**Task 6: Creating and improving the Kitchenware dataset**

Please read the given Task carefully.

Resources

Python code “Task6\_2\_Kitcheware\_Image\_processing.py” and “Task6\_3\_Load\_image\_and\_process.py” as well as basic kitchenware dataset in Moodle.

Task 6 Definition

The aim of this Task 6 is to familiarize you as student on the methods of:

1. Creating a custom image database,
2. Extracting features from these images and
3. Pre-processing them to apply the dataset in an Artificial Neural Network (ANN) for later.

The provided applications were developed using the OpenCV, Keras + Tensorflow.

TensorFlow is an open-source ML platform that allows for the creation, pre-processing, modelling, and training of ANN for various application. It also comes with a high-end Application Protocol Interface (API) called Keras, which makes the creation and pre-processing of the images easy to implement, with a few lines of codes. Furthermore, the computer vision library OpenCV provides an API for image processing. OpenCV can extract features such as edge detection and has various threshing methodologies. NumPy and Matplotlib are required to visualize the images before and after processed images.

Allowed Tools

Tools that can be used:

* Python 3.9/ 2.7
* VS-Code
* TensorFlow + Keras, NumPy, Matplotlib, and OpenCV2.

Task 6.1 Adding Kitchenware images to the provided database

**Resource = “data” in Task6.zip**

Images have been captured and added to the “data” folder as part of the “**Task6.zip**” file on Moodle.

1. Download and create a local copy of the folder.
2. Capture pictures of cups, bowls, and plates using as an example a mobile phone camera.
   1. All images need to be in .jpeg format and have a simple background.
   2. Convert them if required either in code or via online tools.
3. Add your images to **your local “data” folder** to use in the provided code. It is important to follow the naming convention as well as the “data” folder structure**\***.
4. It is important to change the path in the code @ each instance to your local “data” folder’s path in the provided code.

**\*Note:** The categories have already been defined as cups, bowls, and plates as folder names, the number format must be noted and used within the subfolders.

Graphical user interface, application

Description automatically generated

**Your local path**

Figure 2: Screenshot of cup image database.

Expected Outcomes of Task 6.1

1. The provided baseline of the image database needs to be extended by the team with more images ofcups, plates, and dishes [For train and test/validation images]
2. Therefore, for a team of 2, with 3 categories (cups, dishes, and plates), and 4 images for each category needs to be captured and added to the database as a minimum [2 x 3 x 4 = 24 Images (Minimum)]
3. Please use screenshots and document your process thoroughly.

Task 6.2 Exploring and applying Image processing methods for feature extraction

**Resource = “Task6\_2\_Kitchewaren\_Image\_processing.py” in Task6.zip**

See the code for “Task6\_2\_Kitcheware\_Image\_processing.py”. Image processing based on OpenCV2 with edge detection, threshing and histogram-based calculations has been implemented in the provided application.

Again, it is important to change the path in the code @ each instance to the local folder in the provided code. Same as during Task 6, make sure to make the code your own!

1. Load your arranged images into your “Task6\_2\_Kitcheware\_Image\_processing.py” python code and apply various feature detection such as edge detection and thresholding methods to produce the required features.
2. Other image processing techniques can also be used to acquire the results needed, and it is important for you as a team to explore them. For example, changing the color, geometric transformations, smoothing, filtering and many more.
3. Please determine if there are better ways to extract features from your custom image dataset edge detection and thresholding? Show this by comparing the results from the different image processing techniques.

**Hint:**

<https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html>

**Example of expected results after image processing on your dataset**

A picture containing text, cup, coffee, tableware

Description automatically generatedA picture containing text

Description automatically generated

A picture containing company name

Description automatically generated

Figure 3: Before and after feature extraction using OpenCV

Expected Outcomes of Task 7.2

1. The code provided only explored edge detection and threshing techniques using OpenCV, more image processing techniques for feature extraction needs to be explored. (Hint: [Changing colour space, geometric transformations, smoothing…etc.])
2. Visualization before and after your image processing, detailed in your documentation. [Have you found better feature extractions?]

Task 6.3 Preparing the feature extracted images for an ANN

**Resource = “Task6\_3\_Load\_and \_process.py” in Task6.zip**

The code for “Task6\_3\_Load\_and \_process.py” is available via Moodle. The code implements image processing based on Keras TensorFlow with resolution and color space selections.

Again, it is important to change the path in the code @ each instance to the local folder in the provided code.

The feature extracted images are arranged in a 2D matrix and therefore needs to be prepared as a 1D array to be ported into the ANN for training and validation. This task also allows for the exploring further /other option for image processing. The given code has only demonstrated the image resolution and color space processing options.

As in the Task 7.2, you must explore further possibilities to determine the capabilities of Keras (<https://keras.io/api/preprocessing/image/>)

**Example of expected results after processing for Task 7.3:**

Diagram

Description automatically generated with medium confidenceA picture containing text

Description automatically generated

Figure 4: Before and after images using Keras image pre-processing.



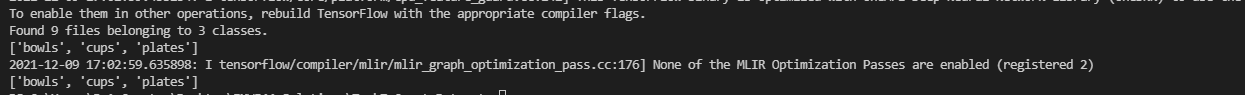


Figure 5: Terminal screenshot of pre-processed images.

Expected Outcomes of Task 6.3

1. Like Task 7.2, you must explore more functionalities with ‘Tensorflow. Keras’ API for image pre-processing
2. Make before and after visualization and document [Have you found better functionalities/methods?]

|  |  |  |  |
| --- | --- | --- | --- |
| *Course:* | **Maschinelles Lernen und künstliche Intelligenz/**  **Machine Learning and Artificial Intelligence** | | |
| *(Group) Member:* | *Nr.* | *Name* | *Matrikel* |
| *1* |  |  |
| *2* |  |  |
| *3* |  |  |

Task Reporting Template

Please read the given Task carefully. You can find all necessary information in Moodle.

1. Definition
   1. Introduction and Explanations

Please summarize here the tasks that you have to fulfill. (eg Bulletpoints are fine, this shall help you to identify the goals of the tasks and how to get there.)

* 1. Allowed Tools

Which tools are allowed, please summarize here.

For example:

It could be,

* Python
* VS-Code or equivalent IDE
* NumPy, Matplotlib and SciPy

1. Your Preparation/ Concept
   1. Structure your Problem

Please show with help of a Sketch and/or Flow Chart and a brief description how you understand the problem and how you try to solve it.

Your sketch/ flowchart/ pseudocode here

Figure 1: Your flow chart

Please also summarize which library functions in detail you will use. E.g. :

from scipy.fft import fft, ifft, fftfreq

from scipy import signal

import numpy as np

import matplotlib.pyplot as plt

* 1. Your Implementation

Please describe your implementation here. Mostly it consists of:

* Code
* Screenshots or plots (you can save/export plots in python easily as eg emf or png)
* Brief description (what was intention, how implemented, result and your judgement)
* Evaluation/ Testing (how have you proofed that all the goals of Chapter 1 are fulfilled?)
* Screen shots showing application working.

The Code is as follow:

PLACE YOUR CODE HERE

And explain (remember to add comments to your code in text)