In [1]:
```
import pandas as pd
```

In [2]:
```
import os
import torch
from pycocotools.coco import COCO
from torch.utils.data import Dataset, DataLoader
import torchvision.transforms as T
from torchvision.models.detection import fasterrcnn_resnet50_fpn
import cv2
```

In [3]:
```
torch.cuda.empty_cache()
```

In [4]:
```
class CarDamageDataset(Dataset):
    def __init__(self, root, annFile, transforms=None):
        """
        root: Directory where images are stored
        annFile: Path to the COCO-format annotation file
        transforms: Any transformations to apply to the images
        """
        self.root = root
        self.coco = COCO(annFile)
        self.ids = list(self.coco.imgs.keys())  # List of image IDs
        self.transforms = transforms

        # Remove images with no annotations (i.e., no bounding boxes)
        self.ids = [img_id for img_id in self.ids if len(self.coco.getAnnIds(imgIds=img_id)) > 0]

    def __getitem__(self, idx):
        # Get image ID and associated file name
        img_id = self.ids[idx]
        img_info = self.coco.loadImgs(img_id)[0]
        path = img_info['file_name']

        # Load image using file name
        img = cv2.imread(os.path.join(self.root, path))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # Get annotations for this image
        ann_ids = self.coco.getAnnIds(imgIds=img_id)
        annotations = self.coco.loadAnns(ann_ids)

        # Prepare bounding boxes and labels
        boxes = []
        labels = []
        areas = []
        iscrowd = []
        for ann in annotations:
            #print(ann['bbox'])
            xmin, ymin, width, height = ann['bbox']
            boxes.append([xmin, ymin, xmin + width, ymin + height])
            labels.append(ann['category_id'])  # category ID for damaged part
            areas.append(ann['area'])          # area of the bounding box
            iscrowd.append(ann['iscrowd'])     # crowd annotation

        # Convert to tensors
        target = {
            "boxes": torch.tensor(boxes, dtype=torch.float32),
            "labels": torch.tensor(labels, dtype=torch.int64),
            "area": torch.tensor(areas, dtype=torch.float32),
            "iscrowd": torch.tensor(iscrowd, dtype=torch.int64)
        }

        # Apply transforms if any
        if self.transforms:
            img = self.transforms(img)

        return img, target

    def __len__(self):
        return len(self.ids)
```

In [5]:
```
# Paths to image directories and COCO-format annotation files for each split
train_root = "Part_Detection_final_dataset/train"
val_root = "Part_Detection_final_dataset/valid"
test_root = "Part_Detection_final_dataset/test"
train_annFile = "Part_Detection_final_dataset/annotations/train_annotations.coco.json"
val_annFile = "Part_Detection_final_dataset/annotations/val_annotations.coco.json"
test_annFile = "Part_Detection_final_dataset/annotations/test_annotations.coco.json"

# Initialize datasets for each split
train_dataset = CarDamageDataset(root=train_root, annFile=train_annFile, transforms=T.ToTensor())
```

```
val_dataset = CarDamageDataset(root=val_root, annFile=val_annFile, transforms=T.ToTensor())
test_dataset = CarDamageDataset(root=test_root, annFile=test_annFile, transforms=T.ToTensor())
```

```
loading annotations into memory...
Done (t=0.18s)
creating index...
index created!
loading annotations into memory...
Done (t=0.03s)
creating index...
index created!
loading annotations into memory...
Done (t=0.01s)
creating index...
index created!
```

Paths to image directories and COCO-format annotation files for each split

train_root = "Part_Detection_final_dataset/train" val_root = "Part_Detection_final_dataset/val" test_root = "Part_Detection_final_dataset/test"

train_annFile = "Part_Detection_final_dataset/annotations/train_annotations.coco.json" val_annFile = "Part_Detection_final_dataset/annotations/val_annotations.coco.json" test_annFile = "Part_Detection_final_dataset/annotations/test_annotations.coco.json"

Initialize datasets for each split

train_dataset = CarDamageDataset(root=train_root, annFile=train_annFile, transforms=T.ToTensor()) val_dataset = CarDamageDataset(root=val_root, annFile=val_annFile, transforms=T.ToTensor()) test_dataset = CarDamageDataset(root=test_root, annFile=test_annFile, transforms=T.ToTensor())

In [6]:
```
# Create data loaders for each dataset split
train_loader = DataLoader(train_dataset, batch_size=4, shuffle=True,pin_memory=True ,collate_fn=lambda x: tuple
val_loader = DataLoader(val_dataset, batch_size=4, shuffle=False, collate_fn=lambda x: tuple(zip(*x)))
test_loader = DataLoader(test_dataset, batch_size=4, shuffle=True, collate_fn=lambda x: tuple(zip(*x)))
```

In [7]:
```
from torchvision.models.detection.faster_rcnn import FastRCNNPredictor
```

In [9]:
```
# Model setup
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
num_classes = 15 # Assuming 6 damage categories + 1 background class
model = fasterrcnn_resnet50_fpn(pretrained=True)
in_features = model.roi_heads.box_predictor.cls_score.in_features
model.roi_heads.box_predictor = FastRCNNPredictor(in_features, num_classes)
model.to(device)

# Define the optimizer
optimizer = torch.optim.SGD(model.parameters(), lr=0.01, momentum=0.9, weight_decay=0.0005)
```

```
C:\Users\harsh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packag
es\Python311\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecat
ed since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
C:\Users\harsh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packag
es\Python311\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight enum or
`None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equival
ent to passing `weights=FasterRCNN_ResNet50_FPN_Weights.COCO_V1`. You can also use `weights=FasterRCNN_ResNet50_
FPN_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
```

In [10]:
```
from tqdm import tqdm

# Training and validation loop
num_epochs = 15
for epoch in range(num_epochs):
    model.train()
    train_loss = 0

    # Add progress bar to the training loop
    progress_bar = tqdm(train_loader, desc=f"Epoch [{epoch+1}/{num_epochs}]")
    for images, targets in progress_bar:
        images = [img.to(device) for img in images]
        targets = [{k: v.to(device) for k, v in t.items()} for t in targets]

        # Forward pass
```

```
        loss_dict = model(images, targets)
        losses = sum(loss for loss in loss_dict.values())
        train_loss += losses.item()

        # Backward pass
        optimizer.zero_grad()
        losses.backward()
        optimizer.step()

        # Update progress bar with current loss
        progress_bar.set_postfix(loss=losses.item())

    print(f"Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss/len(train_loader):.4f}")
```

```
Epoch [1/15]: 100%|██████████| 1501/1501 [1:28:14<00:00,  3.53s/it, loss=0.13]
Epoch [1/15], Train Loss: 0.2202
Epoch [2/15]: 100%|██████████| 1501/1501 [1:27:27<00:00,  3.50s/it, loss=0.158]
Epoch [2/15], Train Loss: 0.1518
Epoch [3/15]: 100%|██████████| 1501/1501 [1:27:40<00:00,  3.50s/it, loss=0.0695]
Epoch [3/15], Train Loss: 0.1292
Epoch [4/15]: 100%|██████████| 1501/1501 [1:27:16<00:00,  3.49s/it, loss=0.156]
Epoch [4/15], Train Loss: 0.1196
Epoch [5/15]: 100%|██████████| 1501/1501 [1:28:33<00:00,  3.54s/it, loss=0.0732]
Epoch [5/15], Train Loss: 0.1122
Epoch [6/15]: 100%|██████████| 1501/1501 [1:27:15<00:00,  3.49s/it, loss=0.0856]
Epoch [6/15], Train Loss: 0.1081
Epoch [7/15]: 100%|██████████| 1501/1501 [1:42:28<00:00,  4.10s/it, loss=0.072]
Epoch [7/15], Train Loss: 0.1057
Epoch [8/15]: 100%|██████████| 1501/1501 [1:43:44<00:00,  4.15s/it, loss=0.0905]
Epoch [8/15], Train Loss: 0.1007
Epoch [9/15]: 100%|██████████| 1501/1501 [1:45:44<00:00,  4.23s/it, loss=0.138]
Epoch [9/15], Train Loss: 0.0983
Epoch [10/15]: 100%|██████████| 1501/1501 [1:39:08<00:00,  3.96s/it, loss=0.102]
Epoch [10/15], Train Loss: 0.0971
Epoch [11/15]: 100%|██████████| 1501/1501 [1:48:17<00:00,  4.33s/it, loss=0.176]
Epoch [11/15], Train Loss: 0.1035
Epoch [12/15]: 100%|██████████| 1501/1501 [1:56:48<00:00,  4.67s/it, loss=0.0851]
Epoch [12/15], Train Loss: 0.1015
Epoch [13/15]: 100%|██████████| 1501/1501 [1:56:54<00:00,  4.67s/it, loss=0.0981]
Epoch [13/15], Train Loss: 0.0918
Epoch [14/15]: 100%|██████████| 1501/1501 [1:56:37<00:00,  4.66s/it, loss=0.103]
Epoch [14/15], Train Loss: 0.0898
Epoch [15/15]: 100%|██████████| 1501/1501 [1:56:50<00:00,  4.67s/it, loss=0.0917]
Epoch [15/15], Train Loss: 0.0898
```

In [13]:
```
print(f"Images format: {type(images)}, Target format: {type(targets)}")
```

```
Images format: <class 'list'>, Target format: <class 'list'>
```

In [ ]:
```
```

In [15]:
```
# Save the model's state_dict to a file
torch.save(model.state_dict(), 'Part_presection_model52_state_dict.pkl')
```

In [16]:
```
torch.save(model, 'Part_predection_model_full_20241125.pkl')
```

In [8]:
```
model = torch.load('Part_predection_model_full_20241125.pkl', map_location=torch.device('cpu'))
device=torch.device('cpu')
model.to(device)
```

Out[8]:
```
FasterRCNN(
  (transform): GeneralizedRCNNTransform(
      Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
      Resize(min_size=(800,), max_size=1333, mode='bilinear')
  )
  (backbone): BackboneWithFPN(
    (body): IntermediateLayerGetter(
      (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
      (bn1): FrozenBatchNorm2d(64, eps=0.0)
      (relu): ReLU(inplace=True)
      (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
      (layer1): Sequential(
        (0): Bottleneck(
          (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(64, eps=0.0)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(64, eps=0.0)
          (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(256, eps=0.0)
```

```
    (relu): ReLU(inplace=True)
    (downsample): Sequential(
      (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): FrozenBatchNorm2d(256, eps=0.0)
    )
  )
  (1): Bottleneck(
    (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(64, eps=0.0)
    (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(64, eps=0.0)
    (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(256, eps=0.0)
    (relu): ReLU(inplace=True)
  )
  (2): Bottleneck(
    (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(64, eps=0.0)
    (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(64, eps=0.0)
    (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(256, eps=0.0)
    (relu): ReLU(inplace=True)
  )
)
(layer2): Sequential(
  (0): Bottleneck(
    (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(128, eps=0.0)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(128, eps=0.0)
    (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(512, eps=0.0)
    (relu): ReLU(inplace=True)
    (downsample): Sequential(
      (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): FrozenBatchNorm2d(512, eps=0.0)
    )
  )
  (1): Bottleneck(
    (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(128, eps=0.0)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(128, eps=0.0)
    (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(512, eps=0.0)
    (relu): ReLU(inplace=True)
  )
  (2): Bottleneck(
    (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(128, eps=0.0)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(128, eps=0.0)
    (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(512, eps=0.0)
    (relu): ReLU(inplace=True)
  )
  (3): Bottleneck(
    (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(128, eps=0.0)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(128, eps=0.0)
    (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(512, eps=0.0)
    (relu): ReLU(inplace=True)
  )
)
(layer3): Sequential(
  (0): Bottleneck(
    (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): FrozenBatchNorm2d(256, eps=0.0)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn2): FrozenBatchNorm2d(256, eps=0.0)
    (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): FrozenBatchNorm2d(1024, eps=0.0)
    (relu): ReLU(inplace=True)
    (downsample): Sequential(
      (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): FrozenBatchNorm2d(1024, eps=0.0)
    )
  )
  (1): Bottleneck(
    (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

```
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (3): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (4): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (5): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
  )
  (layer4): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(512, eps=0.0)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(512, eps=0.0)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(2048, eps=0.0)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): FrozenBatchNorm2d(2048, eps=0.0)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(512, eps=0.0)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(512, eps=0.0)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(2048, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(512, eps=0.0)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(512, eps=0.0)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(2048, eps=0.0)
      (relu): ReLU(inplace=True)
    )
  )
)
(fpn): FeaturePyramidNetwork(
  (inner_blocks): ModuleList(
    (0): Conv2dNormActivation(
      (0): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    )
```

```
        (1): Conv2dNormActivation(
          (0): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): Conv2dNormActivation(
          (0): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (3): Conv2dNormActivation(
          (0): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
        )
      )
      (layer_blocks): ModuleList(
        (0-3): 4 x Conv2dNormActivation(
          (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        )
      )
      (extra_blocks): LastLevelMaxPool()
    )
  )
  (rpn): RegionProposalNetwork(
    (anchor_generator): AnchorGenerator()
    (head): RPNHead(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): ReLU(inplace=True)
        )
      )
      (cls_logits): Conv2d(256, 3, kernel_size=(1, 1), stride=(1, 1))
      (bbox_pred): Conv2d(256, 12, kernel_size=(1, 1), stride=(1, 1))
    )
  )
  (roi_heads): RoIHeads(
    (box_roi_pool): MultiScaleRoIAlign(featmap_names=['0', '1', '2', '3'], output_size=(7, 7), sampling_ratio=2
)
    (box_head): TwoMLPHead(
      (fc6): Linear(in_features=12544, out_features=1024, bias=True)
      (fc7): Linear(in_features=1024, out_features=1024, bias=True)
    )
    (box_predictor): FastRCNNPredictor(
      (cls_score): Linear(in_features=1024, out_features=15, bias=True)
      (bbox_pred): Linear(in_features=1024, out_features=60, bias=True)
    )
  )
)
```

In [9]:
```
import json
```

In [10]:
```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np
from PIL import Image
```

In [20]:
```
# Function to load the COCO label map from an annotation file
def load_label_map(ann_file):
    with open(ann_file, 'r') as f:
        coco_data = json.load(f)

    label_map = {category['id']: category['name'] for category in coco_data['categories']}
    return label_map
```

In [21]:
```
train_label_map = load_label_map(train_annFile)
print("Train label map:", train_label_map)
```

```
Train label map: {0: 'background', 1: 'Damage-Windshield', 2: 'Damage-boots', 3: 'Damage-door', 4: 'Damage-front
-bumper', 5: 'Damage-headlight', 6: 'Damage-hood', 7: 'Damage-left-fender', 8: 'Damage-rear-bumper', 9: 'Damage-
rear-light', 10: 'Damage-right-fender', 11: 'Damage-roof', 12: 'Damage-side-mirror', 13: 'Damage-window-glass',
14: 'flat-tire'}
```

In [22]:
```
# Initialize a persistent color map for labels (only needs to be done once)
def initialize_color_map(label_map):
    # Use a colormap with high contrast (e.g., 'tab20' or 'hsv')
    cmap = plt.cm.get_cmap('hsv', len(label_map))  # Get `len(label_map)` distinct colors
    color_map = {label: cmap(i) for i, label in enumerate(label_map.keys())}
    return color_map


color_map = initialize_color_map(train_label_map)  # Initialize once with consistent colors

# Visualization function for predictions using the pre-defined color map
def visualize_predictions(image, outputs, label_map, color_map, threshold=0.5):
    fig, ax = plt.subplots(1, figsize=(12, 9))
    ax.imshow(image)
```

```
    pred_boxes = outputs['boxes']
    pred_labels = outputs['labels']
    pred_scores = outputs['scores']

    for i, score in enumerate(pred_scores):
        if score > threshold:
            box = pred_boxes[i].cpu().numpy()
            label = pred_labels[i].cpu().item()
            label_name = label_map.get(label, "Unknown")  # Get the class name from the label map
            score = score.cpu().item()

            # Get the pre-defined color for this label
            color = color_map.get(label, 'red')  # Default to red if label is unknown

            # Draw bounding box with the specific color
            rect = patches.Rectangle(
                (box[0], box[1]), box[2] - box[0], box[3] - box[1],
                linewidth=2, edgecolor=color, facecolor='none'
            )
            ax.add_patch(rect)

            # Display label name and score with the same color
            ax.text(
                box[0], box[1], f"{label_name}: {score:.2f}",
                color='black', fontsize=10, bbox=dict(facecolor=color, alpha=0.5)
            )

    plt.show()
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_1872\2259875598.py:4: MatplotlibDeprecationWarning: The get_cmap fun
ction was deprecated in Matplotlib 3.7 and will be removed in 3.11. Use ``matplotlib.colormaps[name]`` or ``matp
lotlib.colormaps.get_cmap()`` or ``pyplot.get_cmap()`` instead.
  cmap = plt.cm.get_cmap('hsv', len(label_map))  # Get `len(label_map)` distinct colors
```

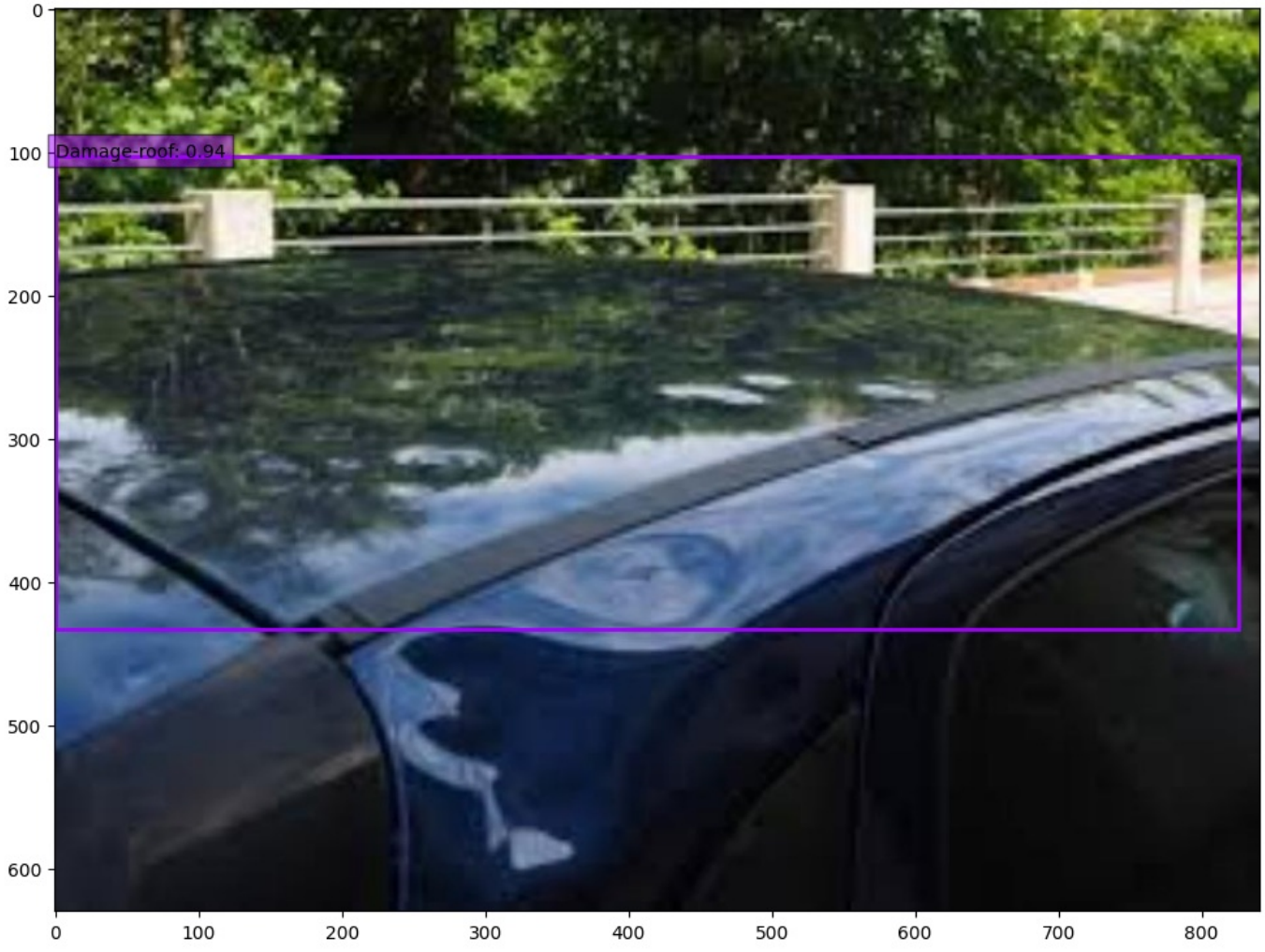
In [23]:

```
# Run inference and visualize predictions
def run_inference(model, data_loader, train_label_map, device):
    model.eval()
    with torch.no_grad():
        for images, _ in data_loader:
            images = [img.to(device) for img in images]
            outputs = model(images)

            for i, output in enumerate(outputs):
                image = images[i].cpu().permute(1, 2, 0).numpy()
                image = np.uint8(image * 255)
                image_pil = Image.fromarray(image)
                visualize_predictions(image_pil, output, train_label_map,color_map)  # Pass the label map

