

23 DECEMBER 2019



USER MANUAL CNNVisTool (CNN VISUALIZATION TOOL)

A SOFTWARE DEVELOPED UNDER PG09
ALGORITHMS AND SOFTWARE FOR OBJECT RECOGNITION

OLEE HOI YING LAM
OLEELAMHY@GMAIL.COM
SPECTORS – HOCHSCHULE RHEIN-WAAL

Table of Contents

ABOUT THIS DOCUMENT	2
OVERVIEW OF THE SOFTWARE	3
Features of the software.....	3
System requirements.....	3
Interface of the software	4
Console Output Window	4
Graphical User Interface (GUI).....	4
GETTING STARTED	6
Obtain the software.....	6
Workflow of the software	6
Preparation of trained model	6
Select/Change model.....	7
Select/Change image directory.....	8
Navigate between images	9
“Save Prediction” Text File	10
ACKNOWLEDGEMENT.....	12
COPYRIGHT INFORMATION.....	12

About this document

This user manual is a guide to all aspects of using CNNVisTool: Convolutional Neural Network Visualization Tool (version 2.2.0), an open source software written in Python 3 that allows users to load different trained convolutional neural network model for classifying images. The software is designed to provide visualization chart to understand the output of the trained model, and to save the predictions for further analysis.

The manual focuses on the practical usage of CNNVisTool – if you are looking for information on the functionality of this software, this is the guide that would provide you the information you need. If you are a developer seeking details of the source code, feel free to visit the GitHub repository (<https://github.com/oleelamhy/CNNVisTool>), which has been made available since version 2.2.0.

Overview of the Software

CNNVisTool stands for “Convolutional Neural Network Visualization Tool”. CNNVisTool is an open source software written in Python 3 that allows users to load different trained convolutional neural network model (VGG16 model using TFlearn) for classifying images. The software is designed to provide visualization chart to understand the output of the trained model, and to save the predictions for further analysis.

CNNVisTool is developed within the scope of an INTERREG V-A Deutschland-Nederland funded project SPECTORS (“Sensor Products for Enterprises Creating Technological Opportunities in airborne Remote Sensing”; project number 143081). And was originally designed for an image classifier trained to classify weed species in grassland, in order to assist volunteers and non-experts in recognizing any targeted weed species in the field.

Features of the software

CNNVisTool consists of the following features:

- Display the prediction outputs of a trained image classifier as bar chart
- Select trained model, display the selected model name, and the number of classes together with the class names which are automatically loaded from the corresponding text file [*Note* currently only VGG16 model trained with TFlearn (TensorFlow) is supported]
- Select folder with images, display the selected folder name, and the first image together with the image name
- Save prediction outputs for all images into a text file
- Navigate images within the selected folder for prediction
- Display the position of the displayed image in the selected folder

System requirements

The followings are the minimum system requirements to run the software.

