1. Normalized Difference Vegetation Index (NDVI): It uses near-infrared (NIR) and red light to assess vegetation health. The formula for NDVI is:

NDVI = (NIR - Red) / (NIR + Red)

1. Weighted Difference Vegetation Index (WDVI): It's a modification of NDVI that attempts to correct for the effects of soil brightness on the NDVI signal. The formula for WDVI is:

WDVI = NIR - slope × Red

The slope is a soil brightness correction factor, often determined empirically. 'slope' is the slope of the soil line in a scatter plot of Near-Infrared (NIR) reflectance versus Red reflectance. The slope of the soil line can be calculated using a linear regression of NIR versus Red reflectance values for a number of bare soil samples. The slope represents how the NIR reflectance changes with respect to Red reflectance for bare soil.

slope or w

**W: Slope of the soil line measurement**



Measure the soil reflectance by AS7265x in various light condition and various places (sunlight, low light, cloudy sky etc.).

Repeatedly measure (NIR/Redsoil) for the soil in various conditions (in laboratory keep the soil under the exposure of full spectrum light source) but change the angle of position of AS7265x and collect the data.

**Plot the Data**: Plot the **NIR reflectance (y-axis)** against the **Red reflectance (x-axis)** for each sample. You should see a scatter of points that roughly form a line, which represents the soil line.



1. Perpendicular Vegetation Index (PVI): It's another index that attempts to minimize soil brightness influence by considering the soil line in the equation:

PVI = (NIR - slope × Red - intercept) / sqrt(slope^2 + 1)

Here, slope and intercept are the parameters of the soil line, usually determined empirically.

1. Difference Vegetation Index (DVI): This index is a simple difference between near-infrared and red reflectances, and is often used in remote sensing applications:

DVI = NIR - Red

1. Transformed Normalized Difference Vegetation Index (TNDVI): This index is a variant of the NDVI that applies a square root transformation to normalize the distribution of NDVI values:

TNDVI = sqrt((NIR - Red) / (NIR + Red) + 0.5)

1. Enhanced Vegetation Index (EVI): This is an optimized vegetation index designed to enhance the vegetation signal in regions with high biomass and improve sensitivity over dense vegetation conditions, while reducing atmosphere influences. The formula for EVI is:

EVI = 2.5 × ((NIR - Red) / (NIR + 6 × Red - 7.5 × Blue + 1))

1. Soil Adjusted Vegetation Index (SAVI): This index attempts to minimize the effect of soil brightness on the vegetation signal by incorporating a soil brightness correction factor, often represented by L. The formula for SAVI is:

SAVI = (1 + L) × (NIR - Red) / (NIR + Red + L)

*'L' in SAVI: The 'L' value in the SAVI calculation is a soil brightness correction factor. It's used to adjust the sensitivity of the SAVI to the presence of soil in the image or the sensor's field of view. 'L' is a constant that you choose based on your understanding of the scene, particularly the amount of vegetation cover.*

*L is a soil brightness correction factor that varies between 0 (for high vegetation cover) and 1 (for bare soil). The L factor is a correction factor that varies from 0 (no correction) to 1 (maximum correction). A commonly used value for L is 0.5.*

*The Soil Adjusted Vegetation Index (SAVI) calculation involves a soil brightness correction factor 'L'. 'L' is not something that can be measured directly with a sensor like the AS7265x. Instead, it is a constant that is chosen to adjust the calculation based on the expected amount of green vegetation in the image or the sensor's field of view.*

1. Green Normalized Difference Vegetation Index (GNDVI): This index substitutes the Red band with the Green, aiming to enhance the sensitivity to variations in chlorophyll content:

GNDVI = (NIR - Green) / (NIR + Green)

Please note that some of these indices require reflectance measurements in bands **not covered by the AS7265x (e.g., Blue or Green),** so you may need a different or additional sensors to calculate them. Additionally, as with the indices you mentioned earlier, accurate calculation of these indices will likely require careful sensor calibration.

1. Triangular Vegetation Index (TVI): This index is a function of the NDVI and is designed to enhance the vegetation signal in areas with high soil background reflectance:

TVI = 0.5 \* (120 \* (NIR - Green) - 200 \* (Red - Green))

Triangular Vegetation Index (TVI): Although the original formulation of TVI uses the NIR band, it can be adapted for use with RGB data.

