Using Object Segmentation to Analyze Nutrient Content of Meals

Brainstorming

In recent months, there is no question that the popularity of AI has experienced a major surge. From the release of OpenAI's ChatGPT and many other chatbots, it seems that everyone and their mother has launched an AI startup. We decided that due to AI currently rising in popularity, it should have further useful applications in *everyday life*. Although AI is used in many professional fields such as healthcare, economics, and cybersecurity, there is limited use for regular people in their day-to-day lives outside of chatbots.

- Text processing, natural language processing (NLP), and chatbots with a specific purpose, much like ChatGPT.
    - Analyzing texts for emotions, recommending emojis based on text.
    - Summarizing texts, generating an outline of key topics in a large paragraph.
- Solving math, specifically word problems (combination of NLP and number processing)
    - Take a picture of a math problem. Technology such as Photomath exists, but ours can solve word problems + equations
    - Original math problem generator that adjusts difficulty based on users' correct and incorrect responses.
    - Graph plotting, any form of equations
- Image recognition, classifying instances recognized in an image, labeling and detecting the location of objects
    - Garbage sorting (waste, recycling, or compost)
    - Wildlife or plant classification
    - **Analyzing nutrients of a meal through a picture of it**
    - Counterfeit money recognition
- Biometric systems (fingerprint, facial recognition, etc)
    - Mental health support, with facial recognition to understand emotions
    - Biometric-based attendance system, using facial or fingerprint recognition
    - Security at home that only recognizes family members, sends an alarm if someone not recognized shows up
- Other
    - Directing traffic based on satellite imaging includes parking spot recommendations
    - Fake passports or ID detection

The project is to analyze a meal and output the percentage of different nutrients in it. This is very helpful to everyday life, and it is a unique idea that has not been widely used. A brief search told us that about 80% of people do not read the nutrition labels on their food items. An app on their smartphones that analyzes nutrients simply by taking a picture will encourage people to learn about nutrition facts, as smartphones are already a necessity for many people. Our project will encourage healthy eating, and possibly prevent obesity, which can lead to many life-threatening diseases including strokes, high blood pressure, and type 2 diabetes. We will create a prototype of this kind of app.

Schedule

December 2023

- Brainstorm project ideas and choose topics
- Take notes of past computer science projects to avoid repetition
- Observe past project winners:
    - Take notes on their topics, fields, and styles.
- Write project question, purpose, hypothesis, materials, and constant variables.
- Make a schedule from January 1st to March 17th (end of March break), by increments of weeks.

January 1st - 7th, Week 1

- Determine the best model to use for object recognition (*Settled on using YOLOv8, You Only Look Once, latest version*)
- Briefly research how to *use and create* object recognition models, don't research technicalities about how object detection works yet.
- Evaluate each other's programming experience and prior knowledge.

January 8th - 21th, Week 2 & 3

- Configure YOLOv8 to work in VS code
    - Install Anaconda, a package installer for large libraries and modules
    - Download dependencies and Ultralytics package from GitHub
- Check YOLOv8 official documentation and various YouTube videos to understand how to use the YOLOv8 library
- Write code in VS code environment to take an image of a meal as input, output each food item that is present, and the area they take up. Further code to associate nutritional info with each food item detected, using Edamam Nutrient Analysis API. Note: Put all functions in classes
- Use a default pre-trained YOLOv8 model (*this can only detect bananas, apples, and oranges*) to begin troubleshooting code.
    - Fix syntax errors, this troubleshooting has nothing to do with adjusting model parameters.

January 22nd - February 11th, Week 4 - 6

- ~~Find a dataset~~ (*no dataset available that fits our needs*)
- Create a dataset that contains labeled and annotated images of 5 food classes (*bread, bagel, banana, bacon, egg*), which will be used to train the YOLOv8 model.
- Use roboflow.com to annotate 1000 instances of each class.
- Export finished dataset to computer's hard drive.

February 12th - 18th, Week 7

- Research how YOLOv8 object ~~detection~~ (*need a more complex model type for this project*) segmentation models work. Specifically, layers of convolutional neural networks. In detail, aim for about 5 pages of research.