# Correct function call with train_label_map
run_inference(model, test_loader, train_label_map, device)
```


Damage-side-mirror: 1.00
Adobe Stock | #442941719
0
100
200
300
400
500
600
0
200
400
600
800

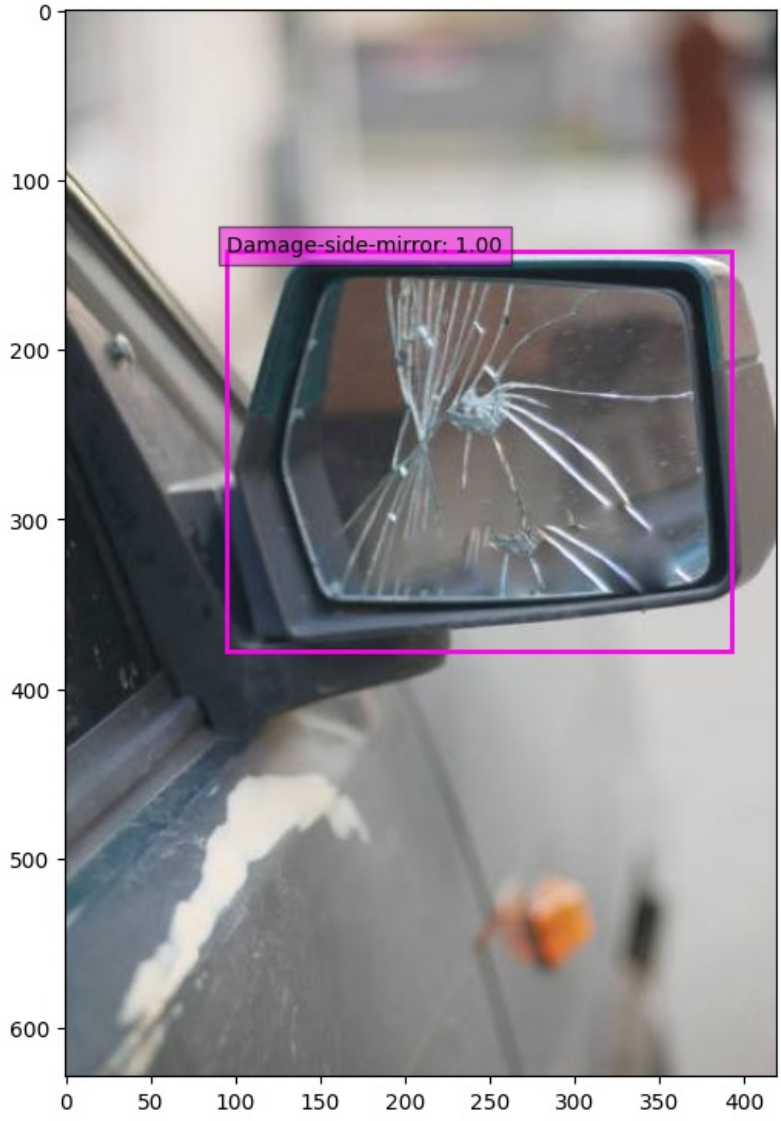

Damage-roof: 0.94
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

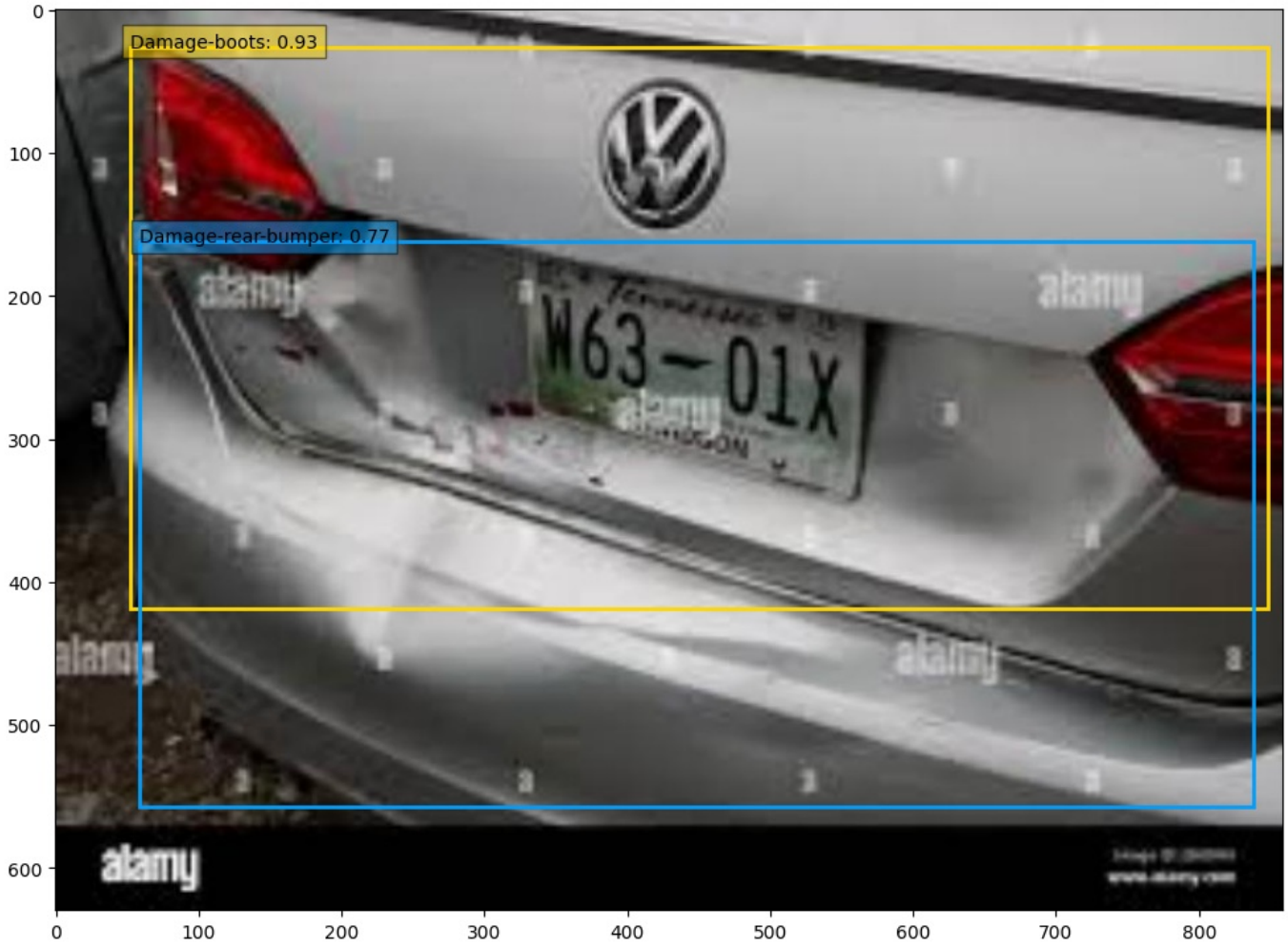

Damage-left-fender: 0.96
0
100
200
300
400
500
600
0
200
400
600
800

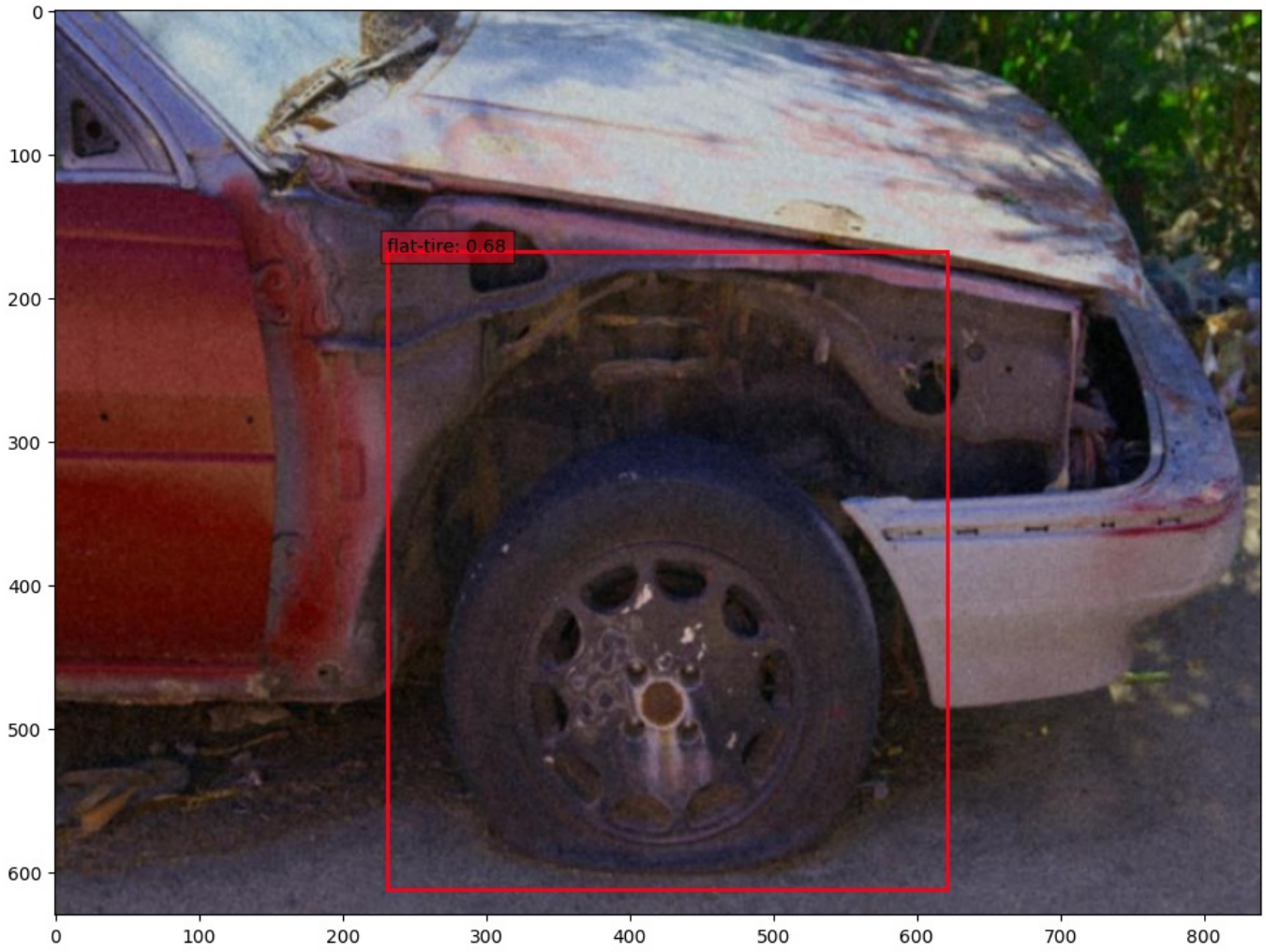

Damage-left-fender: 0.89
0
100
200
300
400
500
600
0
200
400
600
800

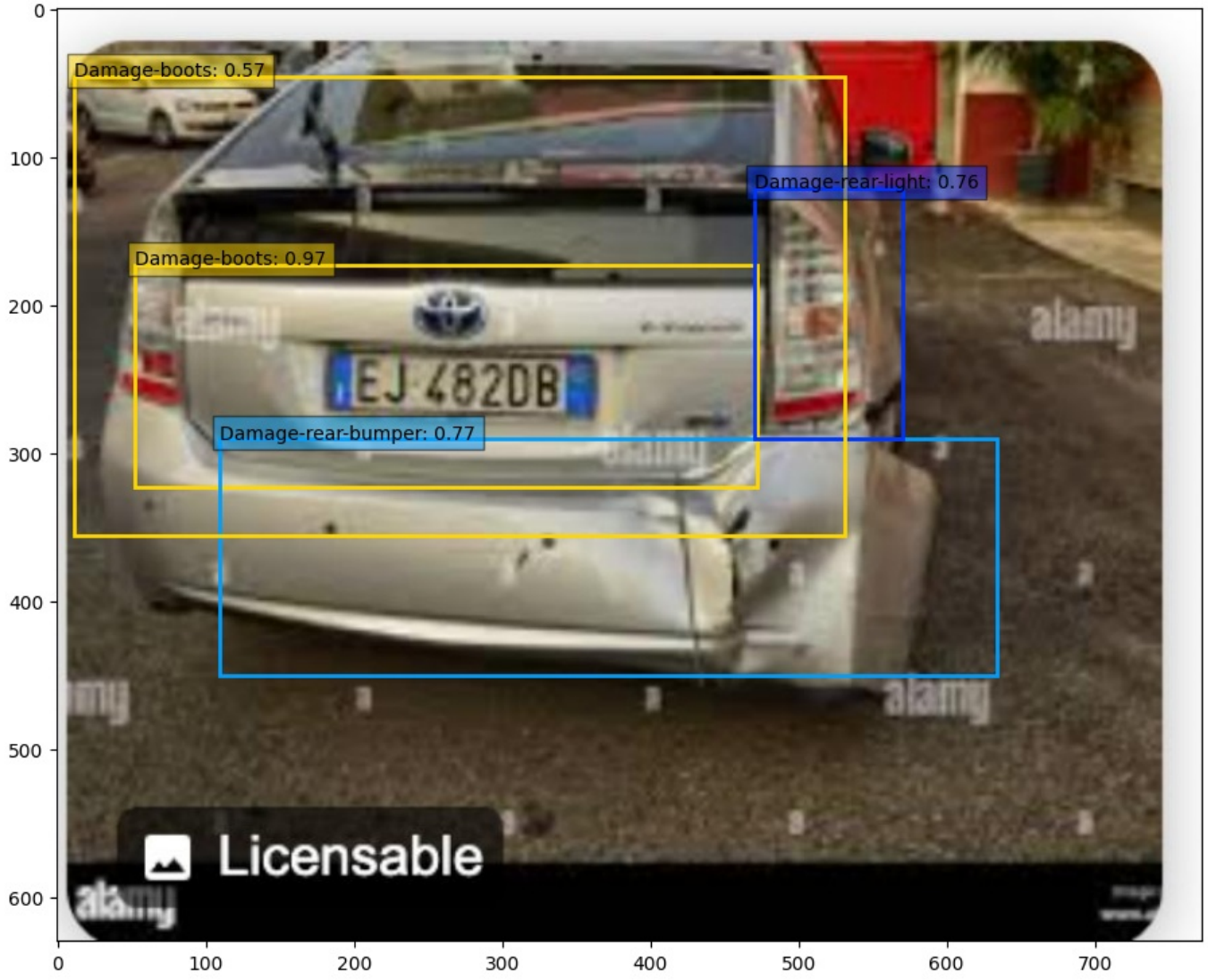

Damage-Windshield: 1.00
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

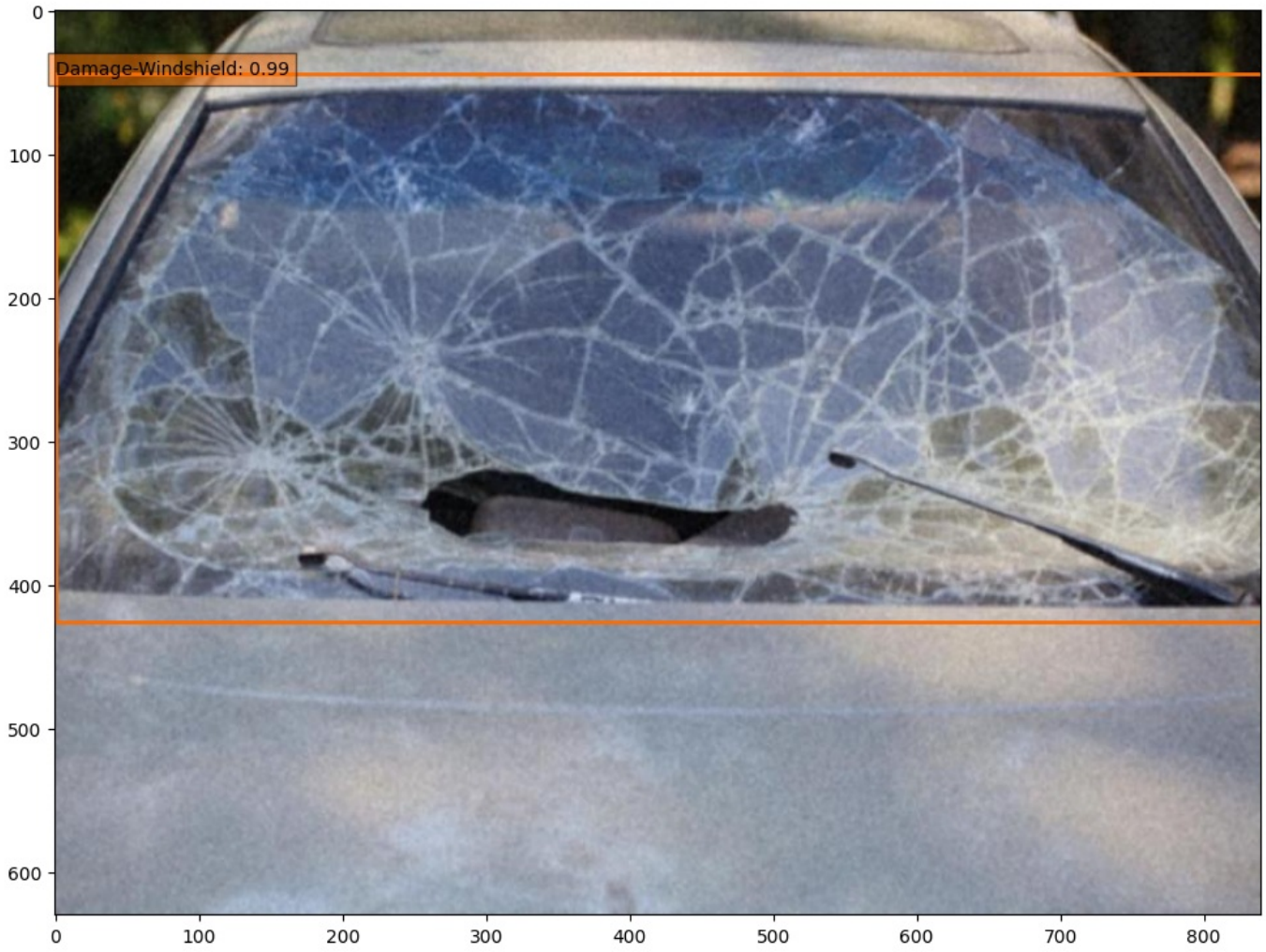

0
100
200
300
400
500
600
0
200
400
600
800

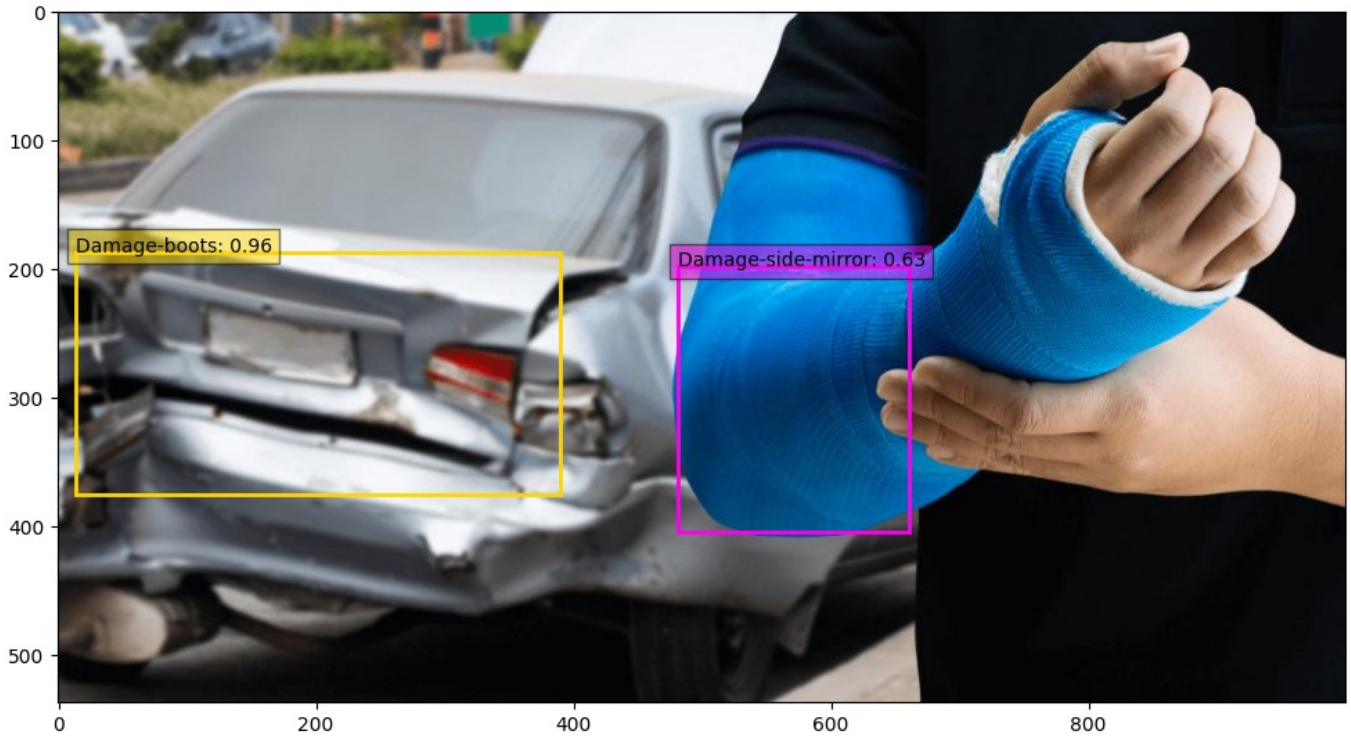

Damage-side-mirror: 1.00
0
100
200
300
400
500
600
0
50
100
150
200
250
300
350
400

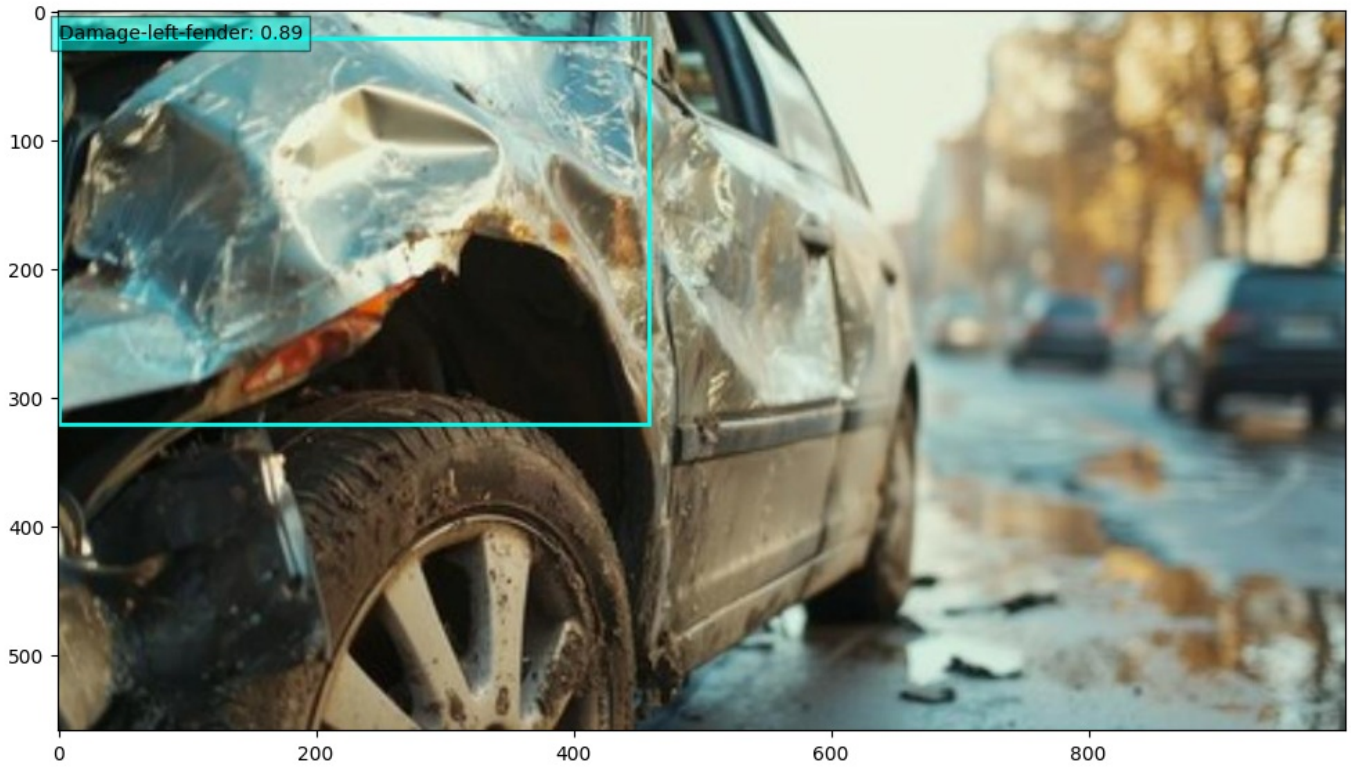

Damage-boots: 0.93
Damage-rear-bumper: 0.77
W63 01X
alamy
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 0.68
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-boots: 0.57
Damage-rear-light: 0.76
Damage-boots: 0.97
Damage-rear-bumper: 0.77
EJ 482DB
alamy
Licensable
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700

