PERSONAL PROJECT FINAL PROJECT:
MATERIAL IDENTIFICATION VIA NEAR-INFRARED SPECTROSCOPY

Lab report presented to
Mr. Daniel Deschênes

written by Luca Scavone

Collège Jean-de-Brébeuf
Montreal
February 24, 2021

**Table of contents**




**Summary**

The following report presents the results of an experiment involving near-infrared spectroscopy as a new and efficient way to identify unknown materials. This technology has the potential to revolutionize the plastic recycling industry which relies on manual labor to sort plastics into their respective categories. This industry is in desperate need of an overhaul as millions of tons of plastic waste are discarded every year in landfills and our oceans. In this experiment, I turned my attention to near-infrared (NIR) spectroscopy as a potential solution to the plastic crisis. This type of spectroscopy analyzes the degree of reflection of 18 electromagnetic frequencies located between 410 and 940 nm. I ran the collected data through a Python program I wrote in order to determine if this technology could be viable in the real world. To test the reliability of my spectrometer, I first added the spectral data of about 20 materials to a database. In a first round of testing, I rescanned the database items to determine reliability. In a second round of testing, I scanned a few new items to assess the sensor's ability to identify unknown materials. From the data collected in this experiment, I established the following conclusion that material identification via NIR spectroscopy is most effective with homogeneous materials with smooth and monochrome surfaces but most definitely has the potential to revolutionize the recycling industry.

## Introduction

In the past few years, a growing population and rising consumption in developed and emerging countries has led to a global waste crisis as millions of tons of plastic waste are dumped into the world's oceans every year. Plastic is a highly advantageous material due to its malleability, high heat capacity and cost. However, the recycling process is lengthy and complicated because several types of plastics are not compatible when recycled. This makes recycling plastic a very costly endeavour which explains why several developed countries export their plastic waste to developing countries. In fact, it is more affordable to simply create new plastic than to recycle it. This results in over 300 million tons of plastic waste being discarded every year in landfills and our oceans.

An updated plastic recycling system is therefore at the heart of this growing threat to our ecosystems. The experiment I have designed makes use of near-infrared technology, a technology that encompasses the electromagnetic spectrum between 800 and 2500 nm to measure light reflection on any material. According to *spectralengines.com*, this technology would make it possible to identify, using a spectroscopic sensor, various materials via their level of reflection of ambient and infrared light which vary from one material to another. If sufficiently developed, such technology could drastically improve the waste sorting system in recycling centers and could also be included in hand-held tools or public recycling bins.

This experiment is both conceptual and experimental. The conceptual phase encompasses the creation and testing of the Python code needed to process the spectroscopic data. The experimental second phase makes use of the Python code coupled with the spectrometer to determine the viability of NIR technology.

During the experimental phase, the programmed Arduino microcomputer will use the spectrometer to take five readings per frequency of a given material (the sensor takes readings at 18 different frequencies). These five readings are then averaged and compared to a database of spectral frequencies using the Euclidean distance between the spectral data of the database and that of the material to be identified. In order to verify the reliability of the sensor at hand, the same materials that are already in the database will be scanned several times by the sensor. If the materials in the database cannot correctly be identified by the spectrometer, there is no use proceeding with the experiment as the instrument will have been deemed unreliable and faulty.

## Materials and methods

- Arduino UNO
- USB-Arduino cable
- Black construction paper
- Tape
- Soldering
- Various plastic or paper materials used for the
- Pycharm[2]
- SparkFun IDE AS7265x Arduino Library[4]
- 4 electrical wires (red, black, blue, green)
- Cardboard
- White lead pencil
- Scissors
- Equipment
- SparkFunTriad Spectroscopy Sensor - AS7265x[1]
- Arduino IDE[3]

1. **Manufacture of the case[5]**:
   i. On thin cardboard, draw the decomposed shape of a prism with a rectangular base whose base measures 3.5 cm x 4.0 cm and the height, 3.8 cm. Do not draw the second base measuring 3.5 cm x 4.0 cm.
   ii. Cut out the shape, fold and glue the sides with adhesive tape to obtain a box where the spectroscopic sensor should fit perfectly. If necessary, secure the sides with additional cardboard. Use the scissors or any other knife.
   iii. Repeat steps 1 and 2 with black construction paper without gluing the sides. This second box should fit easily into the cardboard box. Use scissors.
   iv. Place the sensor in the box and mark the location of the 4 holes next to which are written GND, 3V3, SDA and SCL.
   v. Remove the sensor and cut a thin hole of approximately 0.2 cm x 1 cm at the location of the 4 holes. Cut this whole in both the black lining and the cardboard box. They should line up perfectly.

[1] Consult this link to visit the site of the sensor: [https://www.sparkfun.com/products/15050].
[2] Check this link to download the program: [https://www.jetbrains.com/pycharm/download/#section=mac].
[3] Check this link to download the program: [https://www.arduino.cc/en/software].
[4] Check this link to download the file: [https://github.com/sparkfun/SparkFun_AS7265x_Arduino_Library].
[5] See Appendix III, Figure 1 for an overview.

2. **Circuit assembly**[6]:

   i. Solder the black wire to the hole marked GND passing through the bottom of the sensor. The wire should be able to exit the cardboard case through the bottom hole

   ii. Repeat step 1 with the other three threads. For the sake of convention, the red wire will be attached to the 3V3 and the other two wires can be interchanged, but will be attached to the SDA and SCL.

   iii. Insert the sensor and wires into the housing while making sure the wires all exit through the hole in the base of the housing.

   iv. On the Arduino, insert the black wire into one of the two pins marked GND.

   v. On the Arduino, insert the red wire into the pin marked 3.3V.

   vi. On the Arduino, insert the wire connected to the SDA into the pin marked 4.

   vii. On the Arduino, insert the wire connected to the SCL into the pin marked 5.

   viii. Connect the Arduino to the computer via the USB-Arduino cable.

3. **Programming the Arduino section**

   i. Download the Arduino IDE application and open it.

   ii. Download SparkFun's Computer Library[7].

   iii. Open a new project and copy the code from appendix I.

   iv. Click on the arrow at the top left. This will upload the program to the Arduino.

   v. To test the program, click on the magnifying glass at the top right and type "g". If everything works correctly, a series of 18 digits should appear.

4. **Programming section Python**

   i. Download the Pycharm IDE application and open it. If necessary, consult a video[8] to install it correctly.

   ii. Open the app and click on *Terminal* at the bottom of the screen. This will open a small window at the bottom of the application.

   iii. For the program to work, you will need to install a few additional programs. In the window opened in the previous step, type the following command: *pip install numpy* + [enter].

---

[6] See Appendix III, Figure 2 for an overview.

[7] See this link to download the file: [https://github.com/sparkfun/SparkFun_AS7265x_Arduino_Library].

