**Documentation**

**Google Colab**

This code is using the ultralytics library in Python to perform object detection using the YOLO (You Only Look Once) algorithm. Here's a breakdown of the code:

1. !pip install ultralytics: This line installs the ultralytics library using pip, the Python package manager.
2. from ultralytics import YOLO: This line imports the YOLO class from the ultralytics library.
3. !yolo task=detect mode=predict model=yolov8l.pt conf=0.25 source = 'https://ultralytics.com/images/bus.jpg': This line runs the yolo command-line tool to perform object detection on the image located at https://ultralytics.com/images/bus.jpg. The task parameter is set to detect, which tells the tool to perform object detection. The mode parameter is set to predict, which means that the tool will use a pre-trained model to make predictions on the input image. The model parameter specifies the path to the pre-trained YOLOv8l model, which is located in the yolov8l.pt file. The conf parameter sets the confidence threshold for object detection, which means that only objects with a confidence score above 0.25 will be detected.
4. !yolo task=detect mode=train model=yolov8l.pt data =../content/drive/MyDrive/Datasets/FunctionalBlockDetection/data.yaml epochs=50 imgsz=640: This line runs the yolo command-line tool to train a new YOLOv8l model using the data from the data.yaml file located in the ../content/drive/MyDrive/Datasets/FunctionalBlockDetection directory. The task parameter is set to detect, which tells the tool to perform object detection. The mode parameter is set to train, which means that the tool will train a new model using the provided data. The model parameter specifies the path to the pre-trained YOLOv8l model, which is located in the yolov8l.pt file. The data parameter specifies the path to the data file containing the images and annotations for training. The epochs parameter sets the number of training epochs to 50, which means that the model will be trained for 50 iterations. The imgsz parameter sets the input image size to 640 pixels.

In summary, the first line installs the ultralytics library, and the remaining lines use the yolo command-line tool to perform object detection on an image and train a new YOLOv8l model using a data set.

**PyCharm/Other Code**

# Import necessary libraries

from ultralytics import YOLO

import cv2

import cvzone

import math

# Initialize video capture object

cap = cv2.VideoCapture("../Videos/vid1.mp4") # For Images

# Load YOLO model

model = YOLO("best.pt")

# Define class names for bounding box labels

classNames = ['article', 'chat-panel', 'comments', 'contentinfo', 'control-bar', 'form', 'items-grid', 'items-list',

'login', 'media', 'menu-bar', 'menu-panel', 'option-panel', 'posts', 'search', 'side-panel', 'tab-bar']

# Define color for bounding box labels

myColor = (0, 0, 255)

# Loop over video frames

while True:

# Read a frame from the video

success, img = cap.read()

# Check if the video has ended

if not success:

break

# Run the image through the YOLO model to detect objects

results = model(img, stream=True)

# Loop over the detected objects

for r in results:

boxes = r.boxes

for box in boxes:

# Extract bounding box coordinates and confidence

x1, y1, x2, y2 = box.xyxy[0]

conf = math.ceil((box.conf[0] \* 100)) / 100

# Get class name and color for the bounding box

cls = int(box.cls[0])

currentClass = classNames[cls]

myColor = (255, 1, 0)

# Draw bounding box and label on the image

cvzone.putTextRect(img, f'{currentClass} {conf}',

(max(0, x1), max(35, y1)), scale=2, thickness=2, colorB=myColor,

colorT=(255, 255, 255), colorR=myColor, offset=5)

cv2.rectangle(img, (x1, y1), (x2, y2), myColor, 3)

# Resize the image to a smaller size

resized\_img = cv2.resize(img, (1280, 720)) # Adjust the size as needed

# Display the resized image

cv2.imshow("Image", resized\_img)

# Break the loop if the 'q' pressed

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release the video capture object and close all windows