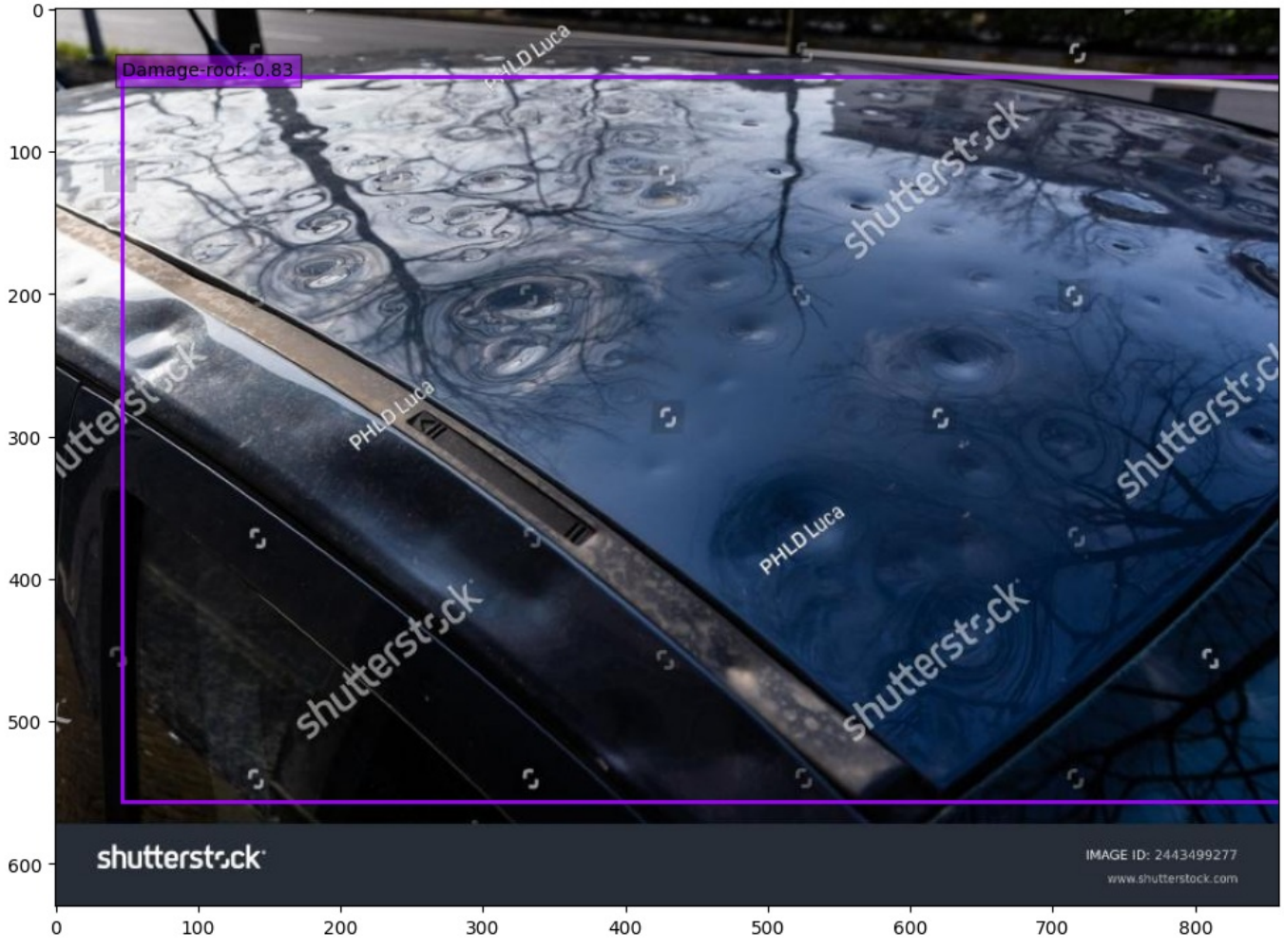

Damage-Windshield: 0.99
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

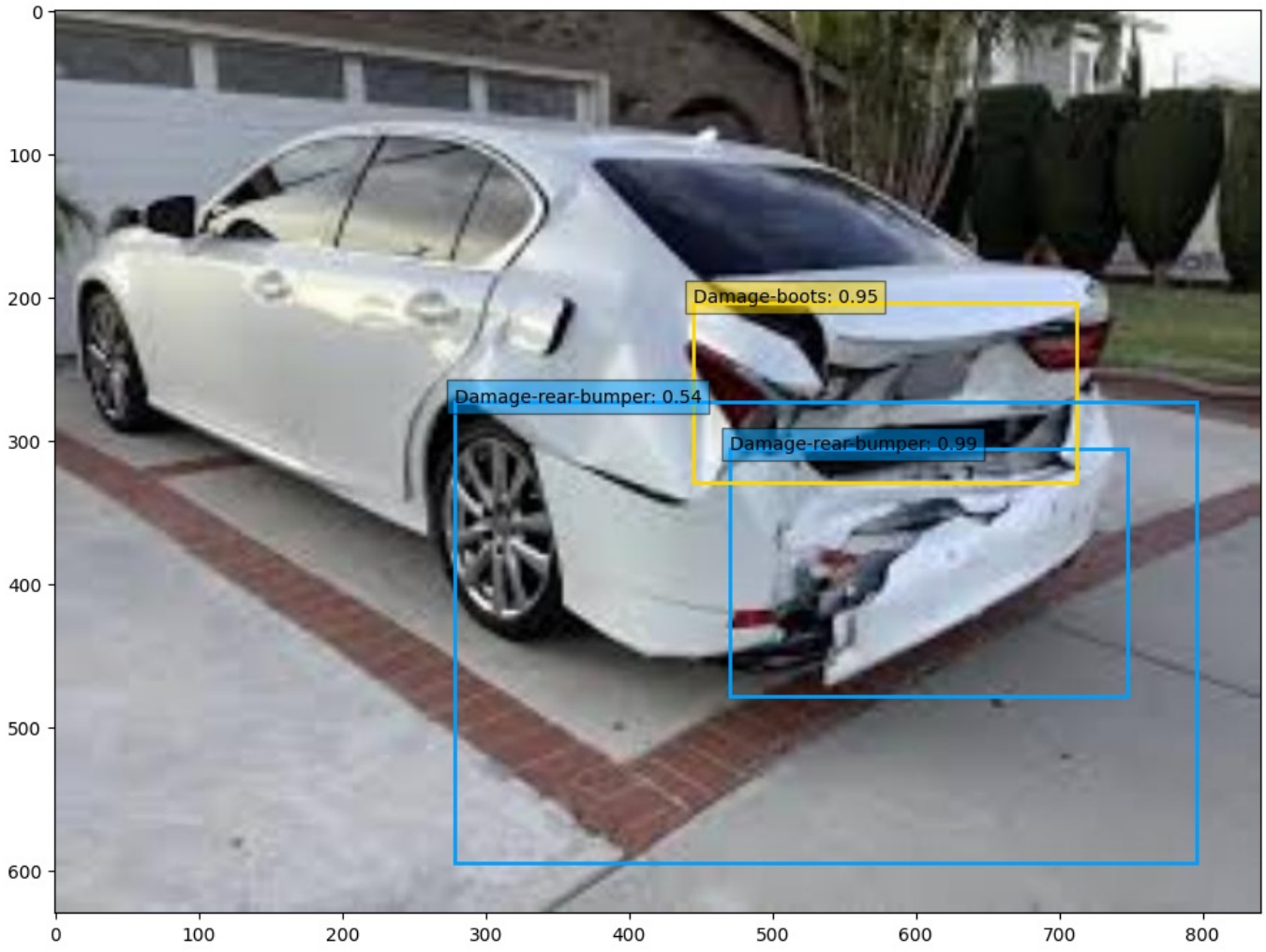

Damage-boots: 0.96
Damage-side-mirror: 0.63
0
100
200
300
400
500
0
200
400
600
800

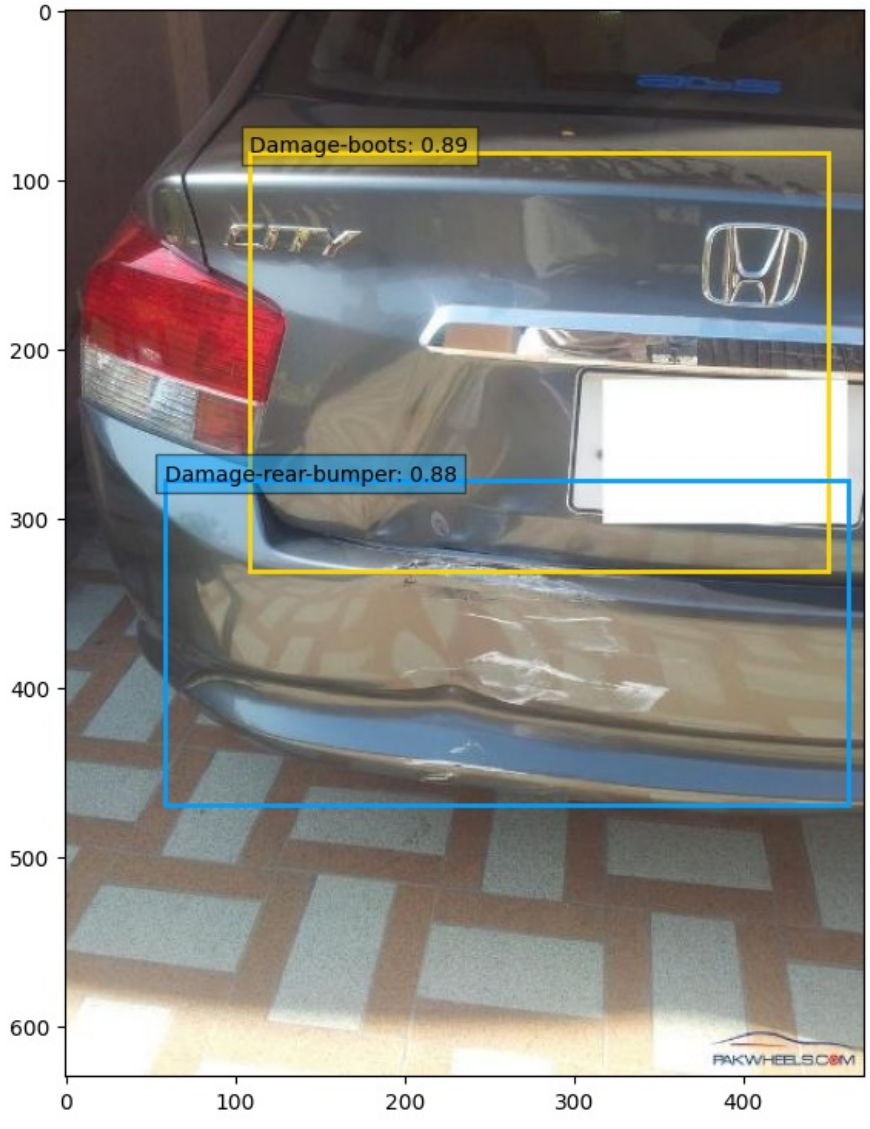

Damage-left-fender: 0.89
0
100
200
300
400
500
0
200
400
600
800


Damage-hood: 0.68
Damage-side-mirror: 0.56
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 1.00
Revisit site, click Upgrade button
to remove watermark and get
1 download plan + 1 more for FREE!
Revisit site, click Upgra
to remove watermark
1 download plan + 1 mor
Image downloaded under free trial
for evaluation purpose only.
dreamstime
0
100
200
300
400
500
600
0
200
400
600
800

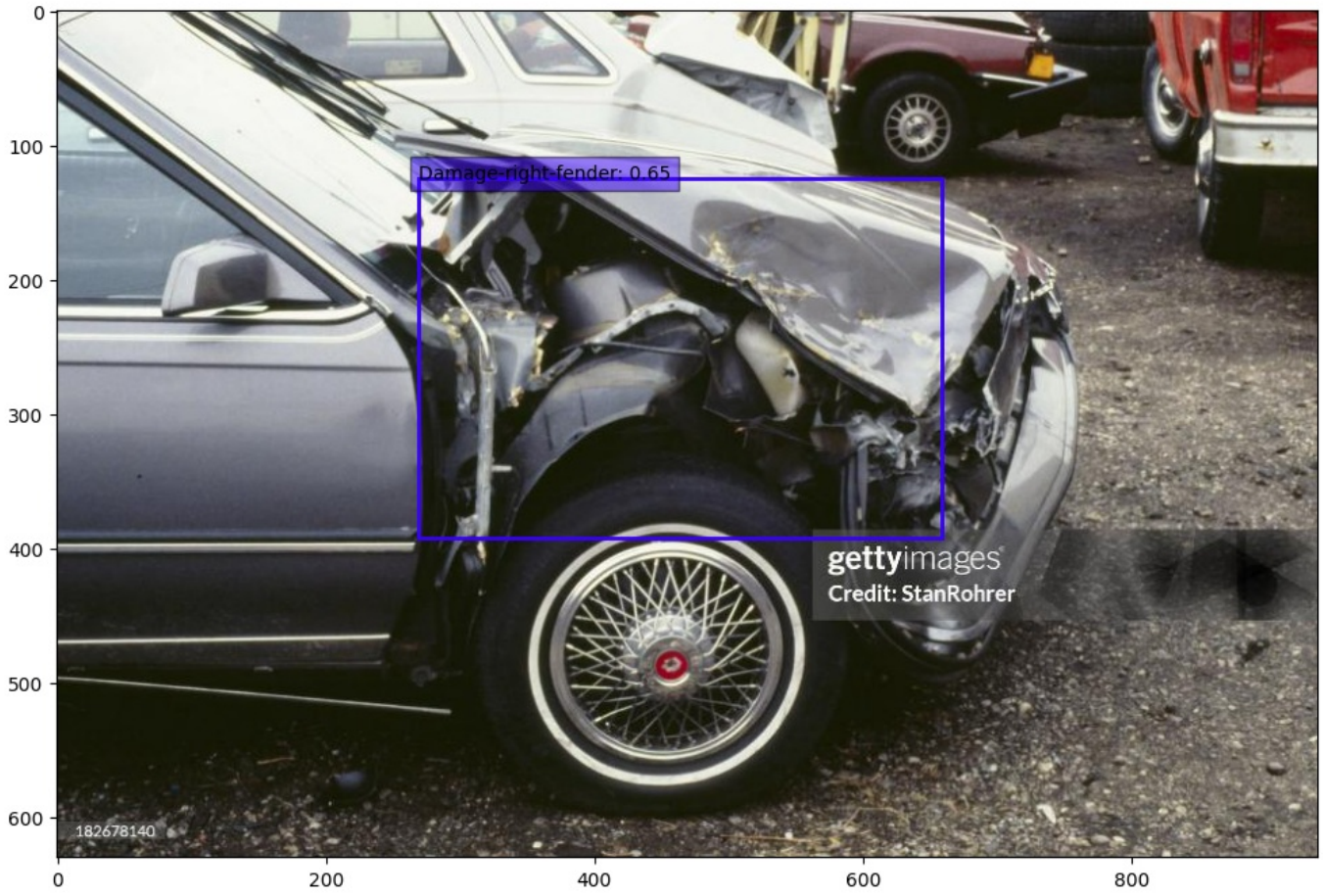

Damage-roof: 0.83
PHLD Luca
shutterstock
IMAGE ID: 2443499277
www.shutterstock.com
0
100
200
300
400
500
600
700
800

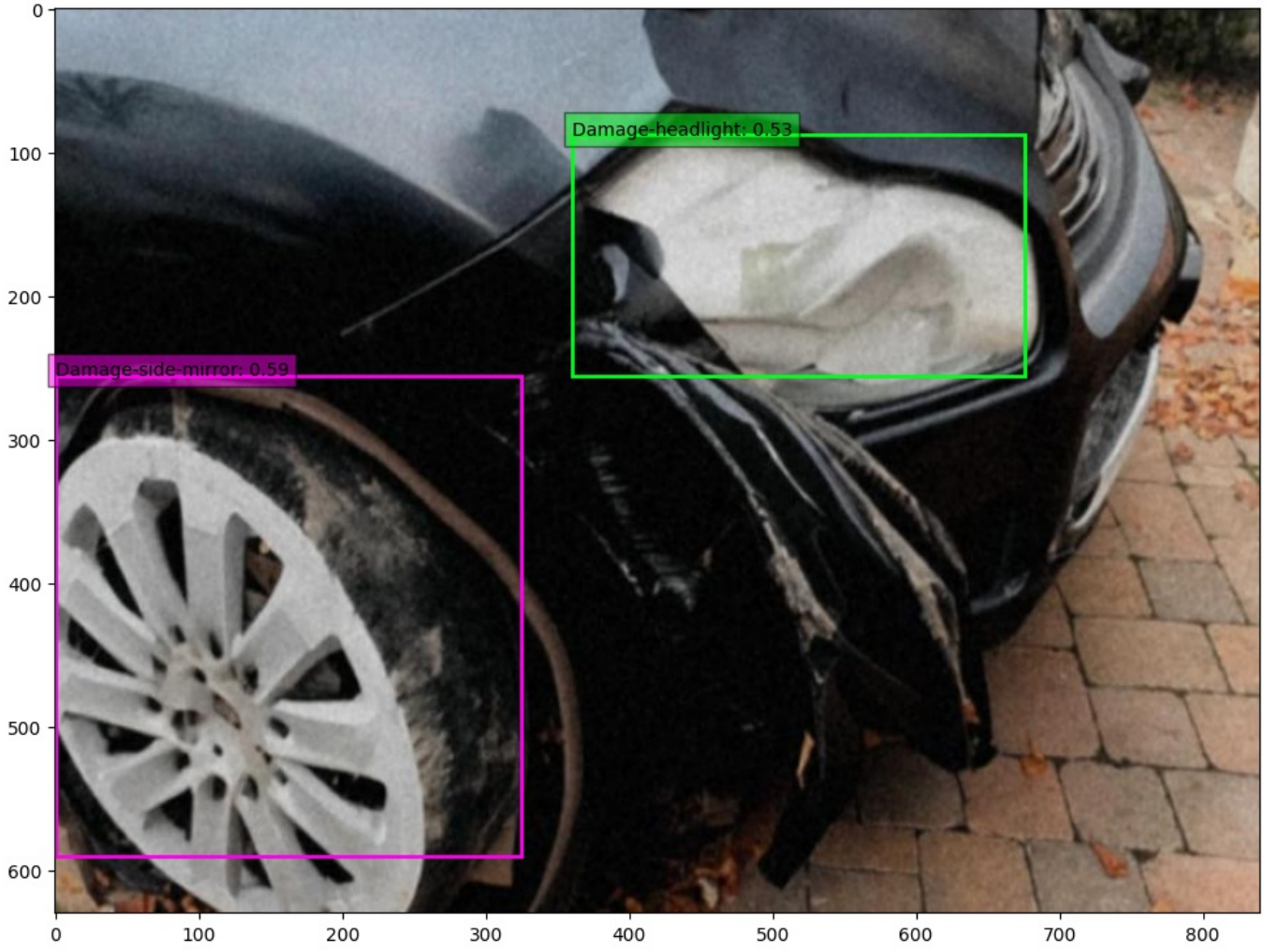

Damage-boots: 0.95
Damage-rear-bumper: 0.54
Damage-rear-bumper: 0.99

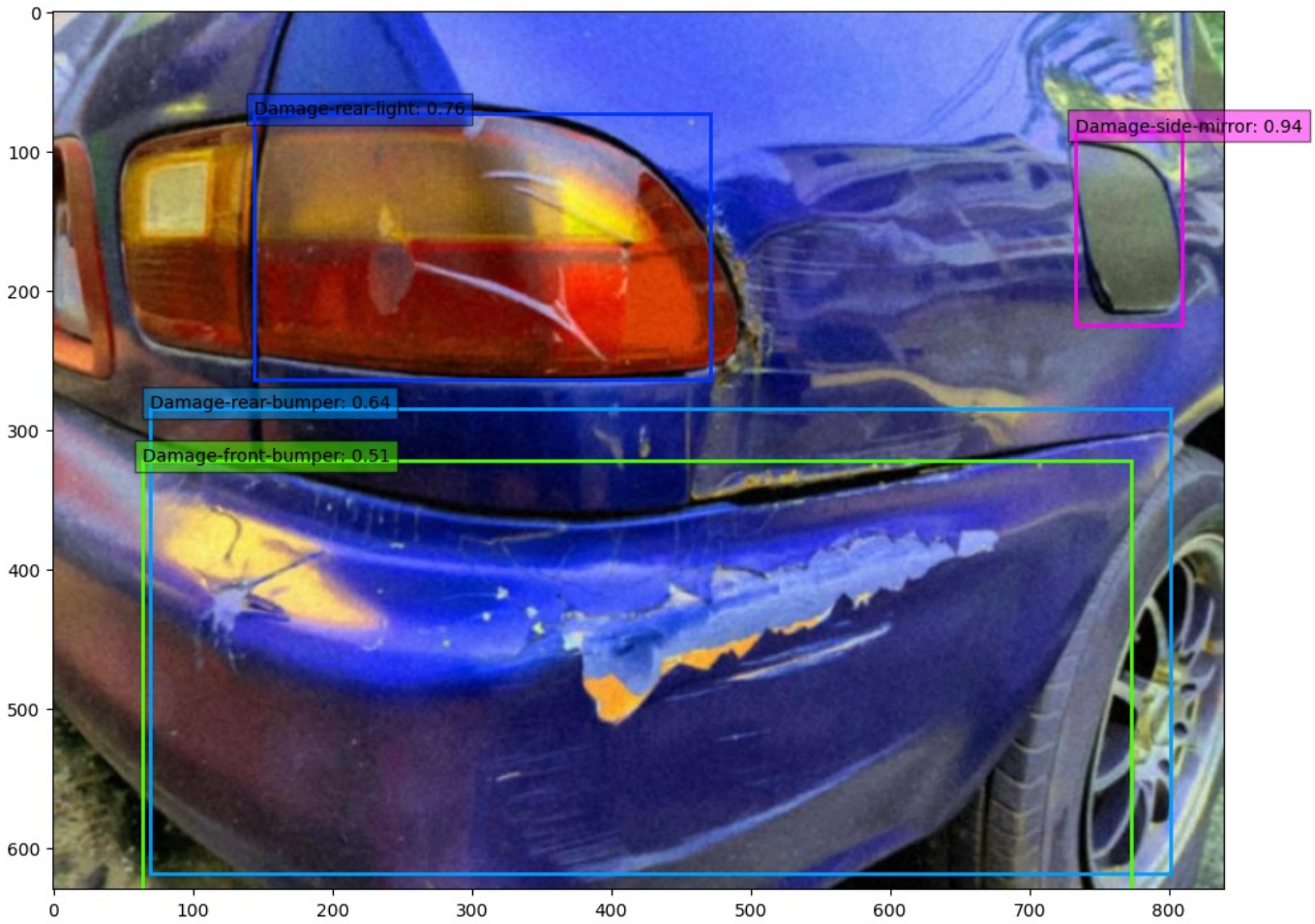

Damage-boots: 0.89
Damage-rear-bumper: 0.88
PAKWHEELS.COM


Damage-boots: 0.81
Damage-boots: 0.61
Damage-side-mirror: 0.87
0
100
200
300
400
500
600

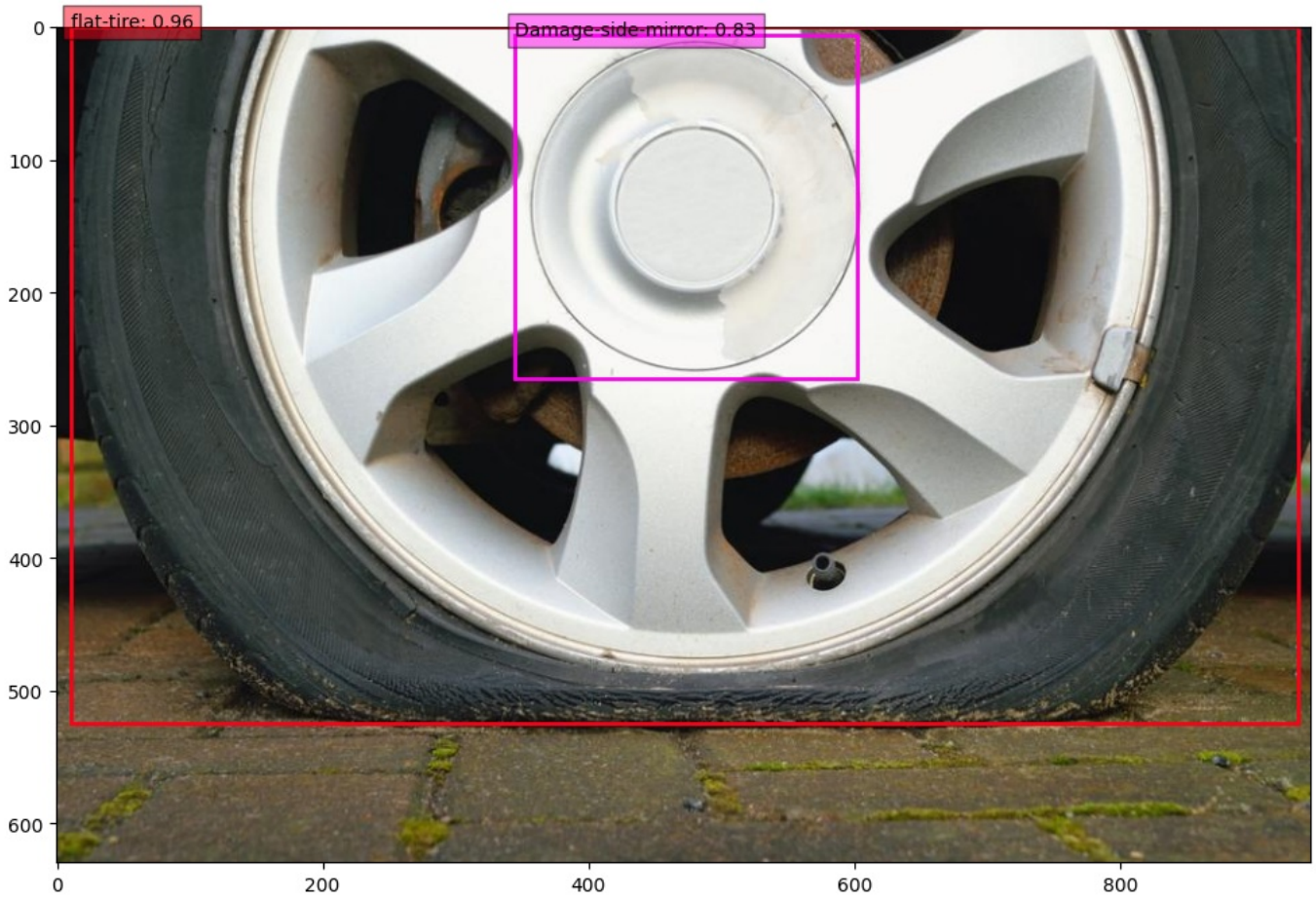

Damage-Windshield: 0.99
0
100
200
300
400
500
600
0
200
400
600
800

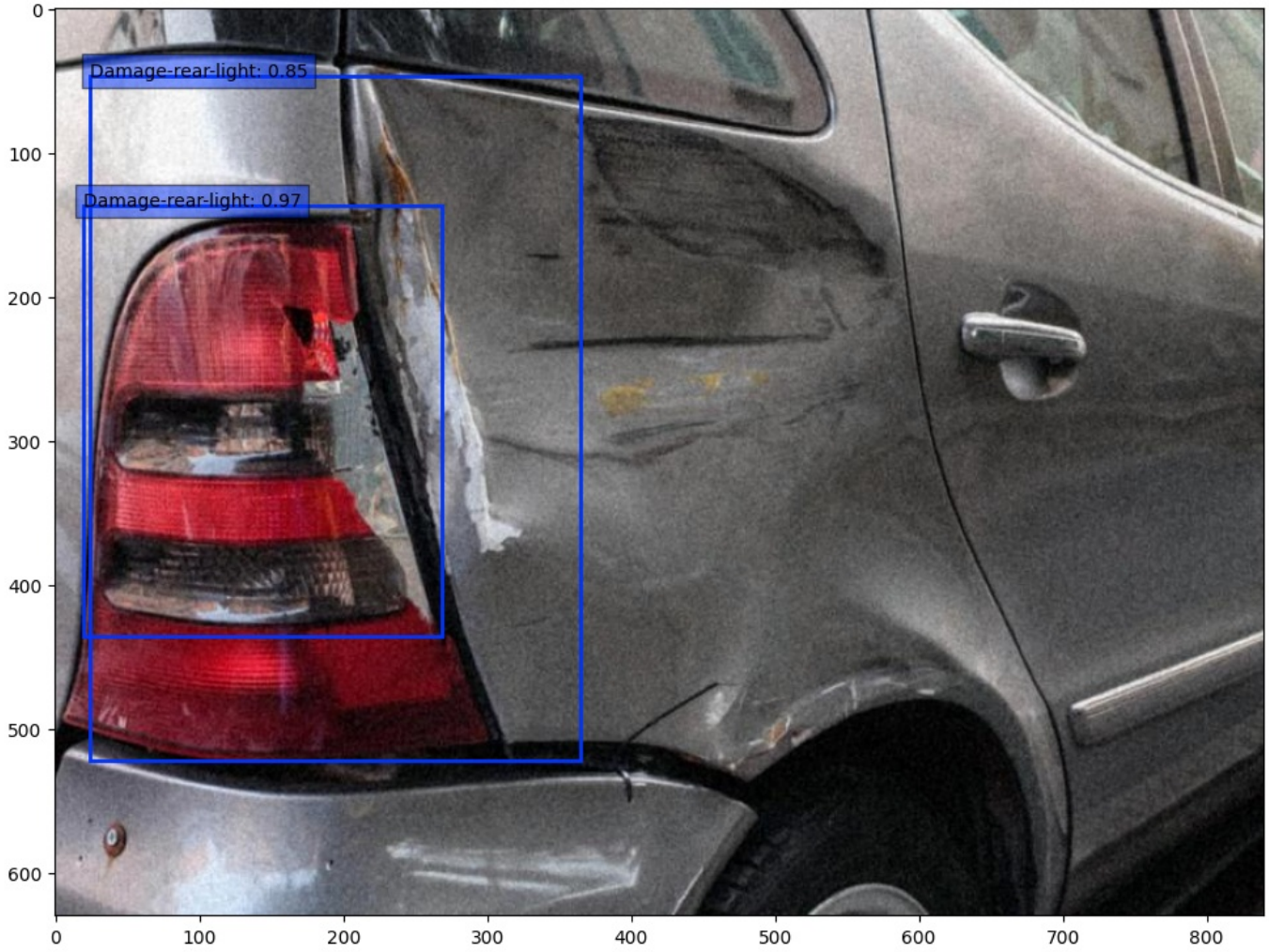

Damage-right-fender: 0.65
gettyimages
Credit: StanRohrer
182678140
0
100
200
300
400
500
600
0
200
400
600
800


Damage-headlight: 0.53
Damage-side-mirror: 0.59
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

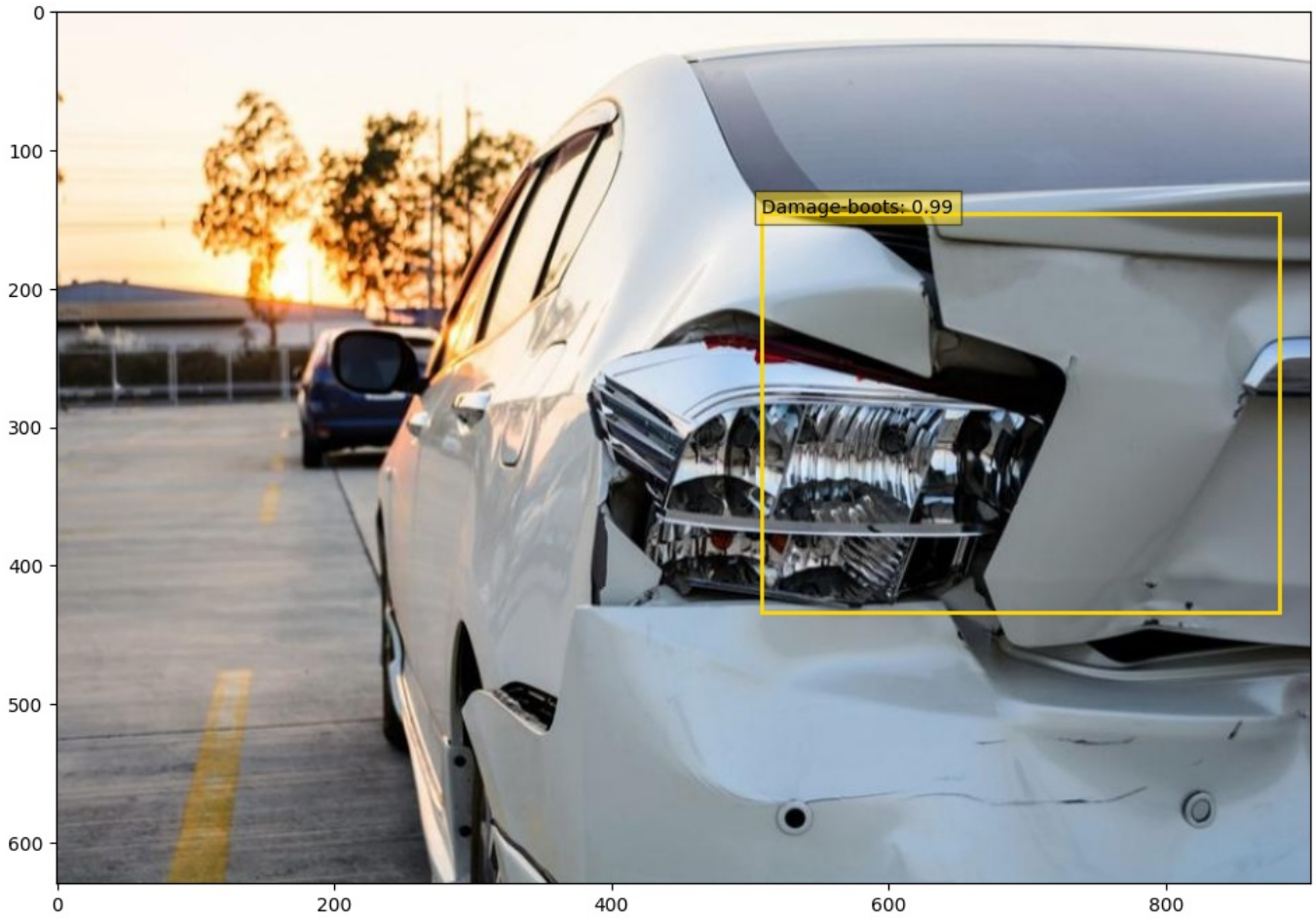

Damage-rear-light: 0.76
Damage-side-mirror: 0.94
Damage-rear-bumper: 0.64
Damage-front-bumper: 0.51
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-front-bumper: 0.69
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700


flat-tire: 0.96
Damage-side-mirror: 0.83
0
100
200
300
400
500
600
0
200
400
600
800


Damage-rear-light: 0.85
Damage-rear-light: 0.97
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-hood: 0.56
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-boots: 0.99
0
100
200
300
400
500
600
0
200
400
600
800


Damage-roof: 0.98
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700

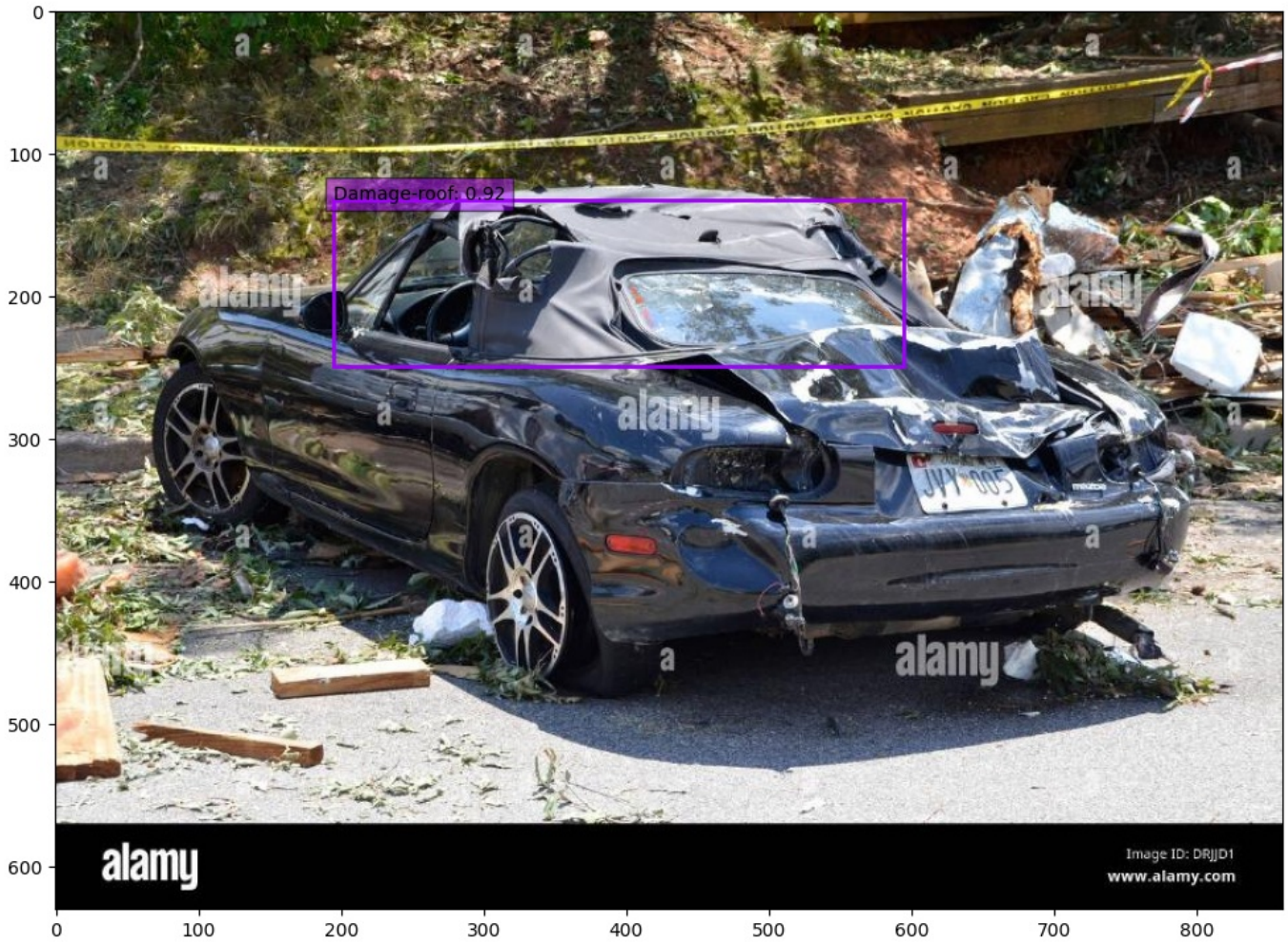

Damage-side-mirror: 0.99
shutterstock.com · 1219505215
0
100
200
300
400
500
600
0
200
400
600
800


Damage-side-mirror: 1.00
0
100
200
300
400
500
600
0
200
400
600
800

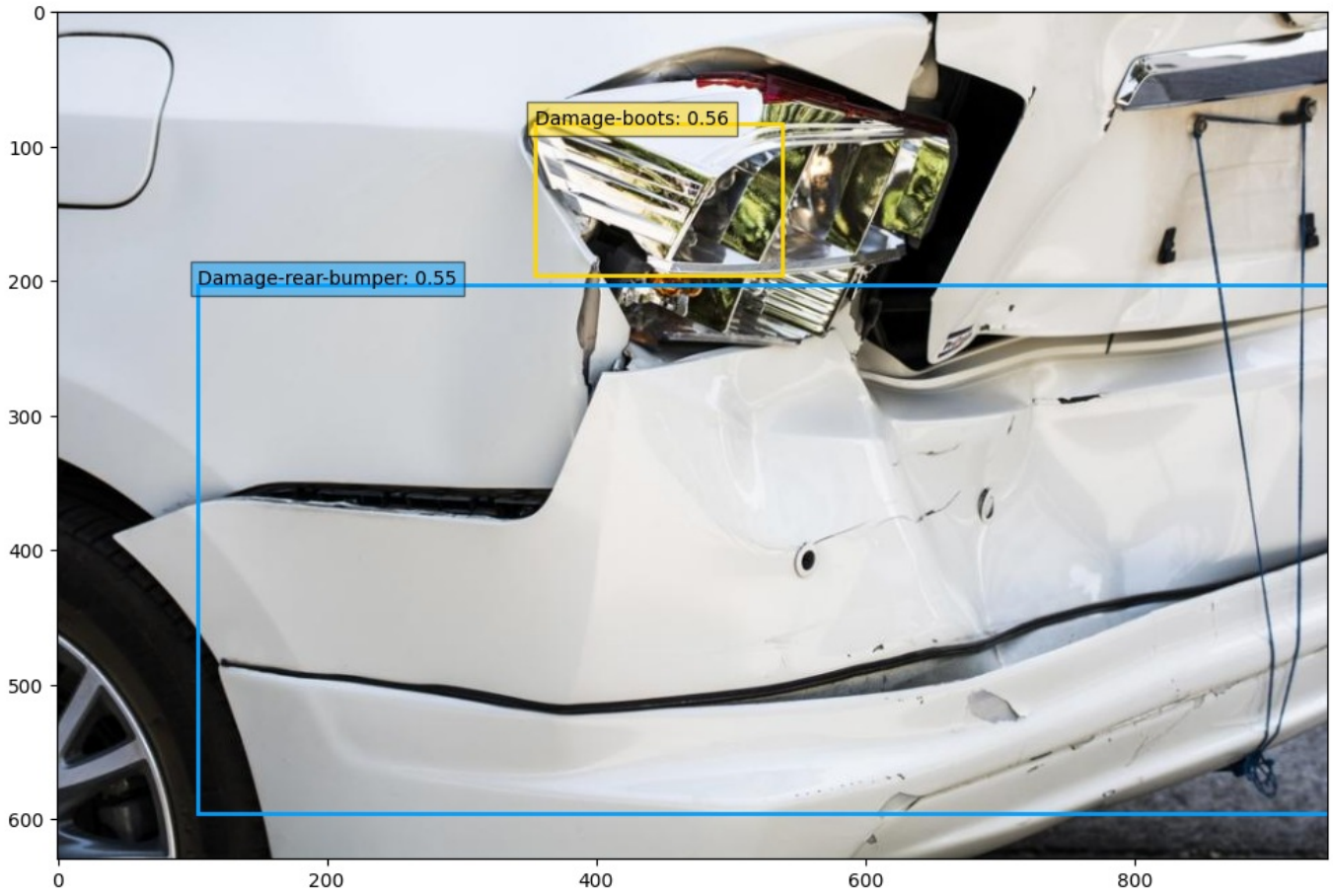

