Neural Networks Based Recognition of the Species and Subspecies of Vegetable Leaves

Ricardo Diamantino Oliveira Belinha

West University of Timișoara
Faculty of Mathematics and Informatics
Master study program: Artificial Intelligence and Distributed Computing

Scientific Coordinator: Conf. Dr. Kaslik Eva

September, 2022

FMI

Overview

Motivation

Thesis structure

Problem Specification

Related Work

Application Functionalities

Results

Conclusion and Future Work

Motivation

- Demonstrate the procedure of creating a neural network capable of correctly classifying vegetable leaves by utilizing MATLAB as the integrated development environment;
- Get familiar with MATLAB;
- Curiosity about leaves and to assist future apprentices in this domain in classifying leaves more easily and quickly.

Thesis structure

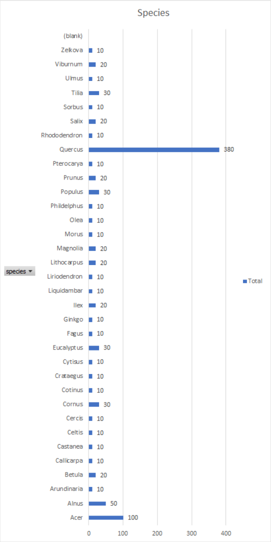
- Problem Description;
  1. Neural network;
  2. Introduction to the project;
  3. Goal of the project.
- Related Work;
  1. Similar products;
  2. Analysis and comparison.
- Results
- Application Functionalities
  1. User guide;
  2. Developer guide.

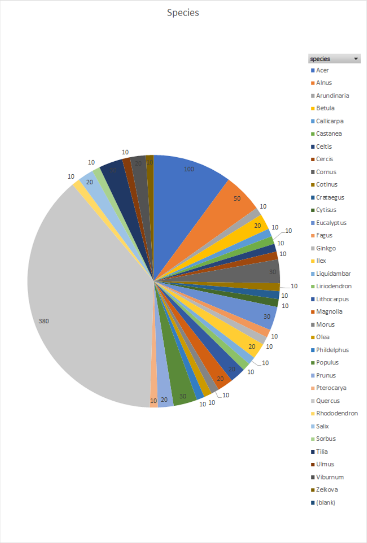
Problem Specification

- The data which will be used to train the neural networks is the content of three folders.
- The folder:
  - ”Leaves_1” contains 99 images;
  - ”Leaves_2” contains 1564 images;
  - ”Leaves_3” contains 20 images;

Problem Specification

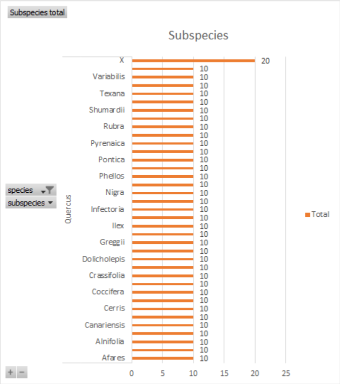
- All the images in the three folders consist of 34 different species.

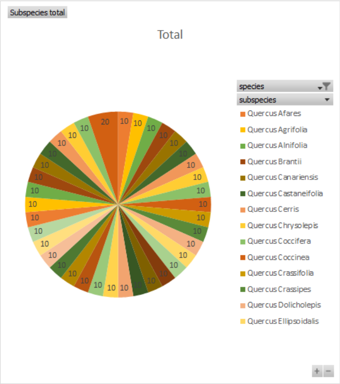




Problem Specification

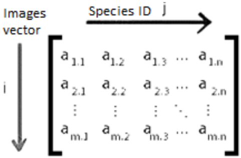
- All the images in the three folders consist of 98 different subspecies.





Problem Specification

- In order to train the neural network with those images, there was a need to map the images inside the folders with their correspondent specie and sub specie.



$$\begin{bmatrix} a_{1.1} & a_{1.2} & a_{1.3} & \cdots & a_{1.n} \\ a_{2.1} & a_{2.2} & a_{2.3} & \cdots & a_{2.n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m.1} & a_{m.2} & a_{m.3} & \cdots & a_{m.n} \end{bmatrix}$$

- And afterwards, with the mapping, the neural network can learn how to classify the leaves by using the images in binary or their characteristics.

Problem Specification

- Leaves characteristics:

Characteristics	Description
ConvexArea	The number of pixels in "ConvexImage" is returned as a scalar.
Eccentricity	The eccentricity of the ellipse sharing the region's second moments returns as a scalar. The eccentricity of an ellipse is defined as the proportion of the distance between its foci to the length of its main axis. The value ranges from 0 to 1 (where 0 and 1 are degraded situations). An ellipse with an eccentricity of 0 is a circle, but one with an eccentricity of 1 is a line segment.