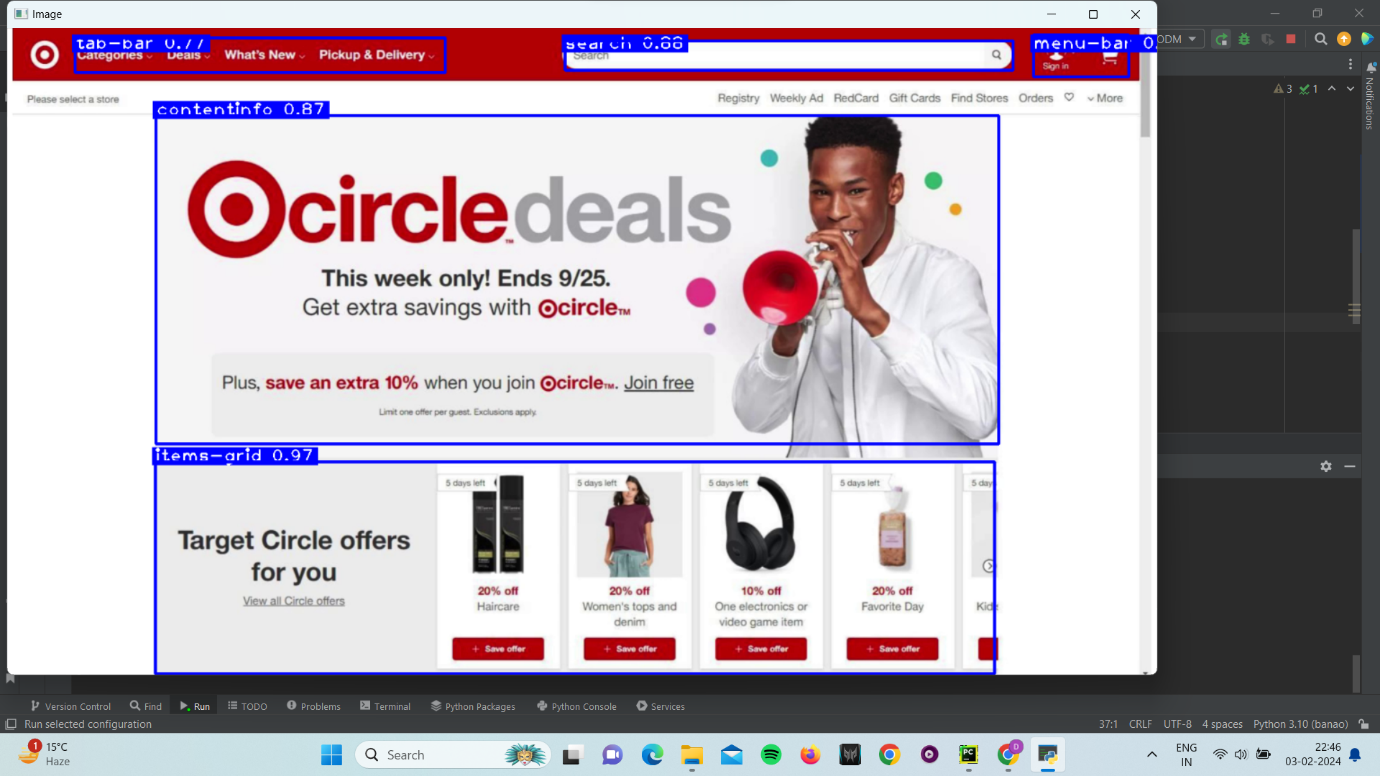
cap.release()

