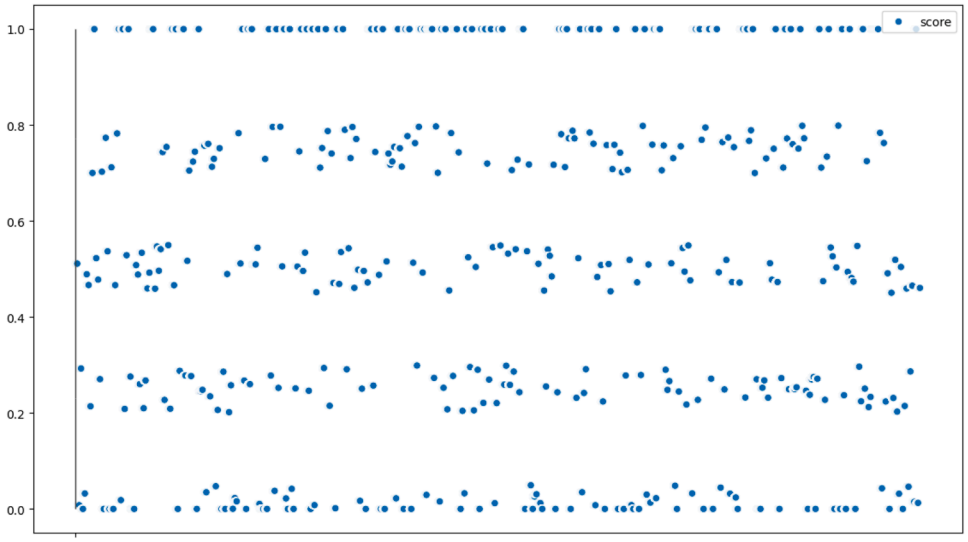

Fig. 4. The distribution of pruning quality scores in our augmented dataset. The vertical axis is the score value and the horizontal axis is the pruning solution index.

6.1. From ranked collections

The simplest option would be to generate a large collection of unique but random pruning solutions that can then be ranked using PruningGraph. It's important that the generated solutions are valid in that they follow the pruning scheme set for that block in the vineyard. For example, the number of bearer canes that are required on each side of the vine head have to be adhered to for any solution to be considered acceptable. Thankfully these pruning schemes are easily interpreted as simple rules that can be applied during the generation of random pruning solutions.

The downside to this approach is that depending on the number of canes, there may be thousands of possible solutions and randomly selecting a small sample of these may miss all the good solutions. This can be mitigated by increasing the size of the samples but it is computationally expensive to generate and then rank the solutions using our PruningGraph model so there is a practical limit on how large this set can be.

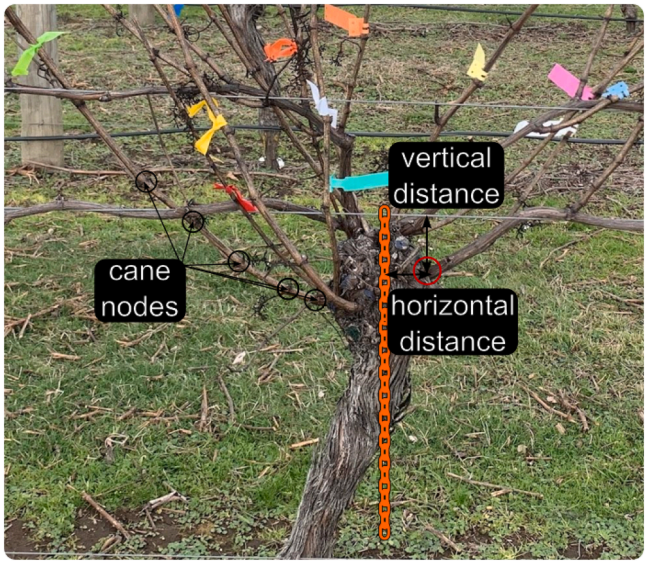

Fig. 5. The manual measurement of three of the most important cane features are illustrated here. The orange chain represents the reference chain that is hung from the bottom wire. The red circle indicates the cane origin from which the vertical and horizontal distance is measured. On the left side of the head the first four nodes of another cane are highlighted. These are counted and recorded as the node count of that cane.