[8] HAMEDANI, Mosh. Software engineer. 2019. "Python Tutorial - Python for Beginners." *Programming with Mosh*. On line. [https://www.youtube.com/watch?v=_uQrJ0TkZlc&ab_channel=ProgrammingwithMosh]. Page accessed October 4, 2020.

iv. After downloading Numpy, type the following command: *pip install pyserial* + [enter].

v. Open a new project and copy the code from appendix II.

vi. In the 13th line of the program, you will have to modify what is between the quotation marks. Return to the Arduino application and determine the port to which the Arduino (microprocessor) is connected. On a Mac, click to *Tools* + *Port* find the port name. Replace " /dev/cu.usbmodem1D11141 " with the name found. Be sure to copy the name literally. Otherwise, the program will not know where to collect the data.

vii. Download the following file: data_bank_example.csv[9].

viii. On Pycharm, right click on the project name at the top left in the directory.

ix. Click on "Reveal in finder" and drag the file into the project directory.

x. On lines 68, 70, 85 and 105, replace " Data_Bank.csv " with the name of the file downloaded in step 7: " Data_Bank_Example.csv ". This file contains a dozen materials already analyzed, but the file is easily editable in Pycharm and could be completely erased.

**5. Experimentation and adding data**

i. On the Pycharm application, click on the small green triangle at the top right to run the program. A small menu will appear showing the possible commands.

ii. To add the spectral data of a material to the database, type "1" then "y", making sure to completely cover the box containing the sensor with the desired material. For more accurate results, cover transparent or semi-transparent materials with a sheet of white paper and maintain a constant illumination on the sensor. This will limit the impact of ambient light on the data.

iii. To compare the data of a material with those of the database, type "2" then "y" while making sure to completely cover the box and the sensor. Type "3" to get a complete list of materials included in the database.

iv. Type "4" for a short description of the program.

v. Type "0" to end the program.

---

[9] Check this link to download the file: [https://drive.google.com/file/d/1UbrQQE_CB15NvumLK9oWos2e5qbjGppV/view?usp=sharing].

## Results

*Table 1: Values measured at 18 different frequencies for 22 materials from a database*

| Matériau: | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardboard - Brown (1mm) | 642.82 | 248.32 | 418.36 | 160.79 | 276.2 | 349.5 | 208.02 | 239.6 | 149.56 | 65.9 | 239.08 | 110.99 | 559.92 | 222.97 | 87.13 | 71.58 | 232.38 | 591.04 |
| Cardboard - Brown (2mm) | 561.43 | 211.16 | 348.48 | 139.41 | 234.2 | 299.73 | 184.15 | 225.95 | 146.37 | 64.21 | 242.07 | 112.91 | 540.83 | 219.29 | 86.28 | 70.73 | 229.17 | 592.95 |
| Cardboard - White (1mm) | 3035.47 | 1045.34 | 1702.47 | 700.04 | 1044.29 | 1174.36 | 718.65 | 788.87 | 527.66 | 201.49 | 428.97 | 199.6 | 1706.96 | 672.99 | 237.48 | 181.51 | 491.48 | 1101.06 |
| Construction Paper - Black | 585.39 | 175.66 | 256.82 | 103.16 | 159.55 | 169.77 | 86.44 | 99.06 | 61.56 | 106.95 | 337.21 | 146.4 | 183.2 | 219.29 | 199.72 | 140.27 | 327.71 | 858 |
| Construction Paper - Purple | 1150.98 | 488.73 | 831.09 | 268.6 | 342.43 | 334.36 | 193.07 | 213.57 | 321.64 | 253.44 | 335.65 | 154.05 | 579.67 | 784.27 | 300.18 | 193.44 | 356.77 | 853.93 |
| Construction Paper - Blue | 1110.38 | 468.13 | 816.02 | 314.51 | 379.6 | 333.14 | 148.79 | 159.92 | 180.11 | 242.97 | 332.41 | 153.1 | 347.75 | 667.68 | 310.26 | 196.85 | 349.46 | 865.17 |
| Construction Paper - Green | 747.38 | 237.1 | 378.43 | 272.32 | 525.31 | 527.18 | 260.22 | 246.38 | 115.36 | 211.63 | 333.7 | 151.18 | 358.51 | 439.2 | 301.72 | 192.59 | 357.48 | 868.99 |
| Construction Paper - Yellow | 726.39 | 248.32 | 350.29 | 315.81 | 776.22 | 946.22 | 666 | 909.63 | 749.63 | 254.88 | 357.97 | 172.23 | 2205.66 | 871.25 | 286.51 | 200.09 | 450.32 | 928.27 |
| Construction Paper - Orange | 848.18 | 293.39 | 433.24 | 266.74 | 395.11 | 410.45 | 273.49 | 615.52 | 856.41 | 277.02 | 344.48 | 166.49 | 2508.22 | 957.01 | 310.26 | 210.31 | 439.98 | 905.32 |
| Construction Paper - Pink | 1166.82 | 545.57 | 929.83 | 361.35 | 472.62 | 515.31 | 366.07 | 726.65 | 838.81 | 274.99 | 344.48 | 167.45 | 2481.24 | 945.78 | 307.53 | 211.17 | 445.15 | 890.02 |
| Paper - White | 2267.3 | 1466.56 | 2006.66 | 860.63 | 1027.13 | 1069.63 | 633.09 | 673.3 | 429.62 | 182.48 | 413.66 | 192.33 | 1445.21 | 606.83 | 220.4 | 167.02 | 457.09 | 1055.17 |
| Food Wrapping - Semi-Clear | 3143.4 | 1258.16 | 1911.55 | 642.78 | 998.53 | 1204.63 | 614.58 | 647.16 | 359.49 | 164.49 | 362.26 | 159.6 | 1197.07 | 512.5 | 186.91 | 145.04 | 394.9 | 859.67 |
| Single-Use Bag - Semi-Clear | 3762.26 | 1312.6 | 2123.54 | 761.19 | 1267.05 | 1562.58 | 459.54 | 587.58 | 425.61 | 167.1 | 388.09 | 177.59 | 1435.34 | 594.58 | 216.81 | 161.91 | 450.14 | 960.29 |
| Styrofoam - White | 2590.69 | 1099.23 | 1644.75 | 664.34 | 916.5 | 997.36 | 597.4 | 628.22 | 385.85 | 169.81 | 361.48 | 166.49 | 1357.23 | 537.62 | 199.89 | 149.98 | 407.91 | 892.65 |
| Thicker Plastic Bag - Semi-Clear | 3902.27 | 1770.8 | 3007.63 | 1116.41 | 2149.24 | 2437.21 | 598.29 | 869.41 | 481.6 | 205.29 | 416.38 | 185.63 | 3617.97 | 876.76 | 409.36 | 269.97 | 621.04 | 937.35 |
| Plastic Sticker Material - White | 3575.32 | 1252.46 | 2012.29 | 681.26 | 1134.6 | 1321.63 | 648.26 | 697.11 | 380.83 | 152.49 | 404.31 | 183.14 | 1627.1 | 534.96 | 194.6 | 155.09 | 469.57 | 1029.84 |
| Plastic Food Container - White (PP-5) | 2687.33 | 1451.12 | 1990.32 | 732.75 | 1039.02 | 1167.68 | 595.06 | 631.71 | 361.13 | 140.24 | 305.67 | 146.4 | 1397.38 | 482.49 | 171.87 | 132.94 | 365.14 | 770.28 |
| Plastic Food Container - Blue (PETE-1) | 1772.41 | 612.34 | 1059.78 | 251.87 | 513.87 | 500.59 | 88.78 | 74.61 | 45.6 | 69.27 | 171.46 | 83.44 | 285.22 | 132.52 | 114.47 | 79.25 | 184.62 | 423.02 |
| Plastic Shrimp Bag - White/Grey | 2008.07 | 680.03 | 1355.99 | 376.04 | 724.29 | 831.54 | 305.84 | 346.92 | 176.19 | 83.55 | 312.55 | 131.85 | 661.5 | 199.69 | 80.98 | 79.93 | 277.11 | 657.72 |
| Plastic Milk Bag - White | 4370.82 | 1326.96 | 1972.9 | 694.83 | 1276.24 | 1413.67 | 765.6 | 973.97 | 513.25 | 192.53 | 296.84 | 150.22 | 2976.43 | 732.4 | 287.54 | 201.28 | 520.35 | 740.17 |
| Plastic Cider Container - Semi-Clear (HDPE-2) | 1536.55 | 471.81 | 751.23 | 253.92 | 409.11 | 471.27 | 220.29 | 248.81 | 175.73 | 71.39 | 99.94 | 57.22 | 572.86 | 222.76 | 88.84 | 60.67 | 120.47 | 214.14 |
| Ziploc Bag - Clear | 3983.07 | 1278.57 | 1772.16 | 783.68 | 1251.55 | 1331.17 | 348.11 | 573.19 | 376.46 | 151.56 | 360.18 | 168.22 | 1722.32 | 389.78 | 191.35 | 128.68 | 365.68 | 563.08 |

*Figure 1: Graph of values[10] measured at 18 different frequencies for 22 materials from a database*

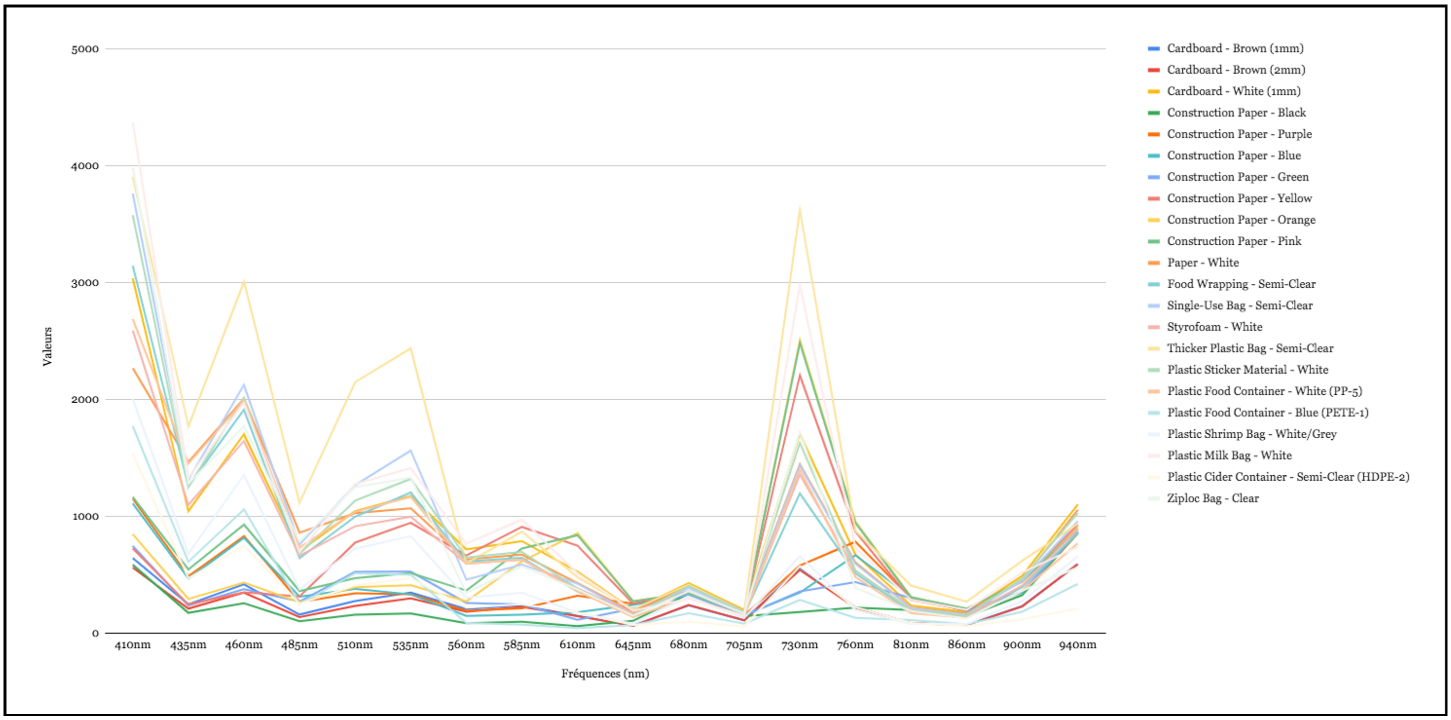


*Table 2: Values measured at 18 different frequencies for brown cardboard (1mm) at five different positions*

| Carton brun - 1mm: | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 648.96 | 247.4 | 406.01 | 161.53 | 269.43 | 342.27 | 192.4 | 231.77 | 151.84 | 65.05 | 238.82 | 111.95 | 548.07 | 229.91 | 87.99 | 71.58 | 222.22 | 590.56 | 1.12 |
| Essai 2: | ☑ | 648.96 | 247.4 | 406.38 | 161.72 | 269.58 | 342.41 | 192.51 | 231.77 | 151.84 | 65.05 | 238.82 | 111.95 | 548.51 | 230.93 | 87.99 | 71.58 | 222.04 | 589.85 | 1.12 |
| Essai 3: | ☑ | 652.72 | 249.24 | 418.18 | 161.53 | 275.45 | 349.09 | 206.35 | 237.38 | 148.01 | 65.9 | 238.95 | 111.95 | 558.38 | 223.58 | 87.13 | 71.58 | 222.22 | 582.91 | 0.28 |
| Essai 4: | ☑ | 640.24 | 247.4 | 411.1 | 167.29 | 276.35 | 346.5 | 199.65 | 239.82 | 153.57 | 65.05 | 238.82 | 111.95 | 564.74 | 224.4 | 87.99 | 71.58 | 221.68 | 585.54 | 0.42 |
| Essai 5: | ☑ | 658.47 | 250.71 | 414.91 | 158 | 271.68 | 346.36 | 197.31 | 229.87 | 147.74 | 65.64 | 238.3 | 111.95 | 543.02 | 226.65 | 87.13 | 71.58 | 222.22 | 587.46 | 0.94 |

[10] The values measured by the sensor are measured in terms of $\mu W/cm^2$, but this unit is inconvenient and difficult to visualize. We will therefore speak of data measured in values (*counts*).

