BB-KI Drone Debris Detection

Jianming Zhou, Manap Shymyr, Sirui Wang ✉, and Yifan Tian

Professorship Big Geospatial Data Management, Department of Aerospace and Geodesy, Technical University of Munich

✉ sirui.wang@tum.de

March 5, 2023

Abstract — This is a project focused on deadwood recognition using Artificial Intelligent (AI) methods. BB-KI-CHIPS project aims at building AI chips and detecting debris by UAV (Unmanned Aerial Vehicle) data. In this project, Random Forest (RF) [1], 1D Convolutional Neural Network (1D CNN), 2D Convolutional Neural Network (2D CNN), Residual Neural Network (Resnet50) [2], Mobile Neural Network (Mobilenet) [3] and Visual Geometry Group Neural Network (VGG16) [4] machine learning methods have been used. And the accuracy was increased to 0.972 by fine tuning work. Finally, the decision fusion achieved better performance with the accuracy of 0.977.

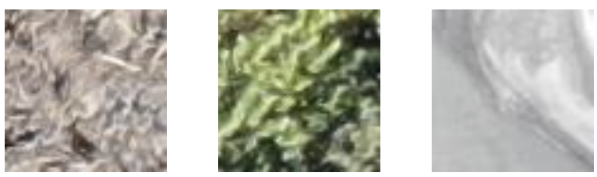
1 Introduction

Deadwood recognition [5] by remote sensing is an efficient technique for disaster monitoring and ecosystem research. Flooding in forests can cause massive damage due to lack of oxygen in the wood roots. Especially the area around the water surface. The debris area usually mixes together with other classes around the boundary of forest and water. The project aims to classify debris, forest, and water into three categories through machine learning methods.

2 Theory

2.1 Machine learning for earth observation

The UAV data contains Red, Green, Blue three channels. Due to the colour difference, forest, debris and water can be separated by machine learning methods. The histograms in Figure 1 show these three classes have different value ranges and peaks at different channels. Therefore, the texture differences can be detected by the CNN algorithms.



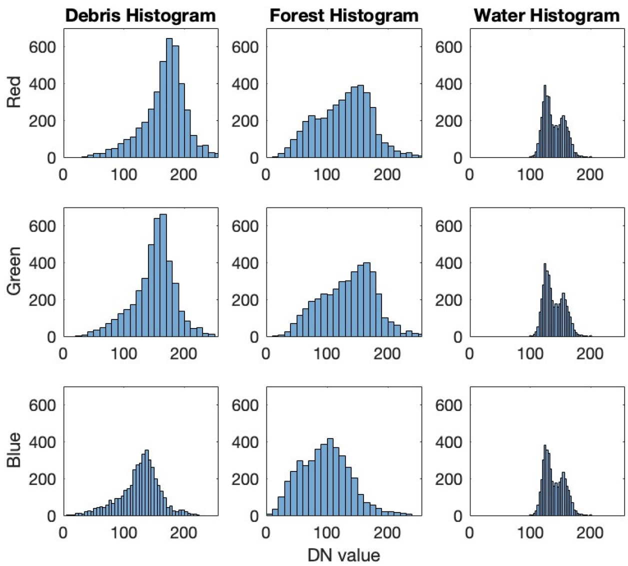


Figure 1 Example tiles and histograms of three classes

3 Experimental Setup

3.1 Datasets preprocessing

The dataset contains 8 images that are obtained by UAV. Each image has the region of deadwood located in the forest. The raw image was visualised by python script and tiled into 64 x 64 small patches.

3.2 Data mining and labelling

In order to get more accurate machine learning models, we labelled each tile for 8 images manually. There are three main steps for data labelling:

1). Open the folder and check how many tiles are there;

2). Check if there are three different sub-folders for classes;

3). Identify the correct classes for tiles and move them into the right sub-folder.

As a result, each sub-folder should contain the right tiles for these classes. We obtained 5879 tiles for forest, 3667 tiles for debris and 3705 tiles for water. The training and validation datasets were splitted randomly by the ratio of 0.8 and 0.2.

3.3 Model selection and model fusion

In this project, six machine learning models with good performances were selected. The methods included in the comparison are summarised as follows:

1). Random Forest [1] is a popular ensemble machine learning algorithm that is used for both regression and classification tasks. The algorithm works by building multiple decision trees.

2). CNNs are specifically designed for image data. They can achieve good results by detecting texture. In this project, we implemented two CNNs by ourselves. We firstly use the simple neural network CNN with a flatten layer as 1D CNN. We have implemented a 2D CNN method by adding 3 stages of 'Conv2D - MaxPool' layers and 2 stages of 'Flattening - Dense' layers.

3). Three pretrained models are also selected in this project. ResNet-50 [2] is one of the deep CNNs. Residual connections that are used to improve the performance. Mobile Network V2 [3] Seeks to perform well on mobile devices. Based on an inverted residual structure where the residual connections are between the bottleneck layers. VGG16 [4] contains 16 layers. It can load a pre-trained version of the network. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

4). Decision fusion [6] is one form of data fusion that combines the decisions of multiple classifiers into a common decision about the activity that occurred [7]. All the neural network models were combined together and achieved a fusion model at the end.

4 Network tuning

4.1 Adjustment of epoch number

Adjustment of epoch number: For deep learning neural networks, it's easy to be over fitting after several epochs. In order to avoid this problem, we defined three callbacks to perform an early stop, they can be seen in Table 1. These three callbacks are suitable for different models and the 'Early stop 3' is usually used for fine tuning. Patience represents the number of epochs with no improvement after which training will be stopped.

Table 1 Options of Early stop callback

Options	Early stop 1	Early stop 2	Early stop 3
Monitor	Validation loss	Validation loss	Validation accuracy
Min delta	0.005	0.005	0.005
Patience	20	10	5
Mode	Min	Min	Min

4.2 Fine tuning

To further improve the performance, the weights of the top layer of the pre-trained model are trained simultaneously. The training process will force the weights to be adjusted from a generic feature map to features associated specifically with the dataset. The goal of fine-tuning is to increase the performance. The example was shown in Figure 2. These are the three main steps for fine tuning, that we conducted:

1) Unfreeze the top layer of the model: Unfreeze the base model and set the bottom layer as non-trainable. Then recompile the model and resume training.

2) Compile model: Use a lower learning rate, otherwise the model may overfit quickly.

3) Continue to train the model: Using early stopping callbacks to detect training convergence improves accuracy by a few percentage points. The loss will decrease and the model will become more stable.

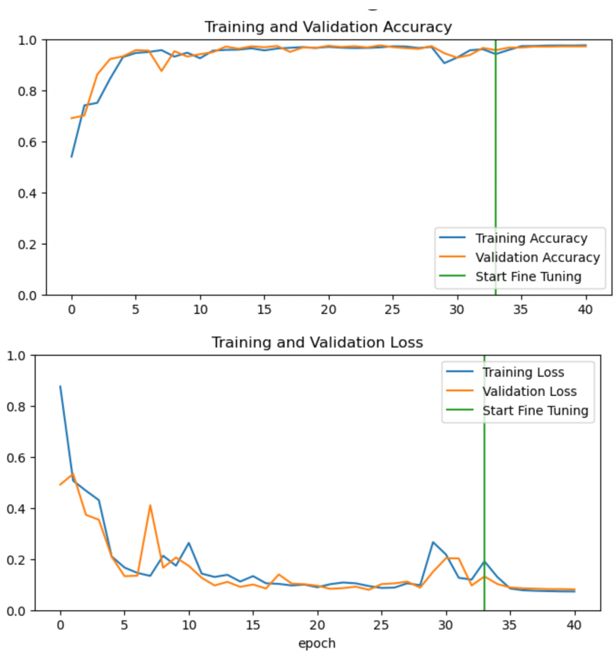


Figure 2 Accuracy and Loss after fine tuning from Mobile Net V2

5 Results

The accuracy results for six machine learning models were shown in Table 2. And the confusion matrix for each deep learning method was shown in Figure 3.

Table 2 Overall accuracy table of different models

Models:	Accuracy:	After tuning:
Random Forest	0.8653	– –
1D Convolutional network (1D CNN)	0.8751	0.9321
Residual Network (ResNet)	0.9634	**0.9727**
2D Convolutional Network (2D CNN)	0.9573	0.9623
Mobile Neural Network (MobileNetV2)	0.9539	**0.9725**
Visual Geometry Group (VGG16)	0.9664	0.9713
Decision fusion	– –	**0.9773**

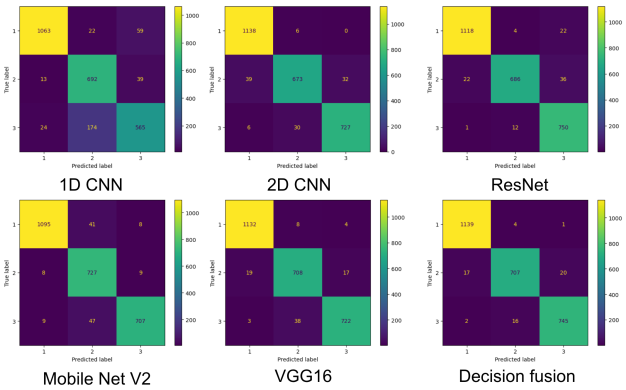


Figure 3 Confusion matrix of each neural networks

6 Conclusion

Random Forest (RF), 1D Convolutional Neural Network (1D CNN), Residual Neural Network (Resnet), 2D Convolutional Neural Network (2D CNN), Mobile Neural Network (Mobilenet) and Visual Geometry Group Neural Network (VGG16) machine learning methods can get good accuracies for UAV data classification. Over fitting problems can be solved by early stopping callbacks. And the accuracy was increased to 0.972 by fine tuning work. Decision fusion achieved better performance 0.977 and can be robust to outliers. During the process of the experiments, we find out the training model works better without mixed patches. For further improvements, we can explore better methods to tile the whole images and decrease the rate of mixed patches. And we can implement more complex networks or try data fusion and feature fusion to increase the quality. More information for our project follows:

1). Group cooperation: We separate the work packages and make a collaboration with each group member in this project. The work packages table was shown in Table 3.

2). Datasets available in:

https://drive.google.com/drive/folders/1f0yXys4zgmnzXfFUFGaAYO9xfc2cY3XJ?usp=sharing

Table 3 Work packages table

Name	Work packages
Jianming Zhou	Make training set(3 images); Analyze results
Manap Shymyr	Data labeling; Model selection; Writing LateX scripts
Sirui Wang	Model training; Model fusion; Fine tuning; Writing LateX
Yifan Tian	Literature research; Data labeling(4 images); Formal analysis

3). Github repositories:

https://github.com/SiruiWang0731/BBKI_drone_debris_detection

References

[1] L. Breiman, "Random forests," pp. 5–32, 2001.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] K. Zielewska-Büttner, P. Adler, S. Kolbe, R. Beck, L. Ganter, B. Koch, and V. Braunisch, "Detection of standing deadwood from aerial imagery products: Two methods for addressing the bare ground misclassification issue," *Forests*, vol. 11, p. 801, 07 2020.

[6] C. Qiu, X. Tong, M. Schmitt, B. Bechtel, and X. X. Zhu, "Multilevel feature fusion-based cnn for local climate zone classification from sentinel-2 images: Benchmark results on the so2sat lcz42 dataset," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2793–2806, 2020.

[7] E. J. Hoffmann, Y. Wang, M. Werner, J. Kang, and X. X. Zhu, "Model fusion for building type classification from aerial and street view images," *Remote Sensing*, vol. 11, 6 2019.