Modified Formula: TVI = 0.5 \* (120\*(Green - Red) - 60\*(Red - Blue))

1. Visible Atmospherically Resistant Index (VARI): This index is used to estimate vegetation fraction using only the visible spectrum:

VARI = (Green - Red) / (Green + Red - Blue)

1. Excess Green Index (ExG): This index was developed to maximize the spectral response of green vegetation.

Formula: ExG = 2\*Green - Red - Blue

1. Excess Red Index (ExR): This is another index that could be calculated using the RGB bands.

Formula: ExR = 1.4\*Red - Green

1. Excess Blue Index (ExB): This is another index that could be calculated using the RGB bands.

Formula: ExB = 1.4\*Blue - Green

1. Normalized Green - Red Difference Index (NGRDI): This index is often used in digital photography for plant health assessment.

Formula: NGRDI = (Green - Red) / (Green + Red)

1. Color Index of Vegetation Extract (CIVE): This index was specifically designed to differentiate between crop and weed.

Formula: CIVE = 0.441Red - 0.811Green + 0.385\*Blue + 18.78745

1. Green Leaf Index (GLI): This index emphasizes the greenness of vegetation and can be used to understand the health of plants.

Formula: GLI = 2\*Green - Red - Blue

1. Chlorophyll Index (CI): This is an indicator of the chlorophyll content in leaves. It can be calculated in different ways using different spectral bands, but one common formulation is:

CI = (NIR / Red) - 1

1. Difference Green Chlorophyll Index (DGCI):

The DGCI is a vegetation index that helps evaluate chlorophyll content in plants. It measures the difference between the reflectance values in the NIR and green bands, providing insights into chlorophyll concentration and plant health. Positive values of DGCI indicate higher chlorophyll content, while negative values suggest lower chlorophyll levels.

DGCI = (NIR - Green) / (NIR + Green)

1. Photochemical Reflectance Index (PRI): This index is used to assess light use efficiency and photosynthetic activity:

PRI = (R531 - R570) / (R531 + R570)

R531 and R570 refer to reflectance at 531 nm and 570 nm wavelengths.

1. Ratio Vegetation Index (RVI): RVI is one of the simplest vegetation indices. It is a ratio of the near-infrared (NIR) reflectance to red reflectance. Here is the formula:

RVI = NIR/Red

1. Modified Non-Linear Index (MNLI): MNLI is a sophisticated vegetation index designed to better manage the saturation problem at high biomass conditions and the soil background influence. The formula is as follows:

MNLI = (1 - 0.5) \* ((1.08\*(NIR-Red))/(1 + sqrt(NIR-Red)))

1. Transformed Chlorophyll Absorption in Reflectance Index (TCARI): This index was developed specifically to correct the Chlorophyll Absorption Ratio Index (CARI) for soil background. It's more effective in fields with exposed soil surfaces:

TCARI = 3 \* ((Green-Red) - 0.2 \* (Green-Blue) \* (Green/Red))

Please note that the Green, Red, and Blue reflectance values correspond to the respective spectral bands.

1. Modified Chlorophyll Absorption Ratio Index (MCARI): MCARI is designed to maximize sensitivity to leaf chlorophyll concentration, and is less sensitive to leaf area index (LAI):

MCARI = ((Red - Green) - 0.2 \* (Red - Blue)) \* (Red/Green)

Again, the Red, Green, and Blue reflectance values correspond to the respective spectral bands.

1. Red Edge Chlorophyll Index (RECI): The RECI was created to estimate the chlorophyll content in higher biomass. The 'red edge' is the steep increase in reflectance between the red and NIR wavelengths:

RECI = (NIR/Red\_edge) - 1

Note: Red\_edge refers to the reflectance at the red edge position, which is a narrow spectral region between the red and the NIR region.

1. Modified Simple Ratio Index (MSRI): MSRI is another vegetation index that includes the blue band to correct for atmospheric effects and soil background:

MSRI = ((NIR/Red) - 1) / (NIR/Blue)

The NIR, Red, and Blue reflectance values correspond to the respective spectral bands.

