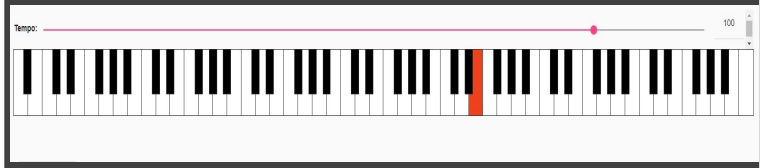
The Masked Scales



2020 NASA SpaceApps Covid19 Challenge Global Winner

Artash Nath (Grade 8)
Arushi Nath (Grade 5)
Toronto, Canada

Acoustics and COVID19 Pandemic
4th International Conference. 5 September 2020

ICA ACOUSTICAL SOCIETY OF NIGERIA
(A MEMBER OF INTERNATIONAL COMMISSION FOR ACOUSTICS)

The Masked Scales

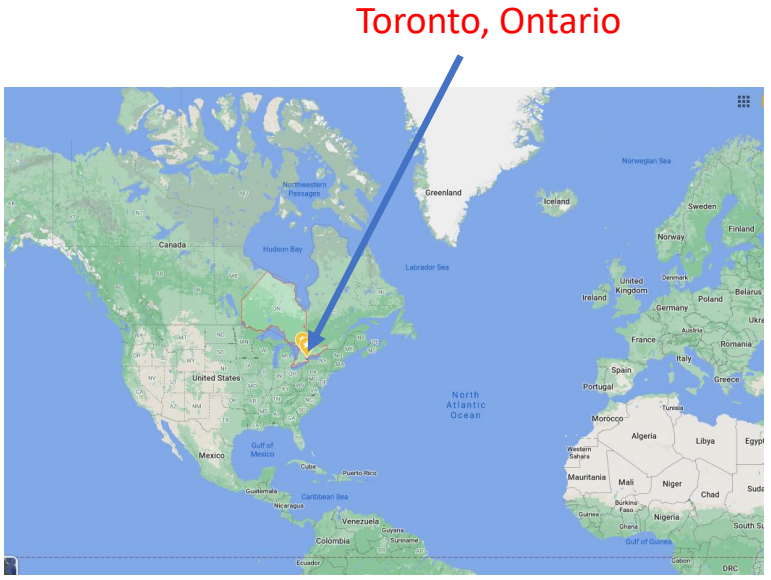
2020 SpaceApps Global Winners from 1422 projects from 150 countries.

COVID 19 Lockdown Timeline

Toronto, Ontario, Canada

- **March 11: WHO declares COVID19 as a pandemic**

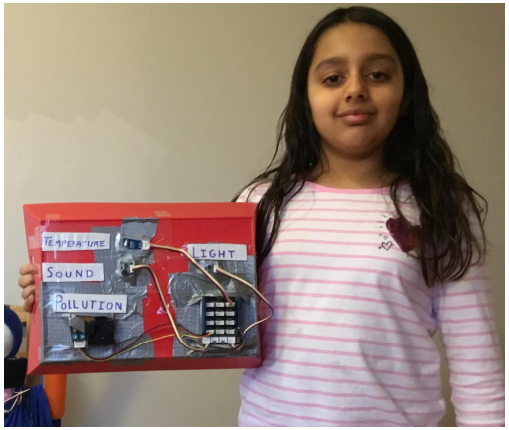
- *March 17: Ontario declares a state of emergency*
- *March 18: Canada - US border closed*
- *March 22: Ontario Schools closed*
- *March 23: Toronto declares state of emergency*
- *March 31: Majority Air Canada flights shut down*



