SUPPLEMENTARY MATERIAL 2: USER MANUAL

1. INTERFACE USER MANUAL

This custom-developed interface enables real-time optical monitoring of electrochromic materials (or any other color changing technology) using a low-cost imaging system based on the ESP32-CAM module. The platform allows users to capture and process images of a device or film under test and extract meaningful transmittance information with both **temporal and spatial resolution**.

Specifically, the interface offers two core functionalities:

- **Real-time reconstruction of full visible transmittance spectra**, generated at regular time intervals defined by the user.
- **Real-time monitoring of transmittance at λ_{max}**, also recorded at user-defined time steps, enabling temporal tracking of optical contrast evolution during electrochemical switching.

Both outputs are displayed on-screen during data acquisition, providing immediate visual feedback of the material's behavior. Upon completion of the measurement, the user can **export the results as PDF and CSV files**, ensuring compatibility with further analysis workflows.

The interface has been implemented in Python and is designed to work on a local environment (e.g., Spyder IDE), making use of libraries such as tkinter, OpenCV, and matplotlib. It is not compatible with cloud-based notebooks (e.g., Google Colab or Jupyter Lab) due to GUI dependencies.

All required files, including the main Python script, auxiliary modules, and documentation, are provided in the GitHub repository:

https://github.com/antoniocs11/Arduino-Camera-Spectrophotometer

This manual describes the functionality and workflow of the interface in detail, covering input configuration, system requirements, and the structure of the outputs.

Subsection **1.1** lists the required libraries and installation instructions, while **1.2** provides a guided walkthrough of the interface usage and settings.

1.1. Spyder software and required libraries

To open the corresponding programming environment, first download the "Anaconda" application (select the last version), available at the following address: https://www.anaconda.com/download , selecting your computer's operating system. Once installed, a set of programming environments are pre-installed, such as "Jupyter Notebook", "Spyder", "RStudio", and so on. In our case, "Spyder" will be used.

Before loading the system-related script, a set of extra libraries must be installed, as only some of them are installed by default. Below are the set of all libraries used and code to install them:

```
import cv2

import os

import datetime as dt

import pandas as pd

import numpy as np

from skimage.io import imread

from skimage.color import rgb2lab, lab2rgb

import matplotlib.pylab as plt

import tensorflow

import joblib

from keras.models import load_model

from matplotlib.backends.backend_pdf import PdfPages

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

from matplotlib.pyplot import figure

from tkinter import *

from pathlib import Path

from tkinter import filedialog

from PIL import Image

from colormath.color_objects import LabColor, sRGBColor

from colormath.color_conversions import convert_color

from scipy import signal

import csv

from tkinter import messagebox
```