*Table 3: Values measured at 18 different frequencies for white cardboard (1mm) at five different positions*

| Carton blanc - 1mm: | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 3024.39 | 1042.77 | 1709 | 686.65 | 1024.27 | 1165.63 | 682.28 | 759.35 | 499.84 | 188.73 | 425.73 | 198.07 | 1632.58 | 620.92 | 221.25 | 170.43 | 462.26 | 1079.07 | 2.34 |
| Essai 2: | ☑ | 3024.39 | 1050.85 | 1739.31 | 679.77 | 1022.01 | 1172.58 | 682.73 | 756.59 | 494.56 | 187.3 | 426.25 | 198.07 | 1615.25 | 618.88 | 220.57 | 170.43 | 459.94 | 1076.92 | 2.32 |
| Essai 3: | ☑ | 3041.02 | 1056.56 | 1759.09 | 662.67 | 1014.34 | 1168.77 | 701.47 | 750.67 | 476.32 | 188.39 | 420.54 | 194.43 | 1626 | 604.79 | 219.54 | 169.58 | 457.62 | 1056.36 | 2.51 |
| Essai 4: | ☑ | 2972.5 | 1032.28 | 1700.83 | 670.85 | 1012.23 | 1153.36 | 704.93 | 770.67 | 492 | 187.13 | 418.33 | 194.24 | 1629.95 | 605.6 | 219.54 | 168.56 | 451.04 | 1041.07 | 3.46 |
| Essai 5: | ☑ | 2970.92 | 1034.85 | 1719.71 | 639.25 | 987.69 | 1151.04 | 662.2 | 712.89 | 456.25 | 183.32 | 409.76 | 190.41 | 1563.69 | 614.39 | 215.96 | 167.71 | 451.39 | 1060.19 | 5.05 |

*Table 4: Values measured at 18 different frequencies for black paper at five different positions*

| Papier noir: | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 583.01 | 175.29 | 258.46 | 101.31 | 158.95 | 170.45 | 85.89 | 96.95 | 60.19 | 100.11 | 334.09 | 144.67 | 181.01 | 216.84 | 177.68 | 126.12 | 307.76 | 842.46 | 2.3 |
| Essai 2: | ☑ | 577.47 | 174.74 | 258.64 | 103.16 | 160.45 | 173.18 | 85.89 | 97.89 | 60.92 | 100.11 | 334.87 | 145.44 | 185.4 | 218.89 | 179.22 | 126.46 | 308.65 | 846.77 | 1.89 |
| Essai 3: | ☑ | 566.97 | 174 | 258.64 | 98.33 | 153.23 | 169.09 | 78.08 | 92.07 | 59.73 | 98 | 329.03 | 142.57 | 170.69 | 218.07 | 176.83 | 125.27 | 304.73 | 833.38 | 4.14 |
| Essai 4: | ☑ | 585.39 | 175.11 | 256.82 | 101.31 | 158.04 | 169.77 | 84.77 | 97.37 | 61.56 | 100.11 | 333.7 | 145.44 | 175.52 | 214.8 | 177.68 | 125.27 | 305.26 | 831.71 | 2.84 |
| Essai 5: | ☑ | 565.39 | 173.45 | 257.73 | 98.52 | 152.02 | 166.36 | 76.96 | 91.02 | 58.82 | 97.15 | 325.92 | 141.61 | 174.43 | 216.44 | 176.83 | 125.27 | 305.44 | 844.38 | 4.16 |

*Table 5: Values measured at 18 different frequencies for white paper at five different positions*

| Papier blanc: | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 2266.51 | 1467.48 | 2004.66 | 869.74 | 1035.11 | 1075.49 | 634.77 | 674.57 | 422.51 | 179.1 | 426.51 | 197.49 | 1450.92 | 585.6 | 215.1 | 164.47 | 452.28 | 1081.22 | 0.19 |
| Essai 2: | ☑ | 2277.99 | 1460.12 | 1957.65 | 879.78 | 1029.84 | 1049.58 | 637.23 | 679.55 | 427.62 | 178.68 | 421.83 | 196.15 | 1443.46 | 576.21 | 212.71 | 161.06 | 441.95 | 1040.12 | 0.68 |
| Essai 3: | ☑ | 2310.47 | 1493.05 | 2020.82 | 883.31 | 1041.43 | 1076.72 | 630.31 | 672.98 | 427.07 | 180.37 | 428.97 | 199.02 | 1444.33 | 585.8 | 216.47 | 164.47 | 450.32 | 1070.7 | 0.8 |
| Essai 4: | ☑ | 2347.9 | 1516.23 | 2058.75 | 892.79 | 1055.43 | 1091.45 | 632.43 | 672.98 | 425.52 | 181.29 | 433.78 | 201.32 | 1456.18 | 594.79 | 218.69 | 167.02 | 455.31 | 1084.56 | 2.02 |
| Essai 5: | ☑ | 2316.21 | 1501.51 | 2038.79 | 878.48 | 1038.57 | 1078.49 | 625.4 | 668.86 | 423.51 | 178.26 | 426.9 | 198.07 | 1443.02 | 587.64 | 216.13 | 165.32 | 450.5 | 1075.96 | 0.9 |

*Table 6: Values measured at 18 different frequencies for blue PETE plastic at five different positions*

| PETE (#1) - bleu | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 1792.21 | 610.86 | 1096.81 | 257.26 | 483.76 | 453.82 | 88.67 | 75.14 | 48.61 | 63.78 | 169.77 | 83.25 | 286.32 | 125.77 | 114.47 | 78.4 | 173.04 | 409.64 | 0.8 |
| Essai 2: | ☑ | 1783.1 | 613.07 | 1098.07 | 257.45 | 484.06 | 454.23 | 88.67 | 75.14 | 48.61 | 63.78 | 169.77 | 83.25 | 286.76 | 127.2 | 114.47 | 78.4 | 173.22 | 410.12 | 0.84 |
| Essai 3: | ☑ | 1799.74 | 651.52 | 1176.12 | 250.57 | 503.78 | 492.68 | 88.67 | 75.14 | 44.23 | 61.67 | 170.03 | 81.33 | 298.39 | 149.05 | 116.18 | 81.81 | 181.06 | 435.21 | 2.92 |
| Essai 4: | ☑ | 1683.29 | 695.48 | 1068.49 | 252.43 | 502.73 | 469.77 | 88.67 | 75.14 | 45.6 | 64.04 | 172.63 | 82.29 | 301.68 | 132.31 | 113.62 | 77.55 | 176.6 | 422.55 | 0.6 |
| Essai 5: | ☑ | 1777.56 | 718.47 | 1156.15 | 259.49 | 508.6 | 497.72 | 88.67 | 75.67 | 45.6 | 65.05 | 170.03 | 83.63 | 331.3 | 137.62 | 115.32 | 80.1 | 187.29 | 418.48 | 3.78 |