Home-Made Instrument with Sensors

Measures Changes in Light, Noise and Pollution during COVID19 Lockdown

Laying out the Sensors on an Old Plastic Tray



Sensors attached and wired

Programming the Sensors using Arduino

Instrument deployed in front of our house

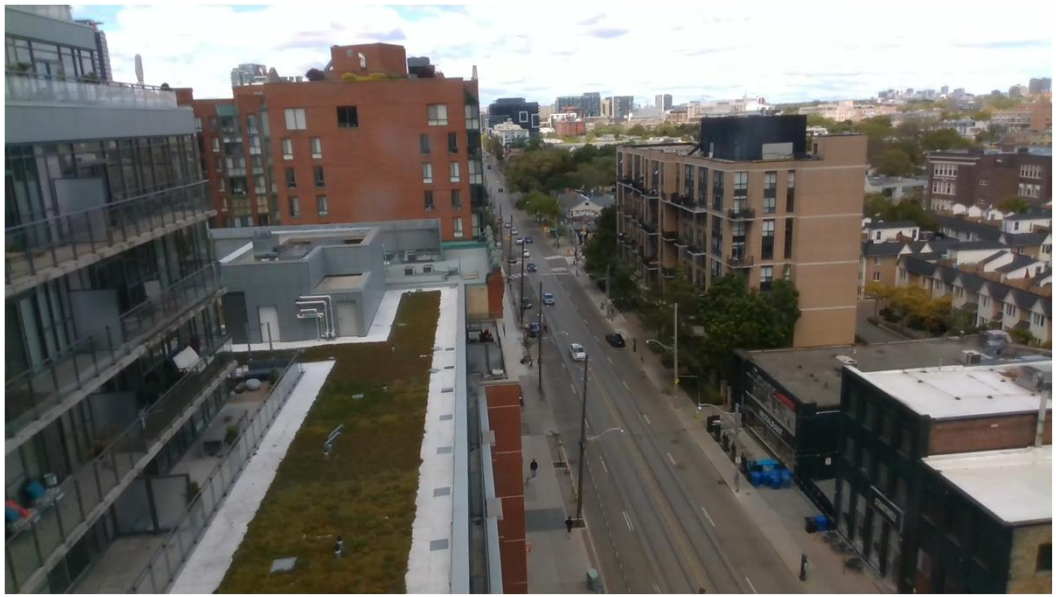
Sound Sensor

Light Sensor

Pollution Sensor

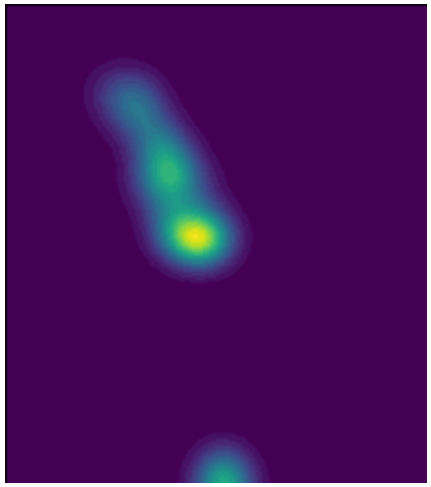
Installed a Camera to Collect Live Traffic Data during COVID19 Lockdown

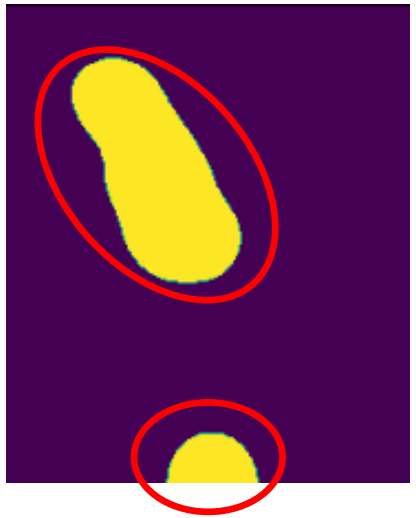
Getting Live Traffic Data on the computer



Live Vehicle Count from Street Camera using Artificial Intelligence / Machine Learning

Image Processing of Live Traffic Data using Machine Learning

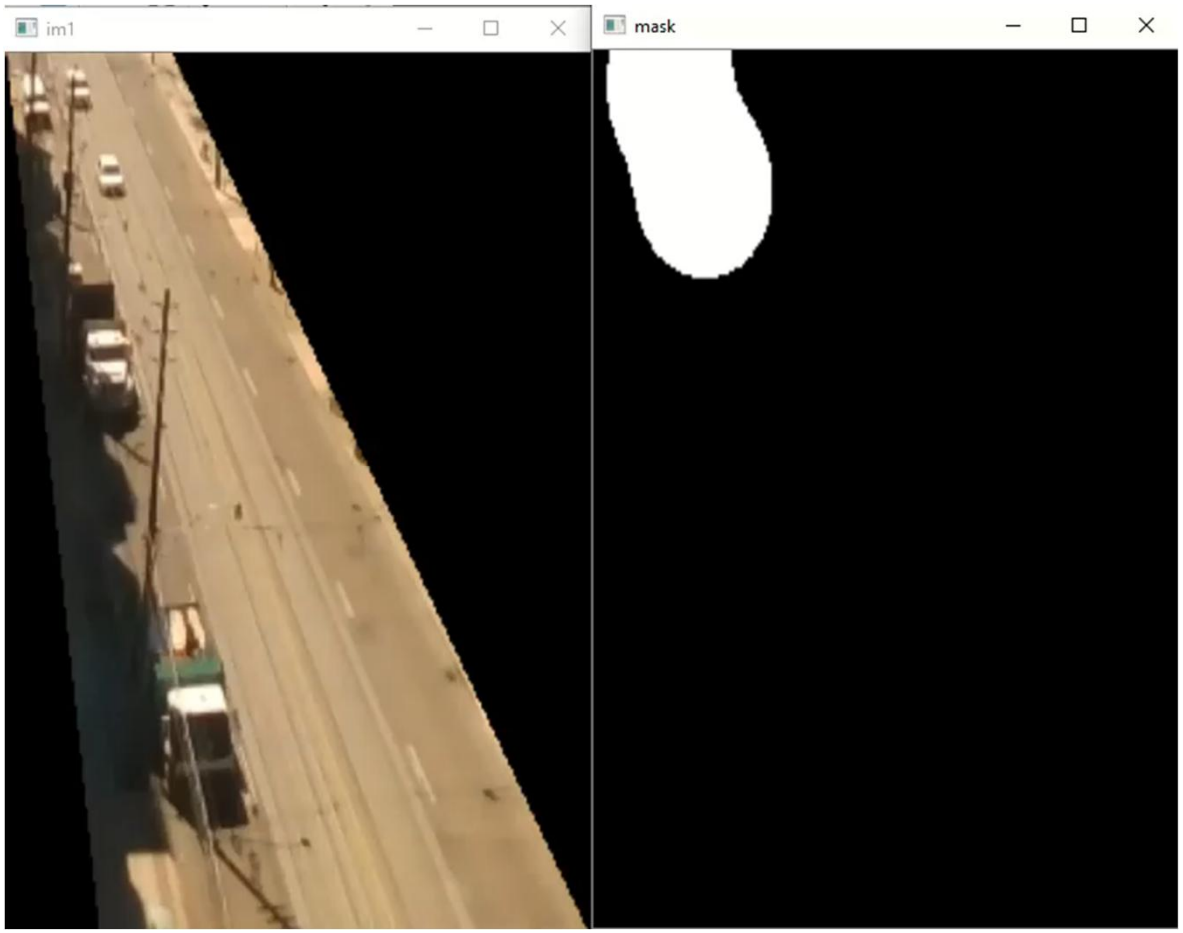




VEHICLE COUNT

2

Machine Learning Demo



6 Datasets Generated

Data Collected from Home-Made Instrument

1. Street Noise (Microphone)
2. Vehicular Emissions (PM 2.5 Sensor)
3. Vehicular Count on Street (Video Camera)
4. Light Intensity Data (Light Sensor)

External Data Sources

1. NASA Night Light Data (Suomi / VIIRS Data)

https://worldview.earthdata.nasa.gov

2. Toronto COVID19 Infection Data (City of Toronto)

https://www.toronto.ca/home/covid-19/covid-19-latest-city-of-toronto-news/covid-19-status-of-cases-in-toronto/

Techniques Used for Data Analysis

- **Arduino to Python Serial Bridge**

To transmit data from Sensors to Python for analysis

- **Machine Learning Algorithm**

for real-time vehicle counting from Street Camera

- **NASA Worldview Compare Tool**

https://worldview.earthdata.nasa.gov

- **Data Sonification**

Musical Algorithm software http://musicalgorithms.org/ to convert data into Music

