ECE 578 Fall 2020

Assignment 3

Michael Escue

Contents

[List of Tables 1](#_Toc58155593)

[List of Figures 1](#_Toc58155594)

[List of Equations 1](#_Toc58155595)

[List Of Functions: 1](#_Toc58155596)

[Summary 2](#_Toc58155597)

[References 2](#_Toc58155598)

# List of Tables

[Table 1 – Used Modules & Description 1](#_Toc58169735)

# List of Figures

[Figure 1 - Orange Canvas View of Processing Algorithm. 2](#_Toc58169728)

# Scripting Environment

Python 3.8.6

# List Of Python Modules:

Table – Used Modules & Description

|  |  |
| --- | --- |
| **module** | **Description** |
| cgi 2.6 | Support module for Common Gateway Interface (CGI) scripts. <https://docs.python.org/3/library/cgi.html> |
| typing 3x | This module provides runtime support for type hints as specified by [**PEP 484**](https://www.python.org/dev/peps/pep-0484), [**PEP 526**](https://www.python.org/dev/peps/pep-0526), [**PEP 544**](https://www.python.org/dev/peps/pep-0544), [**PEP 586**](https://www.python.org/dev/peps/pep-0586), [**PEP 589**](https://www.python.org/dev/peps/pep-0589), and [**PEP 591**](https://www.python.org/dev/peps/pep-0591). <https://docs.python.org/3/library/typing.html> |
| Orange 3.27.1 | Orange is a machine learning and data mining suite for data analysis through Python scriptingand visual programming. <https://orange.biolab.si/citation/> |
| cv2 4.4.0 | Wrapper package for OpenCV python bindings.<https://pypi.org/project/opencv-python/#description> |
| imutils 0.5.3 | A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3. <https://pypi.org/project/imutils/> |
| numpy 1.18.5 | NumPy is the fundamental package for array computing with Python. <https://pypi.org/project/numpy/> |
| sklearn 0.23.2 | *Scikit-learn* is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. <https://scikit-learn.org/stable/about.html> |
| os 3x | This module provides a portable way of using operating system dependent functionality. <https://docs.python.org/3.8/library/os.html> |

# Summary

The goal of this assignment was to recognize ten different postures or hand gestures using any software from the internet or previous projects. This was achieved using all modules listed in table 1, but primarily with the use of functionality supported by cv2 and Orange modules. An initial framework for segmentation was borrowed from “gogul ilango”.[1] Learning, Feature Analysis, and Prediction were performed using Orange supported wrappers to learn from a small data set of provided segmented images.