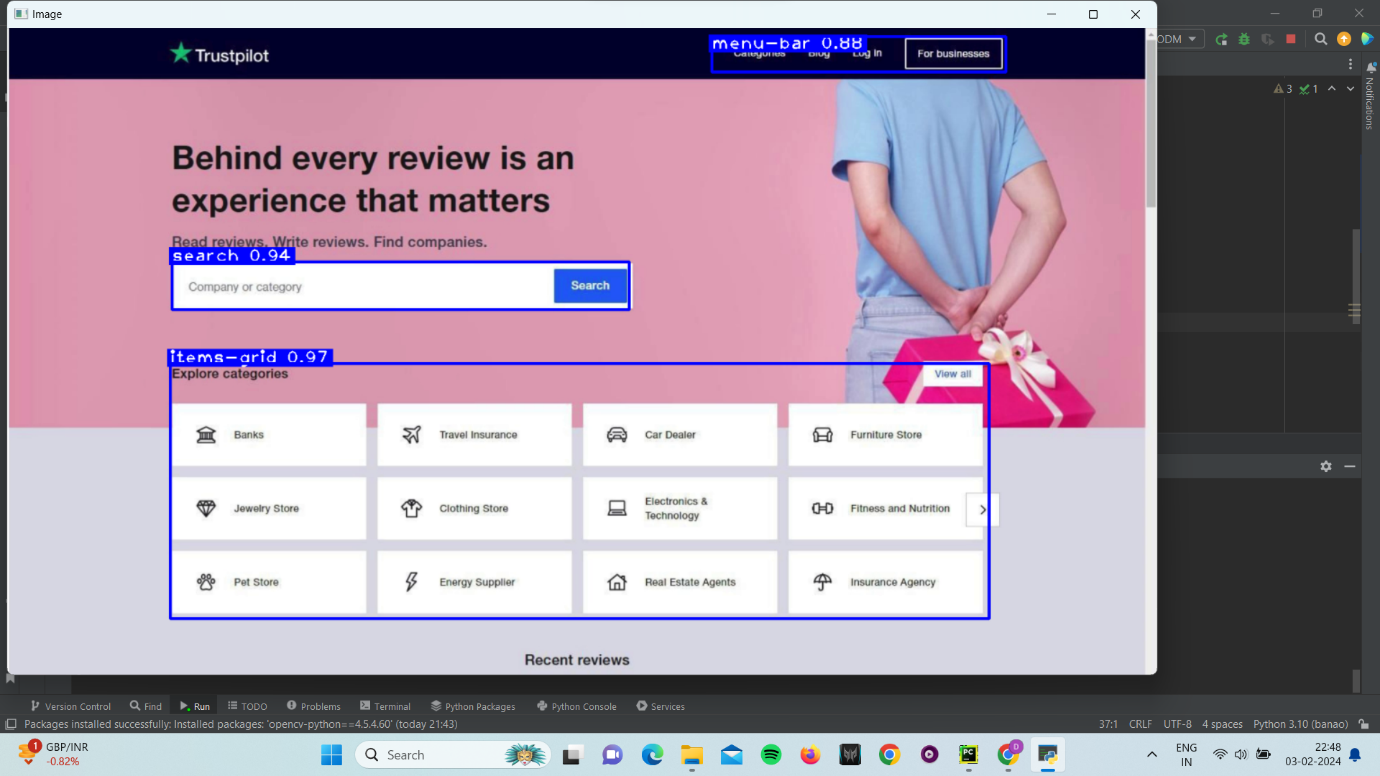
cv2.destroyAllWindows()