6.2. Generative models

Another more interesting approach is to train a generative model based on the PruningGraph model to generate the optimal pruning solution directly. There are several advanced ways to design and train such a model but likely the simplest approach is to use our trained PruningGraph model and invert it so that we provide as input the pruning score and then use an optimiser (similar to how we trained the model) to find the pruning solution that maximises the score. Instead of using the standard gradient **descent** method to minimise the loss function by slowing updating the model weights, we freeze the model weights and use gradient **ascent** to slowly update the model input (the vine feature vector) until we have maximised the pruning score. We only update the last three feature values of each cane as they define the pruning solution and leave all other cane features constant. In this way we can directly find a pruning solution by optimising until the pruning score is sufficiently high.

Unfortunately, this approach is very prone to converge to invalid solutions and post-processing is required to ensure that the final solution adheres to the pruning scheme. We mitigate the risk of the model converging to invalid solutions by starting the optimisation process from a seed solution that is randomly generated but adheres to the pruning scheme. We can take this one step further and combine this approach with the previous idea of generating a ranked list of random solutions by seeding our search with the best solution from the ranked list. The results from these approaches will be elaborated on in Section 8.

7. Data collection

The collection of quality data was a critical limitation of our previous work. We used synthetic data to train our initial models but it became clear early in our analysis that our synthetic data was not realistic and that the model has to be trained on real vine architectures and that pruning solutions should come from expert pruners that physically pruned those vines. This realisation meant that we had to develop a way to accurately and consistently record the cane metrics of Table 1 and then find a practical way to encode the pruning decisions from an expert so these can be associated with our measurements.

7.1. Vine measurements (cane metrics)

The most important features (as it relates to canopy architecture) are the horizontal and vertical distances of the cane origins and it is very important to measure these distances in a consistent manner. A cord or string is used to draw a line straight downward from the bottom guide wire to the cane origin that is being measured. The length of this cord is then measured using a tape measure and recorded as the vertical distance for that cane. For the horizontal distance we first use a thin chain, hung from the bottom guide wire going straight down the centre of the trunk, to serve as a consistent measuring line. Cord is then used to draw a straight horizontal line from the chain to the cane origin and the length of this cord is recorded as the horizontal distance. These measurements are illustrated in Fig. 5.

The cane diameter is recorded using a handheld digital caliper and the measurement is always done between the second and third node from the cane origin. This ensures consistent measurements when multiple people are involved in the collection of measurements. We initially measured the cane length using a flexible cord and a tape measure to measure the full length of the cane from the terminal point to the origin, but later realised that this is unnecessary. The only distance that human expert pruners care about is that the cane is long enough to cover the distance from the trunk to halfway to the next trunk. If a cane can be tied down to the guide wire and is long enough to stretch this distance

	A	B	C	D	E	F	G	H	I	J	K	L
1		Nber	Origin	Vertical Distanc	Horizontal Distanc	Cane Length	Orientation	NodeCount	Health	Diameter		GroundTruth
2	1	Green tape	h	55.7	21.3	97.9	l	16		8.79		bl
3	2	Red tape	s	56.5	22.2	130.7	l	18		13.56		
4	3	White tape	h	59.3	24.8	129	l	18		14.78		
5	4	Pink tape	c	71.9	25.3	125.6	l	16		10.94		tl
6	5	Yellow tape	c	72.8	50.8	104.2	r	15		10.01		rt
7	6	Blue tape	c	67.5	50.2	122.3	r	18		9.43		rb
8	7	Orange tape	c	74.1	57	106.9	r	16		12.79		rs
9	8	Green flag	c	85	30.3	102.2	r	19		9		
10	9	Yellow flag	c	82	56.3	109.1	l	15		12.28		
11	10	Pink flag	c	79.1	60.9	108.3	r	19		7.93		
12	11	White flag										
13	12	Blue flag										
14	13	Orange flag	h	60.8	32.2	97.5	l			15.73		ls
15	14	Red flag										

Fig. 6. In this example multiple canes of the vine have been labelled with coloured tags. These tags allow us to easily and consistently record cane features and pruning decisions as shown in the bottom figure.

it can be considered as an option to be a bearer cane, otherwise it is pruned off. Any extra length beyond this threshold does not influence the pruning decision so all we need to measure is whether the cane is long enough or not and record a boolean value to indicate this. The node count is the simplest feature to measure and is only a record of the number of nodes counted from the cane origin to the terminal point.