*Table 7: Values measured at 18 different frequencies for dark blue PETE at five different positions*

| PETE (#1) - bleu v2 | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 1380.7 | 500.87 | 897.7 | 209.31 | 382.61 | 386.18 | 75.63 | 75.14 | 39.67 | 26.61 | 218.7 | 95.68 | 253.41 | 98.01 | 44.42 | 47.72 | 214.2 | 559.73 | 17.38 |
| Essai 2: | ☑ | 1473.58 | 518.34 | 951.61 | 204.1 | 403.24 | 435.14 | 75.85 | 74.61 | 39.21 | 28.3 | 218.7 | 95.68 | 217.21 | 90.25 | 41.86 | 46.02 | 206.54 | 547.54 | 14.03 |
| Essai 3: | ☑ | 1451.59 | 541.89 | 949.61 | 229.38 | 420.24 | 429.27 | 75.85 | 84.14 | 44.69 | 32.95 | 221.3 | 99.51 | 272.72 | 84.54 | 42.71 | 45.85 | 209.39 | 542.05 | 11.86 |
| Essai 4: | ☑ | 1472.79 | 564.7 | 1087 | 226.59 | 439.51 | 451.63 | 74.73 | 74.61 | 40.58 | 27.03 | 210.92 | 94.73 | 201.19 | 85.76 | 41.86 | 47.72 | 215.27 | 554.71 | 9.33 |
| Essai 5: | ☑ | 1372.18 | 517.42 | 948.16 | 215.62 | 447.19 | 468.68 | 72.5 | 77.47 | 42.41 | 34.21 | 214.81 | 96.64 | 248.58 | 120.88 | 45.28 | 49.43 | 216.88 | 570.25 | 12.03 |

*Table 8: Values measured at 18 different frequencies for HDPE plastic at five different positions*

| HDPE (#2) - semi-tr | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 1409.22 | 446.06 | 737.8 | 235.88 | 397.37 | 473.18 | 223.75 | 242.04 | 166.43 | 69.7 | 98.13 | 55.12 | 558.82 | 208.27 | 80.3 | 56.24 | 116.72 | 218.92 | 4.37 |
| Essai 2: | ☑ | 1373.17 | 431.89 | 699.86 | 224.73 | 367.71 | 437.86 | 217.16 | 232.83 | 160.96 | 67.59 | 92.15 | 53.58 | 548.51 | 204.38 | 79.45 | 55.56 | 116.01 | 214.14 | 8.42 |
| Essai 3: | ☑ | 1493.97 | 458.38 | 700.59 | 264.88 | 417.84 | 473.32 | 224.41 | 254.53 | 190.6 | 70.54 | 99.29 | 56.83 | 599.19 | 224.6 | 84.57 | 57.09 | 117.79 | 221.07 | 0.62 |
| Essai 4: | ☑ | 1501.89 | 460.59 | 730.72 | 258.19 | 410.91 | 478.22 | 228.1 | 259.82 | 181.93 | 71.22 | 100.59 | 56.45 | 594.58 | 225.83 | 83.72 | 57.95 | 119.58 | 218.2 | 0.14 |
| Essai 5: | ☑ | 1467.24 | 454.51 | 738.34 | 230.49 | 387.88 | 471.41 | 215.27 | 229.66 | 162.6 | 69.27 | 97.35 | 54.54 | 549.61 | 218.48 | 80.3 | 57.09 | 117.79 | 224.66 | 3.78 |

*Table 9: Value s measured at 18 different frequencies for PP plastic at five different positions*

| PP (#5) - blanc | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 2628.51 | 1439.34 | 2002.67 | 712.49 | 1031.49 | 1172.04 | 596.4 | 626.64 | 348.37 | 139.82 | 304.37 | 143.91 | 1416.47 | 467.17 | 170 | 131.75 | 354.99 | 761.2 | 0.84 |
| Essai 2: | ☑ | 2629.31 | 1437.87 | 2001.58 | 712.12 | 1031.19 | 1171.22 | 596.62 | 626.74 | 348.37 | 139.82 | 304.37 | 143.72 | 1415.59 | 467.37 | 169.66 | 131.23 | 355.16 | 762.16 | 0.86 |
| Essai 3: | ☑ | 2656.63 | 1432.72 | 1960.56 | 740.55 | 1048.35 | 1173.4 | 596.18 | 634.67 | 359.68 | 140.66 | 308.52 | 146.4 | 1434.46 | 480.65 | 170.85 | 132.09 | 359.61 | 779.61 | 0.09 |
| Essai 4: | ☑ | 2647.32 | 1431.8 | 1961.28 | 719.92 | 1037.36 | 1162.9 | 595.73 | 624.83 | 351.01 | 141.08 | 305.8 | 144.48 | 1437.09 | 480.03 | 170.85 | 132.09 | 359.08 | 773.87 | 0.64 |
| Essai 5: | ☑ | 2683.96 | 1448.17 | 2001.76 | 754.31 | 1059.79 | 1188.95 | 616.36 | 666.74 | 382.56 | 144.46 | 309.17 | 147.73 | 1463.64 | 482.49 | 175.12 | 134.64 | 357.83 | 766.46 | 1.46 |

*Table 10: Values measured at 18 different frequencies for plastic wrap at five different positions*

| Film alimentaire | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | Single-use bag | 3935.54 | 1372.38 | 2210.48 | 703.75 | 1201.43 | 1614.68 | 599.52 | 655.74 | 401.17 | 183.41 | 391.85 | 175.48 | 1629.95 | 623.98 | 211 | 162.25 | 458.16 | 983.23 | 4 |
| Essai 2: | PP (#5) | 2776.84 | 1184.21 | 1765.63 | 729.22 | 1121.35 | 1275.95 | 576.99 | 632.56 | 386.94 | 183.75 | 377.19 | 172.23 | 1553.81 | 546.6 | 197.85 | 147.42 | 416.64 | 904.36 | 2.55 |
| Essai 3: | Ziploc bag | 4631.44 | 1583.18 | 2281.45 | 870.3 | 1361.58 | 1539.94 | 701.36 | 823.9 | 446.4 | 193.04 | 396.01 | 180.27 | 1861.42 | 629.5 | 224.67 | 168.73 | 457.62 | 920.86 | 6.88 |
| Essai 4: | Single-use bag | 3371.74 | 1544.92 | 2425.2 | 852.83 | 1362.48 | 1492.76 | 519.66 | 614.57 | 380.28 | 169.64 | 373.81 | 175.68 | 1652.11 | 586.62 | 213.73 | 161.57 | 420.92 | 881.66 | 2.25 |
| Essai 5: | Plastic sticker ma | 3289.95 | 1331.92 | 2183.08 | 671.22 | 1173.43 | 1460.17 | 622.05 | 623.35 | 353.75 | 170.15 | 389.52 | 173.19 | 1537.8 | 598.46 | 213.56 | 168.22 | 466 | 977.5 | 0.31 |

