Alyana Nelson & Ramprakash Sridharan

Tejaswi Linge Gowda

Grocery Sync

05 December 2023

Object classification and IoT Grocery Sync

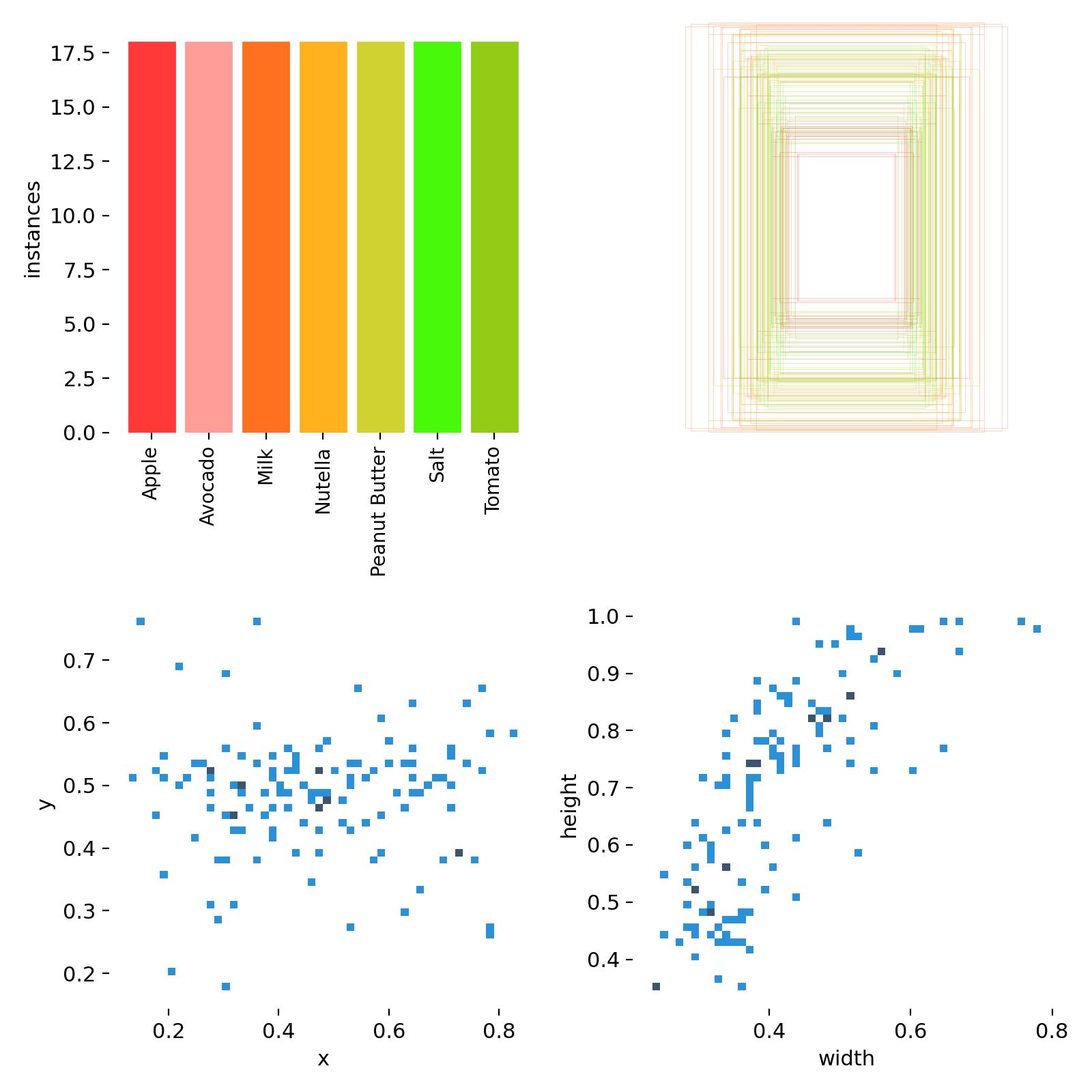
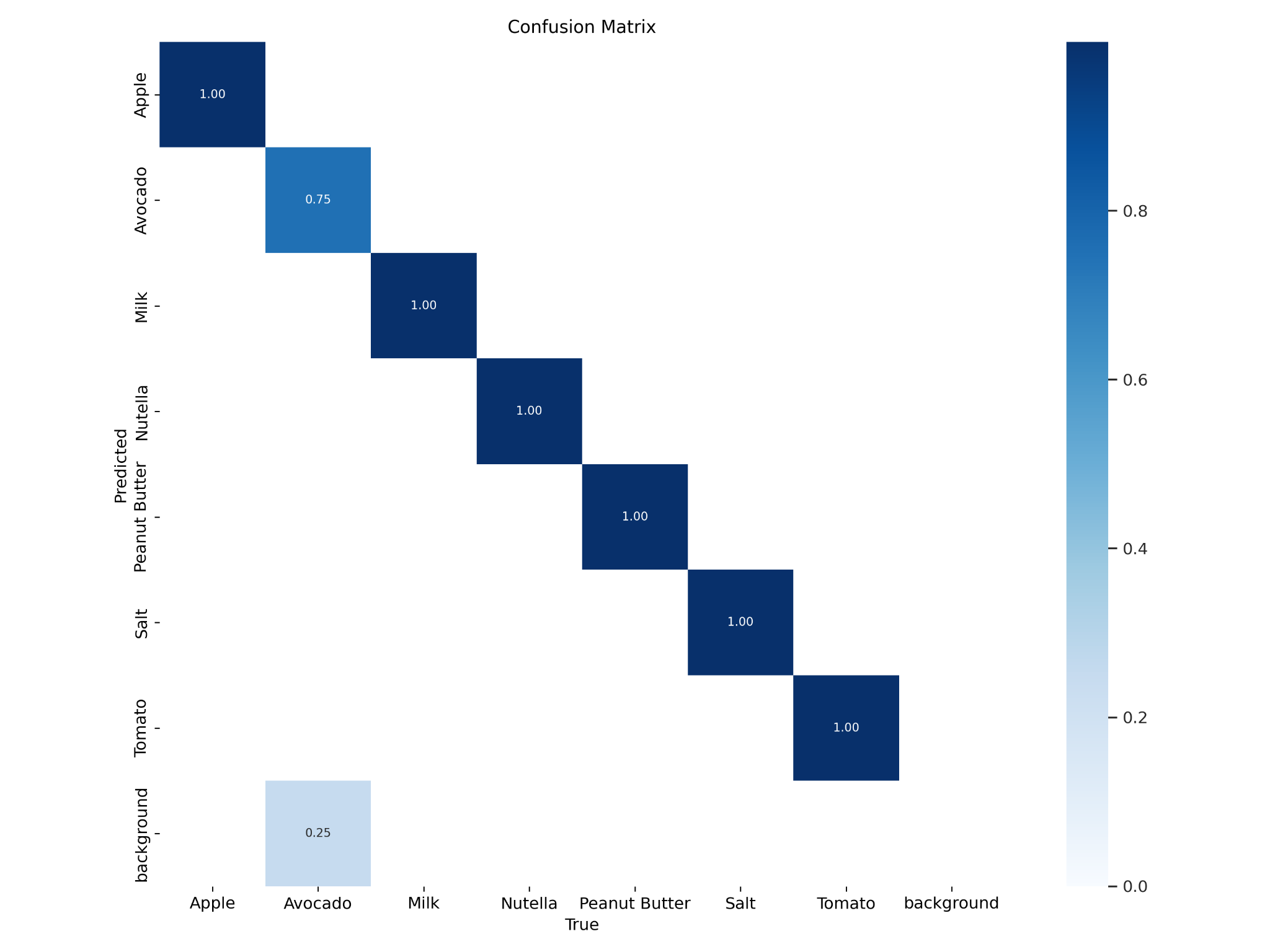
A kitchen-based system that allows you to effortlessly add items to your grocery list is an innovative concept. Simply by showing the food product to the camera, it will automatically update your 'grocery cart', facilitating a streamlined and efficient grocery management process. This system can recognize a wide variety of items, such as apples, water, yogurt, cereal, oil, and much more. It leverages advanced image recognition and machine learning algorithms to identify products and their quantities, ensuring your grocery list is always up-to-date. Additionally, this system can track your consumption patterns and suggest shopping lists based on your usage, making it a smart assistant for your kitchen needs. With its intuitive interface and compatibility with various smart devices, it simplifies the task of grocery shopping, making it a hassle-free experience. This technology represents a significant leap in home management, offering convenience and efficiency in everyday tasks.