import tkinter as tk

import matplotlib.colors as mcolors

from matplotlib.colors import Normalize

from scipy.signal import find_peaks

from scipy.optimize import curve_fit

from flask import Flask, request, jsonify

from datetime import datetime

import requests

import tkinter as tk

import threading

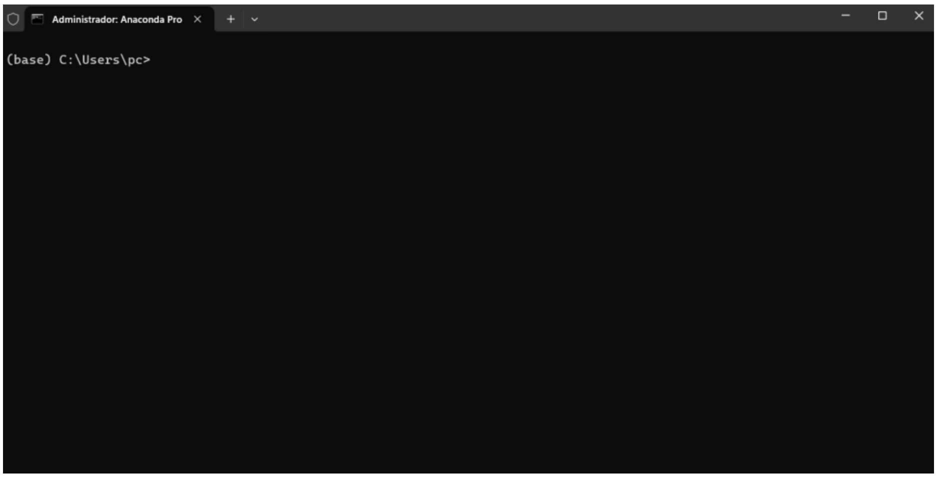
import time

from werkzeug.serving import make_server

from datetime import datetime, timedelta

The vast majority of libraries are installed by default when installing anaconda software. However, there are several that must be installed manually. Regarding the ones used in this particular script, "tensorflow", "colormath" and "opencv" must be installed manually. The steps are described below:

- All libraries are installed in "Anaconda prompt", which comes pre-installed with Anaconda. Therefore, the first step is to open this window:



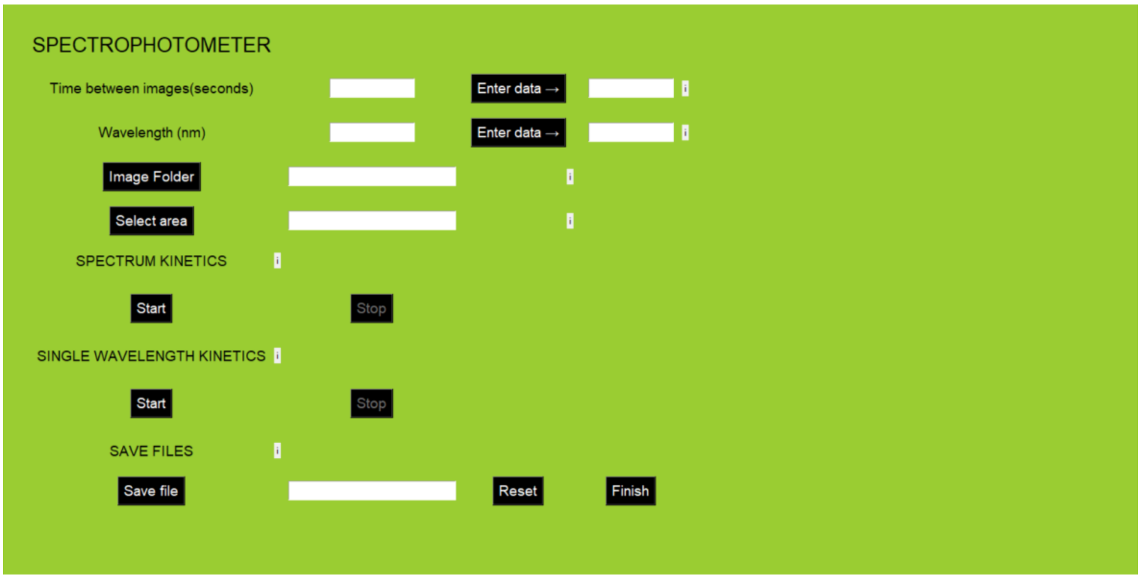
- "Tensorflow" library: this library is installed introducing the following command: "pip install tensorflow".

- “Colormath” library: this library is installed introducing the following command: “conda install conda-forge::colormath”
- “Opencv” library: a file that includes this library must be downloaded. In the following link (https://www.youtube.com/watch?app=desktop&v=ZdMwt9ZZaSc) a video explaining how the installation takes place is shown. During the installation process, the user will download a file including the desired library, which will be saved in a specific folder. Subsequently, in Anaconda Prompt, this file will be installed, implementing the "Opencv" library to the software.

Before opening the script and starting to run, it is necessary to save the file containing the neural network (Neural_network.h5) in the same folder where the program script is located.

1.2. User interface

When the script is executed, The interface shown in the following image opens. There are seven simple steps that the user will have to perform. Each one comprises, at the right side of the text area, a button with the letter "i" for information about that particular step.



TIME BETWEEN IMAGES

This function defines the temporal resolution at which the system captures and analyzes images during a measurement.