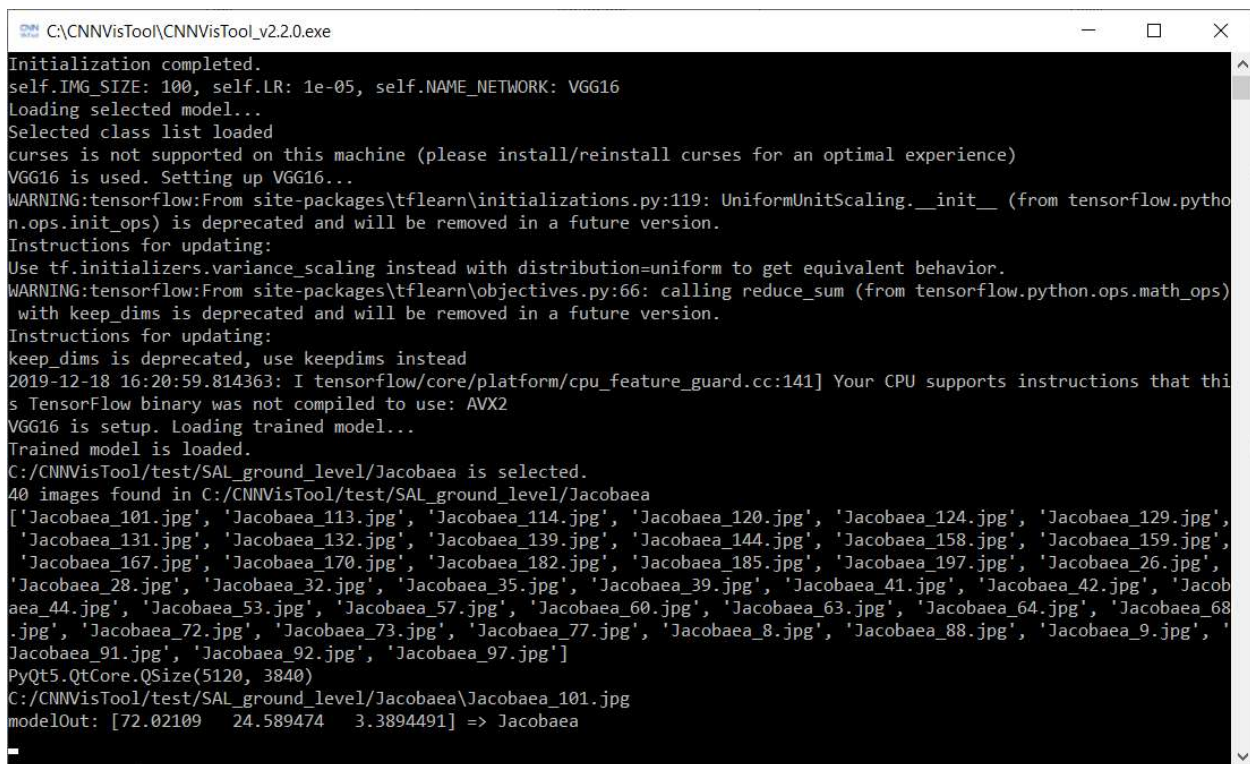
- Windows 7
- i3 processor (CPU)
- 64-bit operating system

Interface of the software

When CNNVisTool is executed, two interfaces will be prompted. The first one is a console output window where the processing output and error messages are displayed. The second one is the graphical user interface (GUI) where all functionalities of the software are performed.

Console Output Window

The console output window is merely acted as an output window for user to better understand the backend process of the software. The window displays the logs of each process, and provides user slightly more feedback on top of the message displayed on the status bar of the GUI.



```
C:\CNNVisTool\CNNVisTool_v2.2.0.exe
Initialization completed.
self.IMG_SIZE: 100, self.LR: 1e-05, self.NAME_NETWORK: VGG16
Loading selected model...
Selected class list loaded
curses is not supported on this machine (please install/reinstall curses for an optimal experience)
VGG16 is used. Setting up VGG16...
WARNING:tensorflow:From site-packages\tflearn\initializations.py:119: UniformUnitScaling.__init__ (from tensorflow.python.ops.init_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.initializers.variance_scaling instead with distribution=uniform to get equivalent behavior.
WARNING:tensorflow:From site-packages\tflearn\objectives.py:66: calling reduce_sum (from tensorflow.python.ops.math_ops) with keep_dims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
2019-12-18 16:20:59.814363: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
VGG16 is setup. Loading trained model...
Trained model is loaded.
C:/CNNVisTool/test/SAL_ground_level/Jacobaea is selected.
40 images found in C:/CNNVisTool/test/SAL_ground_level/Jacobaea
['Jacobaea_101.jpg', 'Jacobaea_113.jpg', 'Jacobaea_114.jpg', 'Jacobaea_120.jpg', 'Jacobaea_124.jpg', 'Jacobaea_129.jpg', 'Jacobaea_131.jpg', 'Jacobaea_132.jpg', 'Jacobaea_139.jpg', 'Jacobaea_144.jpg', 'Jacobaea_158.jpg', 'Jacobaea_159.jpg', 'Jacobaea_167.jpg', 'Jacobaea_170.jpg', 'Jacobaea_182.jpg', 'Jacobaea_185.jpg', 'Jacobaea_197.jpg', 'Jacobaea_26.jpg', 'Jacobaea_28.jpg', 'Jacobaea_32.jpg', 'Jacobaea_35.jpg', 'Jacobaea_39.jpg', 'Jacobaea_41.jpg', 'Jacobaea_42.jpg', 'Jacobaea_44.jpg', 'Jacobaea_53.jpg', 'Jacobaea_57.jpg', 'Jacobaea_60.jpg', 'Jacobaea_63.jpg', 'Jacobaea_64.jpg', 'Jacobaea_68.jpg', 'Jacobaea_72.jpg', 'Jacobaea_73.jpg', 'Jacobaea_77.jpg', 'Jacobaea_8.jpg', 'Jacobaea_88.jpg', 'Jacobaea_9.jpg', 'Jacobaea_91.jpg', 'Jacobaea_92.jpg', 'Jacobaea_97.jpg']
PyQt5.QtCore.QSize(5120, 3840)
C:/CNNVisTool/test/SAL_ground_level/Jacobaea/Jacobaea_101.jpg
modelOut: [72.02109 24.589474 3.3894491] => Jacobaea
```

