
Urban Tree Species Classification Using Aerial Imagery

Emily Waters¹ Mahdi Maktabdar Oghaz¹ Lakshmi Babu Saheer¹

Abstract

Urban trees help regulate temperature, reduce energy consumption, improve urban air quality, reduce wind speeds, and mitigating the urban heat island effect. Urban trees also play a key role in climate change mitigation and global warming by capturing and storing atmospheric carbon-dioxide which is the largest contributor to greenhouse gases. Automated tree detection and species classification using aerial imagery can be a powerful tool for sustainable forest and urban tree management. Hence, This study first offers a pipeline for generating labelled dataset of urban trees using Google Map's aerial images and then investigates how state of the art deep Convolutional Neural Network models such as VGG and ResNet handle the classification problem of urban tree aerial images under different parameters. Experimental results show our best model achieves an average accuracy of 60% over 6 tree species.

1. Introduction

Trees are well recognised for their importance to the planet and human life. Environmentally, trees slow surface runoff from rainfall, reducing flood risk, water pollution and soil erosion (Chandler et al., 2017). They improve overall air quality by absorbing particulate matter, create a cooling effect, and mitigating the heat island effect in urban areas (Manickathan et al., 2017). A study by (Bastin et al., 2019) shows forestation is a possible strategy for mitigating climate change. Trees capture and store atmospheric carbon-dioxide and lock it up for centuries. Trees play a key role in climate change mitigation by capturing, storing and consequently reducing atmospheric CO₂ levels, the main adverse contributor to greenhouse gases and climate change. Studies show, urban trees can cut heating costs by reduc-

ing wind-speed and casting shade around the housing area which indirectly mitigates emission of greenhouse gases (Wolf, 2005). To leverage this potential, effective forest and urban tree management is essential. This requires detailed information about tree species, composition, health and geographical location of each tree in order to create a long term sustainable plan for plantation and forestation sites, pruning schedules and mitigation of potential problems (Baeten & Bruehlheide, 2018). It also helps to monitor tree species diversity and track health and growth rate to creates a more robust ecosystem with better productivity and greater resilience to disease and pests (Gamfeldt & Snall, 2013; Rust, 2016). Such management system demands for an accessible, reliable yet economically and practically viable platform to automatically detect, classify and monitor forests and urban trees. Historically, this has been carried out by experts and volunteers visiting trees on the ground but this is a laborious, time-consuming and expensive approach. Alternatively LiDAR technology, used to estimate the number of trees in an area (Wilkes et al., 2018) and categorise their species (Kim et al., 2008), paved the way to automated urban tree and forest management. However, LiDAR surveying is a costly process mainly due to the speciality equipment and skilled human resource required to collect and interpret it (Rezatec, 2020). Hyperspectral imaging and remote sensing satellites images have advanced significantly over the last couple of decades and are now able to produce high-resolution images which facilitates tree detection and species classification (Frick et al., 2019; Dalponte et al., 2014; Maschler et al., 2018; Clark et al., 2005). There are a limited number of studies looking into the detection classification of trees using RGB aerial images. RGB aerial image surveying can be as costly as other aforementioned approaches however availability of mapping service such as Google Maps and Bing Maps can significantly reduce the cost of surveying and data collection. Studies like (Wegner et al., 2020; Nezami et al., 2020) utilized images from these platforms paired with Convolutional Neural Network (CNN) to create a fully automated yet accurate tree detection and classification model which is pertinent to effective forest and urban tree management.

Having said that, the purpose of this study is to first generate a labelled dataset of urban trees using Google Map's RGB aerial images paired with existing tree inventories to supply

^{*}Equal contribution ¹Faculty of Science and Engineering, Anglia Ruskin University, Cambridge, United Kingdom. Correspondence to: Lakshmi Babu Saheer <lakshmi.babu-saheer@aru.ac.uk>, Mahdi Maktabdar <mahdi.maktabdar@aru.ac.uk>.