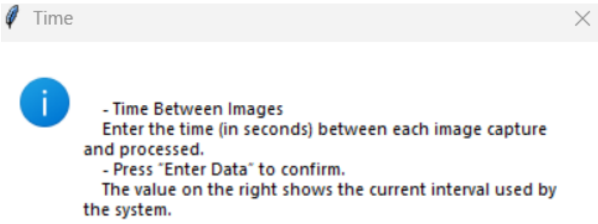
The user must enter the desired **time interval (in seconds)** into the left-side text entry box labeled “Time Between Images”. This value determines two critical aspects of the experiment:

- The time between each image captured and stored by the ESP32-CAM system.
- The frequency at which data points appear on the live transmittance graph during acquisition.

Once the desired interval is written, the user must click the **“Enter Data”** button. Upon doing so, the selected value will appear in the adjacent box on the right, confirming that the system has correctly registered it.
If a value is present in this confirmation box, the system will use that time interval for all subsequent data acquisition steps—until it is manually changed by the user.

This functionality allows flexible temporal control, enabling both high-resolution (short interval) and long-duration (long interval) experiments depending on the electrochemical kinetics under study.

The Info button will display this message when clicked:



WAVELENGTH (nm)

This functionality allows the user to define the specific wavelength (in nm) at which the transmittance evolution will be monitored in real time during the experiment.

To set the desired wavelength (λ_{max}), the user must enter a numeric value between 400 and 700 nm into the left-side input box labeled “Wavelength (λ_{max})”. This value determines the spectral point at which the transmittance will be continuously calculated and plotted throughout the measurement.

Once the value is entered, the user must click the “Enter Data” button. The selected value will then appear in the adjacent confirmation box on the right, indicating that the system has successfully registered the desired wavelength.

This feature is particularly useful for tracking the color switching kinetics at the most responsive part of the spectrum, and it allows for quick assessment of device behavior at a user-defined optical maximum. As with the time interval setting, the wavelength can be updated at any time before starting the acquisition.

The Info button will display this message when clicked:

IMAGE FOLDER

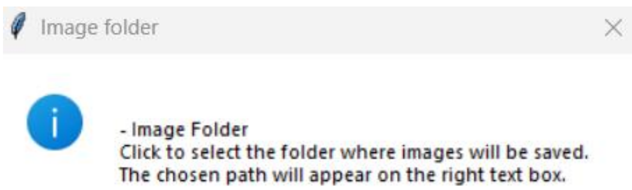
This functionality allows the user to define the destination folder where all the images captured during the real-time measurement will be automatically saved.

By clicking the "Image folder" button, a file explorer window will open. The user must navigate through their directories and select the folder where the images should be stored. Once selected, the full path of the chosen folder will appear in the adjacent text box on the right side of the interface. This confirms that the system has successfully registered the storage location.

During the experiment, each image taken at the defined time interval will be saved in this folder. When the acquisition ends, the image saving process will stop automatically. This ensures organized storage of raw data, which can later be used for post-processing, spectral correction, or detailed optical analysis.

It is important to note that the folder must be selected before starting the measurement to ensure that data is not lost.

The Info button will display this message when clicked:



SELECT AREA

This functionality allows the user to define the specific active region of the image (typically corresponding to the electrochromic film or device under study) that will be processed during real-time transmittance monitoring.

When the "Select area" button is clicked, a new window will open displaying a live image captured at that moment by the camera. Within this image, the user can select a rectangular region that defines the area of interest.

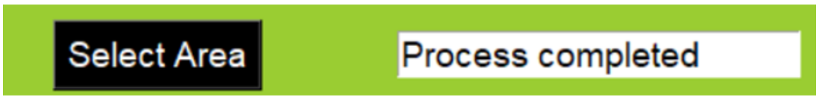
To do so:

- Left-click to mark the top-left corner of the rectangle.
- Right-click to mark the bottom-right corner.

