

Project 3 Object Detection

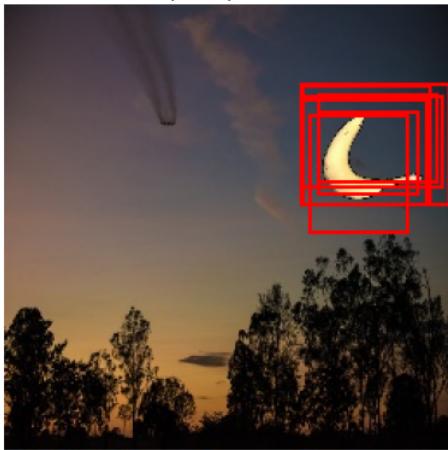
Part 1- Lightweight Object Detection

TinySSD implementation - Training on Banana Detection Dataset

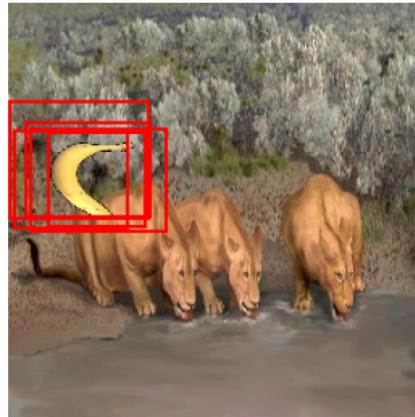
https://d2l.ai/chapter_computer-vision/object-detection-dataset.html