Graphical User Interface (GUI)

The graphical user interface (GUI) is the interface where user performs all the functionalities provided by CNNVisTool. To ensure easy operations, the software is designed to be operated through button clicking.



"Select model" button: This button is on the top of the interface, allows user to select which trained model to be used. The name of the selected trained model is displayed next to the button, whereas the corresponding information on the image categories will be shown above the button.

"Select folder" button: This button is beneath the "Select model" button, allows user to select which image folder to process. The name of the selected folder is displayed next to the button.

"Image Preview" panel: The left panel is used to display the image currently passed to the trained model. Underneath the displayed image, the image name is shown on the left, while on the right indicates the position of the image and the total number of images in the selected folder.

"Predictions" panel: The right panel is used to display the bar chart after the trained model makes a prediction on the displayed image. Through the bar chart, the user is able to observe the probabilities of the displayed image being in each class.

"Previous" and "Next" buttons: These two navigation buttons at the bottom right of the "Predictions" panel allow user to freely navigate between images within the selected folder.

"Save Predictions" button: This button at the bottom left of the "Predictions" panel enables user to save the image names and corresponding predictions of the selected folder into a text file. The text file is saved in the selected folder, with the file name "<date>_<time>_predictions.txt".

Status bar: The status bar at the bottom of the interface aims at providing critical information concerning initialization, model selection, folder selection, and predictions.

Getting started

Obtain the software

The standalone executable (.exe) for Windows can be obtained from the Release page of the GitHub repository: <https://github.com/oleelamhy/CNNVisTool/releases>

For minimum system requirements, please refer to the Section [“System requirements”](#).

Workflow of the software

CNNVisTool is designed to visualize outputs of a trained CNN model of selected images, by simply clicking buttons. The workflow can be described as follow.

1. Select a trained model (currently only supporting VGG16 model based on TFlearn)
2. Select an image directory
3. Navigate through the selected directory to visualize the prediction outputs for each image
4. Save the predictions for all images within the selected directory into a text file

Preparation of trained model

Once an image classifier has been trained inside a Python 3 environment, preferably TensorFlow 1.11.0 and TFlearn 0.3.2, three separate files should be generated when the model is saved, with the extensions “.tfl.data-00000-of-00001”, “.tfl.index” and “.tfl.meta”.

For CNNVisTool to automatically pick up the parameters for setting up the CNN architecture, the trained model files should be named in the format such as “20180418-Salmorth-x-60-0.25-1e-05-VGG16-17epoch_run-1”, which corresponds to the following:

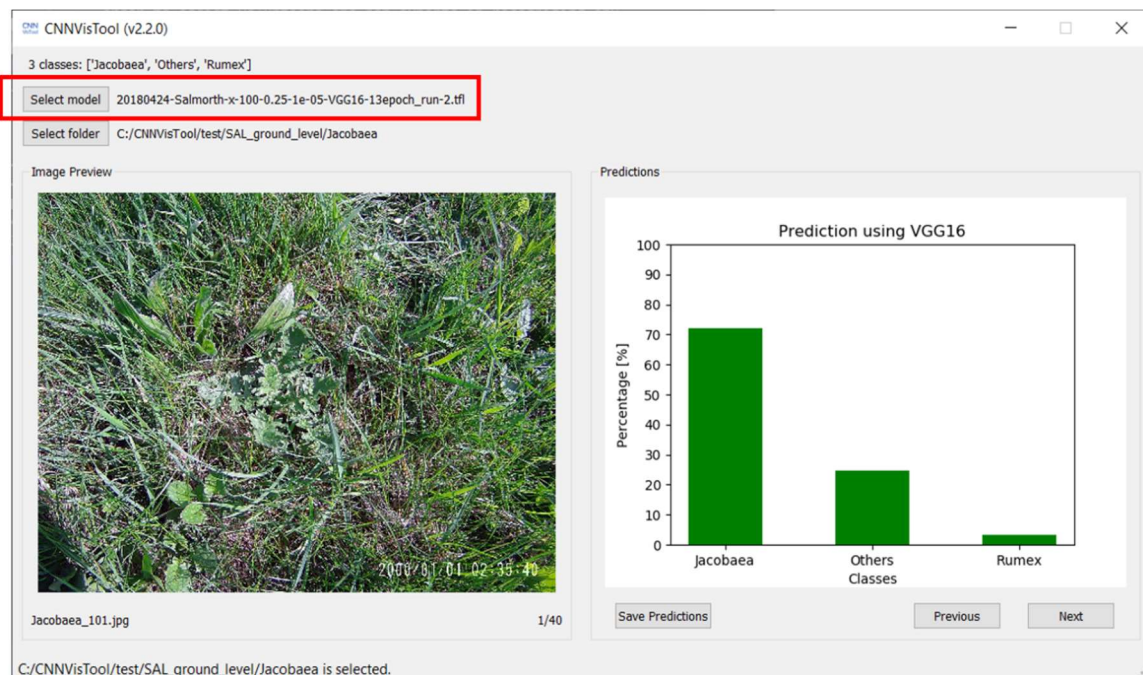
```
“<date_of_training>-<python_script_name>-<smallest_sample_size>-  
<input_image_size>-<train_valid_ratio>-<learning_rate>-<architecture_name>-  
<#>epoch_run-<#>”
```

Despite only the three bolded parameters (input image size, learning rate, CNN architecture’s name) are essential for CNNVisTool version 2.2.0 to load the model properly, the above naming convention is suggested for user to better understand the trained models from the file name. This is useful to help user selecting the most appropriate model for image prediction later on.

A text file having the same naming as the trained model “20180418-Salmorth-x-60-0.25-1e-05-VGG16-17epoch_run-1.txt” is also necessary for CNNVisTool to obtain the list of class

categories for the corresponding trained model. The name of the categories should be written in the first line of the text file using “,” as separators. The order of the classes should be the same as when the model was trained; therefore it is highly suggested to generate this text file during the training. An example of a class list would be “Jacobaea,Others,Rumex”.

Select/Change model



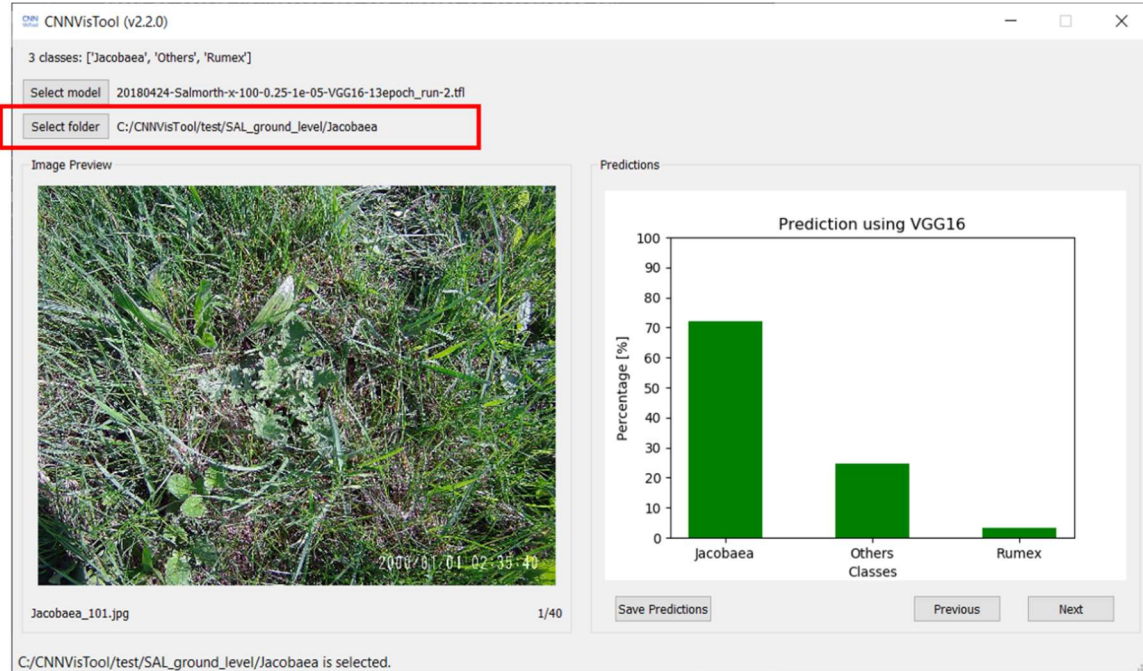
When the “Select model” button is clicked, a dialog will be prompted to select a trained model (.tfl file). Select a trained model that fits the naming conventions stated above in the “[Preparation of trained model](#)” section.

When the trained model (.tfl) is selected –

- (1) The corresponding text file is read and the class list, together with the number of classes, will be shown above the button;
- (2) The software will pick up the parameters such as input image size, learning rate and the name of the CNN architecture to set up the neural network accordingly in order to load the trained model for predictions afterwards; and
- (3) The name of the selected trained model will be displayed next to the button.

Note that user is allowed to change to different trained model when necessary, but up till version 2.2.0, CNNVisTool only supports VGG16 architecture.

Select/Change image directory



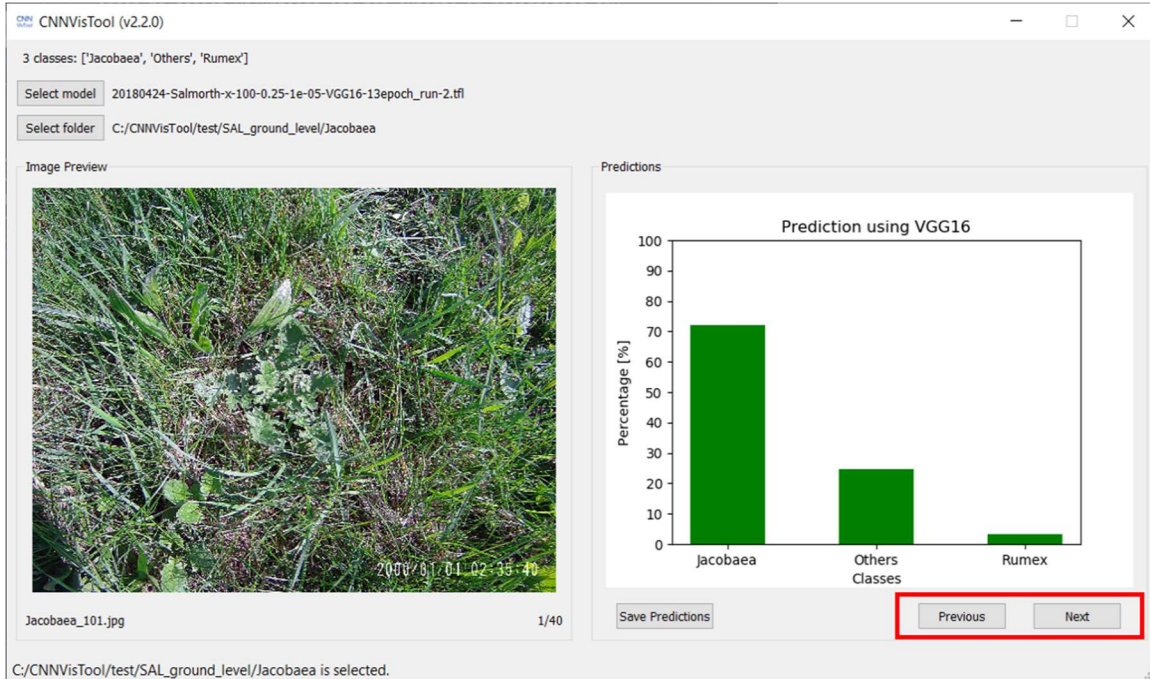
When the “Select folder” button is clicked, a dialog will be promoted to select a folder. Select a folder that contains the images for classification.