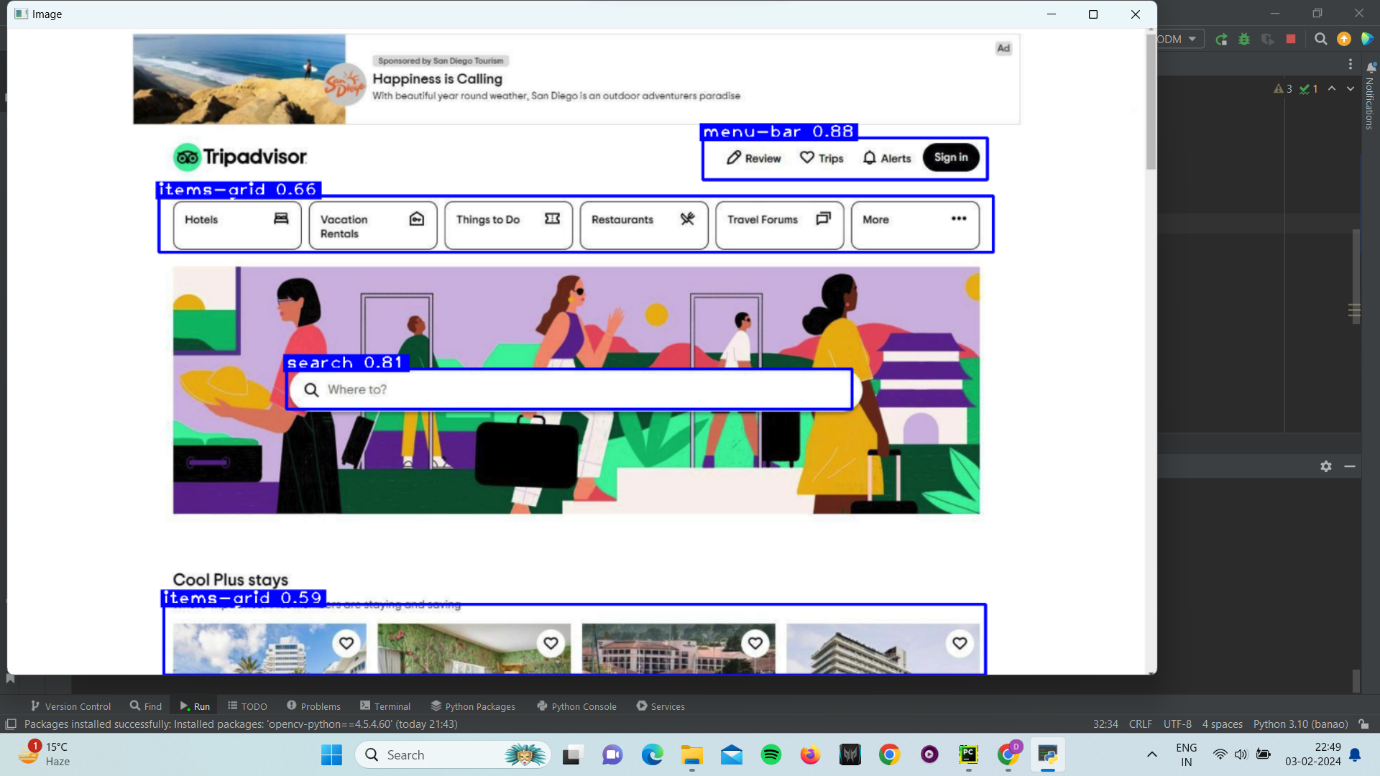
This code uses the YOLO object detection model to detect objects in a video and draw bounding boxes around them. It also displays the class name and confidence level for each detected object. The code uses the OpenCV library to interact with the video and display the resulting image.

The documentation comments provide a brief overview of the code's functionality, inputs, and outputs. They also include information about the libraries used and any relevant parameters or variables. This information can be helpful for other developers who may need to understand or modify the code in the future.