*Table 11: Values measured at 18 different frequencies for a milk bag at five different positions*

| Sac de lait - transpare | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | PP (#5) | 2947.75 | 1125.72 | 1948.4 | 620.85 | 1314.47 | 1544.72 | 386.26 | 452.32 | 270.48 | 135.59 | 372.12 | 143.53 | 1608.67 | 441.24 | 179.39 | 129.53 | 374.41 | 703.13 | 0.88 |
| Essai 2: | Ziploc bag | 4251.6 | 1236.27 | 1857.47 | 429.02 | 1028.03 | 1315.49 | 393.73 | 433.38 | 262.28 | 145.06 | 322.15 | 132.04 | 1370.84 | 479.63 | 188.79 | 138.9 | 385.63 | 757.62 | 4.04 |
| Essai 3: | ☑ | 3798.3 | 1206.84 | 1942.77 | 594.45 | 1225.21 | 1488.67 | 367.41 | 462.49 | 283.53 | 155.11 | 296.58 | 133 | 2675.63 | 535.78 | 275.92 | 175.21 | 384.39 | 675.17 | 16.37 |
| Essai 4: | Styrofoam | 2798.82 | 1008.92 | 1682.32 | 483.29 | 954.73 | 1183.49 | 414.7 | 498.68 | 310.61 | 123.43 | 322.54 | 140.66 | 1326.95 | 485.55 | 223.47 | 159.01 | 393.3 | 751.88 | 3.81 |
| Essai 5: | Single-use bag | 3286.58 | 1300.46 | 2207.94 | 547.79 | 1120.75 | 1316.58 | 394.63 | 491.38 | 305.5 | 134.83 | 328.9 | 144.67 | 1516.3 | 466.76 | 223.98 | 146.57 | 391.52 | 760.72 | 2.14 |

*Table 12: Values measured at 18 different frequencies for thin styrofoam at five different positions*

| Styromousse mince | Identification | 410nm | 435nm | 460nm | 485nm | 510nm | 535nm | 560nm | 585nm | 610nm | 645nm | 680nm | 705nm | 730nm | 760nm | 810nm | 860nm | 900nm | 940nm | % d'écart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Essai 1: | ☑ | 2640.39 | 1112.84 | 1642.21 | 667.69 | 913.64 | 996.13 | 606.21 | 641.23 | 382.2 | 164.74 | 366.67 | 168.98 | 1363.59 | 507.6 | 189.98 | 144.87 | 396.86 | 890.74 | 0.21 |
| Essai 2: | ☑ | 2647.32 | 1120.57 | 1637.49 | 679.77 | 918.01 | 984.4 | 606.88 | 646.21 | 389.13 | 163.89 | 366.28 | 169.55 | 1343.41 | 495.35 | 188.79 | 142.31 | 389.38 | 865.41 | 0.1 |
| Essai 3: | ☑ | 2581.58 | 1087.46 | 1580.13 | 660.62 | 892.27 | 979.08 | 566.06 | 620.07 | 393.96 | 161.78 | 360.05 | 167.45 | 1328.93 | 516.38 | 189.64 | 143.16 | 391.69 | 900.06 | 1.83 |
| Essai 4: | ☑ | 2640.59 | 1097.58 | 1586.67 | 654.12 | 894.07 | 972.95 | 587.81 | 618.17 | 379.46 | 165.08 | 365.5 | 169.17 | 1310.06 | 517.81 | 187.93 | 143.16 | 393.65 | 891.94 | 1.41 |
| Essai 5: | ☑ | 2586.13 | 1108.8 | 1607.54 | 687.2 | 906.41 | 975.68 | 593.5 | 652.56 | 404 | 161.7 | 364.59 | 168.6 | 1362.06 | 511.68 | 190.5 | 143.5 | 392.76 | 889.79 | 0.44 |

This experimental phase mainly took place in two stages. Table 1 and graph 1 illustrate the spectra of about twenty different materials that are part of the program's database. The values collected are an average of five measurements taken by the sensor for each of the 18 frequencies measured between 410 and 940nm. This frequency length includes part of the visible electromagnetic spectrum as well as part of the near infrared. While the sensor measures the amount of reflected light, the greatest differences between materials can be observed at frequencies of 410, 460 and 760nm and the smallest differences at frequencies of 645, 680, 705, 810, 860, 900 and 940nm which suggests that similar results could be obtained without considering these values.

Tables 2 to 12 contain the spectra of 11 different materials scanned by the sensor at five different points on the object to test for local variability. I also noted the percent difference between the database spectra and the spectra of the scanned material in order to quantify this local variability. The checkmarks on the left side of each table indicate whether or not the scanned material was correctly identified by the program.

Tables 2 through 5 show the spectra of pieces of brown and white cardboard as well as black and white paper that I scanned five times at different locations. We see a slight variation in the values with each test, but the results remain very precise since the average percent difference between the values in the database and the values measured rarely exceeds 4%. In addition, the materials were correctly identified every time. However, I note that the brown cardboard and the white paper had significantly lower deviation percentages than the white cardboard and the black paper.

Tables 6 and 7 are unique in that both are nearly identical objects. Table 6 contains the spectrum of a small blue plastic container belonging to Category 1 (polyethylene terephthalate). Note that the spectral data of this object is in the database. For a reliable sensor, we would expect the blue container to be correctly identified every time as shown by the small percent difference. Table 7 contains the spectral data of a very similar container, but whose colour and thickness differ slightly from the container in Table 6. However, despite their similarities, the average percentage difference is much greater, i.e. about 15%. Despite a high average deviation percentage, the two containers were identified correctly.

Tables 8 and 9 group the spectra of hard plastic objects, ie categorie 2 (high-density polyethylene) and 5 (polypropylene) of the CIR. The deviation percentages are relatively small and the two materials were identified without fault. This is the same case for the styrofoam object of table 12.

Finally, tables 10 and 11 contain the spectra of soft plastics: plastic wrap and a bag of milk. These two plastics are more or less transparent and had uneven surfaces that made them difficult to identify. In five attempts, the plastic wrap was never identified correctly and the bag of milk only once. The deviation percentages are also extremely varied. Note also a trend that is maintained through the majority of the spectra collected: the frequencies previously identified as having small differences between the various materials have in fact less great variations in the experimental results.

**Analysis**

During my experiment, I had to control and account for many different factors. Starting with the sensor, it illuminates objects with three diodes of different electromagnetic frequencies: ultraviolet, visible (white) and infrared. These three lights ensure uniform illumination for each object. In order to control the distance at which the spectroscopic measurements are taken, I have fixed the height of the case at approximately 3.8 cm (1.5 inches). In addition, the housing serves to control the ambient light since it can have a considerable impact on the results obtained. However, several of the objects that were part of the experiment were more or less transparent. Thus, the amount of ambient light that reached the sensor varied depending on the transparency and thickness of the material. To account for this variability, I illuminated the whole sensor with a powerful LED, thus rendering the ambient light negligible.

The experiment focuses on the identification of common materials that would be found in a recycling center. Thus, I chose a wide variety of plastic and cardboard items to include in my database: paper, cardboard, plastic containers, juice containers, plastic wrap, milk bags, etc. I then used my program to add their spectra to the database. At the same time, I inserted this data into a graph, as shown in figure 1. As mentioned previously, I noticed that the values for some frequencies are quite similar regardless of the material. It therefore would be possible to redo this same experiment while omitting certain frequencies to reduce the computation required by the program. We would still obtain similar results despite a slight loss of precision in the data.

Following this step, I continued the experiment by scanning these same materials again, as shown in tables 2 through 12. What I wanted to verify was the sensor's ability to consistently identify predetermined materials. If it couldn't identify the same piece of cardboard repeatedly, I knew the experiment would be flawed from the start. Fortunately, this was not the case: the vast majority of materials were identified correctly several times. However, some materials such as the plastic wrap (Table 9) and the milk bag (Table 10) were never correctly identified by the sensor.

This might be explained by the fact that the spectrometer is highly sensitive to surface variation. Since it measures the reflectivity of a material, objects with a smooth and flat surface reflect light more evenly, providing reliable data with every measurement. However, thin malleable plastics such as the plastic wrap reflect light unevenly, especially when crumpled up or flattened. This provides inconsistent data as light is scattered off the material at many different angles. Thus, cardboard, paper and hard plastics are exceptional candidates for this technology as light will evenly scatter across their surfaces.

Following my analysis of the items included in the database, I turned my focus to the sensor's ability to identify unknown materials. I found two similar blue plastic containers made of the same material (PETE) and added only one of the two container's spectra to the database. I then attempted to identify the other container. As shown in tables 6 and 7, both containers were correctly identified despite a significantly larger percent difference for the second container.

This larger percent difference brought me to consider a potential problem with my database system. By calculating the Euclidean distance between experimental data and the spectral data of the database, the computation power required increases with every new material added to the database. More importantly however, the spectra of some materials often closely resemble the spectra of other materials. As such, small inconsistencies in the sensor or small variations in the data could potentially result in some objects being misidentified. This underlines an underlying problem with the current code and the necessity to vastly improve the comparison algorithm in the future. Including a deep-learning algorithm would allow the program to create its own database by grouping similar spectra in the same category.

I believe an improved algorithm could also solve a significant problem I noticed during this experiment: colour. Preliminary tests had shown that colour could have a considerable impact on the data since it directly influences the reflectivity of an object. Knowing that increasingly white surfaces reflect more and more light, lighter colors have higher spectra and vice versa. In short, the colour of an object can vary the spectra of two objects made of the same material.

Lastly, it should be noted that the uncertainty associated with the sensor is quite high at around 12%. This uncertainty is not to be neglected and could explain some results such as high percent differences for some objects. In terms of possible improvements to the experiment, a stricter control of ambient light and an optimized comparison algorithm could drastically improve results, not to mention using a spectroscopic sensor with a smaller uncertainty.

**Conclusion**

During this experiment, I usd a NIR spectroscopic sensor to measure the reflectivity of plastic objects. The spectral data that I obtained was then processed by a Python program I wrote in order to match scanned objects to a database of materials. The goal of this experiment was to investigate NIR technology as an efficient way to identify unknown materials.

In the testing phase of this experiment, the sensor successfully identified 9 of 11 materials with high accuracy, allowing me to conclude that the sensor is both reliable and precise. In the case of the two other materials, these proved to be quite difficult to identify since their surfaces were semi-transparent and reflected light unevenly despite constant illumination. Considering the uncertainty of the sensor and the experimental results presented in this report, I have concluded that NIR technology has the potential to revolutionize the recycling industry. Its ability to reliably identify materials could be a game-changer for this field that still relies heavily on the manual sorting of recyclable materials. However, while much improvement is still needed, most problems reside in the data processing aspect of this technology and not in the spectroscopic sensors themselves. Variables such as colour, degree of transparency and levels of ambient light still need to be addressed before this technology becomes truly reliable and a viable alternative to manual sorting systems.

**References**

UN2019. *Plastics Recycling: Inefficient Sector Ready for Change*. On line. [https://www.unenvironment.org/fr/actualites-et-recits/recit/recyclage-des-plastiques-un-secteur-inefficace-pret-pour-le-changement].

National Geographic. 2018. *7 Things You Didn't Know About Plastic (and Recycling)*. On line. [https://blog.nationalgeographic.org/2018/04/04/7-things-you-didnt-know-about-plastic-and-recycling/].

CAMPION, Juliet. 2019. "Learn Why Plastic Recycling Is Creating A Global Waste Crisis." *FranceInfo*. June 7. On line. [https://www.francetvinfo.fr/sante/environnement-et-sante/on-vous-explication-pourquoi-le-recyclage-du-plastique-est-en-train-de-creer-une-crise-mondiale -des-dechets_3465921.html].

SPECTRAL ENGINES. 2019. *NIR Technology and the Plastic Pollution Crisis.* [https://www.spectralengines.com/blog/nir-technology-and-the-plastic-pollution-crisis].

THERMO FISHER. *NIR Technology*. [https://www.thermofisher.com/ca/en/home/industrial/spectroscopy-elemental-isotope-analysis/spectroscopy-elemental-isotope-analysis-learning-center/molecular-spectroscopy-information/nir-technology.html].

HAMEDANI, Mosh. Software engineer. 2019. "Python Tutorial - Python for Beginners." *Programming with Mosh*. On line. 6 hours, 14 minutes, 6 seconds. Watched on YouTube. [https://www.youtube.com/watch?v=_uQrJ0TkZlc&ab_channel=ProgrammingwithMosh].