February 19th - 25th, Week 8

- Train model(*s*) on dataset
    - Design procedure for the experiment to optimize the model's performance during training (*We decided to train 4 models for different numbers of epochs, and observe results to determine which one is the best model*).
    - To help with designing the experiment, learn indicators that could tell us the performance of a model (*loss, precision, recall, confusion matrices, etc*)
- After training, upload trained weights to the computer's hard drive
- Modify the previous code so that it uses weights from our trained model, rather than the default pre-trained model.

February 26th - March 10th, Week 9 & 10

- Using the results of the experiment and previous research, write the project report.
- Submit BASEF application
- Fill out consent forms and write project abstract, submit them.
- Write and practice presentation script
- Make poster board.

March 11th - 21th, Week 11 & 12

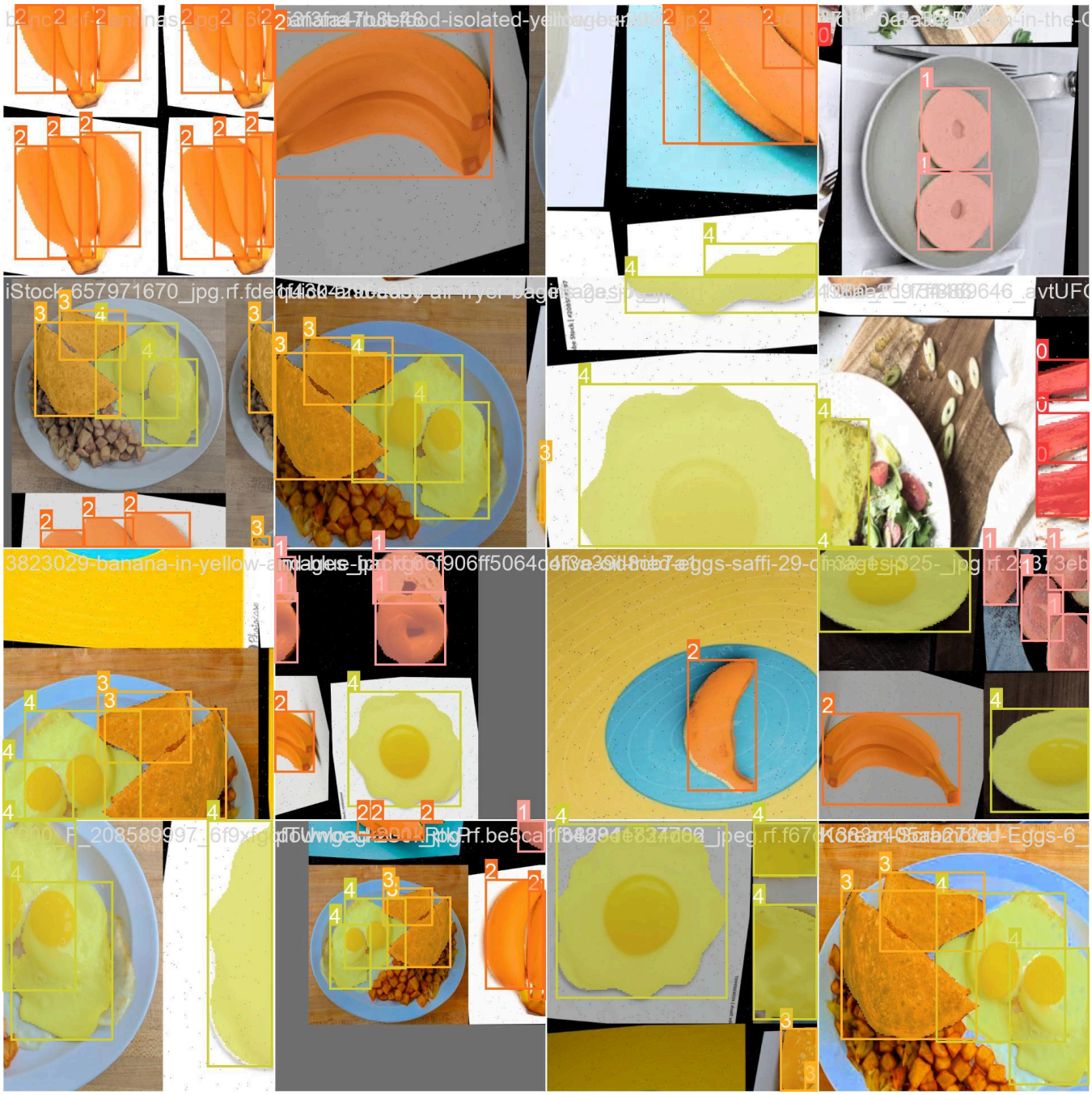
- Revise and edit project reports.
- Practice complete presentation, until confident for competition.
- Mental prepare, relax, and give BASEF our all.