5 Sample Detections (Validation Set)

Val Sample 0 | 7 detections



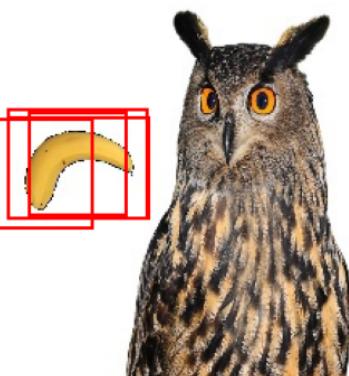
Val Sample 1 | 5 detections



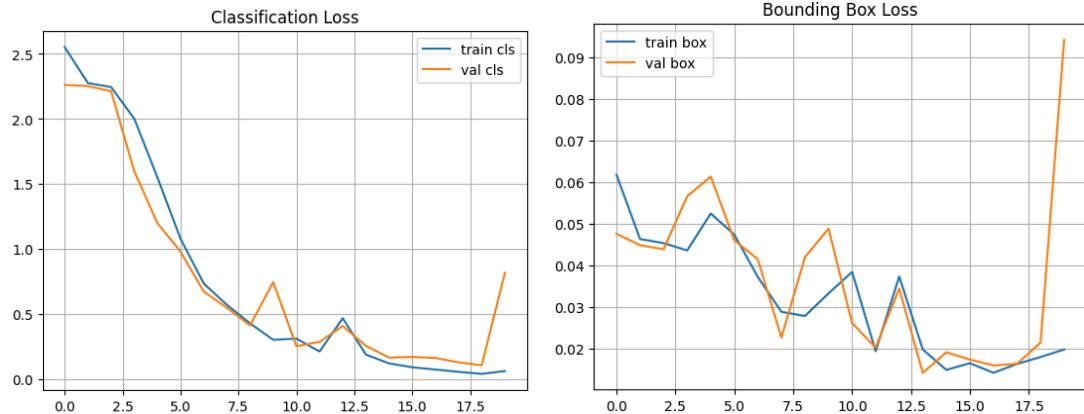
Val Sample 4 | 4 detections



Val Sample 3 | 5 detections

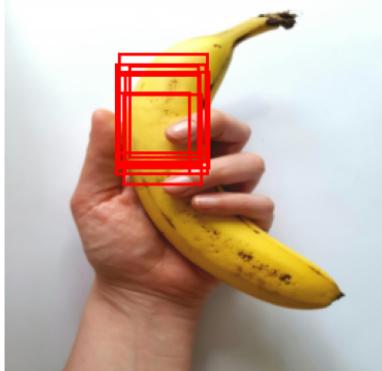


SSD Objectness Loss & Bounding Box Loss



Test on Banana Images outside the Dataset

/content/drive/MyDrive/csci581/bananas/banana1.jpg



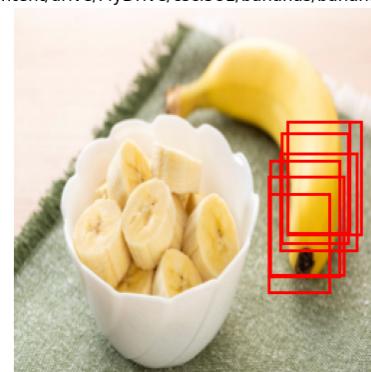
/content/drive/MyDrive/csci581/bananas/banana2.jpg



/content/drive/MyDrive/csci581/bananas/banana3.jpg



/content/drive/MyDrive/csci581/bananas/banana4.jpg

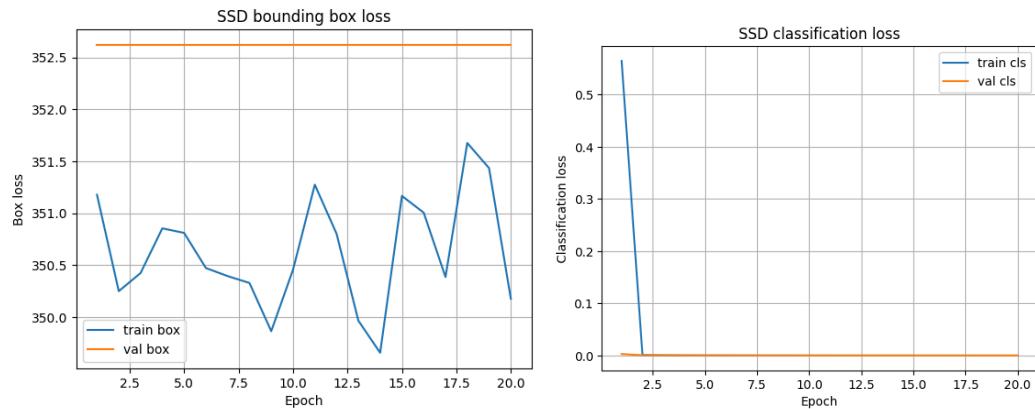


My model performed poorly on my difficult test images but it sheds some light on the banana dataset as well as my model. My model's anchor scales were 10% and 20% which fit the banana dataset well but cause issues with the banana images I found on the internet. These outside test images have bananas that are much larger (relative to the image size). Also several

bananas included in the image also impacts performance. Lastly the image of a peeled banana is so far from the bananas in the training set that the model mis-classifies the image entirely.

To solve these issues I could introduce larger anchor scales adding a 35% and a 50% weights. I could also create another set of images to train on which would include multiple bananas, complex orientations, more zoomed in images as well as some bananas that have already been peeled. Lastly I could implement a stronger backbone in the model architecture.

Initial Convergence Issues, No learning from bounding boxes



Initially my tiny SSD implementation did not correctly handle the SSD encoding of box offsets, I corrected this to successfully predict the offset relative to each Anchor. Also I implemented hard negative mining which showed much better results, keeping a 3:1 ratio of negative to positive anchors and selecting the highest-loss negatives preventing easy negatives from taking over the loss.

Part 2 - Non-Maximum Suppression (NMS)

Results Prior to NMS (7 Detections with high overlap (IoU))

Before NMS



Results After NMS implementation (successful nms suppression)

After custom NMS



Comparison With PyTorch's NMS Implementation (similar box selection)

After torchvision NMS



Both my NMS and PyTorch's NMS suppressed redundant boxes

Purpose and Limitations

The tiny SSD implementation produces several overlapping images which bound the banana in the image. The reason for this is that there are multiple anchors in the feature map which hit on the same object (banana). For each class my NMS implementation sorts boxes by confidence score, keeps the highest bounding box, and then suppresses remaining boxes where the intersection over union (IoU) is greater than the threshold (0.5). As you can see the results in my NMS implementation successfully removes 6 of the 7 detections resulting in one bounding box.