Coding: Arduino, Python and Machine Learning

Complete code available on our GitHub: www.github.com/Artash-N

Arduino Code

```
#include "SHT31.h"
const int pinAdc = A3;

SHT31 sht31 = SHT31();

void setup() {
    Serial.begin(9600);
    pinMode(pin,INPUT);

    while (!Serial);
    Serial.println("begin...");
    starttime = millis();//get the current time
    sht31.begin();
}

void loop() {
   // delay(30000);

    duration = pulseIn(pin, LOW);
    lowpulseoccupancy = lowpulseoccupancy+duration;

    if ((millis()-starttime) > sampletime_ms)//if the sampel time == 30s
    {


    long sum = 0;
    for(int i=0; i<32; i++)
    {
        sum += analogRead(pinAdc);
    }

    sum >>= 5;
    Serial.print("sound: ");

    Serial.println(sum);
    delay(10);
```

Python Serial Bridge Code

```
dust1 = None
dust2 = None
dust3 = None
sound = None
temp = None
hum = None
log_read = None
res = None

while True:
    data = arduino.readline()[:-2] #the last bit gets rid of the new-line chars
    if data:
        print(data)
    csv_string = ''
    if b'Dust: ' in data:
        dust = data.decode()
        dust = (dust[dust.index(':')+1:]).split(',')
        dust1  = dust[0]
        dust2 = dust[1]
        dust3 = dust[2]
    elif b'sound: ' in data:
        sound = data.decode()
        sound = sound[sound.index(':')+1:]
    elif b'Temp: ' in data:
        temp = data.decode()
        temp = temp[temp.index(':')+1:]
    elif b'Hum: ' in data:
        hum = data.decode()
        hum = hum[hum.index(':')+1:]
    elif b'analog read data: 'in data:
        log_read = data.decode()
        log_read = log_read[log_read.index(':')+1:]
    elif b'sensor resistance: ' in data:
        res = data.decode()
        res = res[res.index(':')+1:]
```

Machine Learning Code for Live Traffic Analysis

```
images = []

main_dir =  r'E:\Multimodel Data Rec - Copy\run9\photos'
all_dirs = os.listdir(main_dir)
all_dirs = sorted(all_dirs, key=int)
COUNT = 0

for sub_dir in all_dirs:
    start = time.time()
    sub_dir = os.path.join(main_dir,sub_dir)
    dir_subpaths = os.listdir(sub_dir)
    im1 = cv2.imread(os.path.join(sub_dir, dir_subpaths[0]))[230:, 320:540]
    im2 = cv2.imread(os.path.join(sub_dir, dir_subpaths[1]))[230:, 320:540]

    im2 = remove_background(im2)
    im1 = remove_background(im1)
    cv2.imshow("im1", im1)
    key = cv2.waitKey(1) & 0xFF

    m = binarize(mask(im1,im2,10))
    m = m.astype('float32')
    cv2.imshow("mask", m)
    key = cv2.waitKey(1) & 0xFF

    car_count = list(np.array(connected_components(m)[0])>10).count(True)
    with open('./final-car-count/carcounting-runx9.csv', 'a+', newline='') as write_obj:
        csv_writer = csv.writer(write_obj)
        csv_writer.writerow([car_count])

    print("READING FOLDER : " + str(COUNT) + "| TIME/FOLDER ==> " + str(time.time()-start)+'   CARCOUNT :'+str(car_count))
    COUNT+=1
```

500,000 data values generated over 3 weeks

Biggest Challenge: Merging data from different sensors and Arduinos to a single Time Stamp to create Time Series Graphs

Data coming from Pollution Sensor (every 20 seconds)