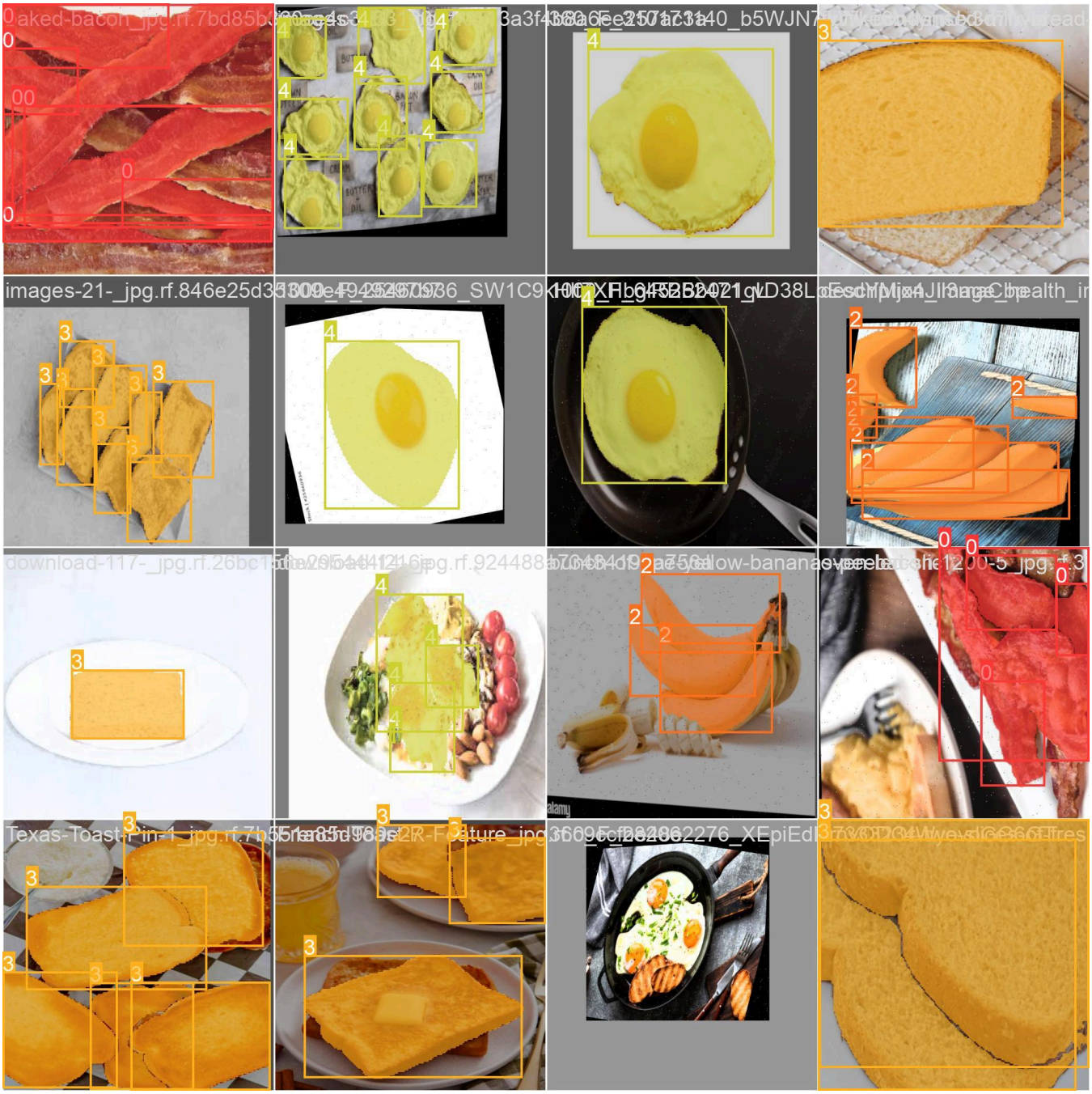
Troubleshooting/Problems Encountered

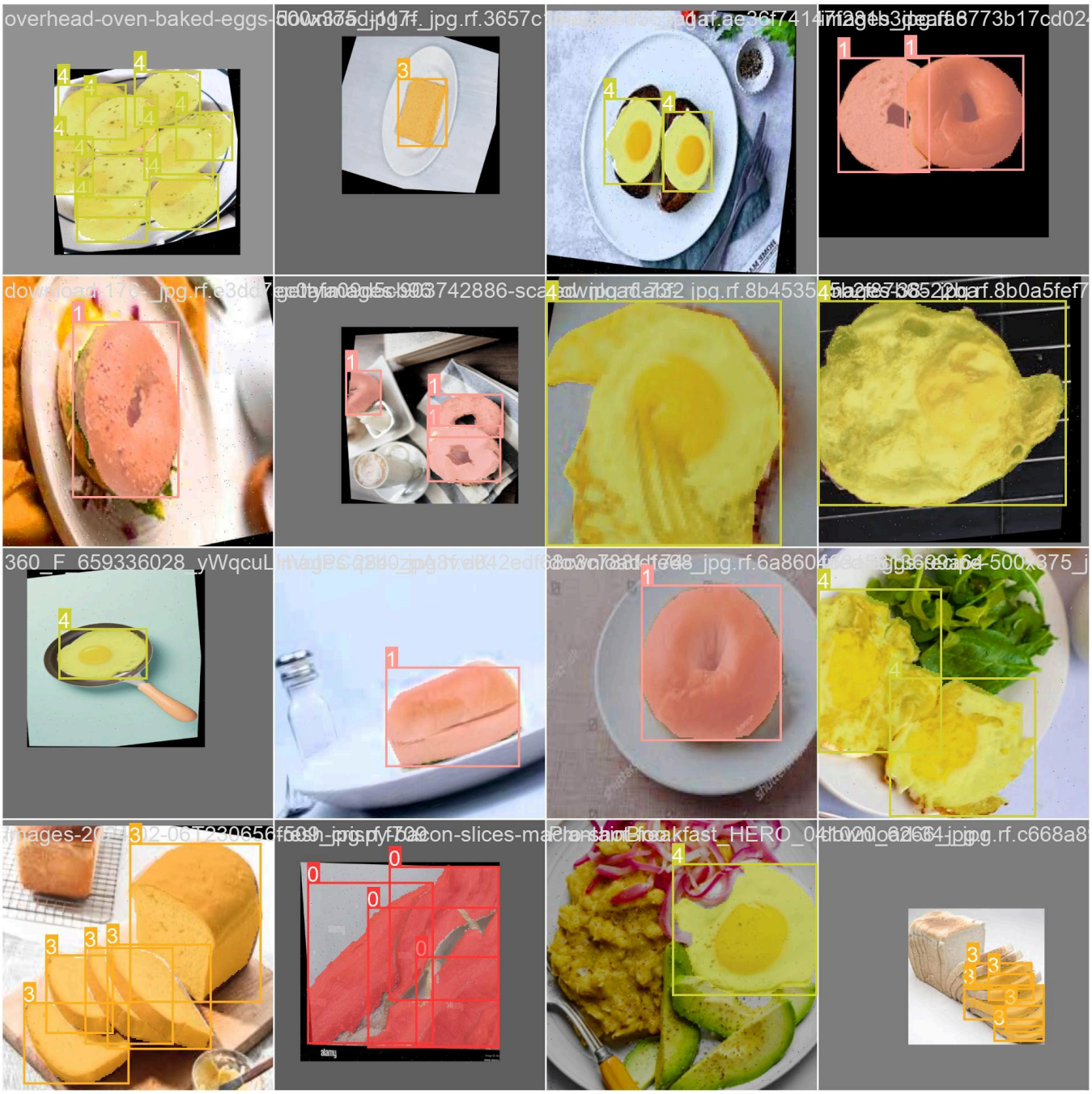
Problem	**Solution**
Inability to download Anaconda, due to incapability with the latest version of Python.	Reinstalled Python as 3.11.3, and added the application to the computer's PATH to function with Anaconda.
Inability to install dependencies of Ultralytics (cannot build "wheel")	Installed dependencies separately instead of having them install automatically, made sure we could install the latest version
Lack of annotated *segmentation* datasets for food that fit our exact needs. However, manually drawing segmentation masks around a total of 5000 instances is too tedious.	Manually labeled and annotated images of 5 food items. Used Roboflow annotator, which is powered by SAM (Segment Anything Model), which means it can draw segmentation masks automatically when you click on the object.
Lack of GPU when training the model, Processing speed too low, training predicted to last days	Used cloud-hosted GPU on Google Colab Notebooks
Google Colab Notebooks had a GPU usage limit, disallowed use after 5 hours per day	Switched to using Kaggle Notebooks environment for double T4 GPU
Training in Kaggle stops midway, usually after 50-70 epochs. Cannot determine why, an issue with the Kaggle environment.	Always make sure the model saves the "latest" weights while training, which means it can always resume training on the latest weights after it is stopped.
Could not figure out how to calculate area (in pixels) efficiently for the food items, as YOLOv8 doesn't output area, only boundary coordinates.	Plotted the coordinates on a Cartesian graph, with the x and y axis representing the width and height of the image in pixels. Used Shoelace Theorem to calculate area on the plot.
The dataset zipped file (Robolfow can only export the dataset as a zipped file) was too large to unpack.	Downloaded WinZip, a tool for unzipping large files
Confusion exists between objects of similar shapes, for example, bacon being confused for the edges of bread. Another example is eggs and bagels being confused with each other, due to their similar round shape.	Red tones in an image are to be highly saturated to enhance that color, before being processed by the model. This process will cause bacon, which consists of many shades of red, to turn significantly red, while bread, which hardly contains any red tones, will not experience major changes.

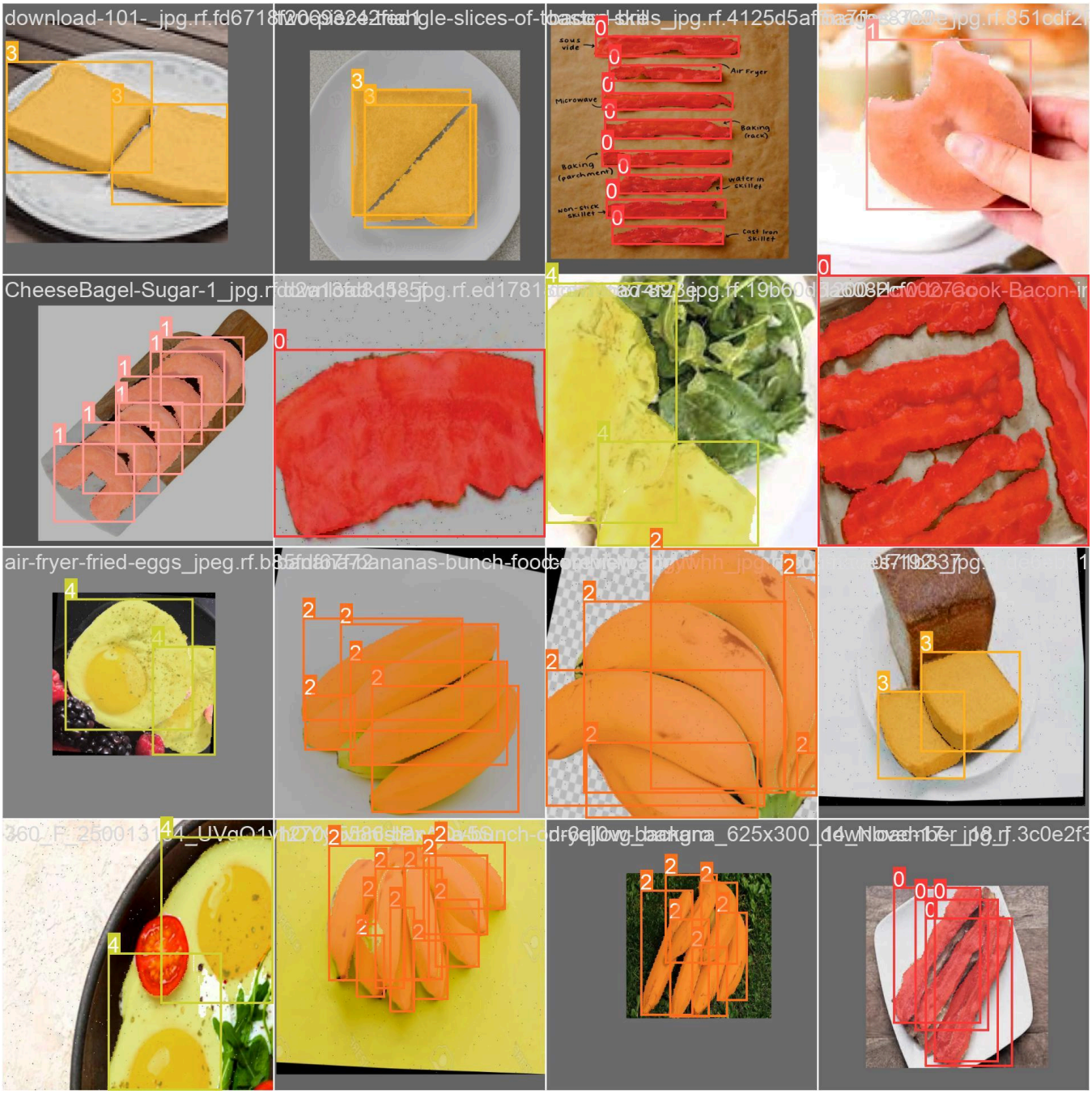
Results of the Best Model

Images of the model analyzing the validation dataset. Performance indicators such as loss