The model is developed using pytorch (python language) to predict grocery items from the image and send it to a node server. The image is taken from the webcam connected to the Raspberry Pi 4 . The raspberry pi runs the yolov5 models , predicts the output and sends it to the node server hosted on AWS.

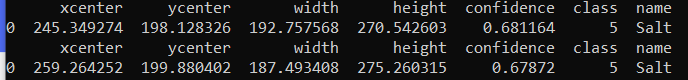
The node server sends the value to Mongodb (Database) using Mongoose library. The database is a free version of the mongodb atlas that has a limit of around 512 MB. The data is then filtered and extracted from the database to add the items on a shopping cart from a custom webpage.

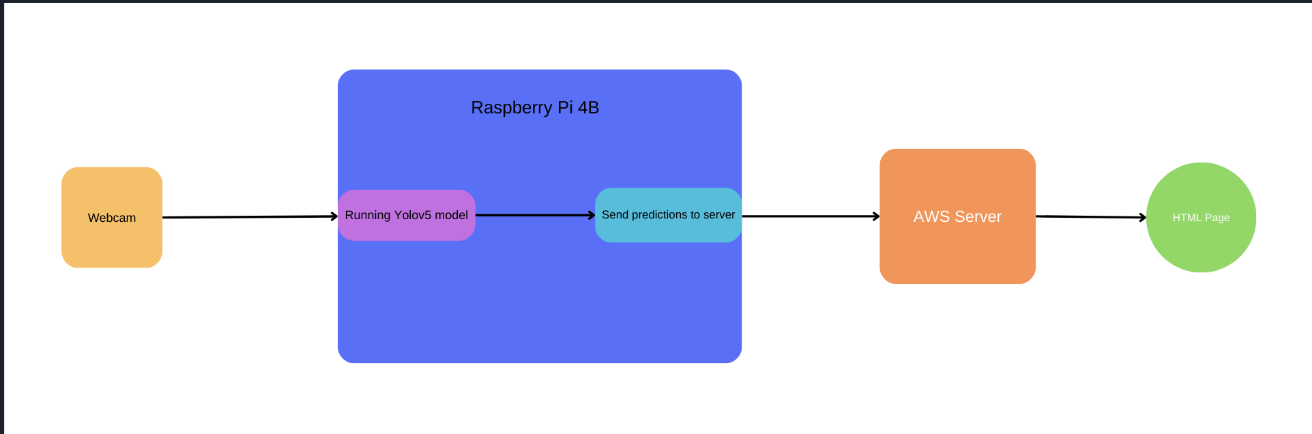
# SUMMARY

* A dataset is created using custom images that represent various grocery items. The grocery items used to train the model are Apple, Avocado, Milk, Nutella, Peanut Butter, Salt and Tomato.
* The model can be trained for all of the possible grocery items in a store , but it takes a lot of data to train and it is complicated to optimize such a huge model. Hence the above grocery items are chosen and the method can be generalized for all the groceries.
* Yolov5 is used to train the Deep learning model. Yolov5 works on top of pytorch (used python ) framework to train and deploy the model.
* The model is trained for 50 epochs for 7 output classes. The results of model training are shown below.

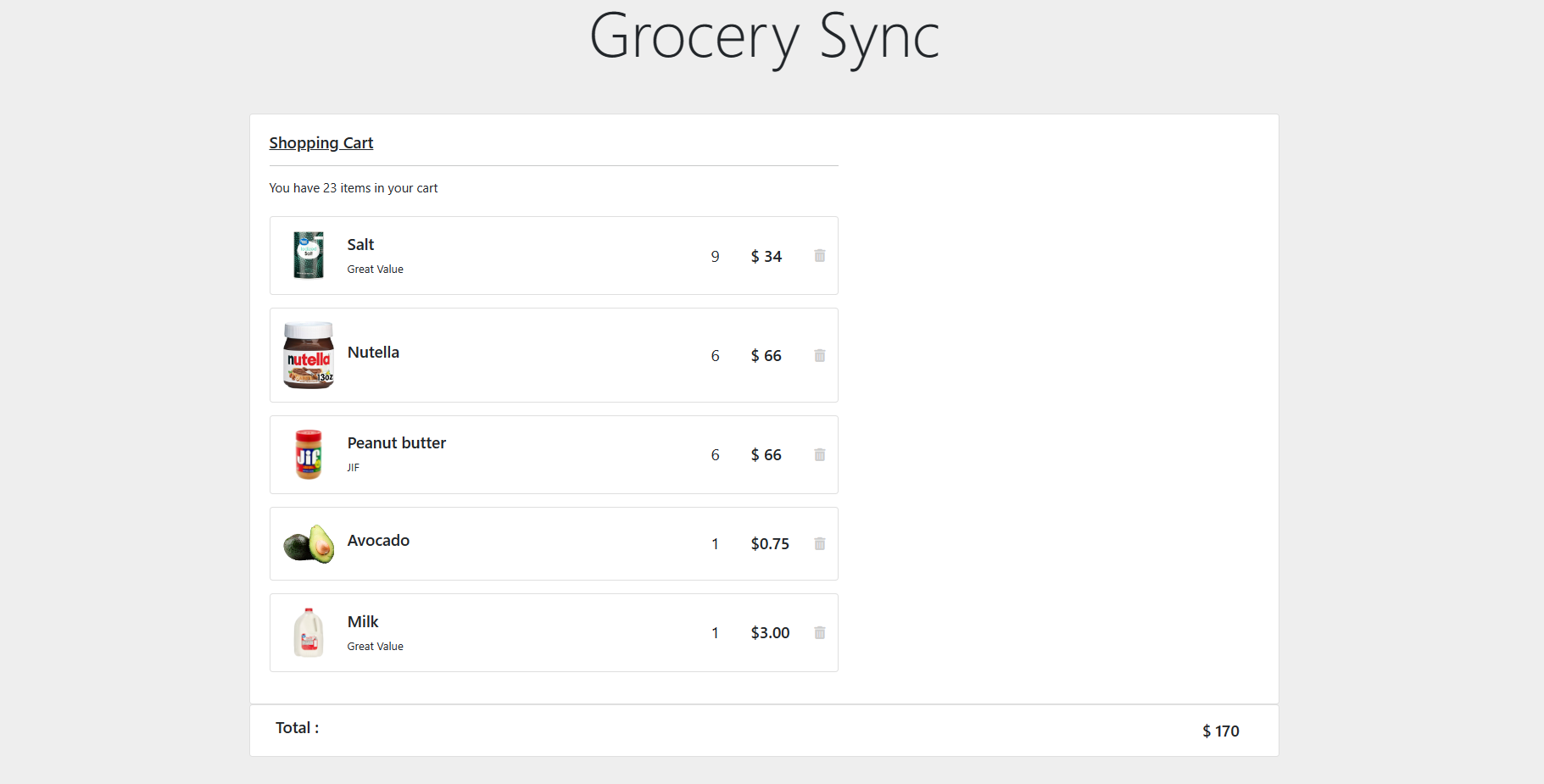


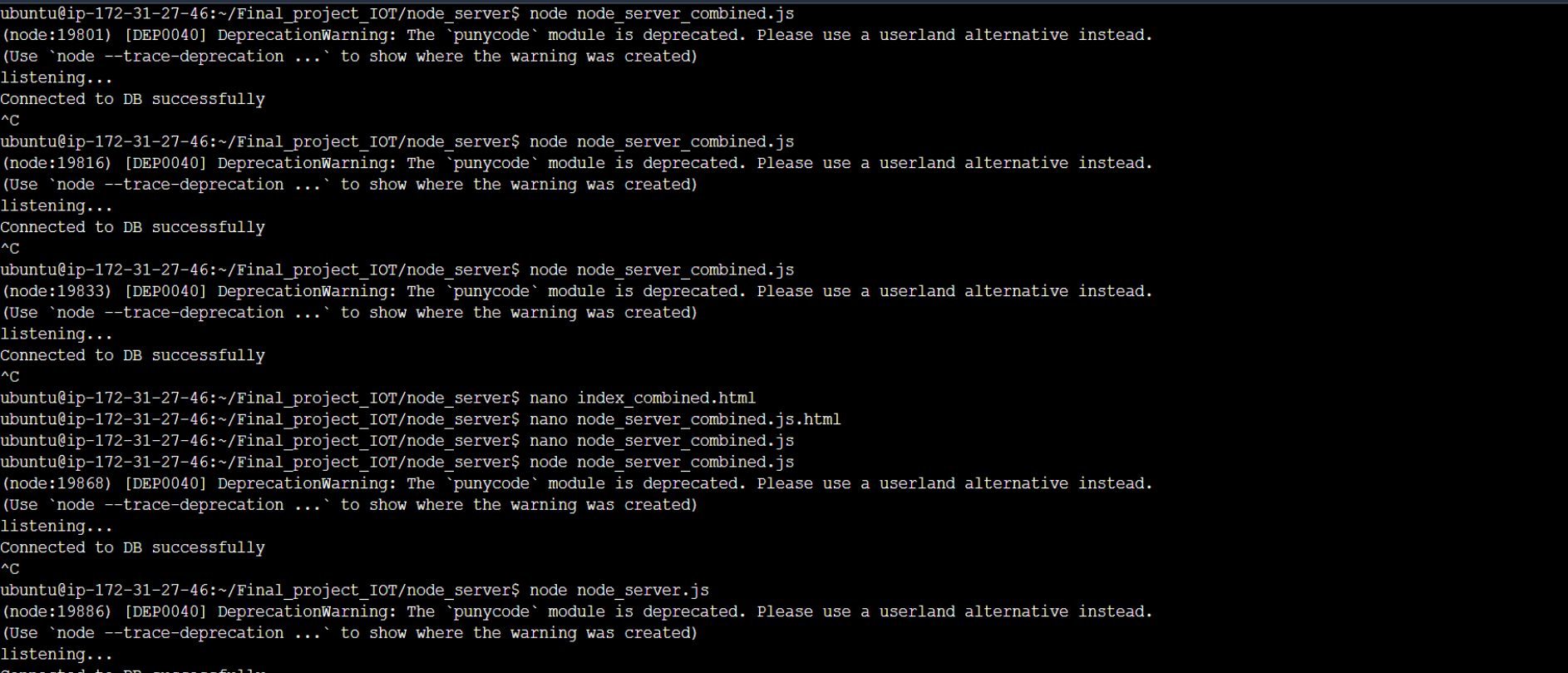


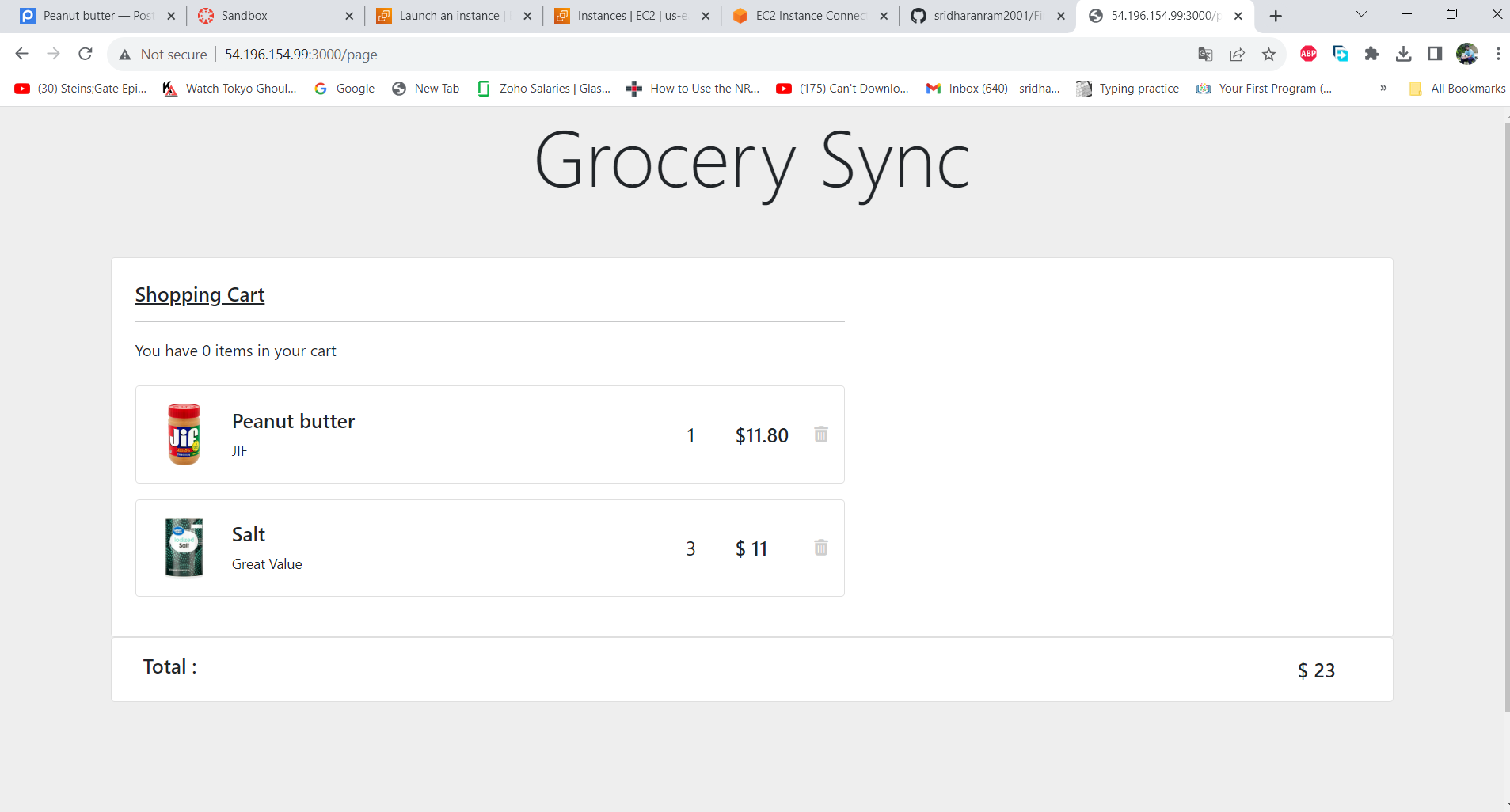
* Opencv is used to collect the data from the webcam and the model is deployed to predict for 7 classes. The resulting prediction is a dataframe consisting of the following data. X and Y center, width and height values of the detected bounding box , output classes name and number and a confidence score which can be used as a threshold for better prediction. 
* The data is converted to json and sent to the node server using a python post request.
* The node AWS server receives the data and sends the data to the Mongodb server (hosted in the cloud). The data is then filtered and the “class” key is used to filter and extract the data.
* The server sends data to the html page. A js script is written to manage the interactions such as “Adding items to cart”, “Update the number of items ”, “Total items in the cart ”, “Price of each item” etc.
* The overall workflow of the system is shown in the block diagram below.

[](https://docs.google.com/presentation/d/1Svx-tTkoo8tO7awWjHJDCXsL8ee3-O2Numrih6zDVe0/edit?usp=sharing)

A sample of the webpage is shown below



The webpage hosted on AWS is shown below 



We opted to maintain the same approach as our presentation, making no further modifications to our project except for removing the API, as we couldn't find one suitable for the Grocery Sync project. Moreover, api's are available only for registered senders to track their own products.

# CONCLUSION

The python code is deployed to send post request to AWS and the webpage is hosted on the AWS server with the database in the cloud. The webpage is rendered as a single html with javascript included in the file as inline javascript. This is necessary because the res.sendfile() can only send a single file and it will not have access to the js files. This also means that the images must be put on a free image hosting website and linked to the webpage. Another approach is to make the file as ejs and render it in the node server.

The workshops and the readings Assisted in constructing the server on AWS (Amazon Web Services) and MongoDB contributed to developing the database for our project. These resources offered valuable insights into best practices for database management and server optimization, enhancing the overall efficiency and functionality of our project.

Has the work in class shifted the way you envision your research goals?

Yes, the coursework, particularly focused on web services, cloud platforms, and databases, has significantly transformed my approach to research goals.

What are your plans for your future research?

The plan for Future research will be to integrate IoT devices to develop more sophisticated solutions, thereby streamlining and enhancing our daily lives.

Works Cited

“Make Sense.” Make Sense, www.makesense.ai/. Accessed 4 Dec. 2023.

“Mongoose.” Mongoose ODM v8.0.2, mongoosejs.com/. Accessed 4 Dec. 2023.

“JavaScript.” JavaScript Tutorial, www.w3schools.com/js/. Accessed 4 Dec. 2023.

Ultralytics. “Ultralytics/Yolov5: Yolov5 🚀 in PyTorch &gt; ONNX &gt; CoreML &gt; TFLite.” GitHub, github.com/ultralytics/yolov5. Accessed 4 Dec. 2023.

“MongoDB Atlas Database: Multi-Cloud Database Service.” MongoDB, www.mongodb.com/atlas/database. Accessed 4 Dec. 2023.