Damage-rear-bumper: 0.51
0
100
200
300
400
500
600
0
200
400
600
800

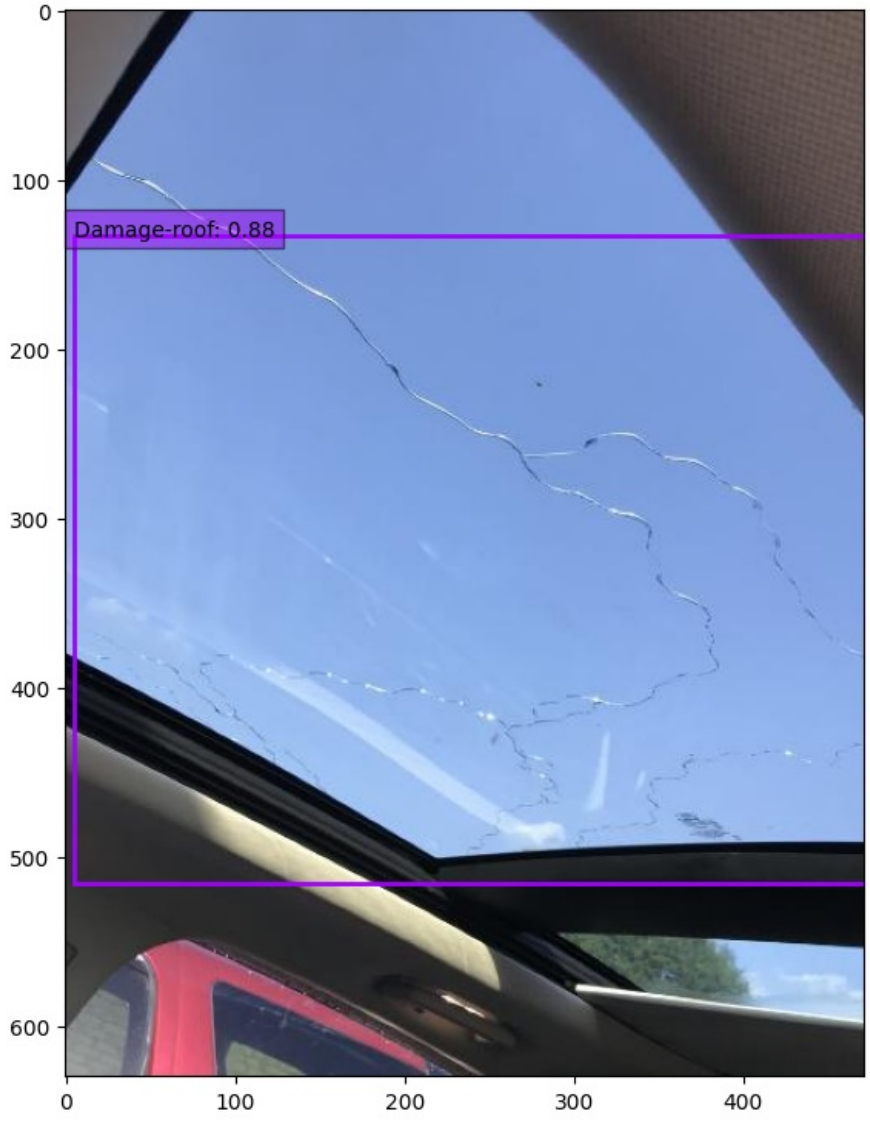

Damage-left-fender: 0.96
Damage-left-fender: 0.71
flat-tire: 0.53
0
100
200
300
400
500
600
0
200
400
600
800


Damage-Windshield: 0.91
0
100
200
300
400
500
600
0
200
400
600
800


Damage-roof: 0.92
alamy
Image ID: DRJJD1
www.alamy.com

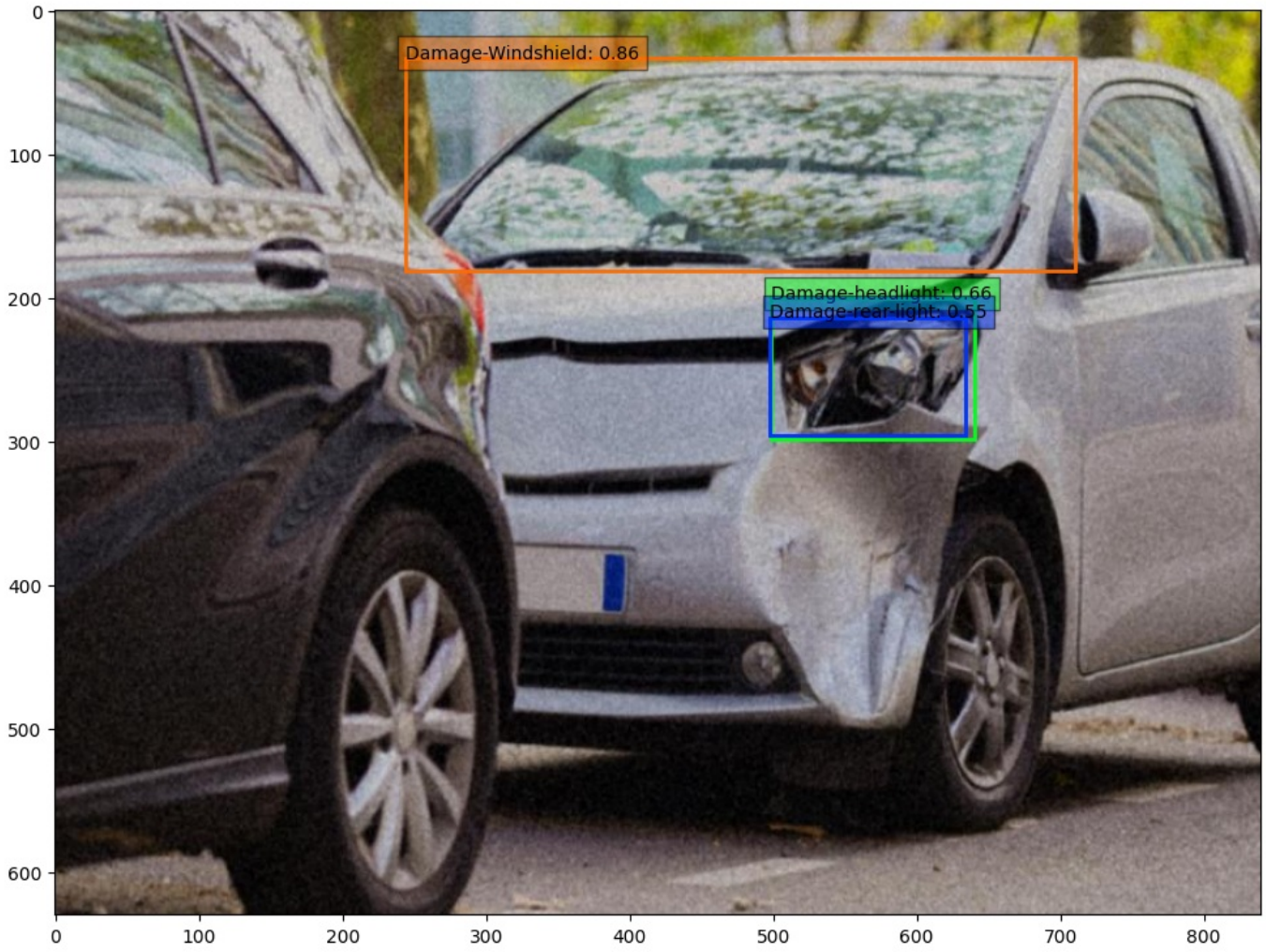

Damage-rear-light: 0.82

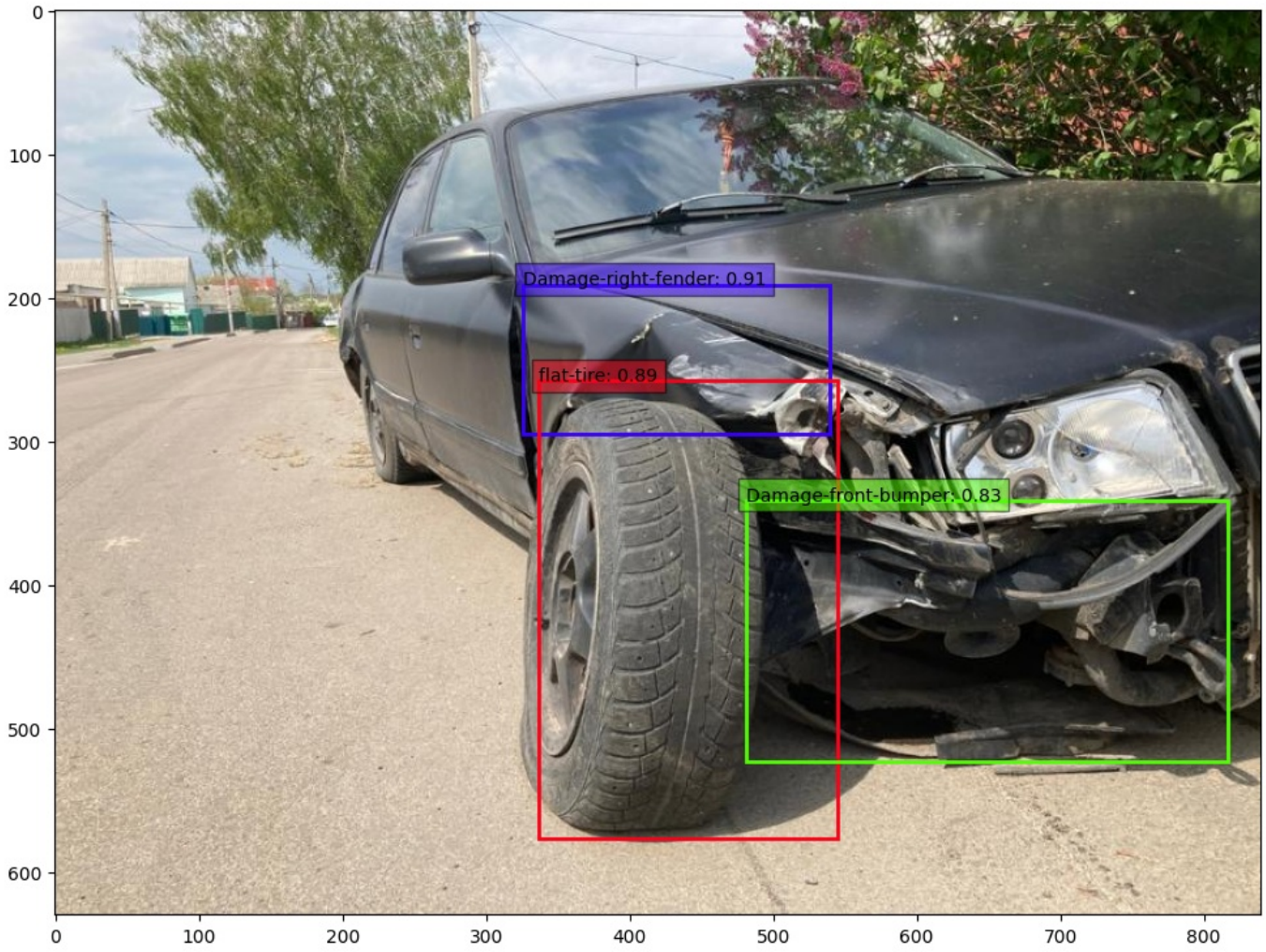

Damage-boots: 0.56
Damage-rear-bumper: 0.55
0
100
200
300
400
500
600
0
200
400
600
800


Damage-roof: 0.88
0
100
200
300
400
500
600
0
100
200
300
400

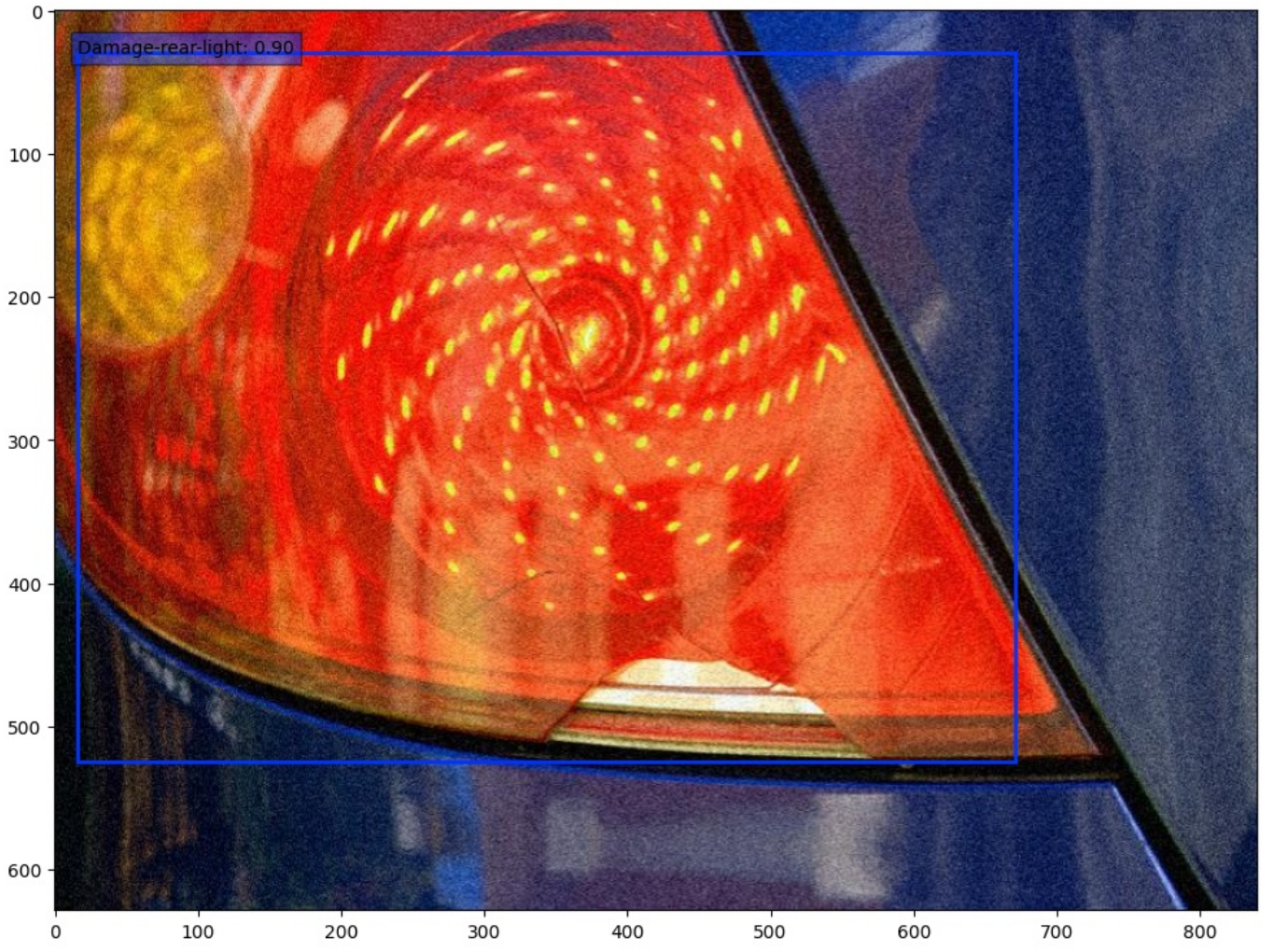

Damage-right-fender: 0.88
0
100
200
300
400
500
600
0
200
400
600
800

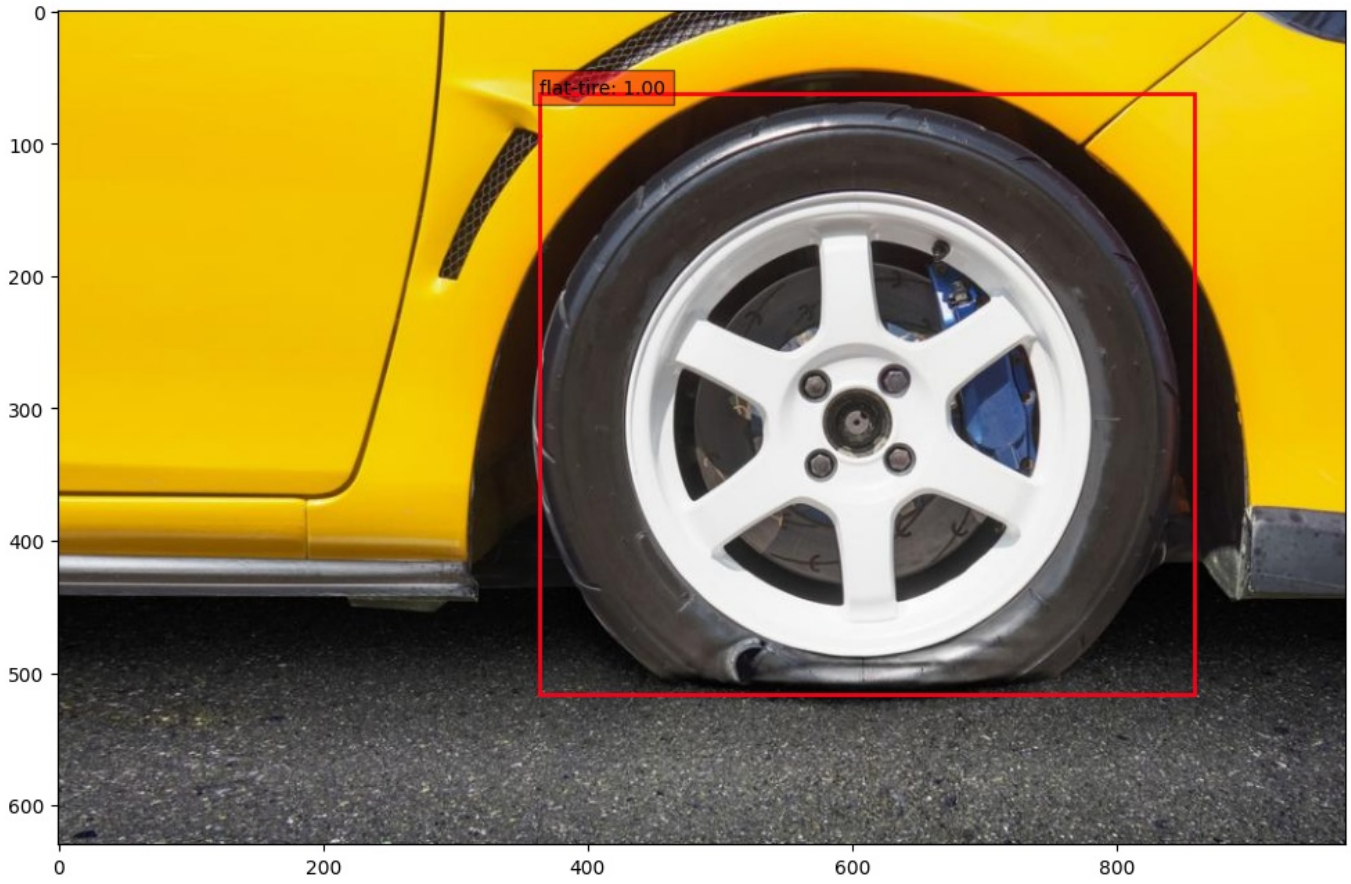

Damage-Windshield: 0.92
0
100
200
300
400
500
600
0
200
400
600
800


Damage-Windshield: 0.86
Damage-headlight: 0.66
Damage-rear-light: 0.55
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-right-fender: 0.91
flat-tire: 0.89
Damage-front-bumper: 0.83
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-Windshield: 0.86
0
100
200
300
400
500
600
0
200
400
600
800

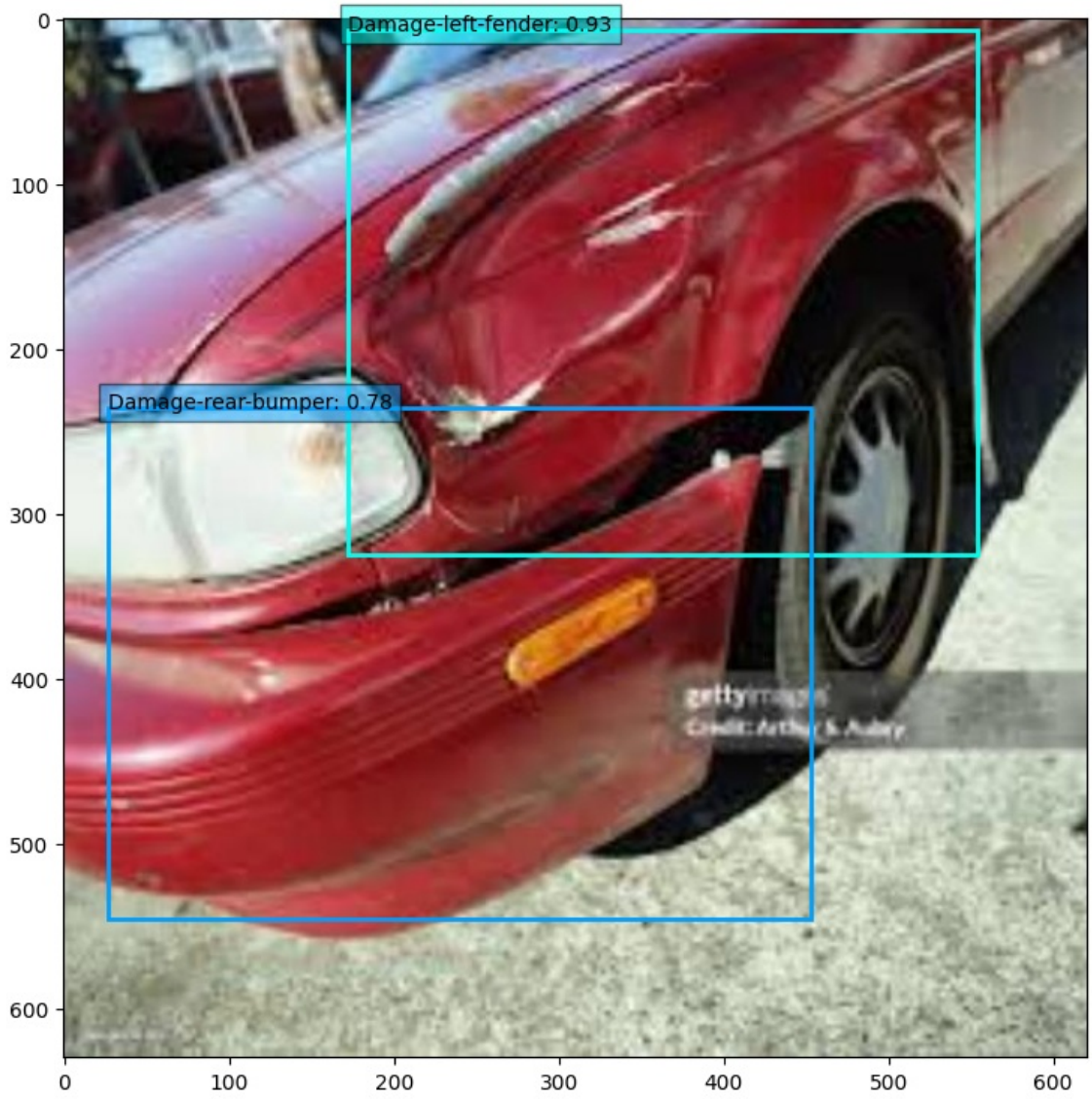

Damage-rear-light: 0.90
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