Some limitations of NMS include overlapping objects (seen in my test images), crowded scenes, and lastly a IoU threshold that is too sensitive. The IoU threshold is extremely important, if it is too low NMS will remove true positives while if it is too high multiple similar bounding boxes will remain and the suppression will fail.

Part 3 - Human–Object Interaction (HOI) Analysis using VLMs

Initial Prompt To ChatGPT to Predict <interaction object>

You are an HOI (Human-Object Interaction) recognizer. For each Image I show you...

1. List each person visible as person 1, person 2, and so on and so forth.
2. For each person describe the objects they are interacting with.
3. Describe each interaction in the form of a <interaction object> with the interaction being a verb (example <ride bicycle>)
4. Only include interactions that involve physical contact or control.

Successful Cases (ChatGpt 5.1)



Person 1 (batter in red uniform):

Objects (baseball bat, baseball, ground (foot contact))

Interactions: (<swing bat>, <hit ball>, <stand_on ground>)



Person 1

Objects interacted with: (snowboard, snow (ground), gloves, boots)

Interactions (<grab snowboard>, <ride snowboard>, <jump snow>, <wear gloves>, <wear boots>)



No person-object interactions



Person 1

Objects (laptop, table, robotic arm (indirect contact via interaction workspace), basket of plastic balls)

Interactions (<touch laptop>, <rest_hand_on table>, <handle ball_basket>)

Person 2 (right person holding controller):

Objects (game controller / joystick, table, robotic arm (indirect control))

Interactions: (<hold controller>, <control robotic_arm>, <sit_on chair>)

Failure Cases and Reasonings

ChatGPT 5.1 did a great job at HOI detection so I dropped the model to ChatGPT 4.0



Person 1 (left, standing)

Objects (Mesh bag of plastic balls (left side of table), Table surface)

Interactions: (<touch mesh bag>, <lean on table>)

Person 2 (right, seated)

Objects (Game controller, Robot arm (indirectly via controller), Chair)

Interactions:(<hold controller>, <control robot>, <sit chair>)

Fixes (Better Prompts / Few-shot examples)

I want to improve my prompt to improve ChatGPT 4.0's HOI performance on the image above. ChatGPT 5.1 successfully identifies person 1 as working on their laptop while ChatGPT 4.0 does not identify this object or interaction.

New Improved Prompt

1. List each person visible as person 1, person 2, and so on and so forth.
2. For each person describe the objects they are interacting with. Describe each interaction in the form of a <interaction object> with the interaction being a verb (example <ride bicycle>).

3. Output interactions based on importance, the most significant object-interaction taken by each person should be identified first in your output).
4. Also factor in distance, some objects can be interacted with even though they aren't physically touching.



Most significant interactions ChatGPT 4.0

Person 1 – Seated (right), red sweater, using controller

Human-Object Interaction: (<control robot>, <hold controller>, <sit chair>, <observe laptop> — Likely monitoring the robot via a nearby laptop screen)

Person 2 – Standing (left), brown shirt, near mesh bag of plastic balls

Human-Object Interaction: <assist with balls>, <touch mesh bag>, <observe robot> — Proximal visual engagement with the robot, possibly guiding or supervising the task, <rest hand table>)

Analysis

This time ChatGPT does a much better job identifying the laptop on the table, for some reason it still isn't connected to person 2 who is clearly utilizing the laptop for observation or changes while testing the robot. Even so, by adding a distance call and a mention of importance based on the scene in the prompt, the model successfully identifies the laptop and its use (even though it's attributed to the wrong person).