4/26/2020	21:38:09	12859	0.06	34.04		21:38:09
4/26/2020	21:38:31	77488	0.39	201.58		21:38:31
4/26/2020	21:38:51	114751	0.57	297.93		21:38:51
4/26/2020	21:39:12	231541	1.16	599.24		21:39:12
4/26/2020	21:39:33	172830	0.86	447.85		21:39:33
4/26/2020	21:39:53	140991	0.7	365.69		21:39:53
4/26/2020	21:40:14	132351	0.66	343.39		21:40:14
4/26/2020	21:40:34	131101	0.66	340.16		21:40:34
4/26/2020	21:40:56	57998	0.29	151.12		21:40:56
4/26/2020	21:41:16	29371	0.15	76.91		21:41:16
4/26/2020	21:41:38	50516	0.25	131.74		21:41:38
4/26/2020	21:41:58	74159	0.37	192.97		21:41:58
4/26/2020	21:42:18	214234	1.07	554.62		21:42:18
4/26/2020	21:42:39	45581	0.23	118.95		21:42:39
4/26/2020	21:43:00	135743	0.68	352.15		21:43:00
4/26/2020	21:43:20	67678	0.34	176.19		21:43:20
4/26/2020	21:43:41	66308	0.33	172.64		21:43:41
4/26/2020	21:44:03	37376	0.19	97.67		21:44:03
4/26/2020	21:44:24	50487	0.25	131.66		21:44:24
4/26/2020	21:44:44	116889	0.58	303.45		21:44:44
4/26/2020	21:45:04	149351	0.75	387.27		21:45:04
4/26/2020	21:45:26	188944	0.94	489.41		21:45:26
4/26/2020	21:45:47	286450	1.43	740.83		21:45:47
4/26/2020	21:46:08	96021	0.48	249.52		21:46:08
4/26/2020	21:46:28	180484	0.9	467.59		21:46:28
4/26/2020	21:46:49	0	0	0.62		21:46:49
4/26/2020	21:47:09	159137	0.8	412.52		21:47:09
4/26/2020	21:47:31	28394	0.14	74.37		21:47:31
4/26/2020	21:47:51	54254	0.27	141.42		21:47:51
4/26/2020	21:48:13	0	0	0.62		21:48:13
4/26/2020	21:48:33	175400	0.88	454.48		21:48:33
4/26/2020	21:48:54	152091	0.76	394.34		21:48:54
4/26/2020	21:49:15	8696	0.04	23.22		21:49:15
4/26/2020	21:49:35	60228	0.3	156.9		21:49:35
4/26/2020	21:49:56	160381	0.8	415.73		21:49:56
4/26/2020	21:50:16	78213	0.39	203.46		21:50:16
4/26/2020	21:50:38	166815	0.83	432.33		21:50:38
4/26/2020	21:50:59	224136	1.12	580.15		21:50:59

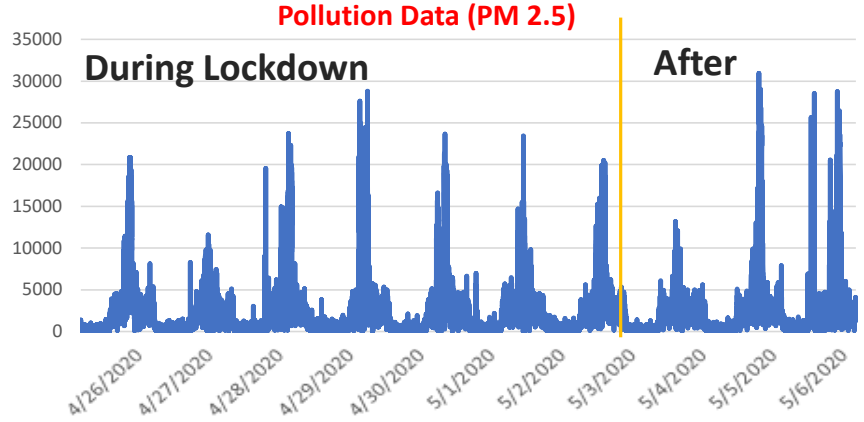
Data coming from Light and Sound Sensors (every 3 seconds)

4/26/2020	21:38:40	100	7.79	73.41	17	591.7647
4/26/2020	21:39:02	182	7.78	73.34	18	558.3333
4/26/2020	21:39:23	260	7.82	73.17	17	591.7647
4/26/2020	21:39:44	782	7.81	73.21	18	558.3333
4/26/2020	21:40:05	68	7.79	73.09	18	558.3333
4/26/2020	21:40:26	164	7.77	73.11	19	528.4211
4/26/2020	21:40:47	202	7.78	73.22	18	558.3333
4/26/2020	21:41:08	389	7.78	73.17	18	558.3333
4/26/2020	21:41:29	68	7.75	72.75	18	558.3333
4/26/2020	21:41:51	205	7.71	72.63	18	558.3333
4/26/2020	21:42:12	177	7.71	72.65	18	558.3333
4/26/2020	21:42:33	326	7.71	72.84	18	558.3333
4/26/2020	21:42:54	177	7.69	72.92	18	558.3333
4/26/2020	21:43:15	246	7.72	73.09	18	558.3333
4/26/2020	21:43:37	126	7.72	73.23	18	558.3333
4/26/2020	21:43:58	201	7.72	73.37	17	591.7647
4/26/2020	21:44:19	228	7.74	73.06	17	591.7647
4/26/2020	21:44:40	185	7.74	73.17	17	591.7647
4/26/2020	21:45:01	185	7.77	73.35	17	591.7647
4/26/2020	21:45:23	204	7.77	73.28	18	558.3333
4/26/2020	21:45:44	91	7.75	73.46	17	591.7647
4/26/2020	21:46:05	277	7.77	73.4	18	558.3333
4/26/2020	21:46:26	188	7.75	72.98	17	591.7647
4/26/2020	21:46:47	70	7.72	73.15	18	558.3333
4/26/2020	21:47:08	73	7.71	72.73	17	591.7647
4/26/2020	21:47:30	240	7.72	72.61	18	558.3333
4/26/2020	21:47:51	70	7.72	72.72	17	591.7647
4/26/2020	21:48:12	113	7.68	72.88	17	591.7647
4/26/2020	21:48:33	514	7.69	72.38	18	558.3333
4/26/2020	21:48:54	70	7.66	72.17	17	591.7647
4/26/2020	21:49:16	70	7.66	72.5	17	591.7647
4/26/2020	21:49:37	137	7.66	72.66	17	591.7647
4/26/2020	21:49:58	134	7.66	72.79	18	558.3333
4/26/2020	21:50:19	158	7.66	72.9	17	591.7647
4/26/2020	21:50:40	71	7.68	72.52	17	591.7647
4/26/2020	21:51:02	212	7.69	72.45	18	558.3333
4/26/2020	21:51:23	75	7.71	72.62	18	558.3333
4/26/2020	21:51:44	178	7.69	72.67	17	591.7647

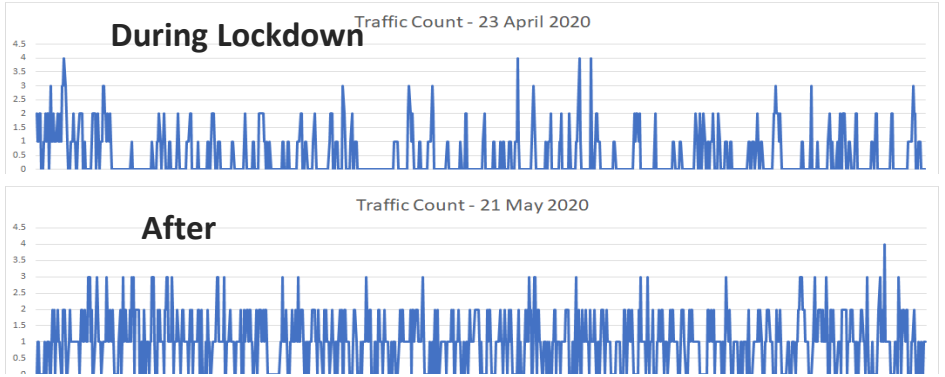
Results 1: Impact of COVID19 Lockdown in Toronto

During Lockdown (26 April – 3 May) **After Lockdown (3 May onwards)**

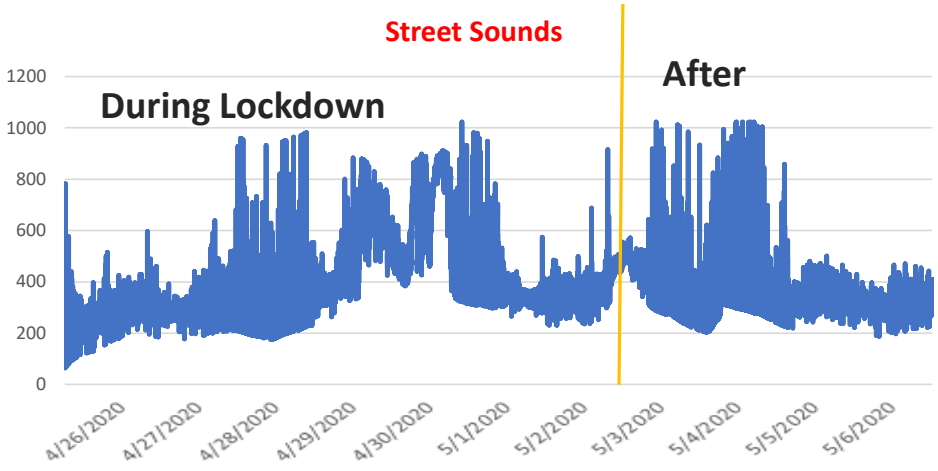
Pollution Data (PM 2.5)



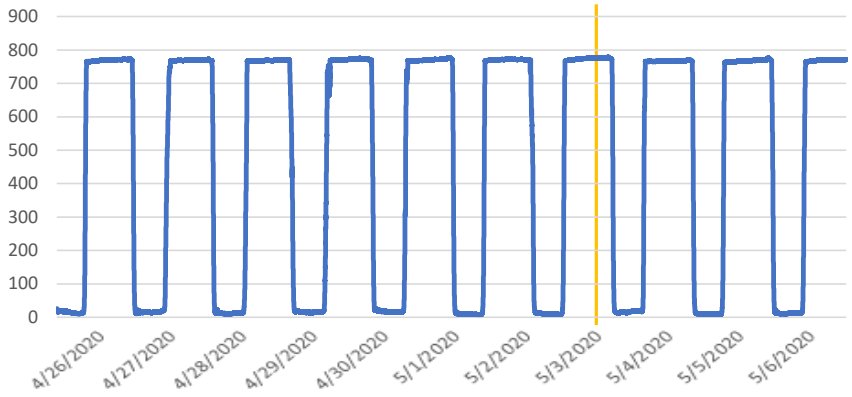
Live Traffic Count (King Street)



Street Sounds



Street Light



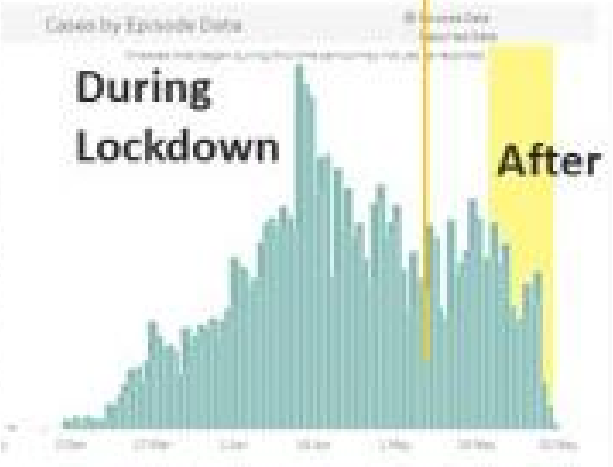
Results 2: Impact of COVID19 Lockdown in Toronto

During Lockdown (26 April – 3 May) **After Lockdown (3 May onwards)**

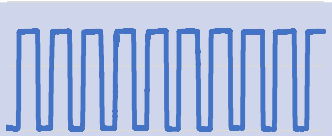
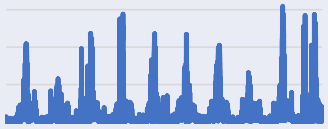
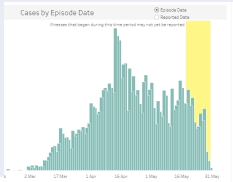
Night Lights in Toronto (NASA Soumi/VIIRS)