# Algorithm

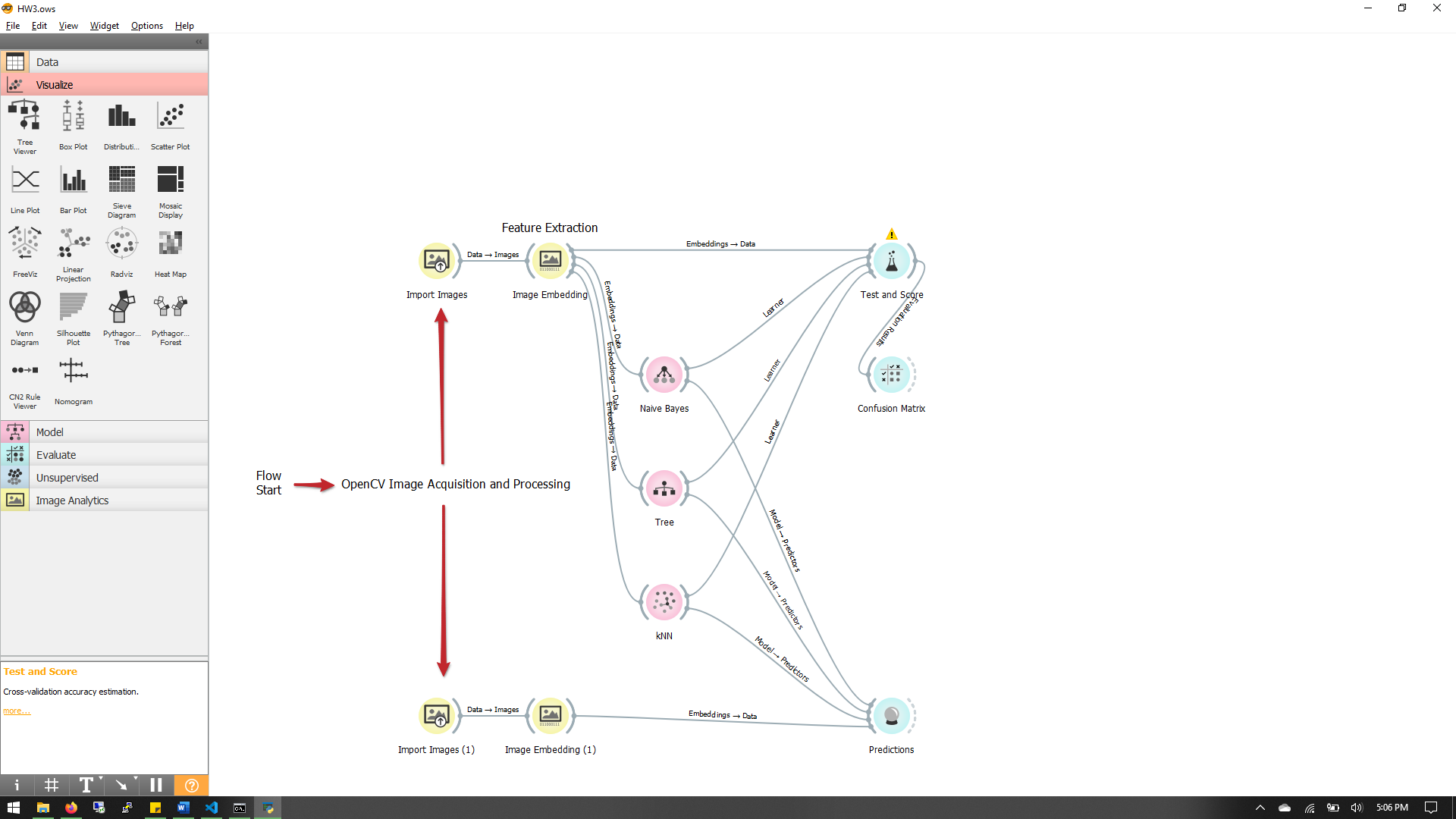


Figure - Orange Canvas View of Processing Algorithm.

## Image Acquisition and Processing

Image acquisition and processing was performed by using a laptop integrated webcam and the cv2 Python module. Individual frames are read then resized before being passed through grayscale modification and a pixel blurring filter. The borrowed code from [1] uses the frame at this stage to gather a weighted average for the background where the hand gestures will be provided. Once the background weighted average is complete, the image processing continues.

The next layer of processing is segmentation of the image. The image is compared to the background data to detect the presence of a hand. This image is then segmented by contours with a static binary threshold, whose level can be adjusted during program execution with “u” and “d” keys. The segmented image with the detected contours is further processed by filling the polygon generated by the contours with solid color. This helps eliminate sensitivity of thresholding to hand texture. This frame is saved to the test image directory for use feature extraction and category prediction.

## Feature Extraction

Once a processed frame is saved as an image, the program proceeds with extracting features used for classification. Learning and test images are imported into an Orange data table format for use with Orange wrapped ML functions. This data table is passed to the Image Embedding function in the Image Analytics module. Many different image embedders which provide image features are available, but “squeezenet” was chosen for its locally based extraction. All other available embedders require online connectivity.

1000 individual features are provided by the squeezenet algorithm and columns inserted into the Orange data table format containing image information. This data is then supplied to

# References

|  |  |
| --- | --- |
|  | Gogul Ilango. (2020) Hand Gesture Recognition using Python and OpenCV - Part 1 – Gogul Ilango. Retrieved December 06, 2020, from <https://gogul.dev/software/hand-gesture-recognition-p1#references> |
|  |  |