When images (.jpg and .png) are detected –

- (1) The name of the selected directory will be shown next to the button;
- (2) The first image and its name will be displayed in the “Image Preview” panel, as well as its position in the selected directory;
- (3) If a trained model has already been selected, the displayed image is resized to the input image size as stated in the trained model file name and passed to the trained image classifier; and
- (4) The trained model makes a prediction of the image, and the probability of that image being in each class is displayed as a bar chart in the “Predictions” panel. The name of the CNN architecture is shown in the title of the bar chart.

User is allowed to change the trained model and image directory at any time of the operations, once a different model or directory has been selected, the “Image Preview” and “Predictions” panels are refreshed and updated accordingly.

Navigate between images



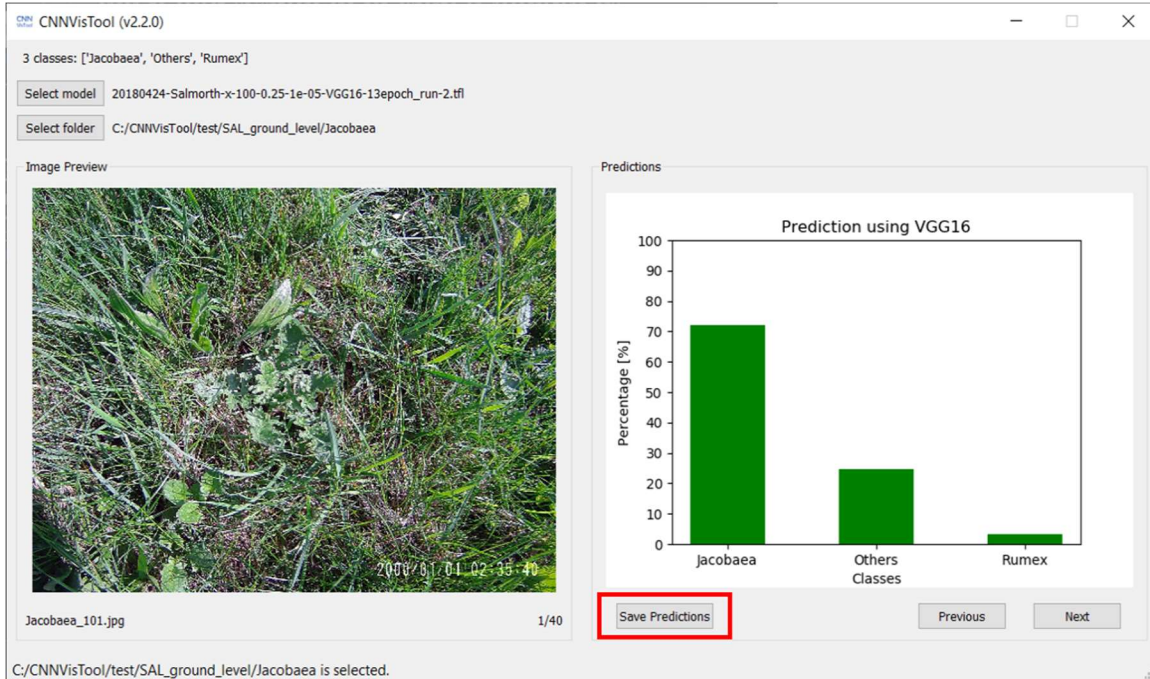
To navigate between different images within the selected directory, the navigation buttons at the bottom right of the “Predictions” panel can be used.

Once the “Previous” or “Next” button is pressed –

- (1) The previous or the next image, if exists, will be displayed on the “Image Preview” panel. The corresponding image name and its position within the selected directory will be updated as well;
- (2) If a trained model has already been selected, the displayed image is resized to the input image size as stated in the trained model file name and passed to the trained image classifier; and
- (3) The trained model makes a prediction of the image, and the probability of that image being in each class is displayed as a bar chart in the “Predictions” panel. The name of the CNN architecture is shown in the title of the bar chart.

User is allowed to change the trained model and image directory at any time of the operations, once a different model or directory has been selected, the “Image Preview” and “Predictions” panels are refreshed and updated accordingly.

“Save Prediction” Text File



Since version 1.1.0, user can choose to save the prediction results of the selected directory into a text file (.txt) by pressing the “Save Predictions” button at the bottom left of the “Predictions” panel.

Provided a trained model is selected, when the button is clicked –

- (1) A text file named “<date>_<time>_predictions.txt” is created under the selected directory, and the header information is written;
- (2) The image list within the selected directory is obtained;
- (3) Starting from the first image of the selected directory, the image is resized to the input image size as stated in the trained model file name and passed to the trained image classifier in the background, i.e. the GUI will not be updated or refreshed during this process;
- (4) The prediction result of the processed image is written in the text file together with the image name;
- (5) The process is continued until the last image of the selected directory has been classified by the image classifier; and
- (6) The number of images classified in each predicted class is written in the text file before saving.

```
20191221_115016_predictions.txt - Notepad
File Edit Format View Help
=====
File created on: 21.12.2019 11:50:16
Application version: 2.2.0 (For more information, please refer to the user manual.)
Model name: 20180424-Salmorth-x-100-0.25-1e-05-VGG16-13epoch_run-2.tfl
Image directory: C:/CNNVisTool/test/SAL_ground_level/Jacobaea

Data description:
1st column: image name
2nd column: an array specifying the prediction percentage of each class ['Jacobaea', 'Others', 'Rumex']
3rd column: predicted class, i.e. the class with the highest percentage
=====
Jacobaea_101.jpg [72.02109 24.589474 3.3894491] Jacobaea
Jacobaea_113.jpg [66.9719 28.819756 4.2083488] Jacobaea
Jacobaea_114.jpg [63.908577 28.869331 7.222093] Jacobaea
Jacobaea_120.jpg [37.160694 9.092867 53.746445] Rumex
Jacobaea_124.jpg [59.084682 39.411087 1.5042264] Jacobaea
Jacobaea_129.jpg [62.79243 1.0610046 36.14657 ] Jacobaea
Jacobaea_131.jpg [93.10926 2.9586153 3.9321249] Jacobaea
Jacobaea_132.jpg [77.8142 3.1759243 19.009874 ] Jacobaea
Jacobaea_139.jpg [75.765045 2.6687968 21.566158 ] Jacobaea
Jacobaea_144.jpg [86.06607 3.2287307 10.705198 ] Jacobaea
Jacobaea_158.jpg [42.178326 1.5325819 56.2891 ] Rumex
Jacobaea_159.jpg [75.68203 1.7853048 22.532671 ] Jacobaea
Jacobaea_167.jpg [81.44271 0.97449523 17.5828 ] Jacobaea
Jacobaea_170.jpg [76.70821 2.3773317 20.914452 ] Jacobaea
Jacobaea_182.jpg [60.041744 1.6292881 38.328964 ] Jacobaea
Jacobaea_185.jpg [25.618616 0.5343341 73.847046 ] Rumex
Jacobaea_197.jpg [49.8856 0.7981764 49.31622 ] Jacobaea
Jacobaea_26.jpg [69.71432 29.551956 0.73371595] Jacobaea
Jacobaea_28.jpg [80.786255 16.201683 3.0120668] Jacobaea
Jacobaea_32.jpg [54.525055 42.429504 3.045439] Jacobaea
Jacobaea_35.jpg [86.6959 12.187236 1.1168597] Jacobaea
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

In the text file, the following data is stored:

- File creation date and time
- Application version
- Name of the selected trained model
- Path of the selected image directory
- Data description
- Prediction results including the image name, prediction output and the predicted class
- The total number of images in the selected directory
- The number of images being classified in each class

Acknowledgement

CNNVisTool is developed as part of the SPECTORS Project, a Dutch-German cooperation project funded by INTERREG V-A Deutschland-Nederland (project number 143081).

Copyright information

© Copyright 2019, Olee Hoi Ying Lam. License under [GNU GPLv3](#). The information contained in this document is subject to change without notice.