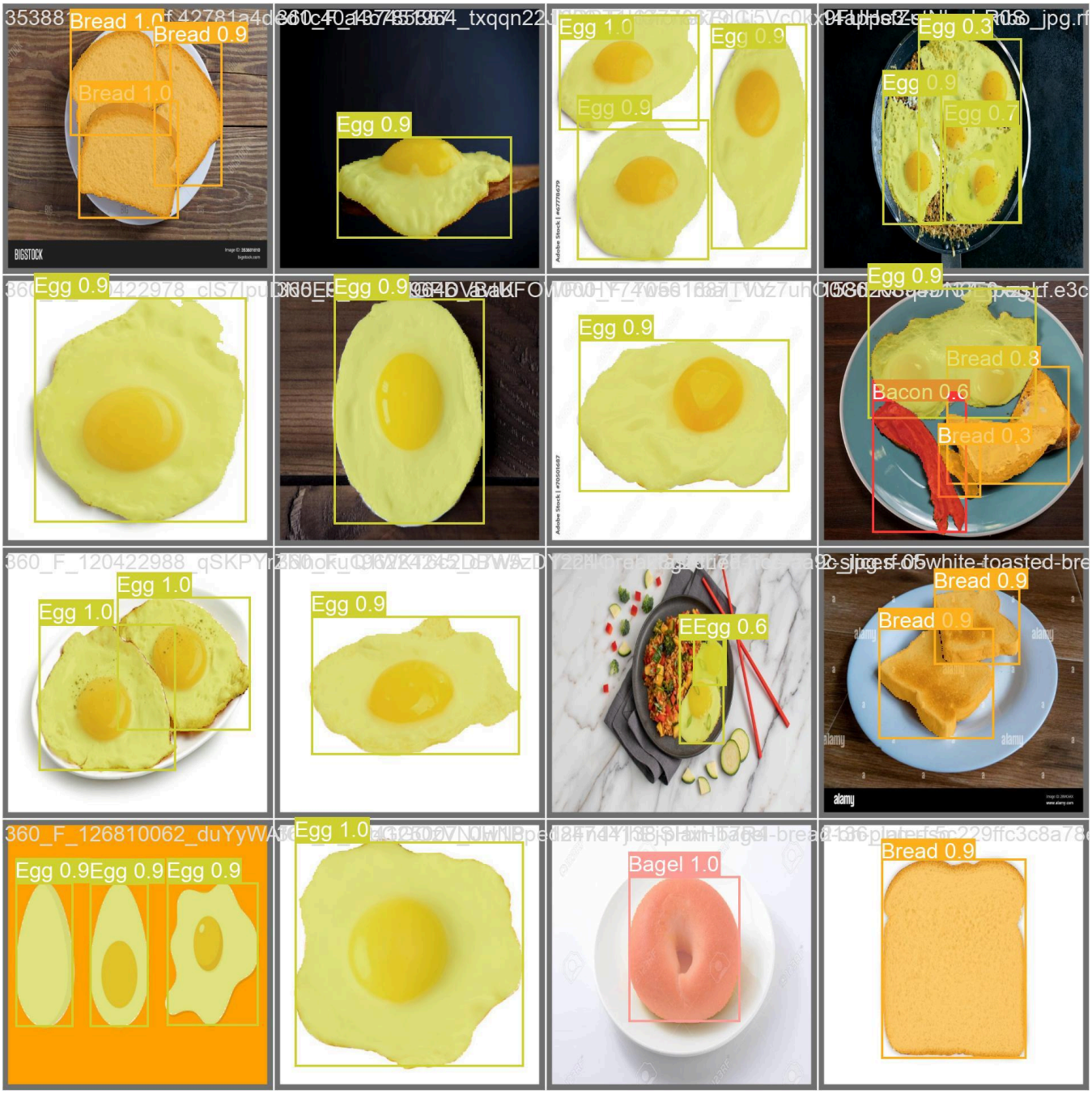







sous vide
Air Fryer
Microwave
Baking (rack)
Baking (parchment)
Wafer in skillet
Non-stick skillet
Cast Iron Skillet


Bread 0.9
Bread 0.9
Bread 0.9
Banana 0.3
Bread 0.8
Bacon 0.9
Bread 1.0
Bread 0.9
Bread 0.9
Bread 0.4
Bread 0.9
Bread 0.6
Bread 0.6
Bread 0.9
Bread 0.8
Bread 0.9
Egg 0.9
Bread 0.9
Bread 0.9
Banana 0.9
Banana 0.9
Banana 0.9
Banana 0.7
Banana 0.9
Banana 0.4
Bread 0.9
Bread 0.9
Bagel 0.9
Bagel 1.0
Bagel 0.9
Bagel 0.9
Bagel 0.9
Bagel 0.9
Banana 0.3
Bread 0.4
Bread 0.4
Bread 1.0
Bread 0.8
Bread 0.9
Bread 0.9
Bread 0.9
Bread 0.3
Bread 0.8
Bread 0.4
Bread 0.6
Bread 0.9
Bread 0.5


Bread 1.0
Bread 0.9
Bread 1.0
Egg 0.9
Egg 1.0
Egg 0.9
Egg 0.9
Egg 0.3
Egg 0.9
Egg 0.7
Egg 0.9
Egg 0.9
Egg 0.9
Egg 0.9
Bread 0.8
Bacon 0.6
Bread 0.3
Egg 1.0
Egg 1.0
Egg 0.9
EEgg 0.6
Bread 0.9
Bread 0.9
Egg 0.9Egg 0.9Egg 0.9
Egg 1.0
Bagel 1.0
Bread 0.9


Bagel
Bagel
BAGEL BLT
nourish nutrition blog
Bacon
Bacon
Banana
Egg
Egg
Egg
Egg
Bagel
Egg
Egg
Bagel
Bagel
Bagel
Bagel
Bagel
Egg
Egg
Egg
Egg
Bread
Bread
Bread
Bagel