**Outputs:**

****

****

****

**Results:**

| **epoch** | **train/box\_loss** | **train/cls\_loss** | **train/dfl\_loss** | **metrics/precision(B)** | **metrics/recall(B)** | **metrics/mAP50(B)** | **metrics/mAP50-95(B)** | **val/box\_loss** | **val/cls\_loss** | **val/dfl\_loss** | **lr/pg0** | **lr/pg1** | **lr/pg2** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1.4121 | 2.7146 | 1.4696 | 0.28491 | 0.29112 | 0.19957 | 0.12039 | 1.4058 | 2.7013 | 1.4831 | 0.00015738 | 0.00015738 | 0.00015738 |
| 2 | 1.3017 | 2.1619 | 1.3995 | 0.56534 | 0.26947 | 0.26053 | 0.16129 | 1.355 | 2.9669 | 1.5325 | 0.00030979 | 0.00030979 | 0.00030979 |
| 3 | 1.285 | 2.0798 | 1.3915 | 0.32388 | 0.37559 | 0.2473 | 0.13807 | 1.5141 | 2.6087 | 1.6097 | 0.00045591 | 0.00045591 | 0.00045591 |
| 4 | 1.2688 | 2.0356 | 1.3851 | 0.59764 | 0.22542 | 0.22017 | 0.13273 | 1.6965 | 2.6635 | 1.7301 | 0.00044773 | 0.00044773 | 0.00044773 |
| 5 | 1.2158 | 1.9507 | 1.3562 | 0.41537 | 0.42853 | 0.27177 | 0.16658 | 1.4726 | 2.3943 | 1.5078 | 0.00044773 | 0.00044773 | 0.00044773 |
| 6 | 1.1615 | 1.8401 | 1.3238 | 0.46102 | 0.41456 | 0.4119 | 0.28604 | 1.2714 | 1.8436 | 1.4051 | 0.0004383 | 0.0004383 | 0.0004383 |
| 7 | 1.1432 | 1.8028 | 1.313 | 0.55323 | 0.46266 | 0.50473 | 0.33251 | 1.2312 | 1.7311 | 1.4033 | 0.00042888 | 0.00042888 | 0.00042888 |
| 8 | 1.0972 | 1.7067 | 1.2856 | 0.42582 | 0.44898 | 0.44974 | 0.30378 | 1.188 | 1.7176 | 1.3619 | 0.00041945 | 0.00041945 | 0.00041945 |
| 9 | 1.0774 | 1.6578 | 1.2779 | 0.50614 | 0.42624 | 0.44829 | 0.31535 | 1.2343 | 1.739 | 1.3542 | 0.00041003 | 0.00041003 | 0.00041003 |
| 10 | 1.0481 | 1.6114 | 1.2547 | 0.55601 | 0.51018 | 0.50135 | 0.3688 | 1.1637 | 1.6722 | 1.3194 | 0.0004006 | 0.0004006 | 0.0004006 |
| 11 | 1.0217 | 1.5589 | 1.2459 | 0.45481 | 0.53337 | 0.46144 | 0.32405 | 1.1405 | 1.5832 | 1.3025 | 0.00039118 | 0.00039118 | 0.00039118 |
| 12 | 0.98814 | 1.5017 | 1.2269 | 0.55475 | 0.4809 | 0.50569 | 0.35874 | 1.1259 | 1.6334 | 1.3105 | 0.00038175 | 0.00038175 | 0.00038175 |
| 13 | 0.96516 | 1.4203 | 1.2082 | 0.624 | 0.40493 | 0.48039 | 0.33656 | 1.1273 | 1.6304 | 1.3094 | 0.00037233 | 0.00037233 | 0.00037233 |
| 14 | 0.96626 | 1.4002 | 1.215 | 0.71753 | 0.39244 | 0.52748 | 0.37763 | 1.1468 | 1.619 | 1.313 | 0.0003629 | 0.0003629 | 0.0003629 |
| 15 | 0.93357 | 1.3564 | 1.1931 | 0.67725 | 0.46911 | 0.62411 | 0.46753 | 1.0733 | 1.443 | 1.25 | 0.00035348 | 0.00035348 | 0.00035348 |
| 16 | 0.93572 | 1.3157 | 1.2033 | 0.58587 | 0.47457 | 0.52538 | 0.4007 | 1.0935 | 1.5817 | 1.3155 | 0.00034405 | 0.00034405 | 0.00034405 |
| 17 | 0.90037 | 1.2757 | 1.1817 | 0.67309 | 0.41212 | 0.48038 | 0.35718 | 1.057 | 1.5593 | 1.2696 | 0.00033463 | 0.00033463 | 0.00033463 |
| 18 | 0.87593 | 1.2235 | 1.1627 | 0.57633 | 0.5154 | 0.55876 | 0.40475 | 1.069 | 1.5558 | 1.3036 | 0.0003252 | 0.0003252 | 0.0003252 |
| 19 | 0.87184 | 1.1952 | 1.1556 | 0.5194 | 0.55708 | 0.55779 | 0.40157 | 1.1203 | 1.576 | 1.2993 | 0.00031578 | 0.00031578 | 0.00031578 |
| 20 | 0.86112 | 1.1995 | 1.1547 | 0.59374 | 0.54067 | 0.62511 | 0.46838 | 1.0383 | 1.4018 | 1.2758 | 0.00030635 | 0.00030635 | 0.00030635 |
| 21 | 0.83642 | 1.1421 | 1.1372 | 0.67186 | 0.52164 | 0.59749 | 0.45587 | 1.0915 | 1.3433 | 1.2785 | 0.00029693 | 0.00029693 | 0.00029693 |
| 22 | 0.83007 | 1.1114 | 1.1377 | 0.55462 | 0.71358 | 0.66433 | 0.49901 | 1.1213 | 1.3084 | 1.3077 | 0.0002875 | 0.0002875 | 0.0002875 |
| 23 | 0.82628 | 1.1011 | 1.1317 | 0.6248 | 0.61176 | 0.63463 | 0.48405 | 1.0374 | 1.3332 | 1.2374 | 0.00027808 | 0.00027808 | 0.00027808 |
| 24 | 0.79496 | 1.0434 | 1.113 | 0.52604 | 0.60249 | 0.58445 | 0.4298 | 1.1028 | 1.533 | 1.3039 | 0.00026865 | 0.00026865 | 0.00026865 |
| 25 | 0.79129 | 1.0358 | 1.1091 | 0.50044 | 0.64009 | 0.61504 | 0.43889 | 1.0796 | 1.4297 | 1.2839 | 0.00025923 | 0.00025923 | 0.00025923 |
| 26 | 0.77785 | 1.0238 | 1.1048 | 0.70099 | 0.51967 | 0.64149 | 0.49384 | 0.99726 | 1.3434 | 1.2197 | 0.0002498 | 0.0002498 | 0.0002498 |
| 27 | 0.77585 | 0.97769 | 1.096 | 0.61743 | 0.54877 | 0.62177 | 0.45734 | 1.0339 | 1.2992 | 1.245 | 0.00024038 | 0.00024038 | 0.00024038 |
| 28 | 0.74961 | 0.96172 | 1.0871 | 0.7538 | 0.53693 | 0.65551 | 0.49759 | 1.0077 | 1.3245 | 1.2389 | 0.00023096 | 0.00023096 | 0.00023096 |
| 29 | 0.74822 | 0.95155 | 1.0848 | 0.74123 | 0.51269 | 0.61636 | 0.46096 | 1.1054 | 1.326 | 1.2589 | 0.00022153 | 0.00022153 | 0.00022153 |
| 30 | 0.73076 | 0.92942 | 1.071 | 0.5967 | 0.70807 | 0.66082 | 0.50426 | 1.0558 | 1.2147 | 1.248 | 0.00021211 | 0.00021211 | 0.00021211 |
| 31 | 0.72835 | 0.91465 | 1.0755 | 0.60197 | 0.63684 | 0.66949 | 0.51704 | 0.97156 | 1.2794 | 1.2171 | 0.00020268 | 0.00020268 | 0.00020268 |
| 32 | 0.7149 | 0.88686 | 1.064 | 0.70765 | 0.55769 | 0.65272 | 0.47804 | 1.094 | 1.3381 | 1.2866 | 0.00019326 | 0.00019326 | 0.00019326 |
| 33 | 0.70481 | 0.88366 | 1.0613 | 0.67926 | 0.58145 | 0.65161 | 0.48479 | 1.053 | 1.3772 | 1.2399 | 0.00018383 | 0.00018383 | 0.00018383 |
