2024

LOG BOOK

SCIENCE FAIR

BY: JOEL AMALDAS

Date: Aug-16-2023

Four years ago, one of our family friends was diagnosed with cancer. He had a brain tumour near his optic nerve and didn't know for a long time. When he finally found out, it was too late to take out the tumour, and he unfortunately passed away.

Similarly, many cancer patients are very unaware of a growing tumour, and skin cancer is the most common type of cancer.

The survival rate is about 25% for stage 4 advanced melanoma skin cancer, but early detection during stage 1 brings it up to 97%.

Recently, I have been learning about artificial intelligence and its impact on society, and how it can aid in healthcare.

If I could use AI as a tool to help and screen, is it possible to find skin cancer tumours earlier using deep CNNs?

Date: September- 18-2023

- Today, I attended my First science fair meeting at school. Ms. Bretner told us about the science fair and CYSF.

- I figured out that my project idea was Innovation. So, I will email Ms. Bretner all the details.

From my research, I can conclude that

- I need to use object detection for my model.
- It can run on a computer.
- To code the program, I can use C ++ or Python.
- First, I need to figure out which coding language I will use.

Date: Sep-30-2023

- I have figured out that the coding language I want to use is Python since also I have more experience in Python.

- Instead of running the code on a computer, I will run it on a Arduino nano microcontroller so that it could be more portable.

When researching, I found that:

- I could “train“ it to recognize the melanoma types.
- It uses “Libraries” which are a part of Python.
- The libraries will be using are YOLOv5, PyTorch
- I am getting very close to starting the project.

Date: Nov-15-2023

- Today, I tried starting my project but soon I found out that it wouldn't work.

- First, I tried doing a basic test in Python on my Arduino. I ran a basic object detection test that I found online, and it did not work.

- The camera model was only for taking photos using the Arduino, and the Arduino was too weak to handle object detection.

- From further research, the Arduino can only be used for basic things, and almost all of my research sources recommend to use of Jetson Nano.

Date: Dec-3-2023

After finding out my Arduino did not work, I have begun researching the Jetson Nano, which is a whole computer with an operating system, but much smaller.

It can run a lightweight version of YOLO-V5.

Before starting, I need to collect the materials.

1. Jetson Nano
2. Camera Module
3. Jumper wires
4. Red Led Light

Jetson Nano	$242.38
Camera Module	$16.49
Jumper Wires	$12.69
Red Led Light	$1.99

Today, my dad ordered Jetson Nano for $242 from the robotics shop online.

Date: Dec- 22-2023

I started my project today.

After research, there are three main steps:

1. Create Dataset
2. Train Model
3. Code software
4. Test
5. If not working, repeat steps 1,2 and 4. (trial & Error)

First, I have to create the dataset, so I can train the screening model.

Date: Jan-13-2024

I may use what I write here in my presentation, along with a few edits.

A dataset is a set of images and text files. These text files contain data on where the object is in each photo of the dataset. This format of text files is specific to the YOLO neural network, as other neural networks such as Pascal VOC CNNs use XML files (Extensive Markup Language).

First, I researched a dataset to try and find one that would be appropriate to screen melanoma. Using a dataset from Kaggle with 10,000 images, they had two object detection classes: malignant and benign. Using this information, we can annotate the dataset.

I annotated each one using a program called labelImg, which labels using bounding boxes. Since there are a total of 10,000 images and 5000 images for each class, this will make my object detection model very accurate. I labelled all 10,000 images, putting a bounding box on each image to show the program where the lesion is, and if it malignant or benign.

Date: Jan-20-2024

Today, I train my model using the YOLO v5 neural network. To train it locally on my Jetson Nano, I first need to select a pre-trained model and re-train it with my own dataset, to get faster results. I chose the second smallest and fastest model available, YOLO v5s.

Before I start training, I need to install the prerequisites. To do that, I need to clone the YOLOv5 repository to get the pre-trained models. After that, I need to install comet_ml which gives training results through pip (preferred installer program). I then import pytorch which is the backbone of the YOLO v5 neural network. While training the model, I used the YOLO v5 neural network as the base of my project. It serves as the core of AI object detection for my program and can do it at high speeds.

YOLO, or You Only Look Once works faster and more efficient than traditional convolutional neural networks. This is because it utilizes single-shot detection, processing it by passing it only a single time.

I use the YOLO v5s pre-trained model as the “base” of my project, building many more layers on top using my annotated dataset.

Date: Jan-27-2024

Finally, I code the program to utilize the previously trained model.

- Import OpenCV: Start by importing OpenCV to handle image processing and capture live video streams from the camera.
- Initialize Classes: Define the object classes 'Benign' and 'Malignant' within the program to classify the different types of melanoma tumors detected by the model.
- Set Up GStreamer: Configure the GStreamer pipeline to ensure the video stream is properly formatted and displayed with the correct quality on the screen.

- Load YOLOv5 Model: Implement code to load the trained YOLOv5 model ('best.pt') so it can process the incoming video stream in real-time.

- Bounding Box Detection: Use OpenCV functions to process the video frames, applying the YOLOv5 model to detect lesions and draw bounding boxes around them.

- Display Results: Code the final step to display the annotated video stream live, showing the detected lesions with labels indicating whether they are benign or malignant.

Date: Feb-3-2024

During the process of training, there was trial and error. During unsuccessful training attempts, the YOLOv5 neural network would try again, adding more images to train until it got the best confidence score, along with the best precision. When the best confidence score and the best precision are achieved, it is checkpointed and saved as best.pt, which can be used for deploying onto real-time object detection.

Date: Feb-17-2024

My real-time melanoma screening and classification software works well and is accurate; however, there are false positive misclassifications such as classifying benign tumours as malignant or classifying certain shapes as malignant tumours. This is caused by issues such as lighting errors, the angle at which the sheet of paper is held, and the focus of the camera (i.e., blurry focus). However, I spoke to a healthcare professional, Dr. Rodel Cenabre who works in the Tom Baker Cancer Center as a family physician about my project. He spoke to me about how false positives are fine, and that false negatives are very serious. “If it is a false negative, things might go unnoticed, which can create more problems. If it is a false positive, it’s OK since we can still check it and confirm, ‘No, it is not melanoma.’ The program should be used as a screening tool, since due to potential false negatives, it cannot diagnose, and Healthcare professionals should be the ones actually diagnosing.”

Date: Feb-18-2024

Future improvements that I could implement to my melanoma screening tool are:

Mobile App Implementation. Using the trained YOLOv5 neural network, we can implement it into a mobile app for easier, and more accessible screening and identification.

Screenshot Feature. When the model detects a tumour, it will take a screenshot and send an email to the doctor's office, specifying the type of tumour detected.

Smart Bandage. With our deep AI, we can implement this into a monitoring device, providing the necessary treatment when required.

I am going to use the writing from my logbook in my presentation, and edit it.

Date: Feb-23-2024

Today, I participated in my school science fair and I did a 3-5 minutes oral presentation to the judges. I won the Innovation Award for my project. I am going to participate in CYSF.

Date: Feb-27-2024

I emailed a dermatologist in Calgary about my project and any feedback.

Date: Feb-28-2024

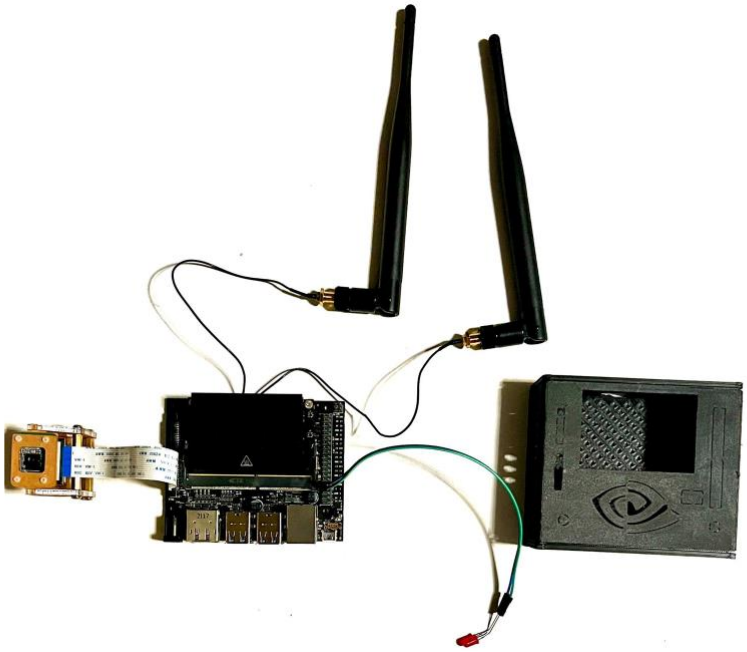
The dermatologist, Dr. Lewinson, emailed me back and gave me clinical insights about the ethics of using AI to detect skin cancer, as well as recommendations to improve my PowerPoint slides.

(I will include the Doctor's email in my logbook after receiving his consent).

Date: March-8-2024

I finished implementing his recommendations, and I will record my video. I am almost done with my CYSF platform.

Picture of my device

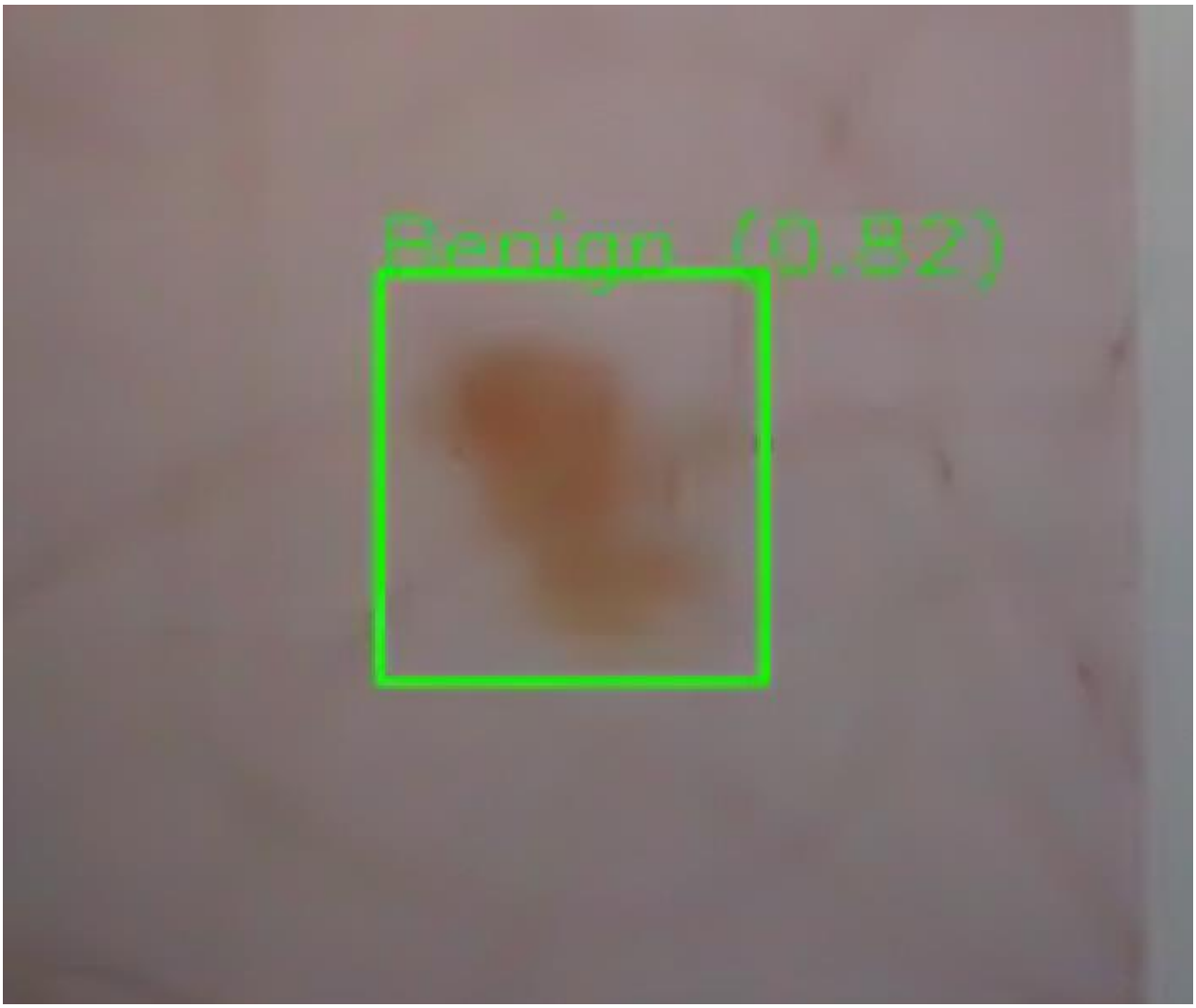


Coding

```
import cv2
import numpy as np
from elements.yolo import OBJ_DETECTION

Object_classes = ['Benign', 'Malignant']

Object_colors = list(np.random.rand(80,3)*255)
Object_detector = OBJ_DETECTION('weights/yolov5s.pt', Object_classes)


cap = cv2.VideoCapture("nvarguscamerasrc ! video/x-raw(memory:NVMM),format=NV12,width=640,height=480,framerate=30/1 ! nvvidconv !
video/x-raw,format=BGRx ! videoconvert ! video/x-raw,format=BGR ! appsink drop=1", cv2.CAP_GSTREAMER)

if cap.isOpened():
    window_handle = cv2.namedWindow("Melanoma Detection", cv2.WINDOW_AUTOSIZE)
    # Window
    while True:
        ret, frame = cap.read()
        if ret:
            # detection process
            objs = Object_detector.detect(frame)

            # plotting
            for obj in objs:
                #print(obj)
                label = obj['label']
                score = obj['score']
                [(xmin,ymin),(xmax,ymax)] = obj['bbox']
                color = Object_colors[Object_classes.index(label)]
                frame = cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), color, 2)
                frame = cv2.putText(frame, f'{label} ({str(score)})', (xmin,ymin), cv2.FONT_HERSHEY_SIMPLEX , 0.75, color, 1,
cv2.LINE_AA)

        cv2.imshow("Melanoma Detection", frame)
        keyCode = cv2.waitKey(30)
        if keyCode == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

```
!git clone https://github.com/ultralytics/yolov5  # clone
%cd yolov5
%pip install -qr requirements.txt comet_ml  # install

import torch
import utils
display = utils.notebook_init()  # checks
```

Data set


Benign (0.82)