Two somewhat subjective measurements are the *healthy* and *orientation* features. The healthy feature is a visual measure that aims to identify visually obvious indicators of poor health, like discolouration, lesions, or breaks. It is recorded as a boolean value to indicate either a cane that cannot be considered as a bearer cane due to poor health, or one that can be considered. The orientation feature captures the natural tendency for a cane to either be easily tied down on the left or on the right of the vine head and is another boolean record. For example, some canes can have their origin on the left side of the head but curve naturally to the right so that the orientation is to the right.

7.2. Ground truth pruning solutions

Our expert human pruners were asked to follow after the cane measurements were done and choose those canes which should be tied down as new bearer canes and those which should be kept as renewal spurs. We labelled each cane that has its features recorded with a coloured ribbon or paper tag. The pruners would then record their decision for each cane using the associated colour. Not all of the canes were measured and considered as candidates for the pruning decisions, as only the canes near the vine head are considered by human pruners. For each vine in the experiment we would choose approximately a dozen of the most promising canes and label them with coloured tags. All the labelled canes will then have their features measured and recorded against that vine identification number and cane colour. The expert pruners will then add their pruning decisions to that record as the ground-truth labels. An example of a vine labelled with coloured tags and an example of a record

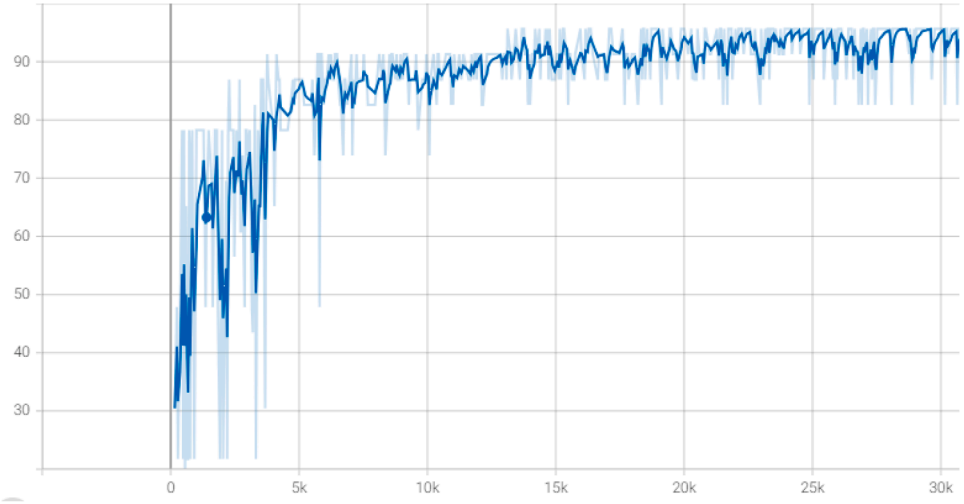

Fig. 7. The average validation accuracy on 2022 dataset. In this graph the x-axis represents the number of training iterations and the y-axis is the average accuracy over the validation set of 23 vines. The graph has been smoothed using a running average so that the dark blue line represents the smoothed average score and the light blue line represents the original scores.