APPENDIX I

*Program Arduino*

```
char userInput;
#include "SparkFun_AS7265X.h"
AS7265X sensor;

#include <Wire.h>

void setup() {
  Serial.begin(9600);

  if(sensor.begin() == false)
  {
    Serial.println("Sensor does not appear to be connected. Please check
wiring. Freezing...");
    while(1);
  }

  sensor.disableIndicator(); //Turn off the blue status LED

}

void loop() {
```

```
  if(Serial.available() > 0){

    userInput = Serial.read();

      if(userInput == 'g'){
        sensor.takeMeasurementsWithBulb(); //This is a hard wait while all 18
channels are measured

        Serial.print(sensor.getCalibratedA());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedB());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedC());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedD());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedE());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedF());
        Serial. print(" ");

        Serial.print(sensor.getCalibratedG());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedH());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedI());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedJ());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedK());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedL());
        Serial. print(" ");

        Serial.print(sensor.getCalibratedR());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedS());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedT());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedU());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedV());
        Serial. print(" ");
        Serial.print(sensor.getCalibratedW());
        Serial. print(" ");

        Serial.println();
      }
   }
   delay(50);
}
```

APPENDIX II

*Python program*

```
from ast import literal_eval
import csv
import serial
import numpy as np
import math
import time

data_bank = [[], [], [], [], [], [], [], [], [] , [], [], [], [], [], [], [],
[], []]
data_bank_average = [[], [], [], [], [], [], [ ], [], [], [], [], [], [], [],
[], [], [], []]
average_list = []
ser = serial.Serial("/dev/cu .usbmodem1D11141", baudrate=9600, timeout=1)


def updating_list():
   counts = 0
   while counts != 5:
       ser.write(b'g')
       data = ser.readline().decode('ascii')
       split_data = data.split()
       material = split_data
       updated_material = []
       for m in material:
           updated_value = float(m)
           updated_material.append(updated_value)
       print(updated_material)

       for d in range(0, len(data_bank_average)):
           data_bank_average[d ].append(updated_material[d])
       counts += 1
       time.sleep(0.2)
   print(data_bank_average)
   return


def finding_average():
   for d in range(0, len(data_bank_average)):
       average = 0
       for i in range( 0, len(data_bank_average[d])):
           average += data_bank_average[d][i]
       average = average / len(data_bank_average[0])
       average = round(average, 2)
       average_list.append(average)
```

```
    return average_list


print('''
Welcome to the Material Identifier. What would you like to do?
> Add a material (1)
> Scan a material (2)
> See a list of the available materials (3)
> About the program (4)
> Terminate (0)
    ''')
while True:
    command = str(input( "Enter command: "))
    if command == '1':
        add_data = input('Add data? (press "y") ').lower()
        if add_data == 'y':
            average_list = []
            data_bank_average = [ [], [], [], [], [], [], [], [], [], [], [],
[], [], [], [], [], [] , []]
            updating_list()
            print(f'The data collected is: {finding_average()}')
            with open('Data_Bank.csv', 'r') as read_file:
                reader = csv.reader(read_file)
                with open( 'Data_Bank.csv', 'a') as write_file:
                    writer = csv.writer(write_file)
                    material_name = input('What material do you wish to add? ')
                    writer.writerow([material_name, average_list])
                for row in reader :
                    print(row)
    elif command == '2':
        add_data = input('Get data? (press "y") ').lower()
        if add_data == 'y':
            average_list = []
            data_bank_average = [[], [], [], [], [], [], [], [], [], [], [],
[], [], [], [], [] , [], []]
            updating_list()
            finding_average()
            np.asarray(average_list)
            print(f'The data collected is: {average_list}')
            with open('Data_Bank.csv', 'r') as read_file:
                reader = csv.reader(read_file)
                next(reader)
                min_final_array = 1000000
                for row in reader:
                    final_array = 0
                    data_row = np.array(literal_eval(row[1]))
                    inter_array = (data_row - average_list)**2
                    for val in inter_array:
```

```
                              final_array += val
                        final_array = math.sqrt(final_array)
                        if final_array <= min_final_array:
                              min_final_array = final_array
                              material_name = row[0]
                              exp_values = 0
                              theo_values = 0
                              for dd in data_row:
                                    exp_values += dd
                              for ddd in average_list:
                                    theo_values += ddd
                              percentage_difference = (abs((theo_values -
exp_values))/theo_values)*100
                              percentage_difference =
round(percentage_difference, 2)
                  print(f'The material was identified as {material_name} with
a {percentage_difference} % deviation.')
   elif command == '3':
       print('Fetching data...')
       with open('Data_Bank.csv', ' r') as read_file:
           reader = csv.reader(read_file)
           for row in reader:
               print(row[0])
   elif command == '4':
       print('''
Luca Scavone created this Python program as part of his
Secondary 5 Personal Project. The program scans materials
using a NIR (near infrared) scanner and compares the
results to a data bank in order to identify various types
of materials (plastic, wood, paper, etc.)
           ''')
   elif command == '0':
       break
   elif command == '':
       continue
   else:
       print('Wrong Command. Try Again.')
       continue
```

APPENDIX III

*Spectroscopic sensor assembly*

*Figure 1: Spectroscopic sensor housing*

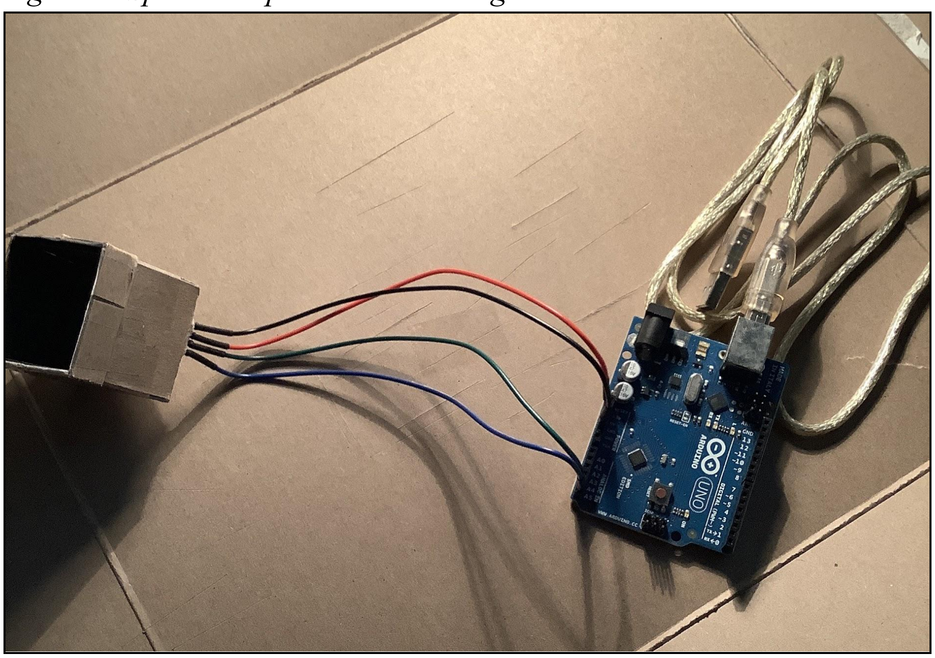

*Figure 2: Spectroscopic sensor fully completed*