| 34 | 0.69512 | 0.84886 | 1.0522 | 0.66185 | 0.58772 | 0.66872 | 0.50476 | 1.0251 | 1.2863 | 1.2222 | 0.00017441 | 0.00017441 | 0.00017441 |
| 35 | 0.67795 | 0.82389 | 1.0455 | 0.69989 | 0.61493 | 0.67396 | 0.50012 | 1.036 | 1.2986 | 1.2542 | 0.00016498 | 0.00016498 | 0.00016498 |
| 36 | 0.675 | 0.80777 | 1.0474 | 0.78432 | 0.55025 | 0.67907 | 0.51221 | 1.031 | 1.2082 | 1.2374 | 0.00015556 | 0.00015556 | 0.00015556 |
| 37 | 0.67016 | 0.81643 | 1.0431 | 0.75628 | 0.56661 | 0.65723 | 0.48798 | 1.0247 | 1.2983 | 1.2555 | 0.00014613 | 0.00014613 | 0.00014613 |
| 38 | 0.65453 | 0.79496 | 1.0322 | 0.68278 | 0.63451 | 0.68355 | 0.51618 | 1.0543 | 1.2392 | 1.2431 | 0.00013671 | 0.00013671 | 0.00013671 |
| 39 | 0.64703 | 0.75935 | 1.0237 | 0.74612 | 0.63446 | 0.72322 | 0.54683 | 1.02 | 1.2292 | 1.235 | 0.00012728 | 0.00012728 | 0.00012728 |
| 40 | 0.63687 | 0.75004 | 1.0217 | 0.69326 | 0.61608 | 0.6806 | 0.50572 | 1.0179 | 1.1999 | 1.2272 | 0.00011786 | 0.00011786 | 0.00011786 |
| 41 | 0.67485 | 0.72373 | 1.0214 | 0.67742 | 0.62855 | 0.67688 | 0.51754 | 0.95223 | 1.2145 | 1.2255 | 0.00010843 | 0.00010843 | 0.00010843 |
| 42 | 0.64661 | 0.6652 | 1.0023 | 0.76852 | 0.61954 | 0.69698 | 0.53891 | 0.93881 | 1.1873 | 1.2097 | 9.9008e-05 | 9.9008e-05 | 9.9008e-05 |
| 43 | 0.63743 | 0.65219 | 0.99532 | 0.64761 | 0.63116 | 0.68576 | 0.53207 | 0.95119 | 1.177 | 1.2179 | 8.9583e-05 | 8.9583e-05 | 8.9583e-05 |
| 44 | 0.62142 | 0.63076 | 0.98868 | 0.78064 | 0.53169 | 0.69357 | 0.55036 | 0.94717 | 1.2254 | 1.1991 | 8.0158e-05 | 8.0158e-05 | 8.0158e-05 |
| 45 | 0.60664 | 0.65076 | 0.97992 | 0.71318 | 0.60186 | 0.68098 | 0.51411 | 0.96152 | 1.2592 | 1.2114 | 7.0734e-05 | 7.0734e-05 | 7.0734e-05 |
| 46 | 0.60544 | 0.60694 | 0.97749 | 0.64346 | 0.68231 | 0.70571 | 0.53654 | 0.95015 | 1.242 | 1.2107 | 6.1309e-05 | 6.1309e-05 | 6.1309e-05 |
| 47 | 0.59287 | 0.59223 | 0.96898 | 0.74734 | 0.61305 | 0.70655 | 0.54682 | 0.92911 | 1.2081 | 1.2072 | 5.1884e-05 | 5.1884e-05 | 5.1884e-05 |
| 48 | 0.58868 | 0.57977 | 0.97517 | 0.68711 | 0.65889 | 0.7128 | 0.53434 | 0.9288 | 1.1812 | 1.2043 | 4.2459e-05 | 4.2459e-05 | 4.2459e-05 |
| 49 | 0.57268 | 0.56228 | 0.95931 | 0.66071 | 0.72058 | 0.70995 | 0.55911 | 0.94919 | 1.1903 | 1.2132 | 3.3034e-05 | 3.3034e-05 | 3.3034e-05 |
| 50 | 0.56415 | 0.55416 | 0.95818 | 0.6476 | 0.68997 | 0.71367 | 0.54676 | 0.93384 | 1.1737 | 1.1955 | 2.361e-05 | 2.361e-05 | 2.361e-05 |