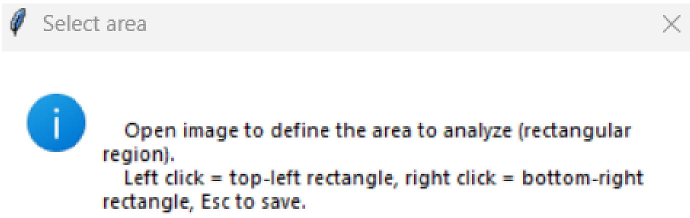
Once both points are selected, the defined rectangle will appear overlaid on the image for confirmation. After verifying the selection, the user must press the Escape (Esc) key to close the window. This action saves the coordinates of the selected region, and they will be used in all subsequent measurements.

Once the region is saved, the message “Process completed” will appear in the text box to the right of the button, confirming that the selected area is now active for the experiment.

This step is essential to ensure that the data extracted corresponds precisely to the region of interest, avoiding irrelevant background or external zones.



The Info button will display this message when clicked:



SPECTRUM KINETICS

This mode allows the real-time monitoring of transmittance spectra from the previously selected active region of the image. The process is governed by the time interval specified by the user and is initiated by pressing the “Start” button.

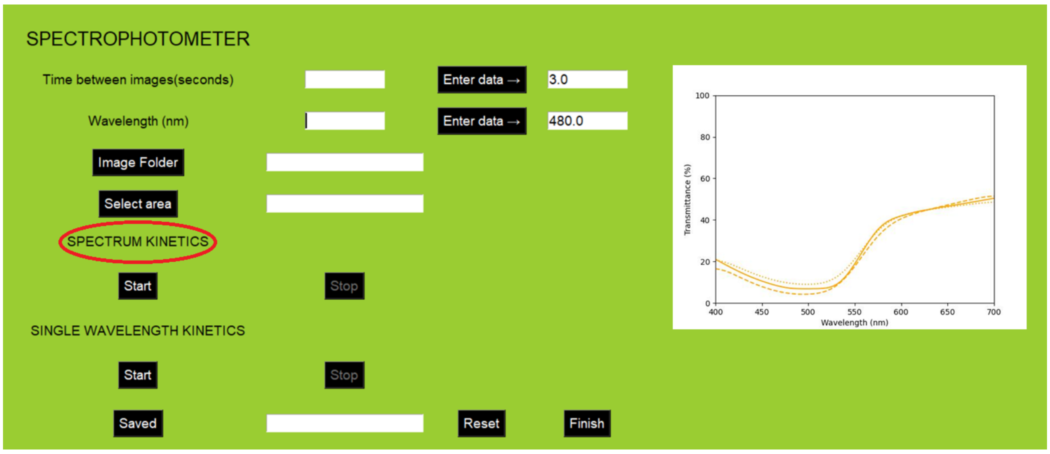
Once activated:

- The camera begins capturing images at the predefined time intervals.
- These images are analyzed on-the-fly to extract the visible transmittance spectrum of the selected area.
- The resulting spectra are plotted sequentially on the graph embedded in the interface, allowing the user to visualize spectral evolution in real time.
- Simultaneously, each captured image is saved into the previously defined folder path, enabling full traceability and optional post-processing.

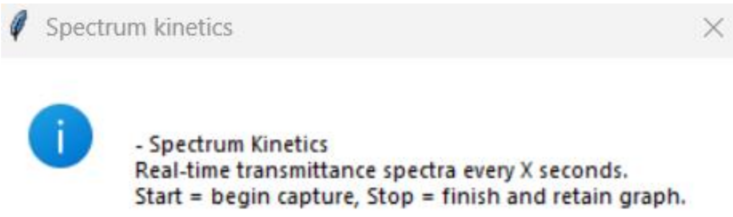
The acquisition continues until the “Stop” button is pressed. When stopped:

- The image capture halts.
- The graph retains the complete set of transmittance spectra acquired during the session, offering a complete picture of the spectral evolution over time.

This function provides both temporal and spectral resolution of dynamic optical processes and is especially useful for assessing how the optical behavior of a film or device changes during electrochemical modulation or light exposure.