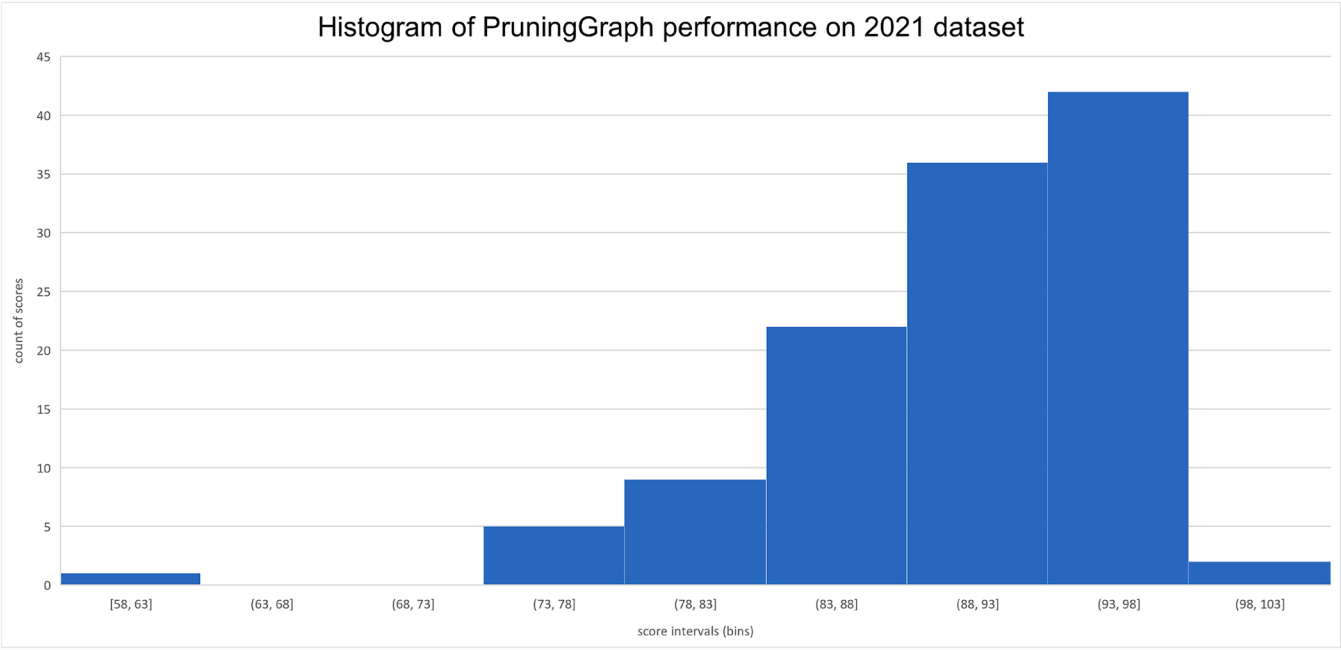

Fig. 8. The pruning score estimates from the PruningGraph model tested on the 2021 dataset.

where the cane features and pruning decisions are recorded is shown in Fig. 6.

8. Results

We initially started by training a cane classifier on synthetic data. This approach is limited severely by the realism of the synthetic data (see Fourie et al. (2021)). However, with the realistic data that we collected 2021 and 2022 we were able to experiment with models trained on real vines and pruning solutions from expert human pruners.

As was explained in Section 4, our initial attempts to train the VineGraph model on realistic data resulted in disappointing results. Instead our goal was first to focus on the PruningGraph model and test the model's ability to accurately score pruning solutions on realistic vines.