Source: https://www.mathworks.com/help/images/ref/regionprops.html

Problem Specification

- Leaves characteristics:

Characteristics	Description
Extent	The ratio of pixels inside the area to those within the overall bounding box is returned as a scalar. Computed as the area of the bounding box divided by the size of the bounding box.
EquivDiameter	Returns the radius of a circle with the same area as the region as a scalar. Calculated as sqrt(4*area/pi).
FilledArea	The number of pixels in the "FilledImage" is returned as a scalar.

Source: https://www.mathworks.com/help/images/ref/regionprops.html

Problem Specification

- Leaves characteristics:

Characteristics	Description
MajorAxisLength	The main axis of the ellipse with the same standardized second central instants as the area is returned as a scalar with its length (in pixels).
MinorAxisLength	The minor axis of the ellipse with the same standardized second central instants as the area is returned as a scalar with its length (in pixels).
Orientation	The angle between the x-axis and the main axis of the ellipse is returned as a scalar with the same second instants as the area. The value is in degrees and ranges from -90 to 90.

Source: https://www.mathworks.com/help/images/ref/regionprops.html

Problem Specification

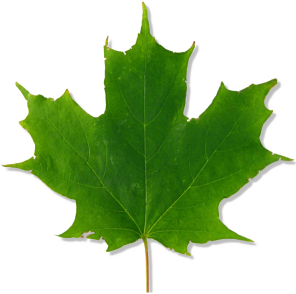
- Leaves characteristics:

Characteristics	Description
Perimeter	The distance around the area's border is returned as a scalar. The perimeter of an area is determined by calculating the distance between each pair of pixels (not broken) along the region's boundary. If the picture includes areas that are not contiguous, regionprops produce surprising results. This graphic depicts the pixels that were used to calculate the perimeter of this item.
Solidity	A scalar representing the ratio of pixels in the convex framework that are also in the area is returned. It is calculated as "Area" or "ConvexArea".

Source: https://www.mathworks.com/help/images/ref/regionprops.html

Problem Specification

- A well trained neural network should identify this image as:
  - Specie: Acer
  - Subspecie: Saccharum

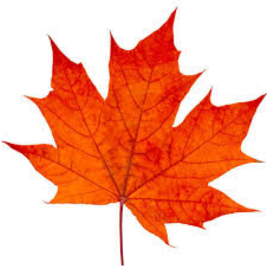


Problem Specification

- A well trained neural network should identify this image as:
  - Specie: Quercus
  - Subspecie: Lobata

Problem Specification

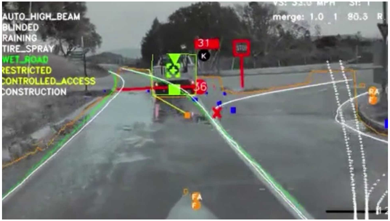
- A well trained neural network should recognize and identify this leaf:
  - Specie: ?
  - Subspecie: ?



Related Work

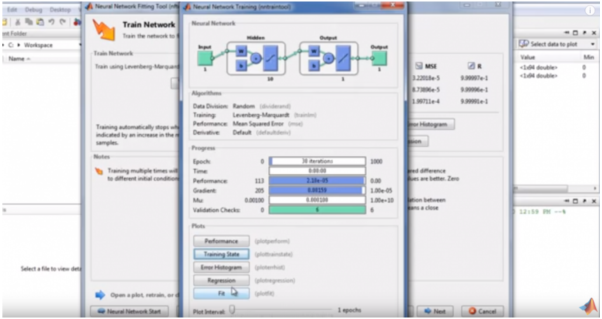
Examples:

- Social networks
- Covid masks verifier
- Tesla: Autopilot
- Shutterstock: Image Composition AI
- Google: Reverse Image Search
- Google: Photos' location



Application Functionalities - Developer guide:

- MATLAB
  - Deep Learn Toolbox
  - MATLAB GUI
  - MATLAB Application Compiler
- Excel



Train Network

Train using Levenberg-Marquard

Training automatically stops wh indicated by an increase in the m samples.