The Info button will display this message when clicked:



SINGLE WAVELENGTH KINETICS

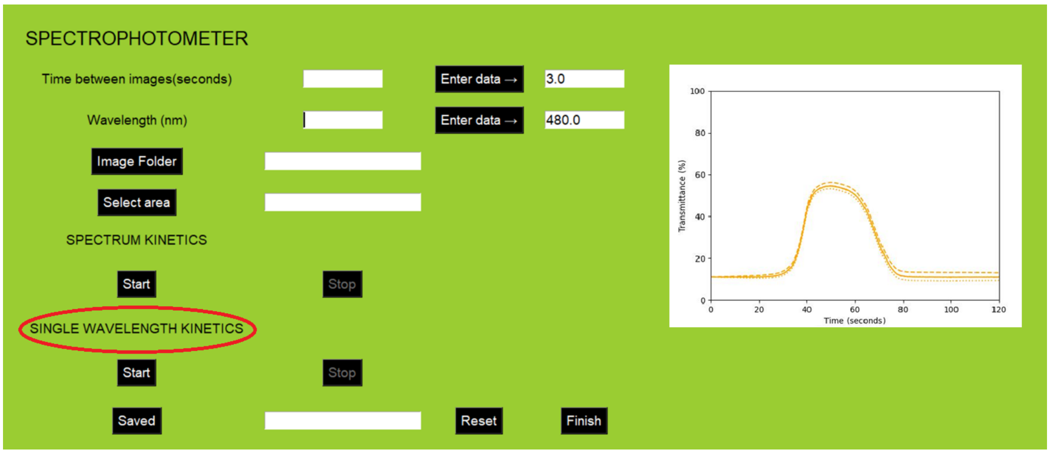
This functionality enables real-time monitoring of transmittance at a specific wavelength over time. As with *Spectrum Kinetics*, the process begins when the user presses the “Start” button and continues until “Stop” is selected.

Once initiated:

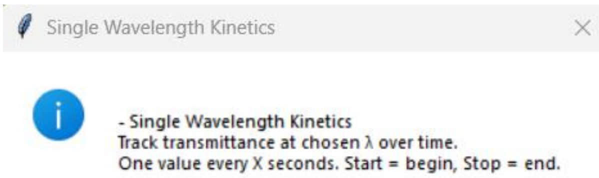
- The system captures and processes images at the user-defined time interval.
- The previously selected area of the image is analyzed in each frame.
- Instead of reconstructing the full spectrum, the software extracts the transmittance value at the selected wavelength (λ), which must fall within the 400–700 nm range.
- This transmittance value is plotted on the real-time graph in the interface, building a kinetic profile that reflects the temporal evolution of optical behavior at that specific λ.

Captured images are saved automatically into the selected folder during the test. When the user presses “Stop”, the acquisition halts and the final plot remains on screen, offering a complete record of the optical response at that wavelength.

This function is especially useful for monitoring optical switching kinetics in electrochromic or photoresponsive systems, where the performance at a characteristic wavelength (e.g., λ_{max}) is of particular interest.



The Info button will display this message when clicked:



SAVE FILES

After completing any of the two experiment types (*Spectrum Kinetics* or *Single Wavelength Kinetics*), the user can export the resulting data by:

- Typing a desired filename into the input box.
- Pressing the “Saved” button.

The following will occur:

- The graph currently displayed in the interface will be saved as a PDF file.
- The corresponding data (either the full spectra or single-wavelength kinetics) will be saved as a .CSV file.
- Both files are stored in the same directory as the Python script.
- Upon successful saving, the word "Saved" will appear in the output text box to confirm completion.

Pressing the "Reset" button clears all previously entered values in the interface (e.g., time interval, wavelength, selected folder, selected area), and resets the system internally. This is useful to start a new experiment from scratch without restarting the program.

To exit the interface and properly close the application, the user can press the "Finish" button. This will terminate all running processes and shut down the window cleanly.

The Info button will display this message when clicked: