RUTGERS

April 5, 2022 Update

CoralDX

- Strengthened the image detection algorithm
- Able to save files in a organized manner, with each row representing a specific coral and each column representing a specific data type of the coral (such as r value, g value, etc.)
- Able to open the data in a clear excel file using denormalized function

CoralDX Continued

Here is an example of my code vs. ImageJ

My Code: (165, 145, 117)
ImageJ: (163, 142, 115)

Comparing these two colors:

CoralDX Pictures

```
3April2022 Sample1 234 037 243 123 213 241
3April2022 Sample2 251 244 145 167 237 095
3April2022 Sample3 213 222 256 189 177 111
```

Top picture is the file generated form the android app. Each row represents a different coral.

1								
2	3-Apr-22	Sample1	234	37	243	123	213	241
3	3-Apr-22	Sample2	251	244	145	167	237	95
4	3-Apr-22	Sample3	213	222	256	189	177	111
5								
6								

This bottom picture is the same data in an excel file. I took the .txt file and put it into excel.

CoralDX Coding Part

```
public String dataStringGenerator(String sampleName, int coralRedValue, in

    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:
    LocalDateTime now = LocalDateTime.now();
    System.out.println(dtf.format(now));

    return dtf.format(now) + " " + sampleName + " " + blockRedValue + " "
}
```

```
public void writeToTxt(ArrayList<String> stringList)

    File file = new File("coralDatatest.txt");
    FileWriter fileWriter = new FileWriter(file);
    PrintWriter printWriter = new PrintWriter(fileWri

    for (int i = 0; i < stringList.size(); i++) {
        printWriter.println(stringList.get(i));
    }

    printWriter.close();
}
```

Future Plans with CoralDX

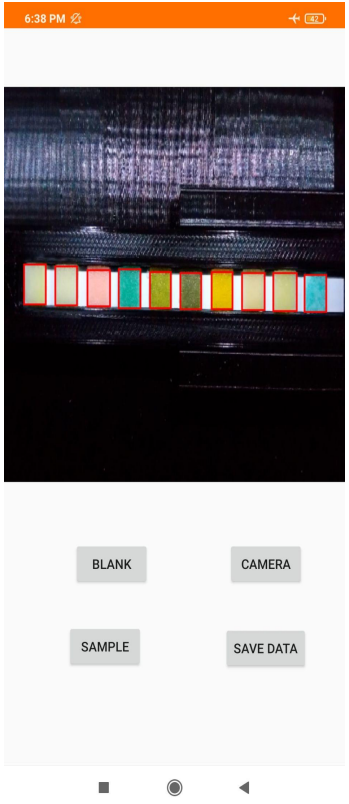
- Need to crop the colors from the background boxes to get their colors.
  - Once I have this, I can find the norm values
    - The norm values will be upto 9/10 decimal places for accuracy

- Need to add a input button which can ask the user for the name of the sample so once the name is inputted, the data can be stored with the name

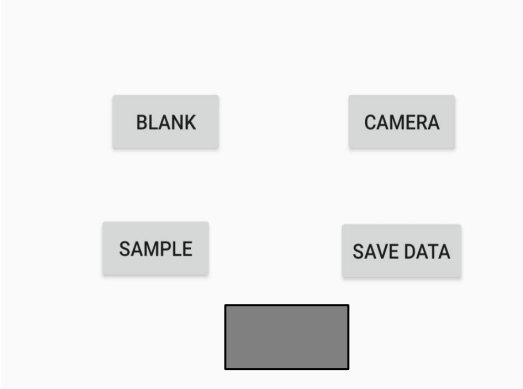
TestStripDX

- Created the buttons and half of the functionality for the buttons on for the app
- Need to be able to crop the images from yolov4 and find the rgb values from them
- Need to build the internal timer for the sample button
- Data saving is the same as what it was for CoralDX, but need to generate the data in a different way here

TestStripDX Pictures



- This is what the UI looks like as of right now.
- Yolov4 is able to identity each part distinctly. We have to crop each part and analyze each components RGB value.
- I would like to put a timer on the screen for the sample button functionality.



Wanted to put the timer down here for the sample button

Future plans with TestStripDX

- The main thing to do is cropping the images from yolov4 and then analyzing them
- Need to code the functionality of the sample button and the internal timer that comes along with it