Damage-Windshield: 0.76
0
100
200
300
400
500
0
200
400
600
800

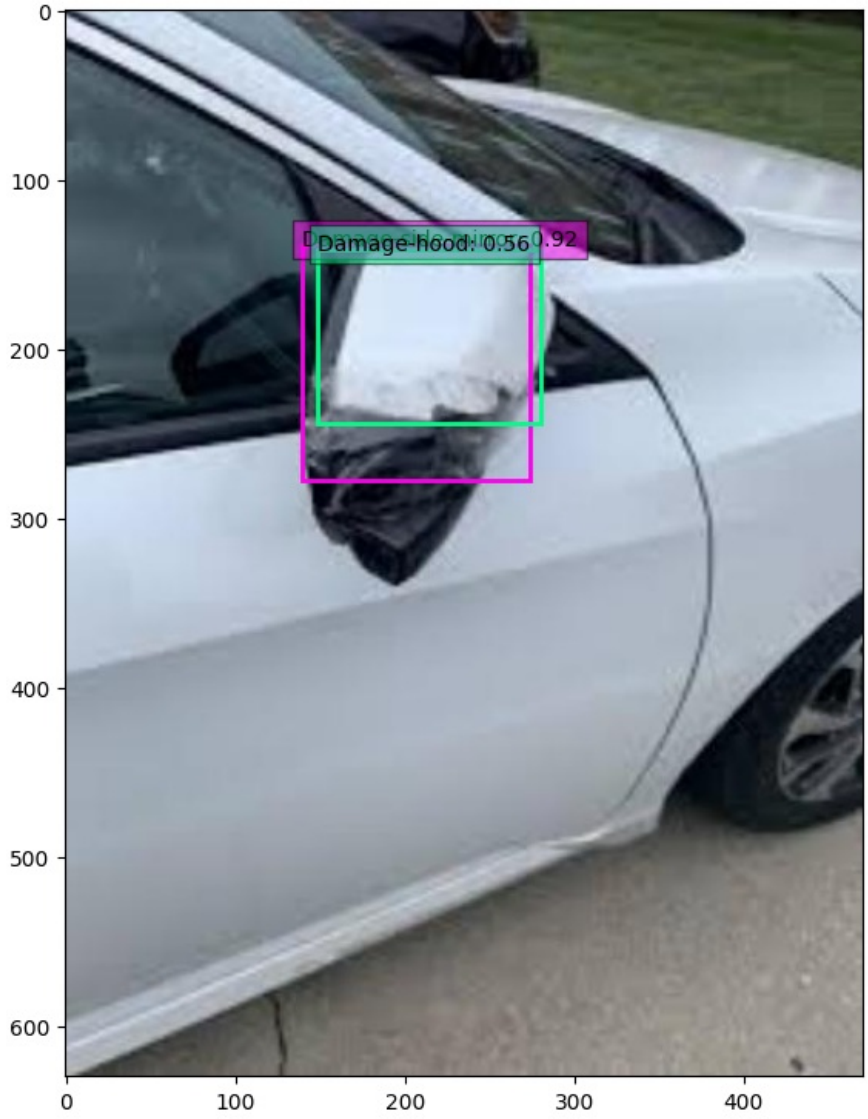

Damage-front-bumper: 0.90
0
100
200
300
400
500
600
0
200
400
600
800

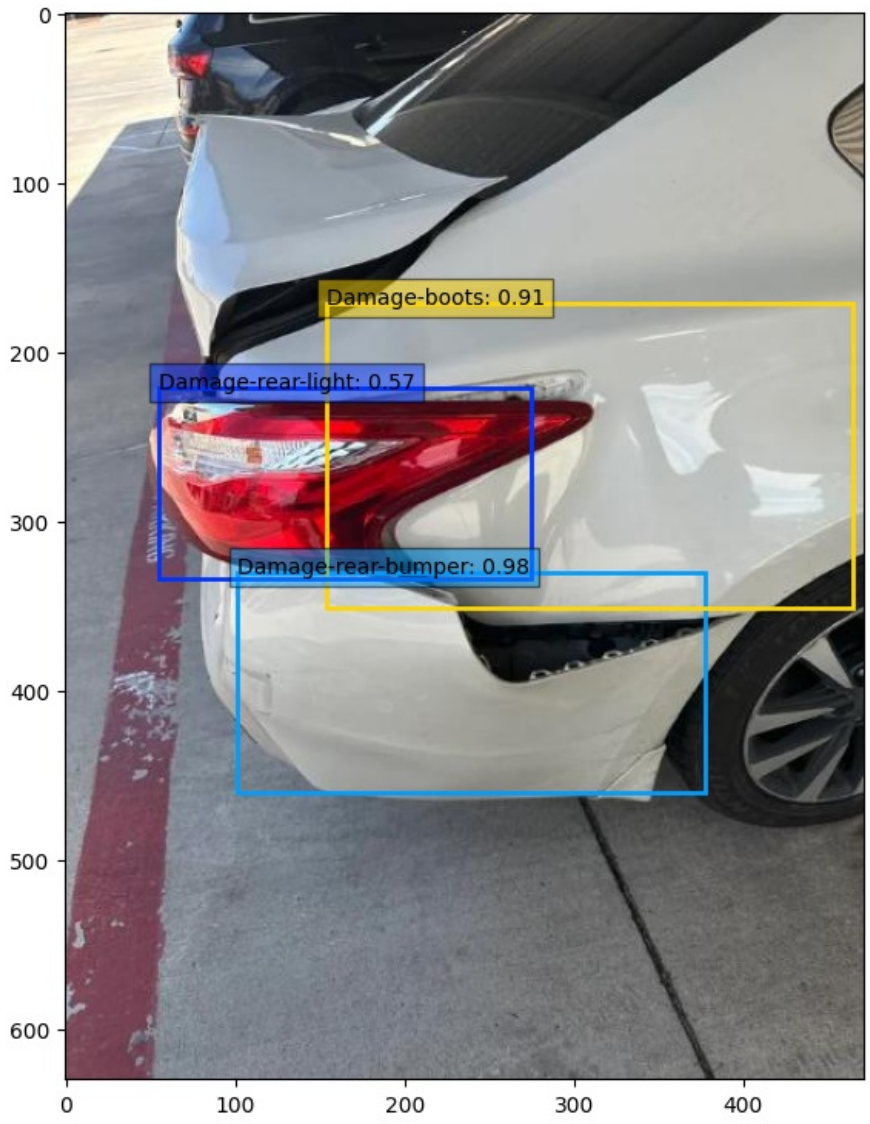

Damage-roof: 0.58
0
100
200
300
400
500
600
0
200
400
600
800

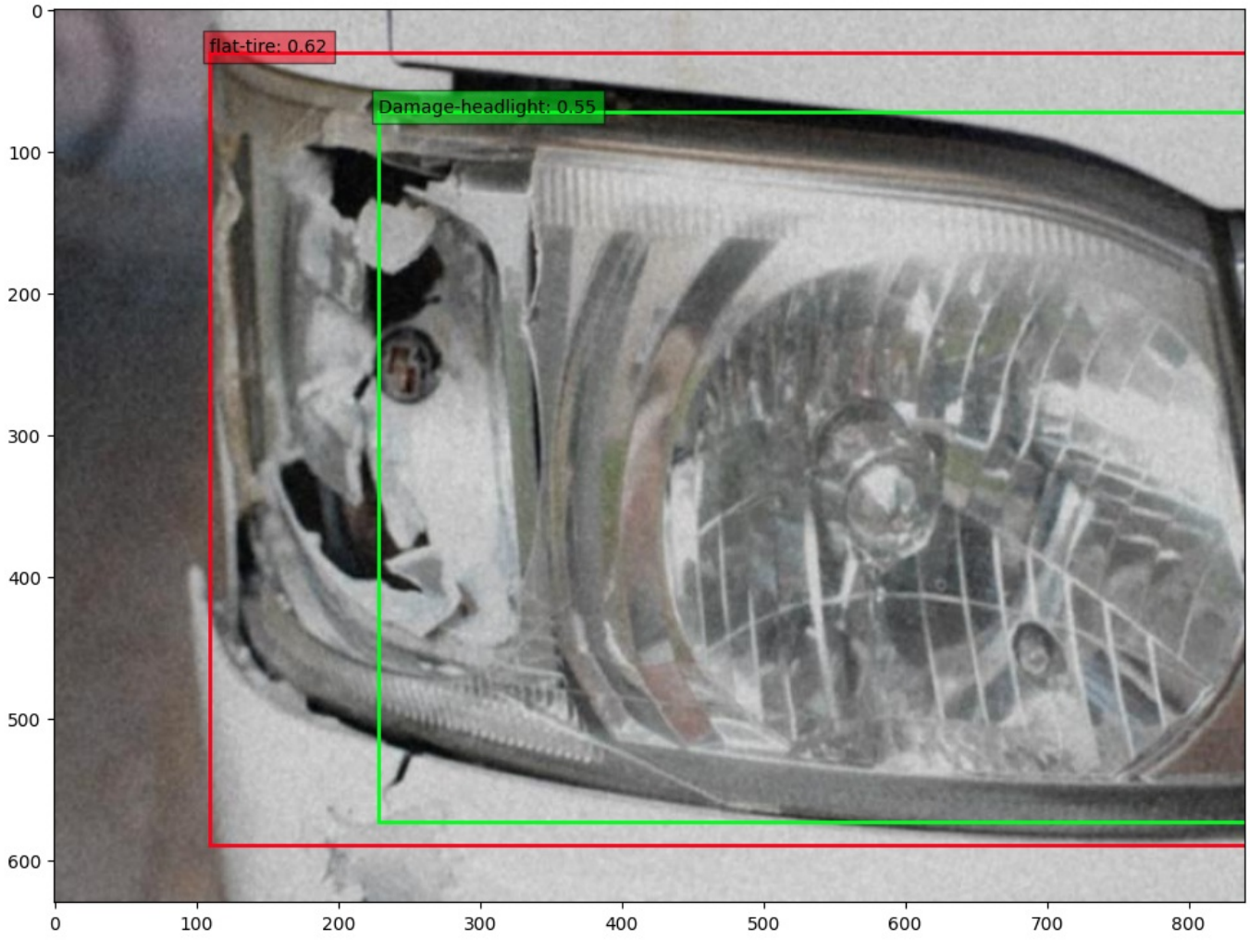

Damage-left-fender: 0.93
Damage-rear-bumper: 0.78
0
100
200
300
400
500
600


Damage-Windshield: 0.73
2596-17
0
100
200
300
400
500
600
800

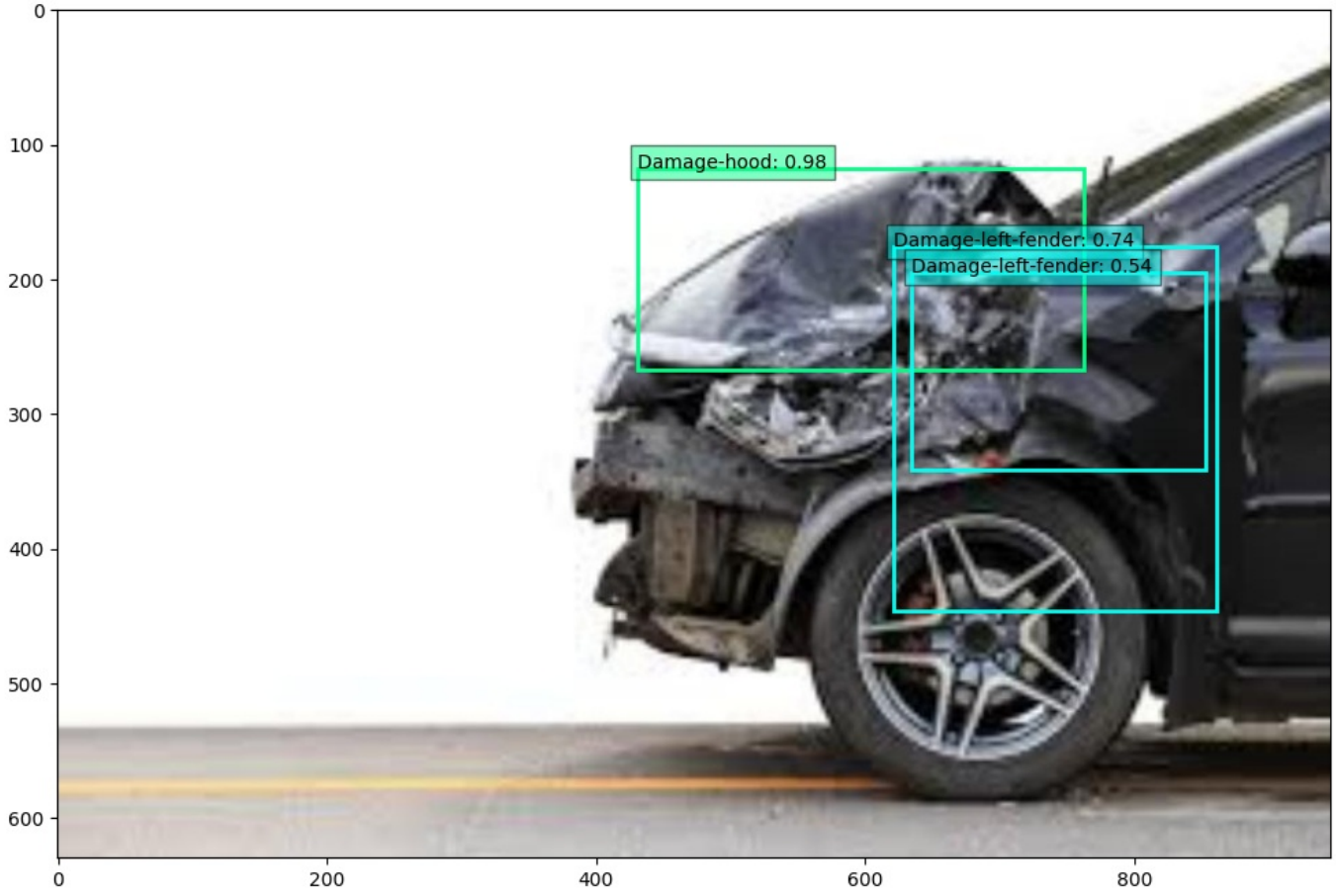

Damage-boots: 0.73
0
100
200
300
400
500
600
0
200
400
600
800

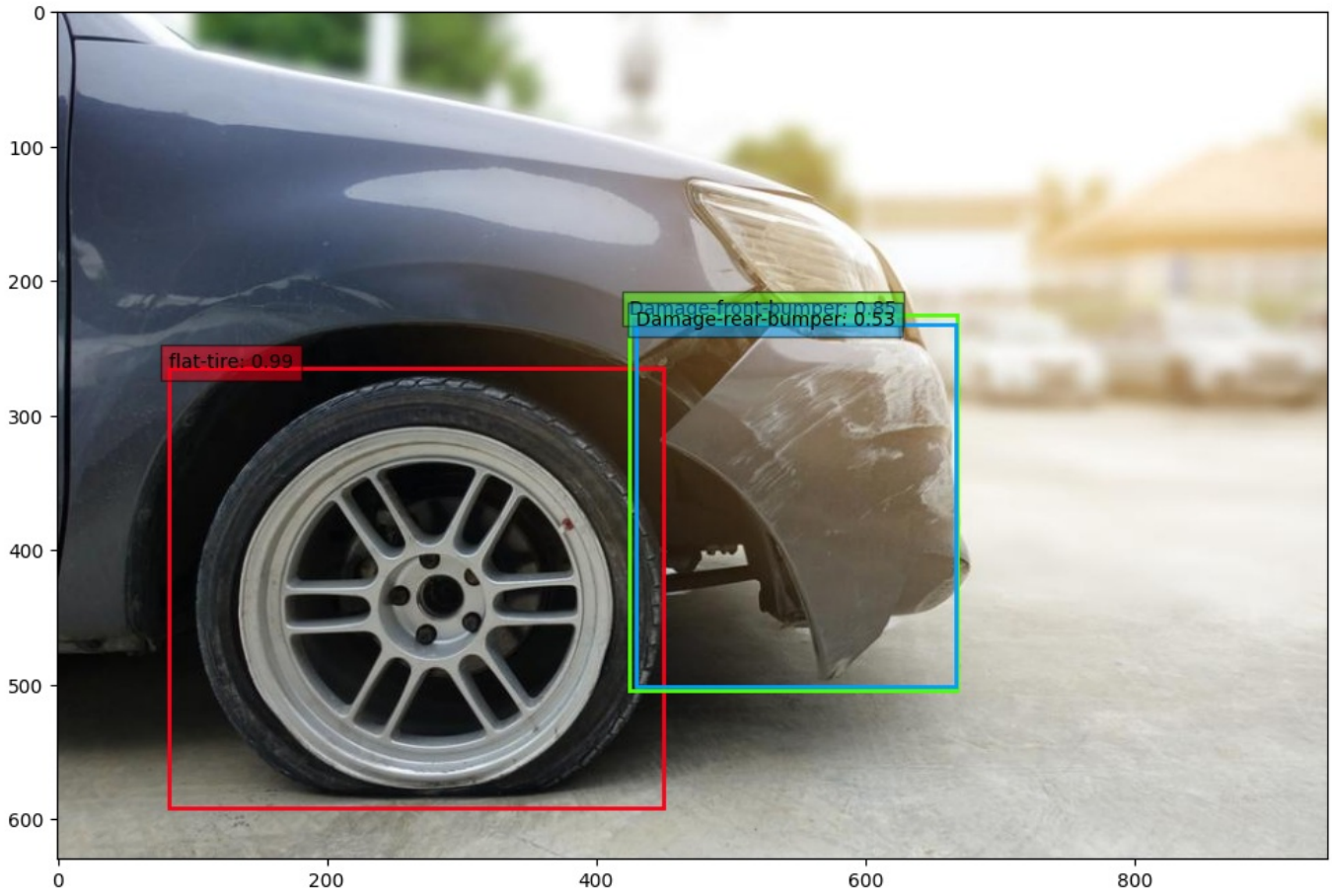

Damage-hood: 0.56
0
100
200
300
400
500
600
0
100
200
300
400

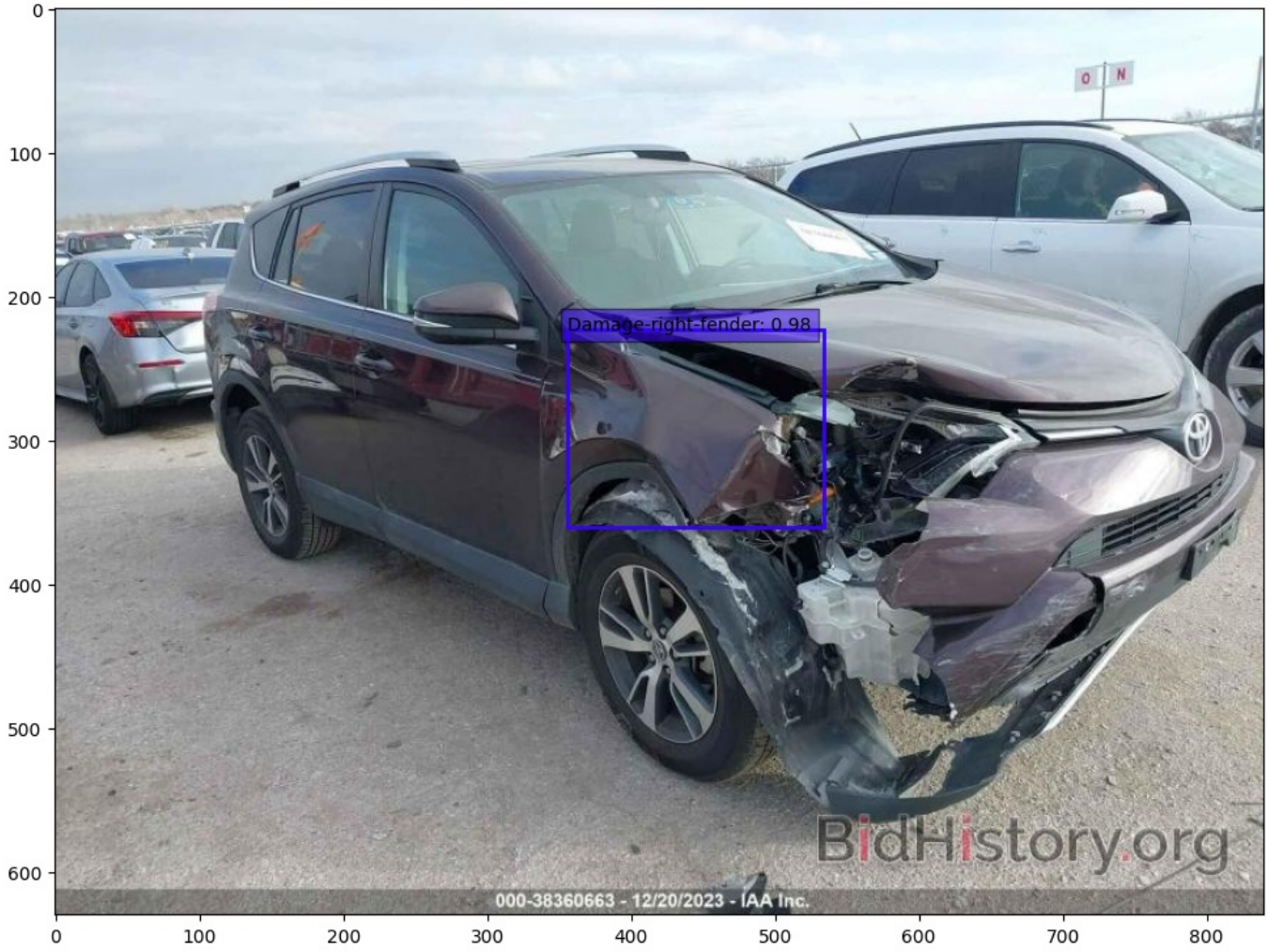

Damage-boots: 0.91
Damage-rear-light: 0.57
Damage-rear-bumper: 0.98
0
100
200
300
400
500
600

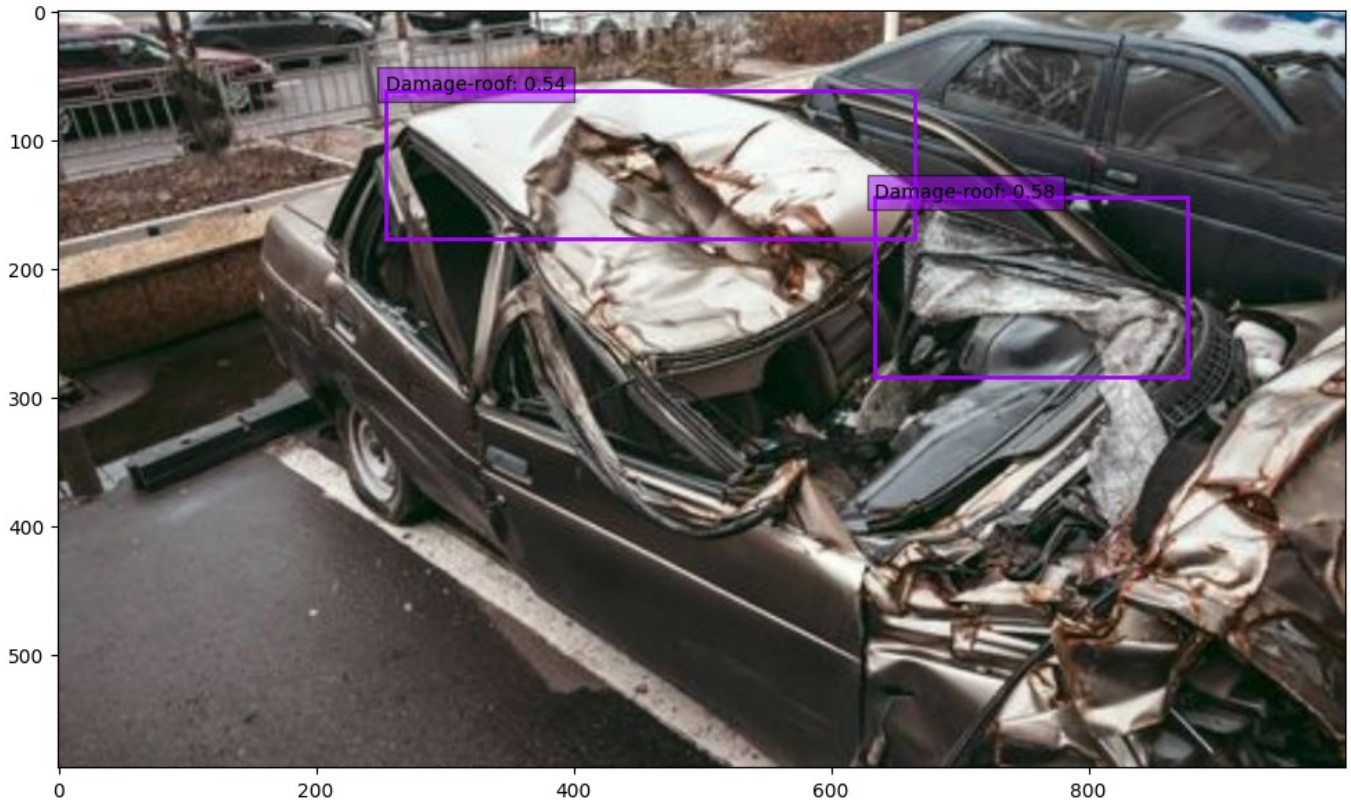

flat-tire: 0.62
Damage-headlight: 0.55
0
100
200
300
400
500
600
700
800

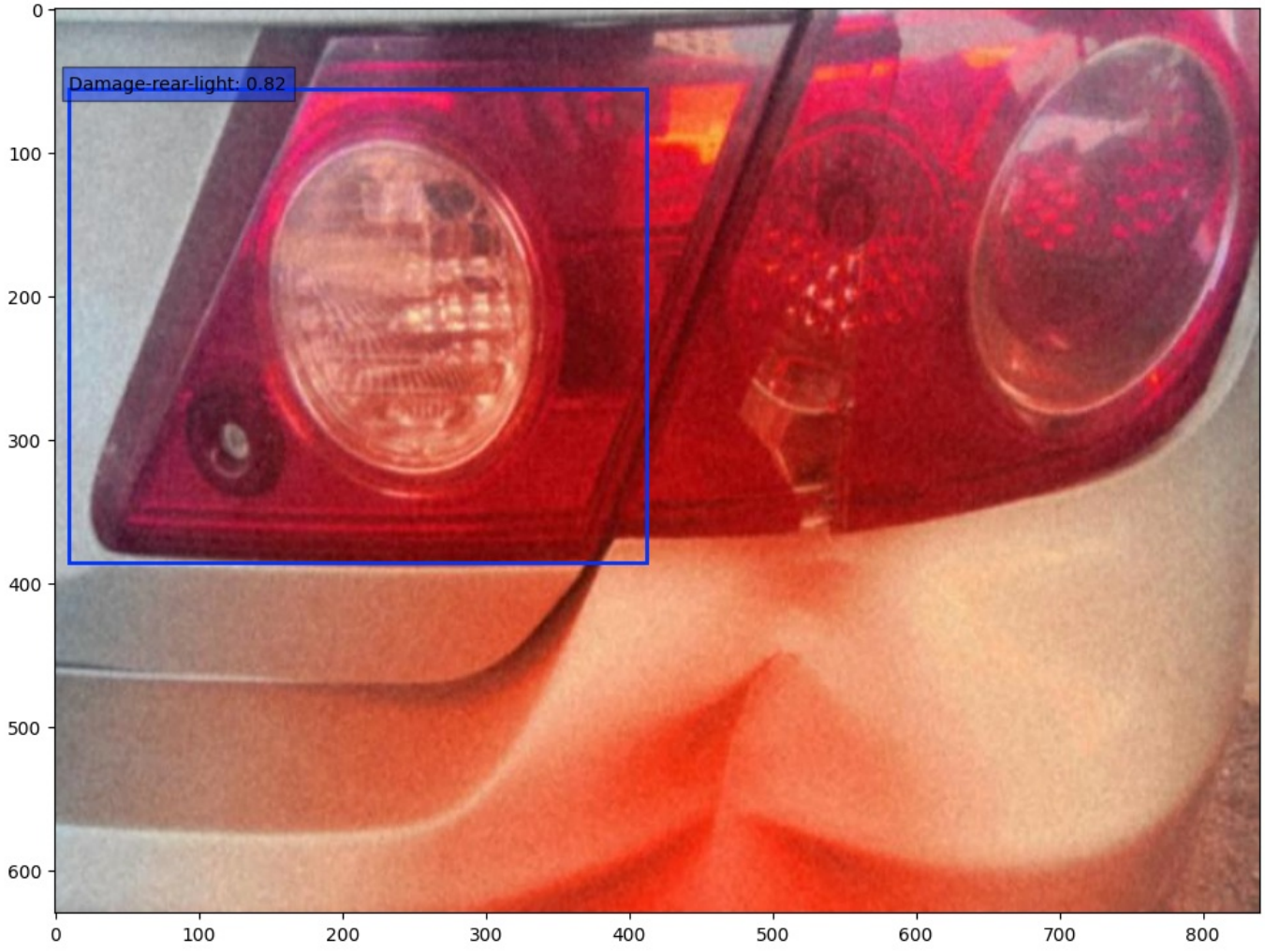

Damage-door: 0.83
0
100
200
300
400
500
600
0
200
400
600
800

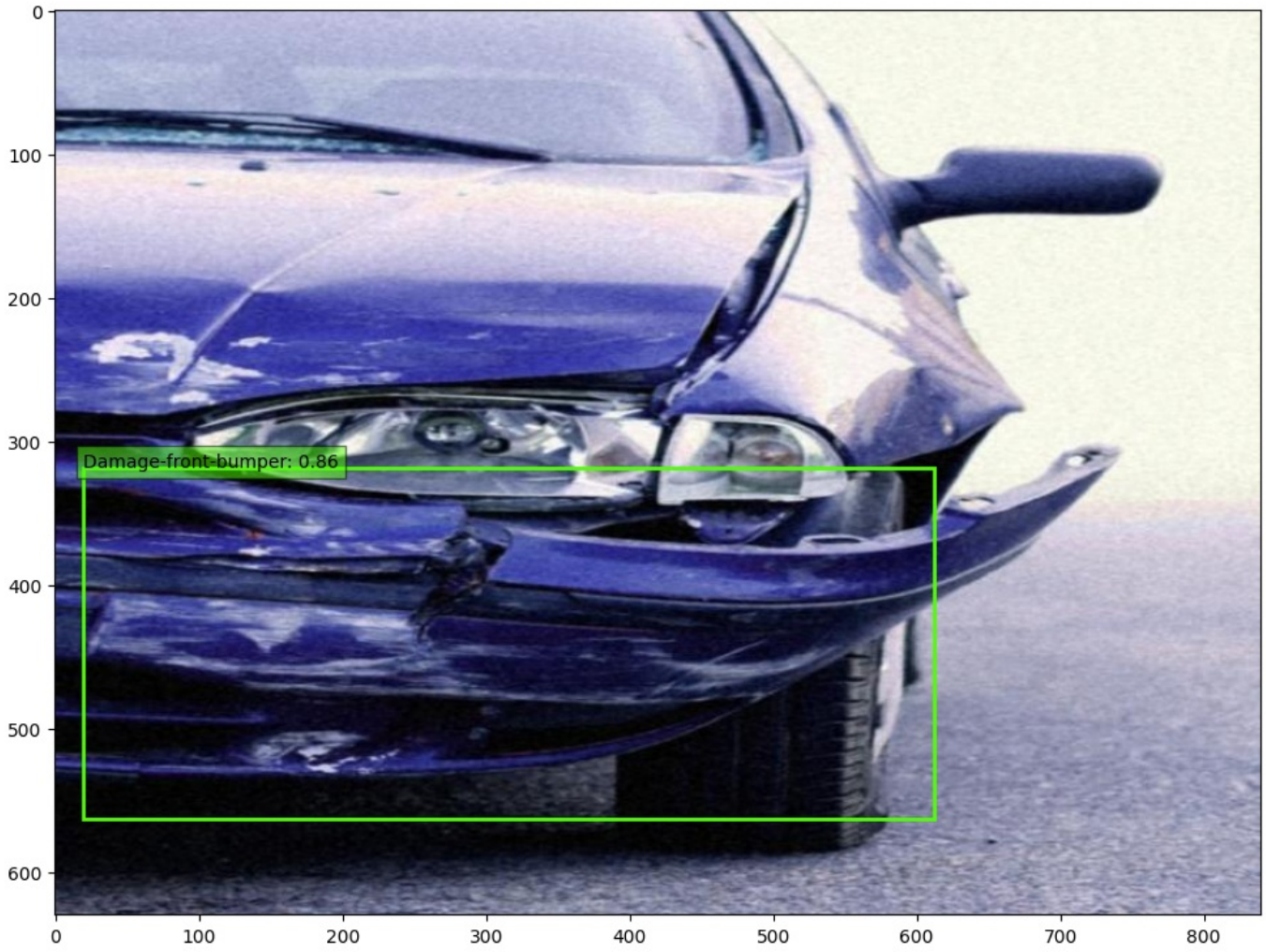

Damage-hood: 0.98
Damage-left-fender: 0.74
Damage-left-fender: 0.54
0
100
200
300
400
500
600
0
200
400
600
800


flat-tire: 0.99
Damage-front-bumper: 0.85
Damage-rear-bumper: 0.53

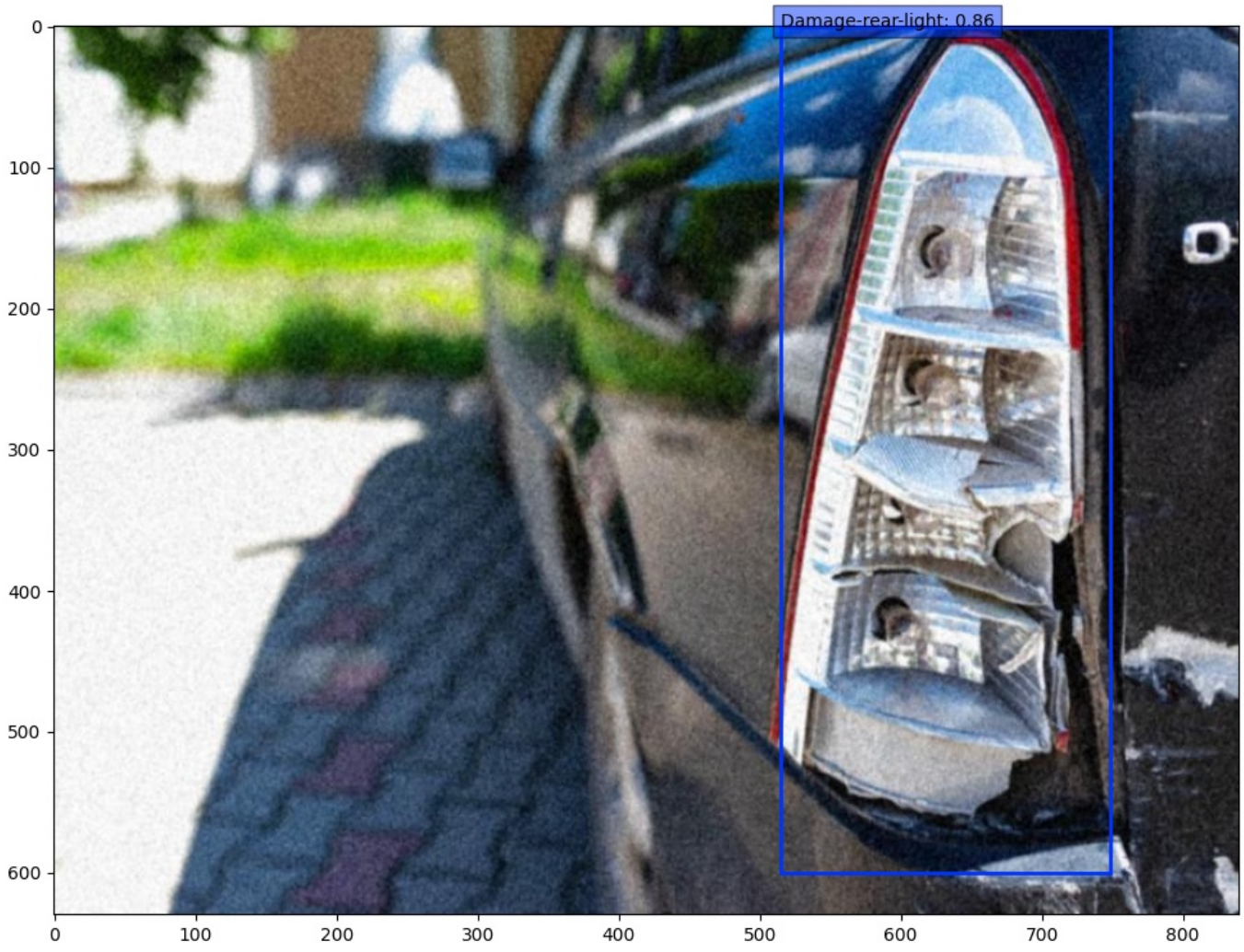

Damage-right-fender: 0.98
BidHistory.org
000-38360663 - 12/20/2023 - IAA Inc.


Damage-roof: 0.54
Damage-roof: 0.58
0
100
200
300
400
500
0
200
400
600
800


Damage-rear-light: 0.82
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

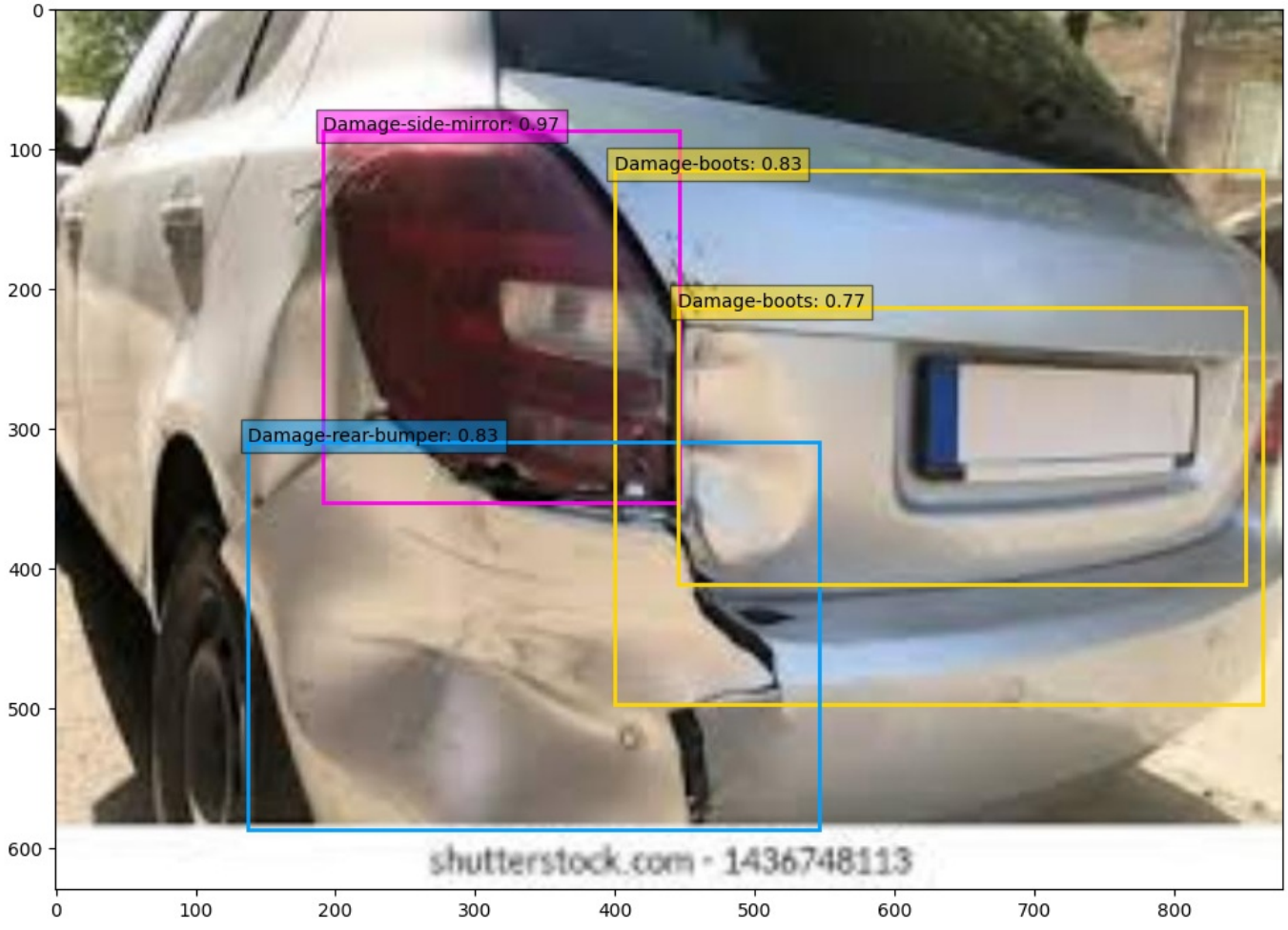

Damage-front-bumper: 0.86
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-boots: 0.57
Damage-rear-bumper: 0.82
shutterstock.com · 2326453211
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

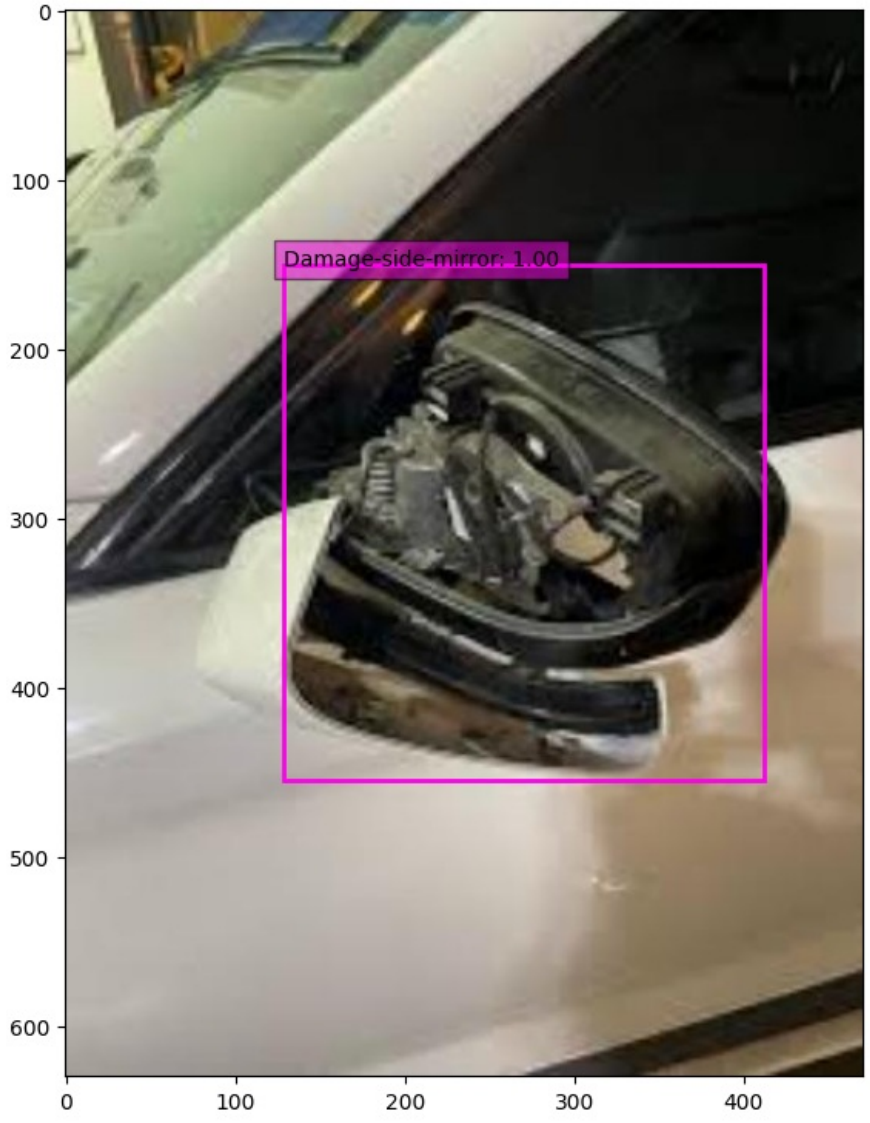

Damage-rear-light: 0.86
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

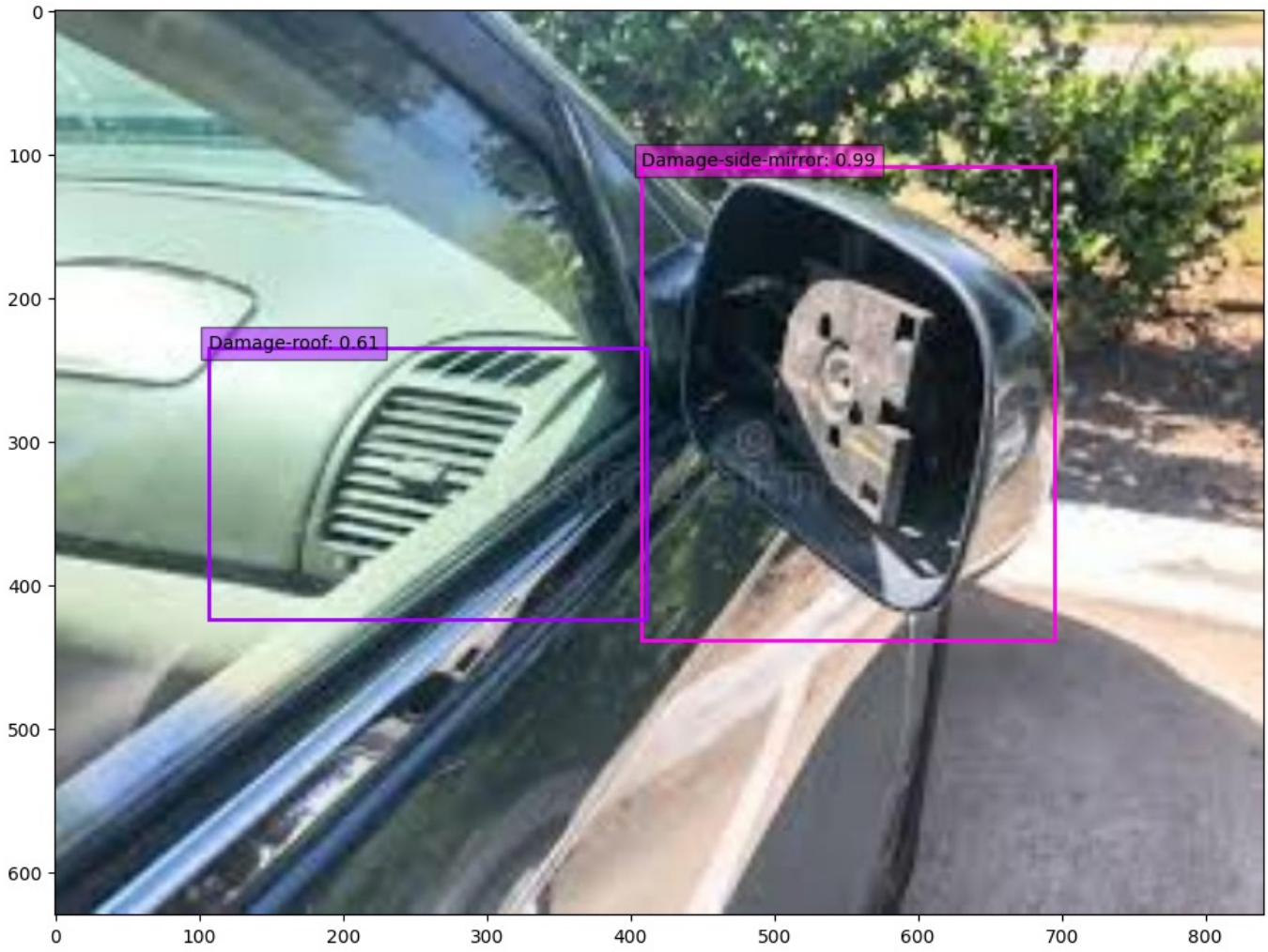

Damage-hood: 0.53
Damage-side-mirror: 0.72
Damage-side-mirror: 0.94
shutterstock.com · 2337867049
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


Damage-side-mirror: 0.97
Damage-boots: 0.83
Damage-boots: 0.77
Damage-rear-bumper: 0.83
shutterstock.com · 1436748113
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


Damage-side-mirror: 1.00
0
100
200
300
400
500
600
0
100
200
300
400

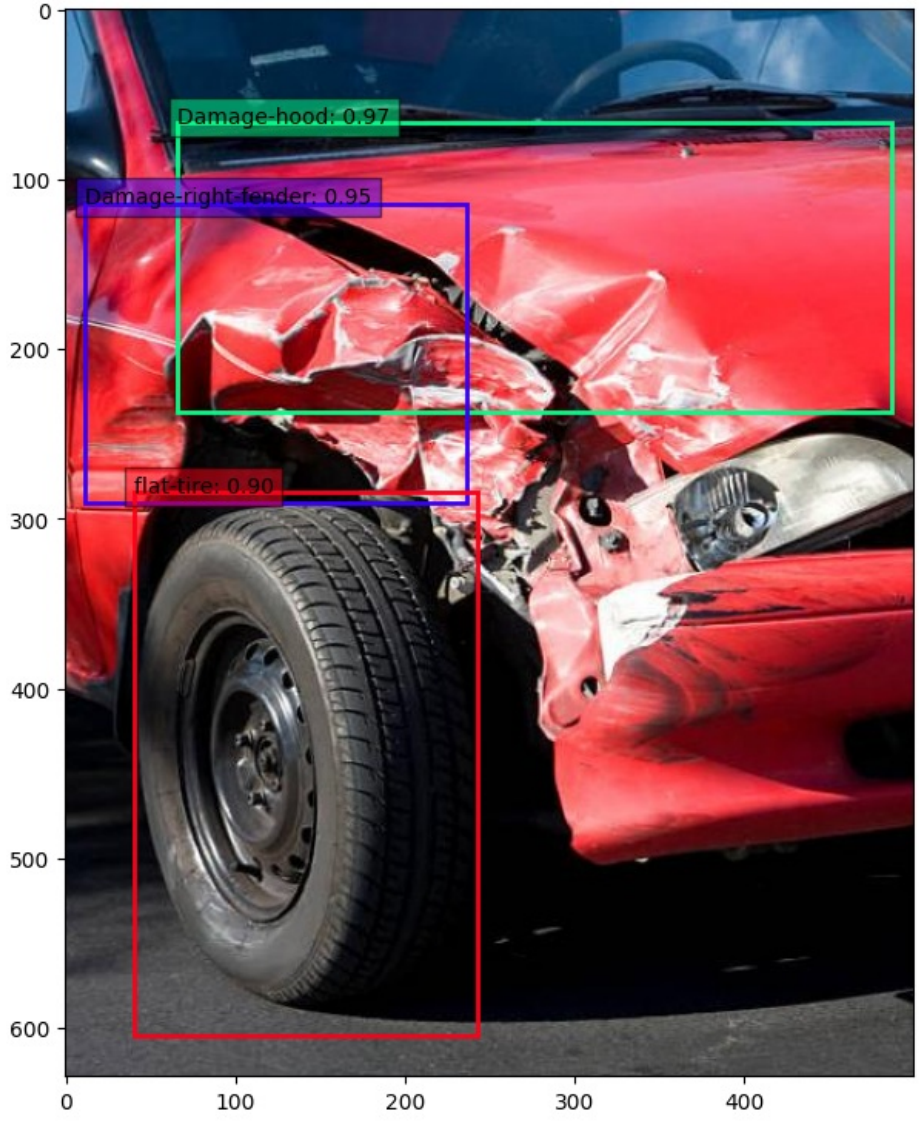

Damage-side-mirror: 0.99
Damage-roof: 0.61
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

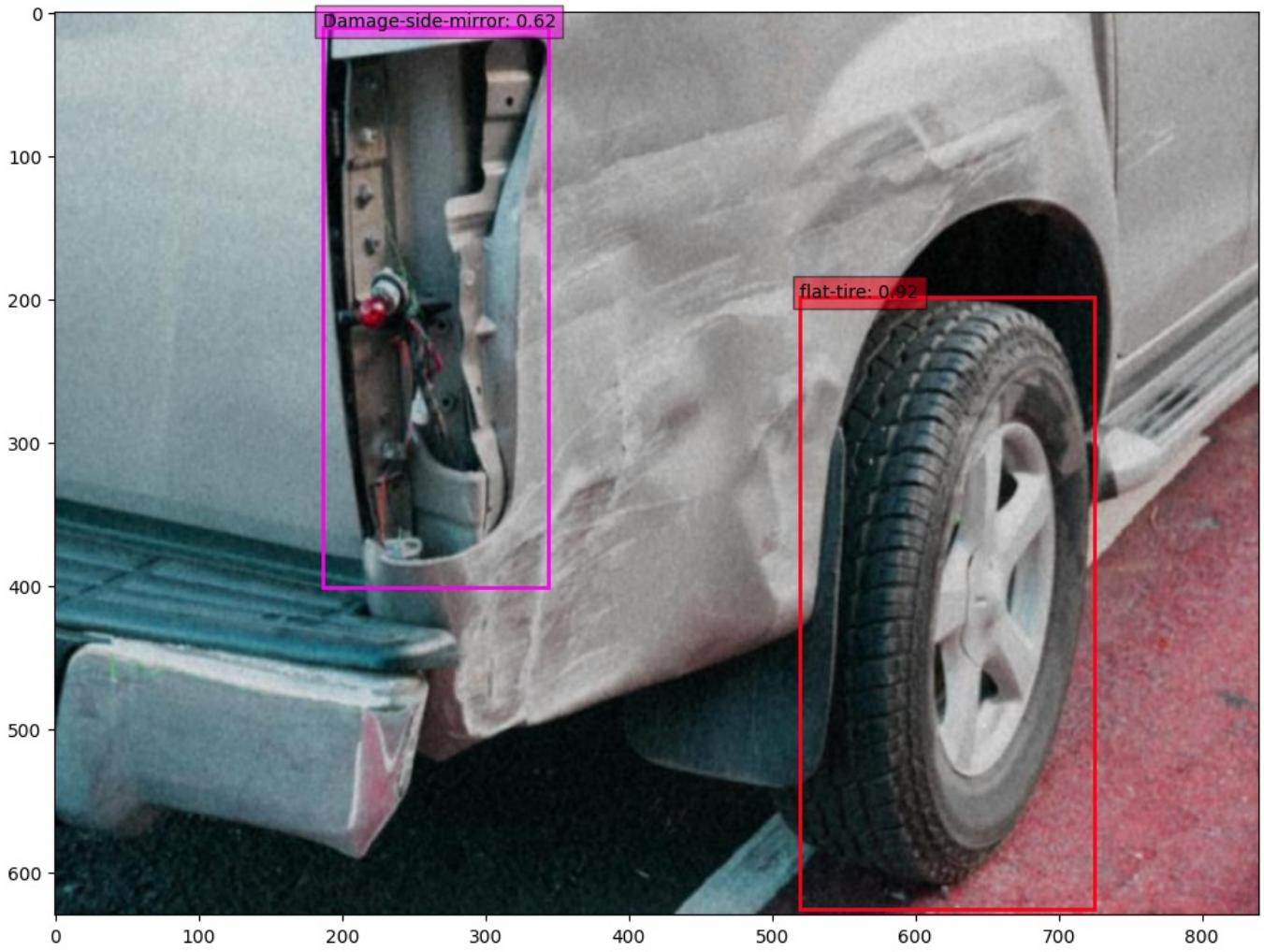