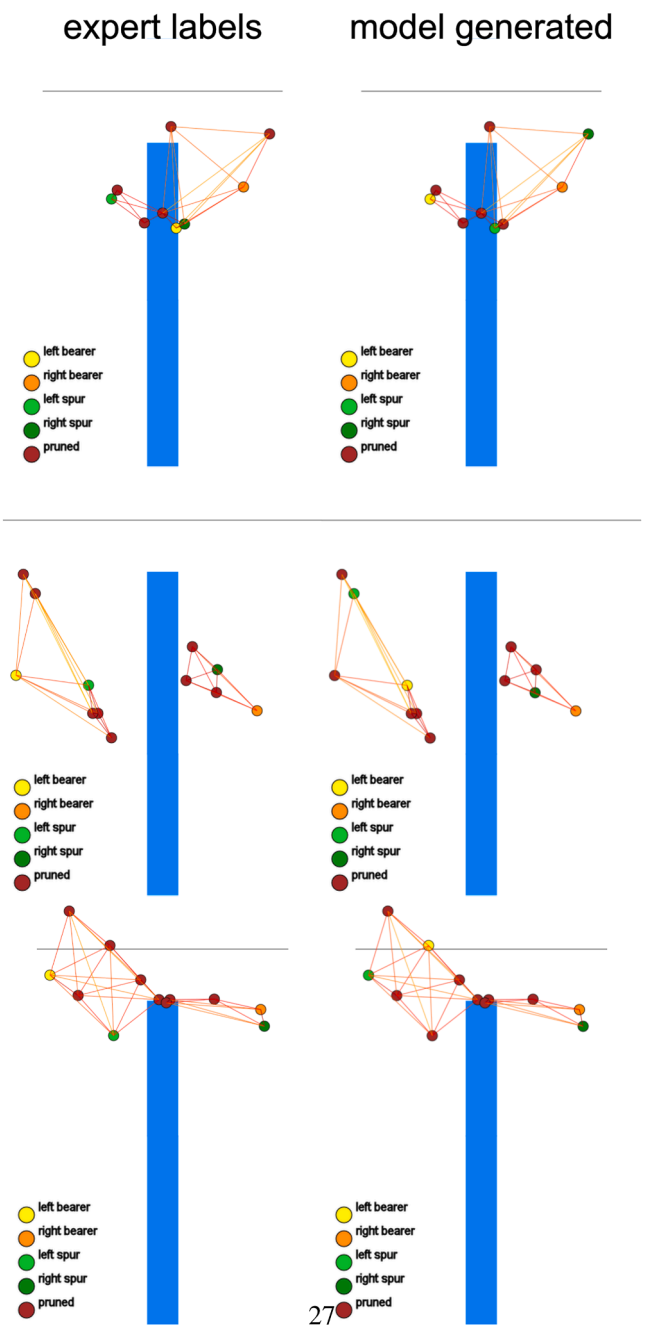

Fig. 9. Three challenging examples of pruning solutions from our PruningGraph model vs the expert human solution.

8.1. PruningGraph model performance

We collected two datasets that both contained cane features and ground-truth pruning solutions from expert pruners. We collected both datasets from the Stoneleigh Vineyard in Blenheim, New Zealand[2] during the 2021 and 2022 winter pruning seasons. The first dataset was collected during the 2021 season and the second in the 2022 season.

We trained the PruningGraph model on the 2022 dataset. After filtering out bad data we were left with 90 vines that have had their canes measured and have been pruned by one of the expert pruners. By applying the data augmentation elaborated on in Section 5.2 we transformed this into a dataset of 450 vines that have balanced pruning scores to prevent training bias. We split this into a training set of 427 vines and a validation set of 23 vines. We optimised the model hyper-parameters based on the performance on the validation set and settled on a training strategy that uses the Adam optimiser Kingma and Ba (2015) with a learning rate of 10^{-3}, a decay rate of 10^{-5}, and an epsilon value of 10^{-2}.

This model accurately estimated the pruning score on the validation set and was able to consistently get 95 % average accuracy on the validation set after less than 20,000 training iterations (see Fig. 7). However, by itself this is not a good indicator of performance as the model was trained on an augmented dataset where the majority of the data points have synthetic labels and the validation set is not completely independent as it was used to optimise the hyper-parameters.

A much better test of the performance is to apply this trained model to the vines from the 2021 dataset. This dataset consists of independent vines with measurements taken a season apart and labelled using different pruners. The 2021 dataset contained 117 vines after we filtered out invalid data points. For this test we did not apply any data augmentation as training was not our intention but rather to test on a realistic unaltered dataset. As all the vines have been expertly pruned we set the ground-truth label at 1.0 for all the vines and compared with the score estimated by the model. The results from the 117 test vines are presented as a histogram of scores in Fig. 8. The average score across the 117 vines was 89.6 % and the minimum score was 57.9 %. We set the score threshold for *good* pruning solutions at 75 % so any pruning solution that has a higher score is considered an acceptable pruning solution. This threshold was set in collaboration with our expert pruners. Only three of 117 vines were estimated by the model to have pruning scores below this threshold and were therefore considered to be incorrectly classified. 72 of the vines had scores larger than 90 %. We therefore concluded that the model was successful and was able to identify the features of a good pruning, even across seasons and with different pruners providing the pruning solutions. This alone does not show that the model is able to generalise across seasons, but rather gives confidence that this approach could generalise given sufficiently diverse datasets across multiple seasons.

Rigorous statistical testing of the model performance using precision, recall, and F1 scores was not possible. Our model does not generate categorical classifications, but rather scores a solution which we then threshold based on subjective advice from pruners to label pruning solutions as *good* or not. Even if we forced our results into categories using the threshold, all our labelled data has an identical score (1.0), since all these solutions are considered as optimal since they were generated by our expert human pruners. Additionally, we do not intend to compare, but rather to introduce the concept of a GNN in this area of decision-support systems.

8.2. Generative models to create optimal prunings

As was previously mentioned our original goal was to train a model that can generate pruning solutions, as apposed to just scoring them. In Section 6 we elaborated on methods to generate pruning solutions using

[2] Located at coordinate 41.4851 S, 173.8731 E

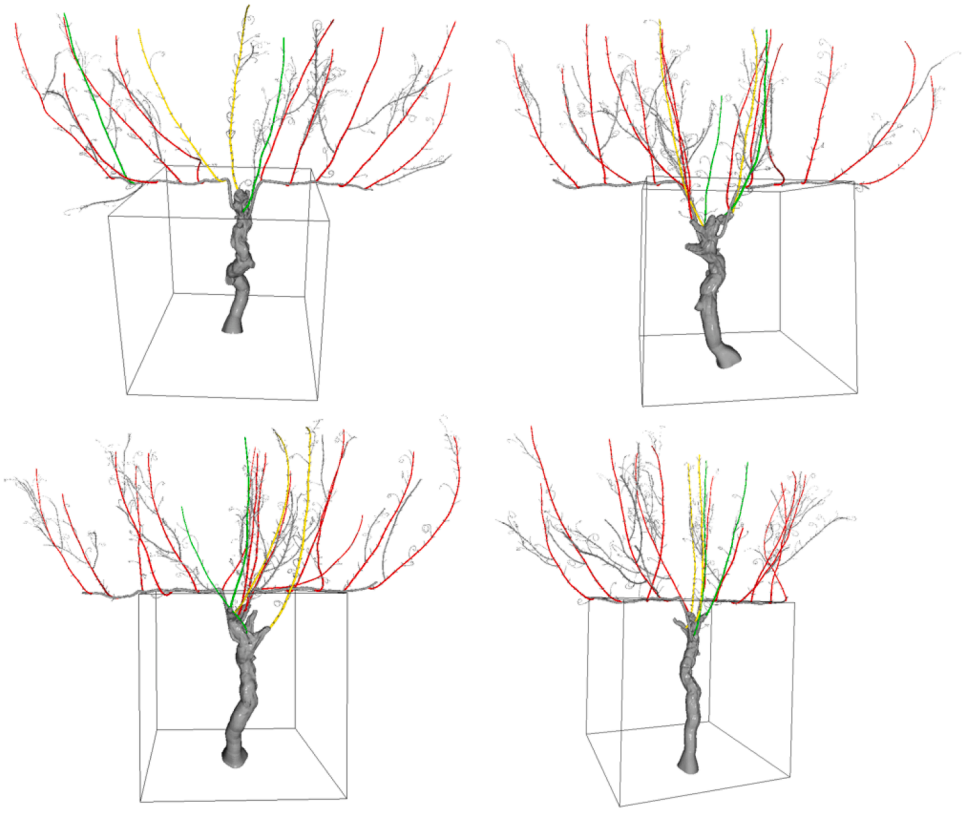

Fig. 10. These are examples of 3D rendered realistic vines that may be used to efficiently generate a realistic dataset of pruning solutions (reproduced with permission from Fourie et al. (2021)).

a trained scoring model and in this section we present example prunings generated using these techniques.

As was mentioned in Section 6 we found that the best results from the generative model is achieved when we generate a sample of random (but valid according to the pruning scheme) pruning solutions to seed the model. We found that the following procedure achieves the best results:

1. Generate 25 random pruning solutions for each vine
2. Enforce the pruning scheme by ensuring the correct number of bearers are present on each side of the vine head. Randomly replace missing or redundant bearer canes with alternatives until the pruning scheme is adhered to.
3. Rank the corrected list of random pruning solutions using the trained PruningGraph model.
4. Pick the pruning solution with the highest pruning score and use it as the first input to the modified PruningGraph model (see Section 6)
5. Optimise using gradient ascent. We found converge to a high scoring pruning solution to be remarkably quick and stop optimising after only 10 iterations.
6. Ensure that the pruning scheme is adhered to on the final result by picking only the highest scoring n bearer canes and renewal spurs from each side of the vine head where n is the number of bearers specified by the pruning scheme.

This algorithm results in good pruning solutions that are generally very similar or identical to the pruning choices made by the expert pruners. We tested our method on the same 2021 dataset that was used to test the scoring performance. In all cases we let the model converge to pruning solutions that were scored to better than 0.95 using the PruningGraph model. It's very difficult to directly compare the quality of our model pruning solutions to the expert solutions as we only have the model scores but we can visually compare the graph representations to verify that the model produces similar solutions when they are not identical to the expert solution. We present a few challenging examples of where the model produced similar but not identical solutions in Fig. 9.

In all three examples the model produced a pruning solution that mostly seem reasonable given the general pruning good practices, like generally preferring the lower canes as bearers and renewal spurs, and preferentially picking canes that are closest to the centre of the vine head. However, there are some examples where the model makes unexpected and potentially dubious decisions. For example, in the first example from the top of Fig. 9 we see the model pick the cane that's furthest away from the vine head as the right renewal spur while the expert pruner chose an option much closer to the vine head. The model's choice could be due to cane diameter, node count or another pattern that may be a more complex combination of features. We see a similar situation in the last example at the bottom of the page where the model picks a left bearer cane that's very high, even above the guide wire. It's unlikely that a human pruner would make such choice and this is likely a mistake by the model. It's very difficult to evaluate whether choices like this would make a good pruning into a bad one without consulting expert pruners.

9. Conclusion

In our previous publication we highlighted the fact that realistic vine data and pruning solutions from expert pruners were required to address some of the original model's limitations (Fourie et al., 2021). We collected this data but subsequent analysis of our model's disappointing results led to a major redesign of the original model and a reevaluation of how the performance of such a model should be measured. By addressing two incorrect assumptions, see Sections 4 and 4.2, we improved our results and developed a more useful tool (see the results from Fig. 4).

We therefore abandoned our original model and developed the PruningGraph model based on the insights gained from testing the original VineGraph model on realistic data. The PruningGraph model does not attempt to classify canes but rather scores a given pruning solution based on analysis of the whole vine. We showed that by augmenting a set of expert pruning solutions with simple modifications that turn a good solution into a poor one, we can train this new model to accurately score pruning solutions (see Section 8.1). We can then use this scoring model to generate new pruning solutions on any vine input (see Section 8.2).

However, some limitations to our approach still remain and some questions regarding the model's performance compared to the expert human pruners have not been addressed. A fundamental problem that remains with the PruningGraph model is that all our labels are only of good pruning solutions and therefore we need to augment our dataset with synthetic poor pruning solutions in order to train an unbiased model. The way that we do this is currently ad-hoc and does not fully capture the features of the pruning solution that make it good or bad. Our model is therefore limited by it's training data which leads to inconsistent scoring of some similar pruning solutions and subsequently the generation of dubious solutions.

9.1. Future work

The next stage of this research should focus on addressing the largest limitation which is the lack of expert labels for pruning solutions that are NOT good solutions. Ideally, we require a dataset made from multiple variations of a known good pruning solution. One way this could be accomplished is to start with an expert pruner's solution on a real vine and then modify this solution in various ways by substituting chosen canes with pruned ones. The pruner will then need to score (on a scale of 1 to 10) the quality of each variation compared to the original. This will need to be repeated for multiple pruners and multiple vine architectures. The goal is to create a balanced dataset with a full range of pruning scores without having to resort to synthetic solutions that may not capture the relevant features accurately.

This process could be accelerated if realistic synthetic vines are digitally rendered and presented to pruners in a way that is similar to how they would evaluate real vines. By using virtual reality (VR) tools, realistic digital vines like those presented in Fig. 10 can be evaluated by pruners much faster and multiple variations on a pruning solution can quickly be shown to the pruner by using colours or icons to indicate the changes. The pruning score could then be recorded digitally using the same system.

Furthermore, it should be noted that the current model is specific to a particular pruning scheme. Our current model is trained to score two-cane pruning systems where one cane is selected on each side of the vine head as the new bearer cane, and one renewal spur is selected on each side as well. This is more restrictive than it should be as expert pruners may choose only one renewal spur if it is impractical to add another, even within the restrictions of the pruning scheme. Currently, the model will have to be retrained and the post-processing rules adjusted to accommodate any changes in the pruning scheme. The implementation of the pruning scheme should be less restrictive and should include rules that allow for some variation within the strict number of canes that should be chosen on each side of the vine head.

There is also a lot of potential improvement to the generative model that is currently used to generate pruning solutions based on scores from the PruningGraph model. As was mentioned in Section 6.2 more advanced models can be trained by using pre-trained PruningGraph layers as a basis and adding custom generative layers. A completely separate model, for example a generative adversarial network (GAN), could also be trained to generate a better list of candidate solutions that can then be ranked by the PruningGraph model.

CRediT authorship contribution statement

Jaco Fourie: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization; **Jeffrey Hsiao:** Methodology, Software, Writing – review & editing; **Oliver Batchelor:** Software, Writing – review & editing; **Kevin Langbroek:** Software, Writing – review & editing; **Henry Williams:** Conceptualization, Resources, Supervision, Project administration; **Richard Green:** Resources, Supervision, Project administration, Funding acquisition; **Armin Werner:** Resources, Supervision, Project administration, Funding acquisition, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jaco Fourie reports financial support was provided by New Zealand Ministry of Business Innovation and Employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This publication is supported by the collaborative MaaraTech Human-Assist project funded by the New Zealand Ministry of Business, Innovation and Employment (MBIE) (grant UOAX1810). Multiple research institutions contribute to this project including University of Auckland (lead), University of Waikato, University of Canterbury, University of Otago, Lincoln Agritech, and Plant and Food Research.

References

Archer, E., & Fouche, G. W. (2017). Effect of bud load and rootstock cultivar on the performance of v. vinifera l. cv. red muscadel (muscat noir). *South African Journal of Enology and Viticulture*, *8*(1), 6–10. https://doi.org/10.21548/8-1-2322

Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., & Corbett-Davies, S. (2017). A robot system for pruning grape vines. *Journal of Field Robotics*, *34*(6), 1100–1122. https://doi.org/10.1002/rob.21680

Bramley, R. G. V., Trought, M. C. T., & Praat, J.-P. (2011). Vineyard variability in marlborough, New Zealand: characterising variation in vineyard performance and options for the implementation of precision viticulture. *Australian Journal of Grape and Wine Research*, *17*(1), 72–78.

Corbett-Davies, S., Botterill, T., Green, R., & Saxton, V. (2012). An expert system for automatically pruning vines. In *Proceedings of the 27th conference on image and vision computing New Zealand* IVCNZ '12 (p. 55–60). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2425836.2425849

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Epee, P. T. M., Schelezki, O. J., Parker, A. K., Trought, M. C. T., Werner, A., Hofmann, R. W., Almond, P., & Fourie, J. (2022). Characterising retained dormant shoot attributes to support automated cane pruning on vitis vinifera l. cv. sauvignon blanc. *Australian Journal of Grape and Wine Research*, *28*(3), 508–520.

Fourie, J., Bateman, C., Hsiao, J., Pahalawatta, K., Batchelor, O., Misse, P. E., & Werner, A. (2021). Towards automated grape vine pruning: Learning by example using recurrent graph neural networks. *International Journal of Intelligent Systems*, *36*(2), 715–735.

Gittoes, W., Botterill, T., & Green, R. (2011). Quantitative analysis of skeletonisation algorithms for modelling of branches. In *International joint conferences on artificial intelligence (IJCAI)*. Auckland, New Zealand.

Greven, M. M., Bennett, J. S., & Neal, S. M. (2014). Influence of retained node number on sauvignon blanc grapevine vegetative growth and yield. *Australian Journal of Grape and Wine Research*, *20*(2), 263–271. https://doi.org/10.1111/ajgw.12074

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations (ICLR)*. San Diego, CA, USA.

Paulin, S., Botterill, T., Chen, X. Q., & Green, R. (2015). A specialised collision detector for grape vines. In *Australasian conference on robotics and automation*. Canberra, Australia.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv:1706.03762.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Williams, H., Smith, D., Shahabi, J., Gee, T., Nejati, M., McGuinness, B., Black, K., Tobias, J., Jangali, R., Lim, H., Duke, M., Bachelor, O., McCulloch, J., Green, R., O'Connor, M., Gounder, S., Ndaka, A., Burch, K., Fourie, J., Hsiao, J., Werner, A., Agnew, R., Oliver, R., & MacDonald, B. A. (2023). Modelling wine grapevines for autonomous robotic cane pruning. *Biosystems Engineering*, *235*, 31–49. https://doi.org/10.1016/j.biosystemseng.2023.09.006