Notes

Training multiple times will to different initial condition

Open a plot, retrain, or d

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	38 iterations	1000
Time:		0:00:00	
Performance:	113	2.18e-05	0.00
Gradient:	205	0.00159	1.00e-05
Mu:	0.00100	0.000100	1.00e-10
Validation Checks:	0	6	6

MSE	R
3.22018e-5	9.99997e-1
8.73896e-5	9.99996e-1
1.99711e-4	9.99991e-1

ared difference
lues are better. Zero

lation between
eans a close

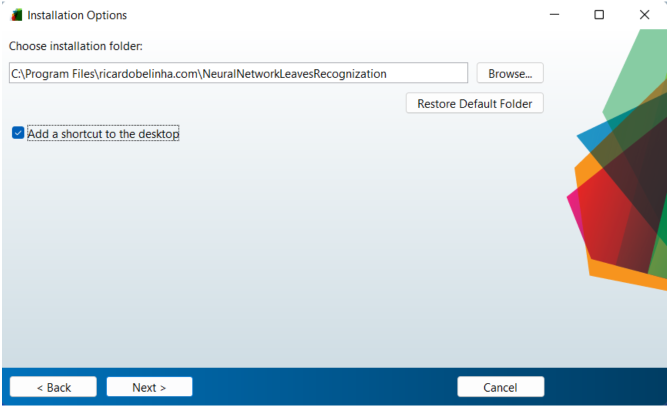
Value	Min
<1x94 double>	0
<1x94 double>	0

12:59 PM --%

Source: https://www.mathworks.com/discovery/matlab-gui.html

Application Functionalities - User guide: Installer

- The user opens the installer through its executable, and the installer will install:
  - all the prerequisites to run the software;
  - the software itself.



Installation Options

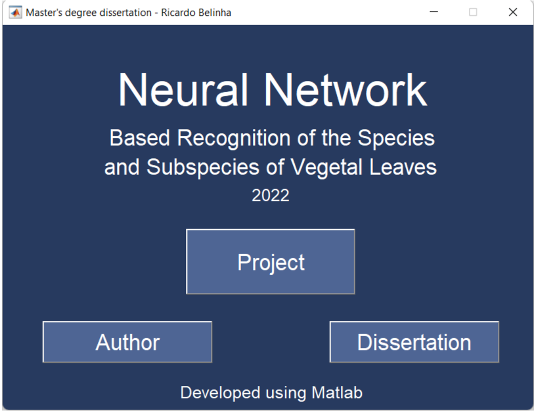
Choose installation folder:

C:\Program Files\ricardobelinha.com\NeuralNetworkLeavesRecognization

☑ Add a shortcut to the desktop

Application Functionalities - User guide: Launcher

- When the user opens the application, a main menu is shown, and it is possible to:
  - Run the application;
  - Check the author's details;
  - Open the dissertation paper in ".PDF" format.



Master's degree dissertation - Ricardo Belinha

Neural Network

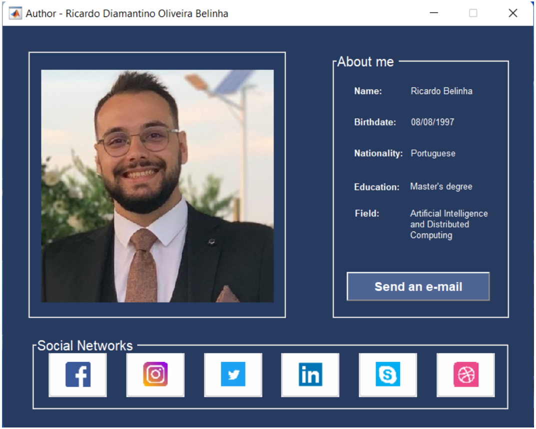
Based Recognition of the Species and Subspecies of Vegetal Leaves

2022

Developed using Matlab

Application Functionalities - User guide: Author's details

- If the user chooses to show the author's details, the output below is shown:



Author - Ricardo Diamantino Oliveira Belinha

About me

Name: Ricardo Belinha

Birthdate: 08/08/1997

Nationality: Portuguese

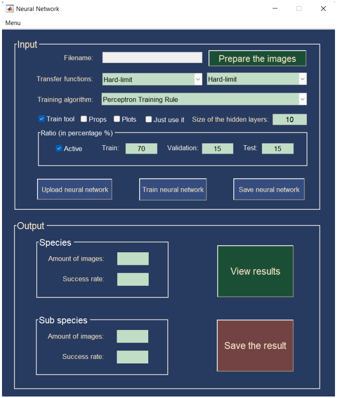
Education: Master's degree

Field: Artificial Intelligence and Distributed Computing

Application Functionalities - User guide: Neural network

Before training a neural network, the user can:

1. Upload an existing neural network;
2. Configure the entire training:
   - input;
   - transfer functions for both layers;
   - training algorithm;
   - train with the images in binary or characteristics of the leaves;
   - Size of the hidden layers;
   - Ratios for training, validating, and testing.
3. Start the training for the neural network;



Filename:

Transfer functions: Hard-limit | Hard-limit

Training algorithm: Perceptron Training Rule

☑ Train tool ☐ Props ☐ Plots ☐ Just use it Size of the hidden layers: 10

Ratio (in percentage %)

☑ Active Train: 70 Validation: 15 Test: 15

Species

Amount of images:

Success rate:

Sub species

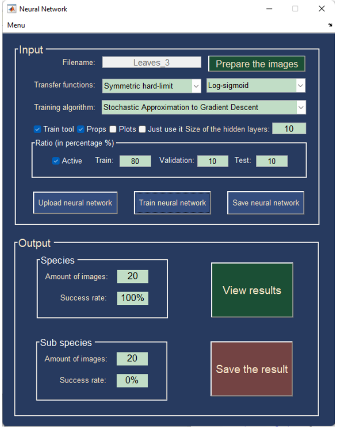
Amount of images:

Success rate:

Application Functionalities - User guide: Neural network

After training a neural network, the user can:

1. Check both output results for the species and subspecies;
2. Save the neural network;
3. Save the output results in the excel file;
4. Open the excel file and check the result for all performed trainings.



Filename: Leaves_3

Transfer functions: Symmetric hard-limit | Log-sigmoid

Training algorithm: Stochastic Approximation to Gradient Descent

☑ Train tool ☑ Props ☐ Plots ☐ Just use it Size of the hidden layers: 10

Ratio (in percentage %)

☑ Active Train: 80 Validation: 10 Test: 10

Species

Amount of images: 20

Success rate: 100%

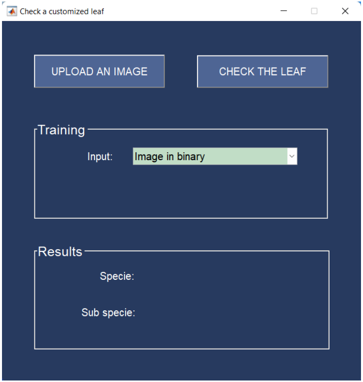
Sub species

Amount of images: 20

Success rate: 0%

Application Functionalities - User guide: Neural network

- The user can also upload a leaf and classify it by using one of the saved neural networks.



Check a customized leaf

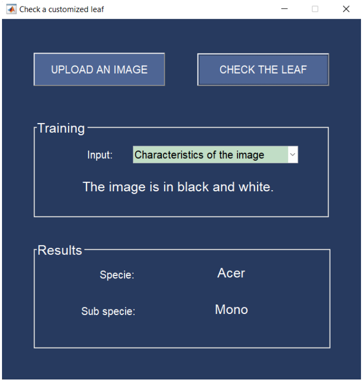
Training

Input: Image in binary

Results

Specie:

Sub specie:



Check a customized leaf

Training

Input: Characteristics of the image

The image is in black and white.

Results

Specie: Acer

Sub specie: Mono

Results

- Both neural networks:
  - tested 1082 times in total, with different types of configurations;
  - only achieved a 100% success rate when classifying the species for the leaves from the folder "Leaves_3".

Results for the neural network using as input the images in binary

- Achieved 13 times, 100% success rate in total;
- 10 times of those 13, were using Perceptron Training Rule and Stochastic Approximation to Gradient Descent as training algorithms (5 times each of them);
- The transfer functions that most achieved 100% success rate were the Linear and the Hyperbolic tangent sigmoid;
- Using 10 as hidden layer size achieved one more 100% success rate than using only 2.

Results for the neural network using as input the characteristics of the images

- Achieved 11 times, 100% success rate in total;
- 6 times of those 11, were using Stochastic Approximation to Gradient Descent as training algorithm;
- The transfer function that most achieved 100% success rate was the Hyperbolic tangent sigmoid which achieved it 6 times;
- Using 10 as size for the hidden layers achieved 7 times 100% success rate.

Results - Conclusion

- Since the input data was exactly the same to train both neural networks, it can be seen that Stochastic Approximation to Gradient Descent was the best training algorithm, Hyperbolic tangent sigmoid was the best transfer function, and 10 was the best size for the hidden layers.
- All the completed trainings were recorded in the Excel file and also added as an appendix to the dissertation paper.

Conclusion and Future Work

Conclusion

- Exhibited the process of developing a neural network that recognizes the specie and sub specie of a vegetable leaf;
- Produced two fully functional neural networks that are capable of training using numerous configurations and classifying the specie and sub specie of a leaf as a final result;
- Made an improvement in phytomorphology, combining technology and biology, contributing to the continuous development of science.

Future work

- Adapt this script to be able to run on all platforms and add augmented reality to it;
- For individuals who are passionate about leaves but have acrophobia, this software will also be great because it will be available to be used by drones.