**Materials**

| **AS7265x :** The **AS7265x Spectral Sensor** is one such example of a light sensor they carry which is designed for hobbyist applications. This sensor can measure light across multiple bands, **but it doesn't cover SWIR.** |
| --- |
| **TCS34725:** Similar to SparkFun, they have a selection of light and color sensors, **such as the TCS34725,** which can provide **RGB color** sensing, but they also do not offer specific narrow band SWIR sensors. |
| **APDS9960: RGB +** Gesture |
|  |

Texture Feature Parameters

Below are the formulas for the texture feature parameters you mentioned. Note that these are all calculated from the grayscale levels in the image. If you have **a color image, you will need to convert it to grayscale** before performing these calculations.

1. Mean (MEAN): The mean of an image refers to the average pixel intensity in the image. To calculate the mean, sum the intensity values of all pixels and divide by the total number of pixels. If the image is I and it has N pixels, the mean is:

MEAN = (1/N) \* sum(I)

2. Variance (VA): Variance measures the spread of pixel intensities around the mean. Variance is the average of the squared differences from the Mean. For an image I with mean MEAN and N pixels, the variance is:

VA = (1/N) \* sum((I - MEAN)^2)

3. Entropy (EN): Entropy measures the randomness or complexity in an image. In texture analysis, it's often computed from the GLCM. A higher entropy value indicates more complexity or less regularity in the texture.

Entropy measures the randomness or complexity of the image. It's typically calculated from the Gray-Level Co-occurrence Matrix (GLCM) P of the image. For a GLCM P with i rows and j columns, the entropy is:

EN = -sum(sum(P(i, j) \* log2(P(i, j)))) for all i and j such that P(i, j) != 0

4. Angular Second Moment (ASM): Also known as energy, this parameter measures the uniformity of the image. It's calculated from the GLCM as the sum of the squares of each element in the matrix. Also known as energy, ASM measures the uniformity of the image. It is calculated from the GLCM P of the image. For a GLCM P with i rows and j columns, the ASM is:

ASM = sum(sum(P(i, j)^2)) for all i and j

5. Difference (DI): The difference or contrast is a measure of the intensity contrast between a pixel and its neighbor over the whole image. It's calculated from the GLCM. The difference or contrast measures the intensity contrast between a pixel and its neighbor over the whole image. It's calculated from the GLCM P. For a GLCM P with i rows and j columns, the contrast is:

DI = sum(sum(|i-j|^2 \* P(i, j))) for all i and j

Note: The Gray-Level Co-occurrence Matrix (GLCM) is a histogram of co-occurring grayscale values at a given offset over an image. This means it measures how often different combinations of pixel brightness values (grayscale values, gray levels) occur in an image.

In Python, you can use the skimage.feature.greycomatrix function to calculate the GLCM, and the skimage.feature.greycoprops function to calculate texture features such as contrast, ASM (under the name 'energy'), and others. The entropy can be calculated using the scipy.stats.entropy function.

Below is a Python example for calculating the texture feature parameters using a Raspberry Pi Camera. We will use the picamera library for capturing the image, and the scikit-image and numpy libraries for image processing.

Before starting, make sure you have installed these libraries on your Raspberry Pi:

Install Picamera:

sudo apt-get install python3-picamera

Install Scikit-image:

sudo apt-get install python3-skimage

Install numpy:

sudo apt-get install python3-numpy

**Now here is the Python code to capture an image and calculate the texture features:**

import numpy as np

import cv2

from picamera import PiCamera

from picamera.array import PiRGBArray

from skimage.feature import greycomatrix, greycoprops

from skimage import img\_as\_ubyte

from scipy.stats import entropy

# Initialize the camera

camera = PiCamera()

# Capture an image

rawCapture = PiRGBArray(camera)

camera.capture(rawCapture, format="bgr")

image = rawCapture.array

# Convert the image to grayscale

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Normalize the grayscale image

gray\_image = img\_as\_ubyte((gray\_image - np.min(gray\_image)) / (np.max(gray\_image) - np.min(gray\_image)))

# Calculate the GLCM

glcm = greycomatrix(gray\_image, distances=[1], angles=[0], levels=256, symmetric=True, normed=True)

# Calculate the texture feature parameters

mean = np.mean(gray\_image)

variance = np.var(gray\_image)

entropy\_val = entropy(np.histogram(gray\_image, bins=256, density=True)[0])

asm = greycoprops(glcm, 'ASM')[0, 0]

difference = greycoprops(glcm, 'contrast')[0, 0]

print('Mean:', mean)

print('Variance:', variance)

print('Entropy:', entropy\_val)

print('ASM:', asm)

print('Difference:', difference)



1. PAR: (µmol/m²/s)
2. DLI (Daily Light Integral): This is a measure of the total amount of PAR that is received each day. It's usually measured in (mol/m²/day). To calculate DLI from PAR, you would integrate the PAR over the course of a day.
3. Photosynthetic photon efficacy (PPE): PPE is defined as the number of photosynthetically active photons (expressed in micromoles per second, or µmol/s) emitted by a light source per watt of electrical power consumed. Thus, to calculate PPE, you need to know two things: the output of the light source in terms of PAR (which you can measure using your sensor, once calibrated), and the electrical power consumption of the light source.