GPS coordinates and species information. This study uses the Camden tree inventory (Couch, 2021) to acquire GPS coordinates and species details. This study also aims to build a supervised model capable to detect and classify tree species accurately. Several state of the art pre-trained CNNs models including VGG and ResNet variants along with some custom models have been investigated, compared and analysed.

2. Dataset Generation

The proposed dataset generator pipeline uses Google Map’s static API to source trees’ aerial images and Camden tree inventory (Couch, 2021) to supply tree’s GPS location and species information. Camden tree inventory contains over 23,000 GPS locations (Latitude and Longitude) of up to date (over 99.9% of records dated 2016 or later) Council owned trees on highways and in parks and open spaces in London Borough of Camden. Cleaning process performed by removing entries with missing locations, vacant plots or unknown species. Each data point contains tree species, height, spread, diameter at breast height (DBH), and maturity. An automated process, goes through all entries in the Camden inventory and downloads aerial image from Google Map’s static API. The latitude and longitude co-ordinates of each tree were used as the centre point for each aerial image of 200x200 and zoom level of 20. While Camden tree inventory consists of hundreds of different tree species, this study only investigates top 6 species with the highest frequencies including Ash, Silver Birch, Common Lime, London Plane, Norway Maple and Sycamore. The data is split into subsets with 70% for training, 20% for validation and 10% reserved for testing. Images were labelled and categorized based on their species and then organized into train, test and validate sub-sets. The proportional representation of each species is preserved across the subsets so that any class imbalance is retained at each stage. As it can be observed in the Figure 1, the number of entries in the training set is fairly limited for an effective train of a deep convolutional model. Hence, this study employed image augmentation technique (Rotation, width and height shift, horizontal flip, zoom and brightness) to over sample and expand the training set with new, plausible examples as shown in the Figure 2 (Krizhevsky et al., 2017).

3. CNN for Tree Species Classification

This research investigates and evaluates 3 possibilities including VGG-16, ResNet50 and a group of custom deep models to find an optimal CNN model for tree species classification. The VGG-16 (Simonyan & Zisserman, 2014) was the chosen model in similar tree species classification studies by Branson, et al. (Branson et al., 2018) and Lang (Lang, 2020). As per these studies, the VGG-16 network was pre-

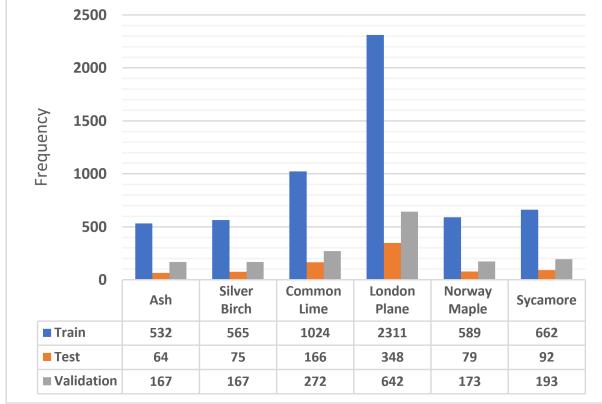


Figure 1. Training, validation and testing sets counts across top 6 species in Camden dataset

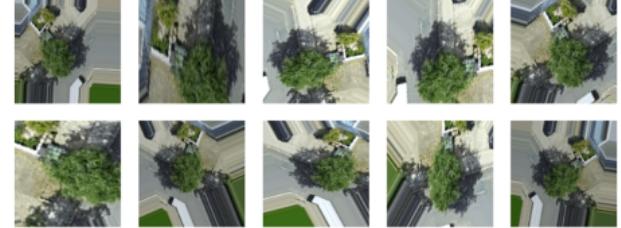


Figure 2. Example of the augmentation applied to images in the training data subset

trained with ImageNet dataset (Russakovsky et al., 2015) and then being fine-tuned and optimized on our dataset of tree aerial images. We paired VGG-16 model with Adam optimiser which besides being computationally efficient was also used in similar studies like Lang (Lang, 2020). The parameters to be varied are dropout and class weights. Class weights applied to compensate for imbalance class sizes. All the models considered in this study are trained and tested based on the training, validation and testing sets shown in the Figure 1. This study used categorical cross-entropy loss function across all models in this study while optimiser choice varied to see which has the greatest impact on model performance. The maximum number of training epochs is set to 100. During training, the model with the smallest loss is saved and used for comparison with other models. To reinforce the evaluation process, the top 5 models with the smallest loss and higher accuracy have further evaluated using 5-fold cross validation to obtain more reliable results. This study also investigates the performance of pre-trained ResNet50 model for tree species classification using aerial images. Many similar studies including (Natesan et al., 2019; Cao & Zhang, 2020) used this model for similar purposes. Deep structure of Resnet50 facilitates modeling of complex features while skip connections avoid issues like vanishing gradients. ResNet50 has been paired

with Adam optimizer to achieve efficient training and timely convergence. Various dropout and class weight ratios have been examined and optimized during the training process. In addition to aforementioned pre-trained models, this study investigates a range of Custom CNNs models to identify possibility of achieving accurate tree species classification using a less complex model. The template for construction of these custom models is illustrated in equation 1.

$$\begin{aligned}
 & \text{INPUT} \rightarrow \\
 & [[\text{CONV} \rightarrow \text{RELU}] * 2 \rightarrow \text{MAXPOOL}] * N \rightarrow \\
 & [\text{FC} \rightarrow \text{RELU}] \rightarrow \text{FC}, \\
 & \text{where } N \in \{1, 2, 3, 4, 5, 6\}
 \end{aligned} \tag{1}$$

where N is the number of Convolutional blocks, ranges between 1 and 6, with each block consisting of two Convolutional layers (CONV) with a ReLU activation function followed by a Maxpooling layer. The size of the kernel and choice of kernel initialiser within the CONV layers are to be varied between models. Dropout is added after each Convolutional block and after the penultimate fully connected (FC) layer. Optimiser choice varied to identify its impact on model performance.

4. Results and Discussion

The training process is conducted using the 6 tree species (Ash, Silver Birch, Common Lime, London Plane, Norway Maple and Sycamore) with the largest number of samples. The VGG-16, ResNet50 and a range of Custom CNNs have been trained with a combination of different parameters including dropout ratio, optimiser, class balanced weight to identify the top performer model. A further model re-evaluation using 5-fold stratified cross validation helped to obtain more reliable accuracy figures.

The VGG-16 model is trained with various dropout and class weights values. The performance measures obtained by the VGG-16 model are presented in Table 1. The VGG-16 base model achieves an accuracy of 56.55%, only outperformed by the VGG-16 model with 20% dropout that increases the score by 0.61%. The accuracy differences for the VGG-16 models with or without dropout appear to be marginal, however the precision gains almost 5% for the model with 20% dropout. Note that the class balanced weight model under-performs other models with a considerable margin. Similar to VGG-16, ResNet50 model is trained with various dropout and class weights values. The performance measures obtained by the ResNet50 model are recorded in Table 2. The standard ResNet50 model managed to achieve accuracy of 59.03% which is already higher than any figure achieved by the VGG-16 model. Adding a 20% dropout, marginally raised ResNet50 accuracy to 59.92%. Moreover,

Table 1. Comparison of results for the VGG-16 variants

Model	Loss	Accu (%)	Ave Class Recall (%)	Ave Class Precision (%)	No Epochs
VGG-16 (Standard)	1.1934	56.55	42.23	40.94	68
VGG-16 (Balanced W)	1.7900	42.23	16.67	7.04	13
VGG-16 (10% dropout)	1.1914	55.83	41.80	40.30	49
VGG-16 (20% dropout)	1.1649	57.16	42.65	45.30	88

Table 2. Comparison of results for the ResNet50 variants

Model	Loss	Accu (%)	Ave Class Recall (%)	Ave Class Precision (%)	No Epochs
ResNet50 (Standard)	0.86	59.03	51.13	49.37	43
ResNet50 (Balanced W)	1.03	58.96	50.66	48.82	41
ResNet50 (10% dropout)	0.88	59.14	51.24	49.44	46
ResNet50 (20% dropout)	0.73	59.92	54.07	52.46	62

Average Class Precision raised by almost 3% for the model with 20% dropout. Other models including Balanced class weights and 10% dropout perform more or less the same as the standard ResNet50 model.

Apart from pre-trained VGG-16 and ResNet50, we have trained and evaluated a range of Custom CNNs models to identify possibility of achieving accurate tree species classification using a less complex model. All custom models are constructed as per the formula in equation 1. The baseline custom model has one convolutional block with 3x3 kernels, "He uniform" initialiser and SGD optimiser. This model is then compared with a few other custom models which mainly differ by having 2 to 6 convolutional blocks, different dropout ratio, kernel size, optimizer and initialiser. A detailed summary of notable results are presented in the Table 3. The choice of optimiser was limited to what Tensorflow library offers. We have explored different optimisers including Adadelta, Adagrad, Adam, Adamax, Ftrl, Nadam, RMSprop and SGD. Experiment results shows the Adamax optimiser consistently outperformed other optimisers in this comparison. Similarly, the initialisers are taken from the Tensorflow offerings including constant, Glorot normal, Glorot uniform, He normal, He uniform, Lecun normal, Lecun uniform and random normal. Results shows "He normal" marginally outperforms other initialisers in this comparison. According to the Table 3, The top performing model has 6 convolutional blocks paired with the "He normal" kernel initialiser and is optimised using Adamax – a variant of the Adam algorithm. This model achieves accuracy of 69.5%, recall of 57.4% and precision of 62.8%. The top performing model re-evaluate using 5-fold stratified cross validation which led to a considerable drop across majority of the metrics. Cross validation Result can be observed at the bottom

Table 3. Comparison of results for custom CNN models

Model	Loss	Accu (%)	Ave Class Recall (%)	Ave Class Precision (%)	No Epochs
x1 Conv block (Baseline)	1.2495	52.79	35.99	38.37	44
x2 Conv block	1.1929	55.95	38.53	40.55	62
x3 Conv block	1.1219	58.50	42.81	54.06	57
x3 Conv block (20% dropout)	1.2326	54.98	40.12	42.27	48
x3 Conv block (30% dropout)	1.2604	51.46	33.49	36.33	67
x4 Conv block	1.1644	58.13	42.81	45.54	35
x5 Conv block	1.0752	62.37	49.37	52.89	54
x3 Conv block (5x5 Kernel)	1.2031	54.98	39.17	44.19	28
x3 Conv block (Adam)	1.0072	65.53	49.38	52.89	49
x3 Conv block (Glorot uniform)	1.2121	55.58	40.73	45.61	97
x6 Conv block (adamax he_normal)	0.8836	69.54	57.41	62.75	58
x6 Conv block (adamax lecun_normal)	0.9299	69.17	58.00	62.54	69
x5 Conv block (adamax glorot_normal)	0.9088	69.17	57.16	61.67	75
x5 Conv block (adamax truncated_normal)	0.9418	68.33	55.43	61.75	69
x6 Conv block (adamax he_uniform)	0.9228	67.72	55.10	56.82	46
x5 Conv block (adamax he_normal)	0.9254	67.11	55.74	58.97	55
x6 Conv block (adamax truncated_normal)	0.9490	66.75	53.31	58.93	85
x6 Conv block (adamax lecun_uniform)	0.9440	66.14	51.56	60.76	61
x5 Conv block (nadam he_normal)	0.9882	66.14	51.39	59.57	59
x6 Conv block (adagrad he_uniform)	0.9633	65.53	50.37	58.22	70
x6 Conv block (adamax he_normal)	-NA-	60.29	46.57	56.18	100
5-fold Cross Val					

row of the Table 3. Qualitative results of the top model can be found in the Appendix 1.

5. Discussion

The VGG-16 network (with up to 20% dropout) can identify tree species accurately 56% of the time. The literature indicated that the VGG-16 architecture would generalise well to new classification problems and benefit from being pre-loaded with ImageNet weights. However, our custom CNN models with 3 or more convolutional blocks, consistently outperformed the VGG-16 variants that had been trained. The ResNet50 performed slightly better than the VGG-16 however its performance was inferior to our custom made models. This was a surprising result, perhaps indicating that VGG-16 and ResNet50 were over complex for the task which negatively impacted generalization. The top performing custom CNN model consists of 6 convolutional blocks paired with the Adamax optimiser and He Normal kernel initialiser, which achieved 69% accuracy on the test set and average 60.29 accuracy on 5-fold stratified cross validation. The Adamax optimiser was a common parameter across the most successful 8 models. Datasets with many outliers or which are noisy in terms of gradient updates can benefit from Adamax over Adam (Kingma & Ba, 2014). Adamax is a sparse implementation of Adam (Kingma & Ba, 2014)

and in this case was shown to be superior. The VGG-16 and ResNet50 models were only optimised using Adam, so future experiments could explore the effects of using different optimisers here too. Other known architectures such as AlexNet could be trialled in addition to VGG-16 and ResNet50. More convolutional layers increase the number of parameters and have the effect of allowing the model to extract more features – up to a point – after which overfitting tends to occur. This could be a reason for the VGG-16 or ResNet50 networks failing to achieve superior results. Amongst the top models, a pattern emerged to identify that kernel initialisers with normal distributions tended to outperform uniform distributions but its impact was almost negligible. The strategy for constructing a custom CNN could be extended to explore other possibilities such as altering convolutional blocks to contain 3 convolutional layers instead of 2, varying other parameters such as batch size and learning rate. We realized 5-fold stratified cross validation led to a considerable drop across majority of the metrics. This implies that our dataset is not large and homogeneous enough to generate reliable results in hold-out test method. Another reason for the disparity could be that the 5-fold validation process uses 20% of the data for each model to be tested on, whereas 10% is retained for the hold-out test method.

Further investigation shows the top performer model struggles at identifying some tree species such as Ash. We believe this is mainly due to limited number of training samples. This could be mitigated by setting up a hierarchical tree species classification model where a top-level model classifies tree's species family while a separate sub-model will be trained to distinguish between each species of a family. Alternatively, ensemble modelling could be employed – where several models are trained on the data, and their predictions are aggregated to produce a final prediction.

6. Conclusion

This work examined the possibility of generating a labelled tree species dataset using Google Maps aerial images and publicly available tree inventories to supply GPS coordinates and tree species information. Moreover, this study offered a deep convolutional neural network model capable to successfully classify tree species using the proposed dataset. The work involved looking at both transfer learning approach using the VGG-16 and ResNet networks and constructing a series of custom CNN models. The top performer model in this research managed to classify up to 6 different tree species with over 60% average accuracy. Future work such as investigating other pre-trained models under different parameters could likely to improve the metrics. Furthermore, genetic algorithm technique could be adopted to optimise and evolve model parameters and identify the

best performing architecture.

References

Baeten, L. and Bruelheide, H. Identifying the tree species compositions that maximize ecosystem functioning in european forests. *Journal of Applied Ecology*, 56(3), 2018.

Bastin, J.-F., Finegold, Y., Garcia, C., Mollicone, D., Rezende, M., Routh, D., Zohner, C. M., and Crowther, T. W. The global tree restoration potential. *Science*, 365 (6448):76–79, 2019.

Branson, S., Wegner, J. D., Hall, D., Lang, N., Schindler, K., and Perona, P. From Google Maps to a Fine-Grained Catalog of Street trees. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135, 2018.

Cao, K. and Zhang, X. An improved res-unet model for tree species classification using airborne high-resolution images. *Remote Sensing*, 12(7):1128, 2020.

Chandler, K., Stevens, C., Binley, A., and Keith, A. Influence of tree species and forest land use on soil hydraulic conductivity and implications for surface runoff generation. *Geoderma*, 310:120–127, 2017.

Clark, M. L., Roberts, D. A., and Clark, D. B. Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales. *Remote sensing of environment*, 96(3-4): 375–398, 2005.

Council, L. B. o. C. Trees in camden: Open data portal, May 2021. URL <https://opendata.camden.gov.uk/Environment/Trees-In-Camden/csop-kdss>.

Dalponte, M., Ørka, H. O., Ene, L. T., Gobakken, T., and Næsset, E. Tree crown delineation and tree species classification in boreal forests using hyperspectral and als data. *Remote sensing of environment*, 140:306–317, 2014.

Fricker, G. A., Ventura, J. D., Wolf, J. A., North, M. P., Davis, F. W., and Franklin, J. A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. *Remote Sensing*, 11 (19), 2019.

Gamfeldt, L. and Snäll, T. Higher levels of multiple ecosystem services are found in forests with more tree species. *Nature Communications*, 4, 2013.

Kim, S., Schreuder, G., McGaughey, R., and Andersen, H. E. Individual tree species identification using LiDAR intensity data. *Portland: ASPRS 2008 Annual Conference.*, 2008.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv:1412, 6980.*, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Lang, N. Deep learning and google maps for tree monitoring. 2020.

Manickathan, L., Defraeye, T., Allegrini, J., Derome, D., and Carmeliet, J. Parametric study of the influence of environmental factors and tree properties on the transpirative cooling effect of trees. *Agricultural and Forest Meteorology*, 248:259–274, 2017.

Maschler, J., Atzberger, C., and Immitzer, M. Individual tree crown segmentation and classification of 13 tree species using airborne hyperspectral data. *Remote Sensing*, 10 (8):1218, 2018.

Natesan, S., Armenakis, C., and Vepakomma, U. Resnet-based tree species classification using uav images. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019.

Nezami, S., Khoramshahi, E., Nevalainen, O., I., P., and Honkavaara, E. Tree species classification of drone hyperspectral and rgb imagery with deep learning convolutional neural networks. *Remote Sensing*, 12(7), 2020.

Rezatec. Satellites vs. lidar for forestry management? 2020.

Russakovsky, O., Deng, J., and Su, H. e. *ImageNet Large Scale Visual Recognition Challenge*. *International Journal of Computer Vision*, 115, 2015.

Rust, S. Tree inventory, risk assessment and management. In Roloff, A. (ed.), *Urban Tree Management : For the Sustainable Development of Green Cities*, pp. 178–210. John Wiley & Sons, Gottingen, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409, 1556.*, 2014.

Wegner, J. D., Branson, S., Hall, D., Schindler, K., Perona, P., Zurich, E., and Technology, C. I. June). *Cataloging Public Objects Using Aerial and Street-Level Images – Urban Trees*. Retrieved May, 1, 2020. URL <http://www.vision.caltech.edu/publications/ CVPR2016-WegnerBransonEtAl.pdf>.

Wilkes, P., Disney, M., Vicari, M., Calders, K., and Burt, A. Estimating urban above ground biomass with multi-scale lidar. *Carbon Balance Manage*, 13(10), 2018.

Wolf, K. L. Business district streetscapes, trees, and consumer response. *Journal of Forestry*, 103(8):396–400, 2005.

7. Appendices

Appendix 1

Species	Correctly Classified Samples			Misclassified Samples		
Ash						
Silver Birch						
Common Lime						
London Plane						
Norway Maple						
Sycamore						

Qualitative results generated by the top performer model with the 6 convolutional block, Adamax optimiser and He normal kernel initialiser