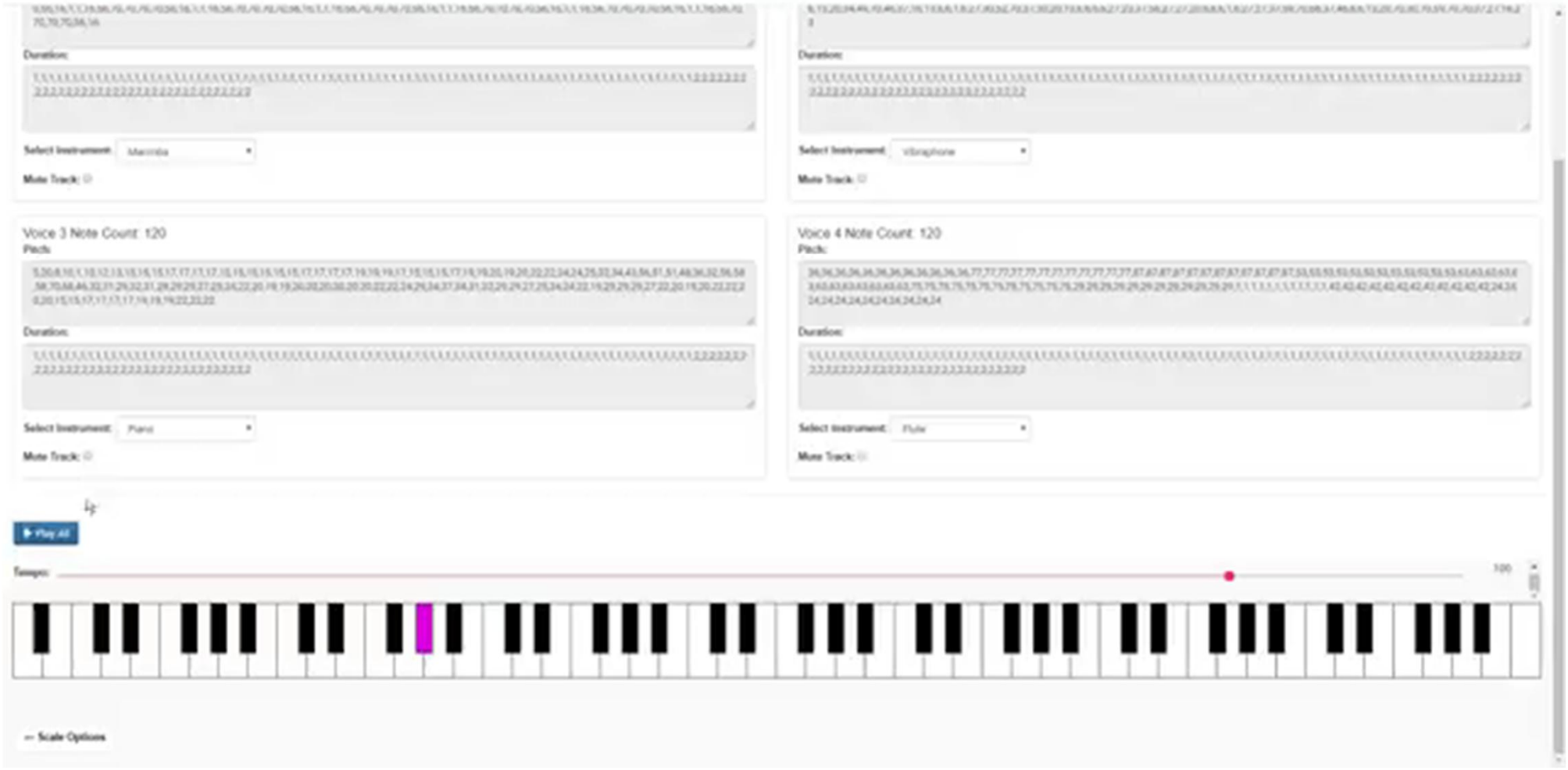
During Lockdown (on 3 April) After (on 30 May)

COVID19 Cases (Toronto)



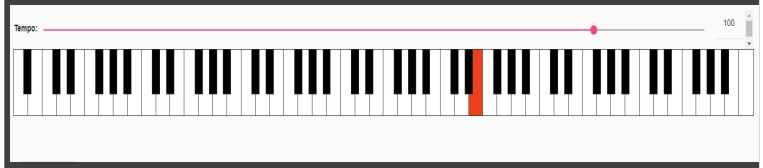
Musical Instruments and Notes Used:
To Sonify Data and create the Lockdown Musical

Instruments	Data Sonified	Graphs	Notes
 **Marimba**	Passage of Days and Nights		G Major
 **Vibraphone**	Vehicle Emissions		G Major
 **Piano**	Street Noise		E Major
 **Flute**	COVID19 Infection Rates		A Major



http://musicalgorithms.org/4.1/app/

The Masked Scales



2020 NASA SpaceApps Covid19 Challenge Global Winner

Artash Nath (Grade 8)
Arushi Nath (Grade 5)
Toronto, Canada

Acoustics and COVID19 Pandemic
4th International Conference. 5 September 2020

ICA ACOUSTICAL SOCIETY OF NIGERIA
(A MEMBER OF INTERNATIONAL COMMISSION FOR ACOUSTICS)