The electrical power consumption is typically provided by the manufacturer of the light source. It's important to note that the power used in the PPE calculation is the total power consumed by the light source, including any power consumed by components other than the light-emitting part itself, such as fans or power supply units. Here is the formula for PPE again, for reference:

**PPE = PAR / Power (μmol/W)**

where:

PAR is the photosynthetically active radiation, in µmol/s.

Power is the electrical power consumption of the light source, in W for **full spectrum light** for example 30W.

| #include <Wire.h>  // Include your specific sensor library  // #include <YourLightSensorLibrary.h>  // Replace this with the actual power consumption of your light source  #define LIGHT\_SOURCE\_POWER\_WATTS 30.0  // Initialize the light sensor  // Replace "LightSensor" with the class name of your sensor  // LightSensor lightSensor;  // Variables to store the DLI and last measurement time  float dli = 0.0;  unsigned long lastMeasurementTime;  void setup() {  // Initialize serial communication  Serial.begin(9600);  // Initialize the light sensor  // lightSensor.begin();  // Store the current time  lastMeasurementTime = millis();  }  void loop() {  // Get the current PAR value from the light sensor  // float par = lightSensor.getPAR();  float par = 0.0; // Replace this with actual PAR reading  // Calculate the elapsed time since the last measurement in seconds  float elapsedTime = (millis() - lastMeasurementTime) / 1000.0;  lastMeasurementTime = millis();  // Add the current PAR value to the DLI (convert from µmol/m²/s to mol/m²/day)  dli += par \* elapsedTime \* 0.0864;  // Calculate PPE (µmol/W)  float ppe = par / LIGHT\_SOURCE\_POWER\_WATTS;  // Print the current PAR, DLI, and PPE values  Serial.print("PAR: ");  Serial.print(par);  Serial.print(" µmol/m²/s, DLI: ");  Serial.print(dli);  Serial.print(" mol/m²/day, PPE: ");  Serial.print(ppe);  Serial.println(" µmol/W");  // Wait for 1 second  delay(1000);  } |
| --- |

**Lux Sensors for PAR, DLI, PPE**

| **BH1750FVI:**  High-resolution mode: 1 - 65,535 lux  Low-resolution mode: 4 - 65,535 lux | **Calibration With PAR Sensor**  Apogee MQ Series MQ-200, MQ-210, MQ-500, or MQ-510  or  Li Cor 190R |
| --- | --- |
| **MAX44009:**  Typical range: 0.045 - 188,000 lux |
| **VEML7700:**  Range: 0.0036 - 120,000 lux |
| **TSL2591:** |  |

1. **Photobiologically Active Radiation (PBAR):** Photobiologically Active Radiation (PBAR) is defined as radiation that is biologically active, and typically includes wavelengths from about 280nm to 800nm. Given that the AS7265X covers a range from 410nm to 940nm, it's capable of measuring most of the PBAR range. However, it does not cover the entire PBAR range, especially the lower wavelengths below 410nm (UV-B and some UV-A radiation) which are also biologically active. So while it can be used for a majority of the PBAR range, it won't cover the full spectrum.

???????????????????????????????????????????????????????

**NO FORMULA**

**Sensors for PBAR**

| **AS7265x** | **Calibration With PAR Sensor**  Apogee MQ Series MQ-200, MQ-210, MQ-500, or MQ-510  or  Li Cor 190R |
| --- | --- |
| **US Sensors:** |
| **ML8511** |
| **VEML6075** |
| **LTR390** |

Leaf Area Index (LAI): This is a measure of the amount of leaf area per unit of ground area, and is an indicator of plant growth and productivity. LAI is typically estimated using complex models and a combination of different vegetation indices.

LCab

CCC = LAI x LCab

Plant Area Indea (PAI)