0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

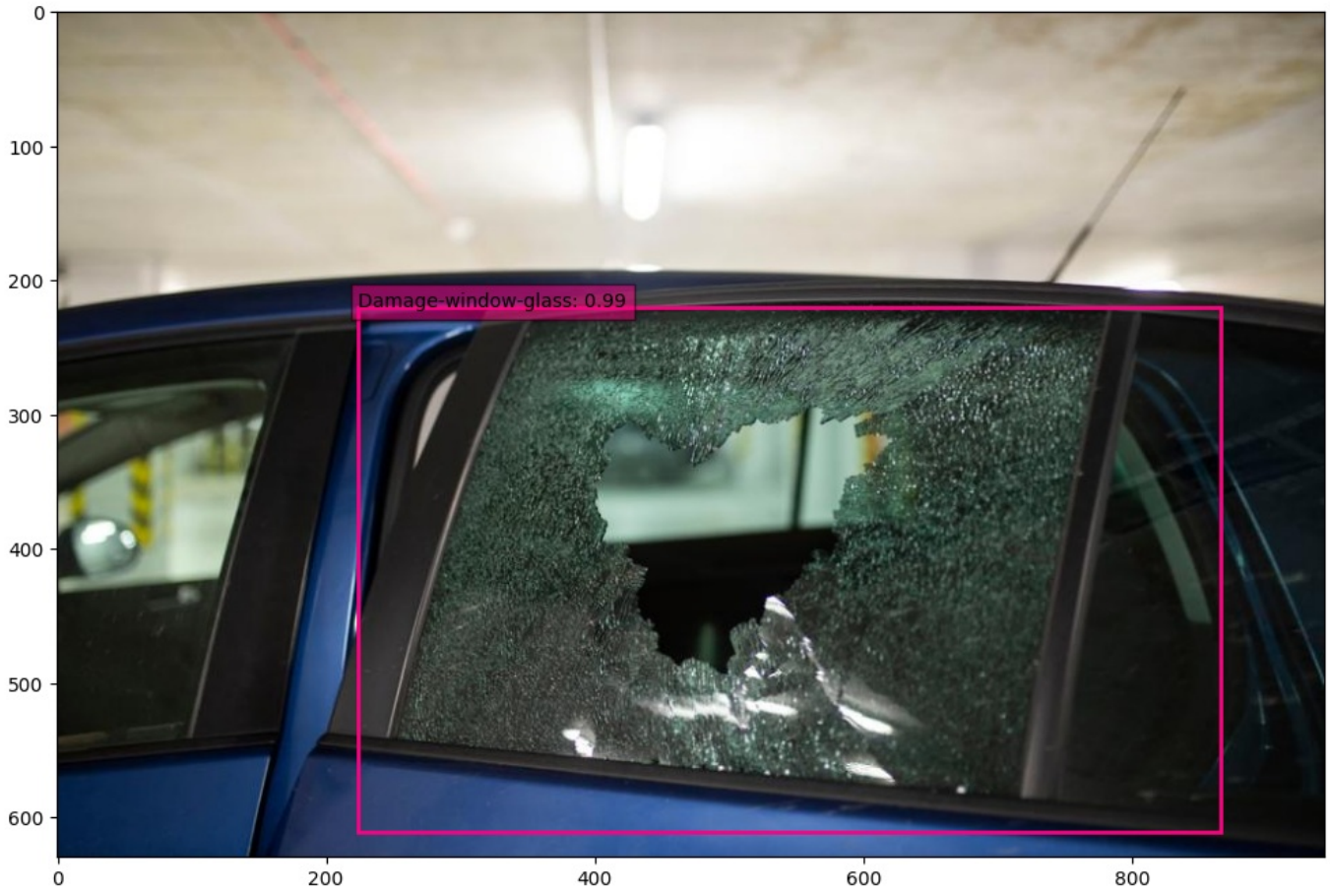

flat-tire: 1.00
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

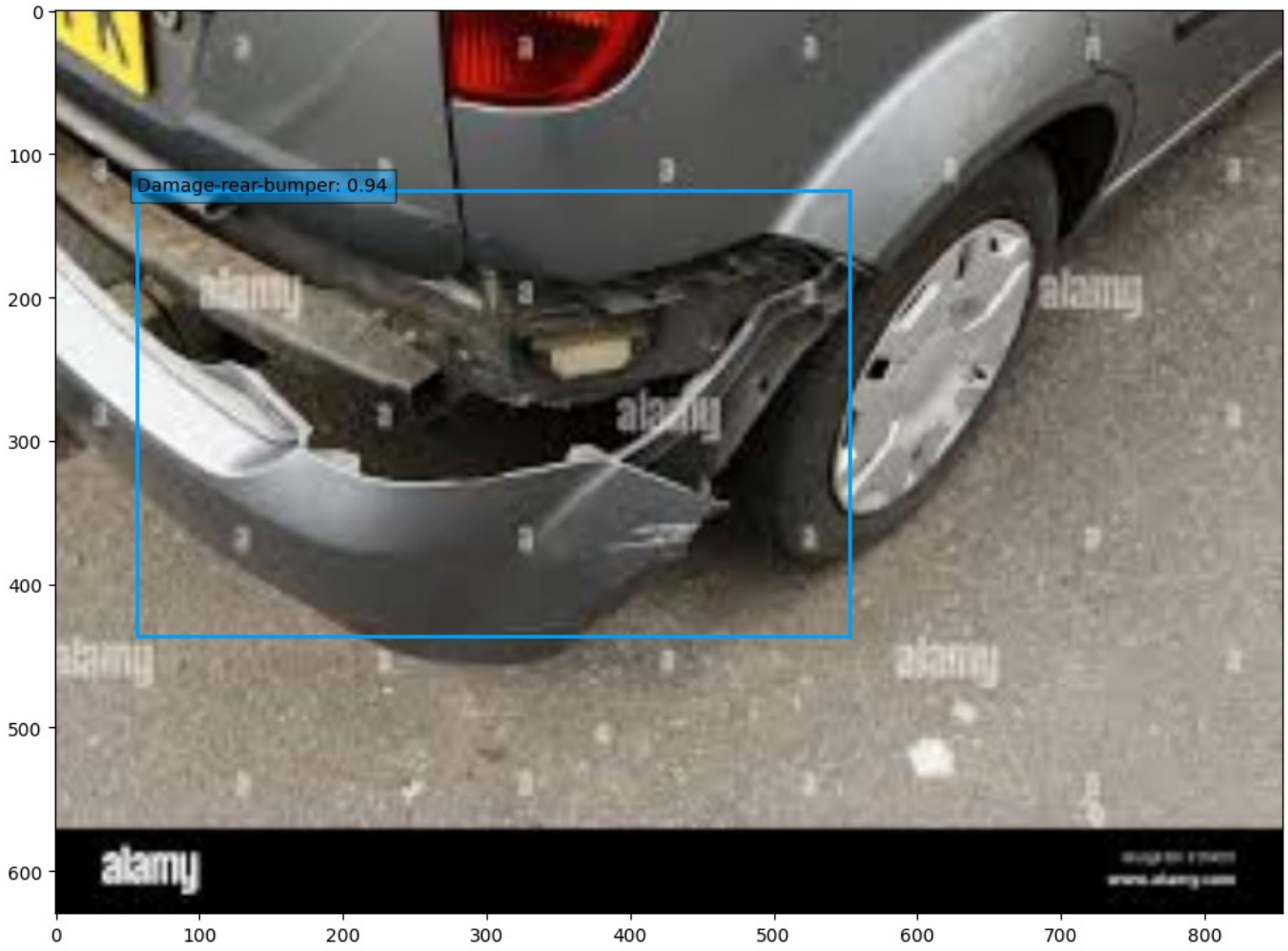

Damage-side-mirror: 0.76
Damage-side-mirror: 0.54
Licensable
shutterstock.com · 2357825167
0
100
200
300
400
500
600
0
200
400
600
800


Damage-front-bumper: 0.97
0
100
200
300
400
500
600
0
200
400
600
800

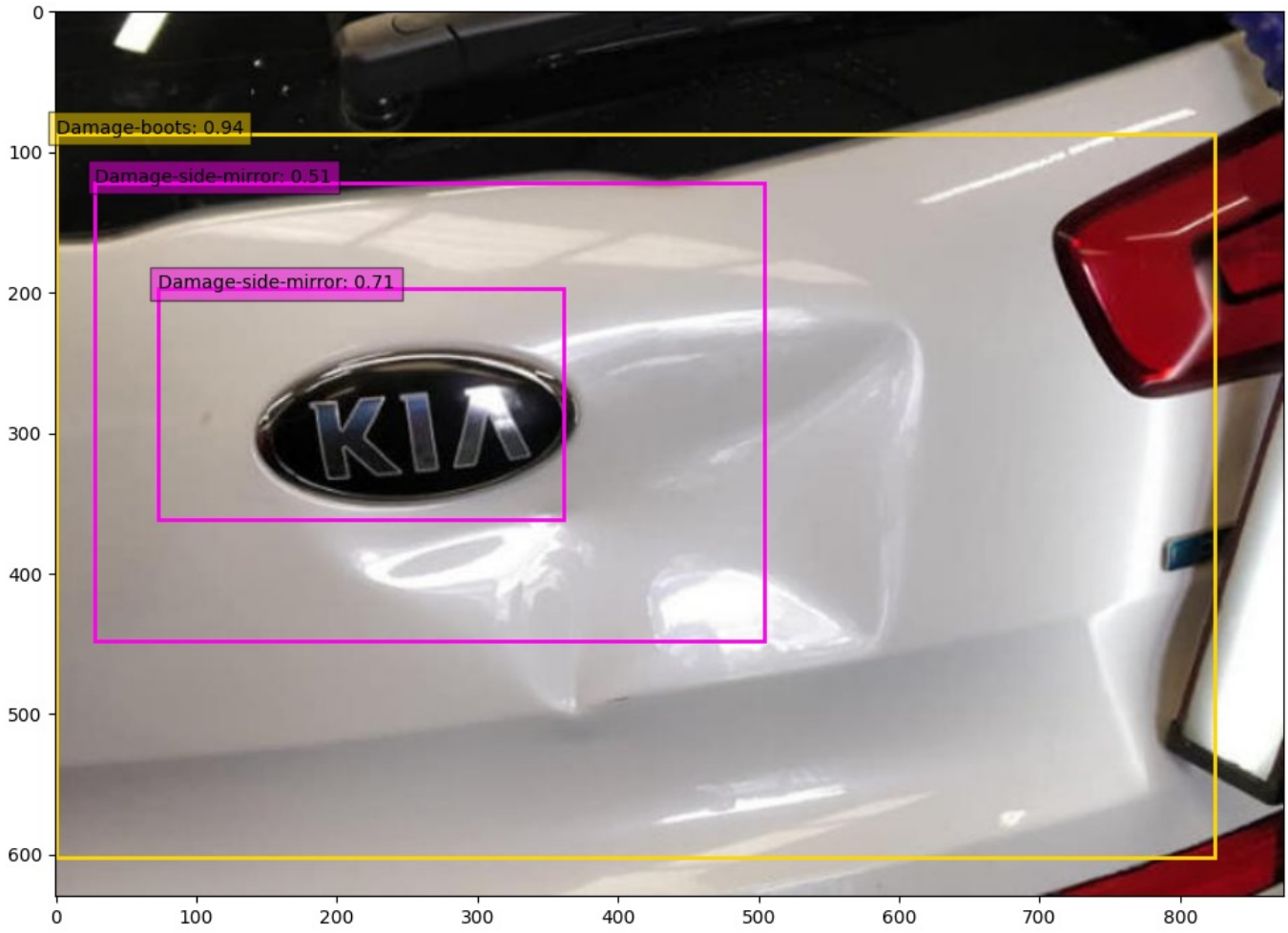

Damage-hood: 0.97
Damage-right-fender: 0.95
flat-tire: 0.90
0
100
200
300
400
500
600
0
100
200
300
400

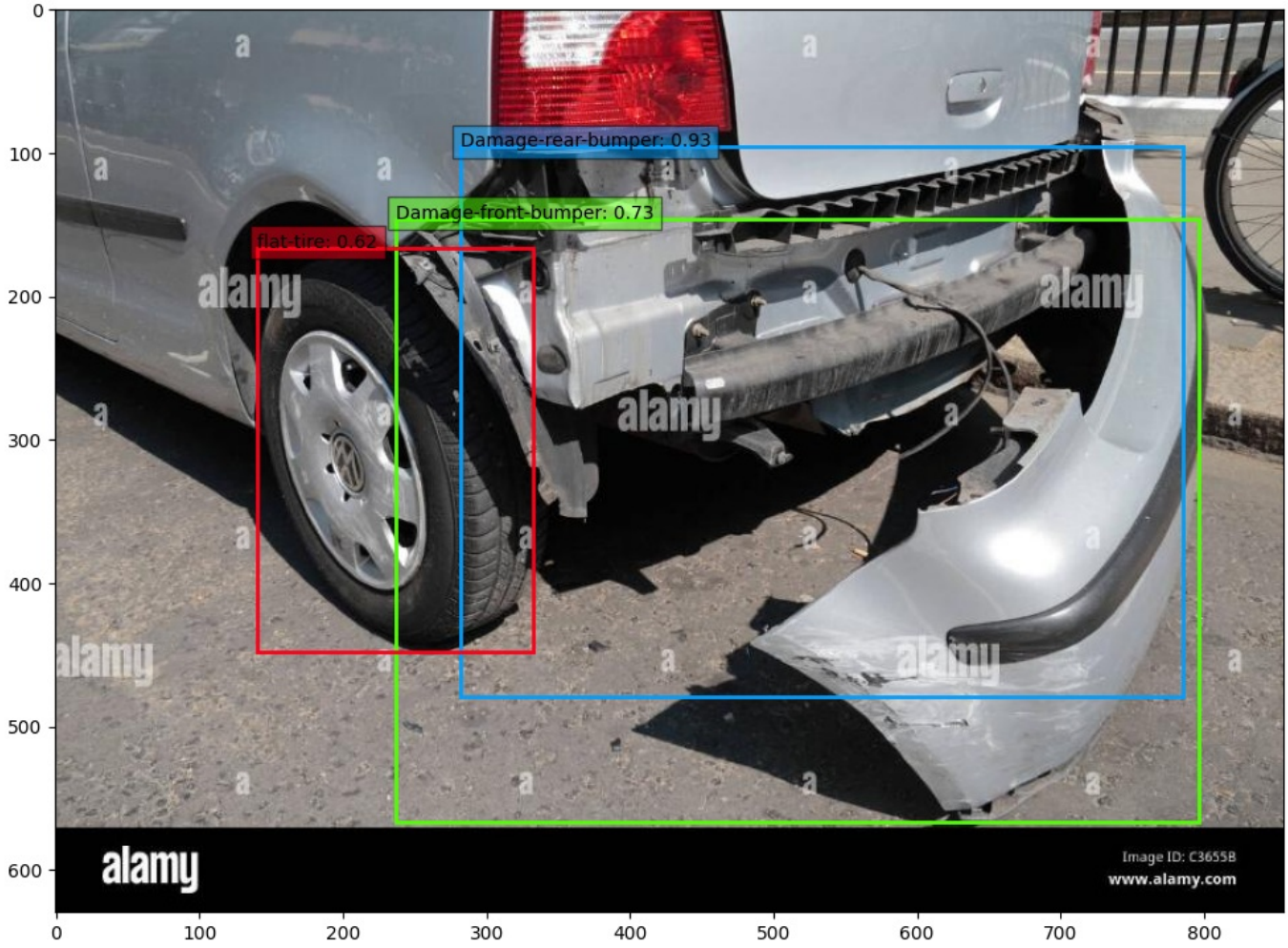

Damage-side-mirror: 0.62
flat-tire: 0.92
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-window-glass: 0.99
0
100
200
300
400
500
600
0
200
400
600
800


Damage-rear-bumper: 0.94
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

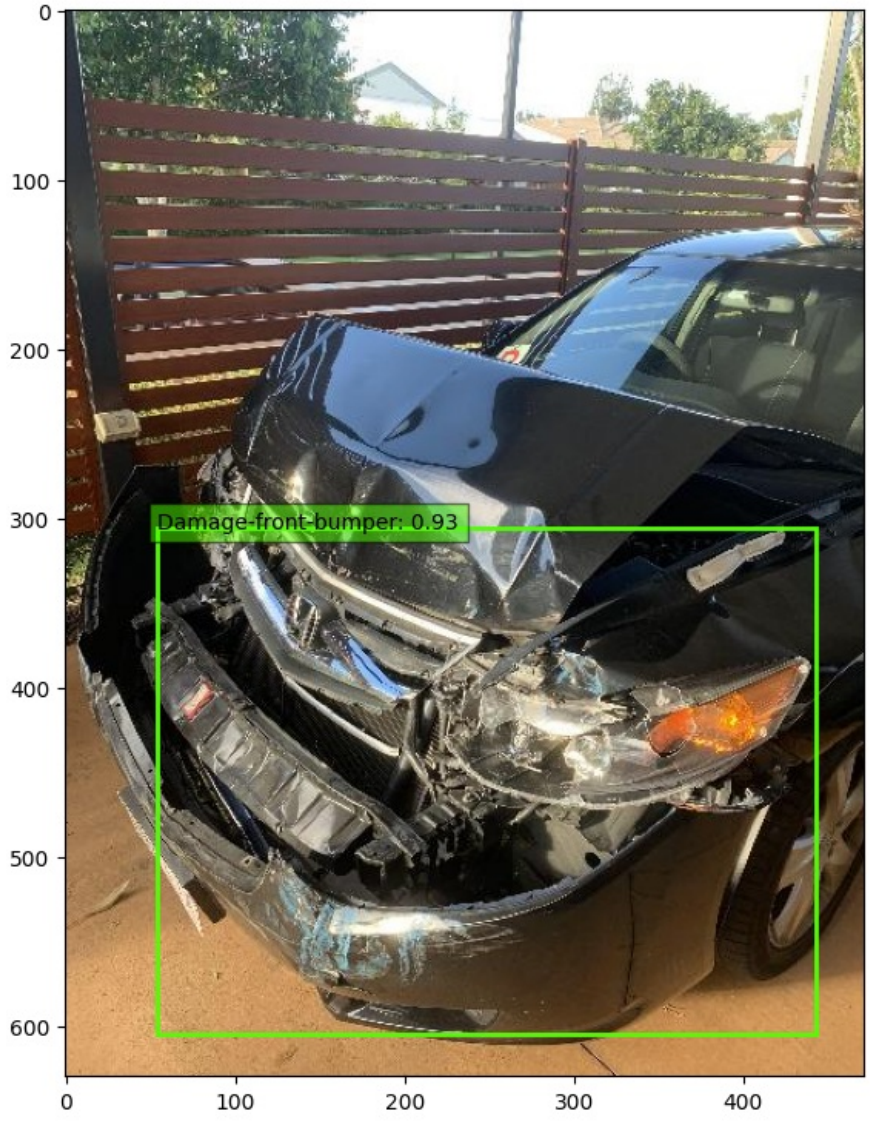

flat-tire: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


Damage-boots: 0.94
Damage-side-mirror: 0.51
Damage-side-mirror: 0.71
KIA


Damage-rear-bumper: 0.93
Damage-front-bumper: 0.73
flat-tire: 0.62
alamy
Image ID: C3655B
www.alamy.com


Damage-window-glass: 0.88
Damage-side-mirror: 0.79
0
100
200
300
400
500
600
0
100
200
300
400
500
600

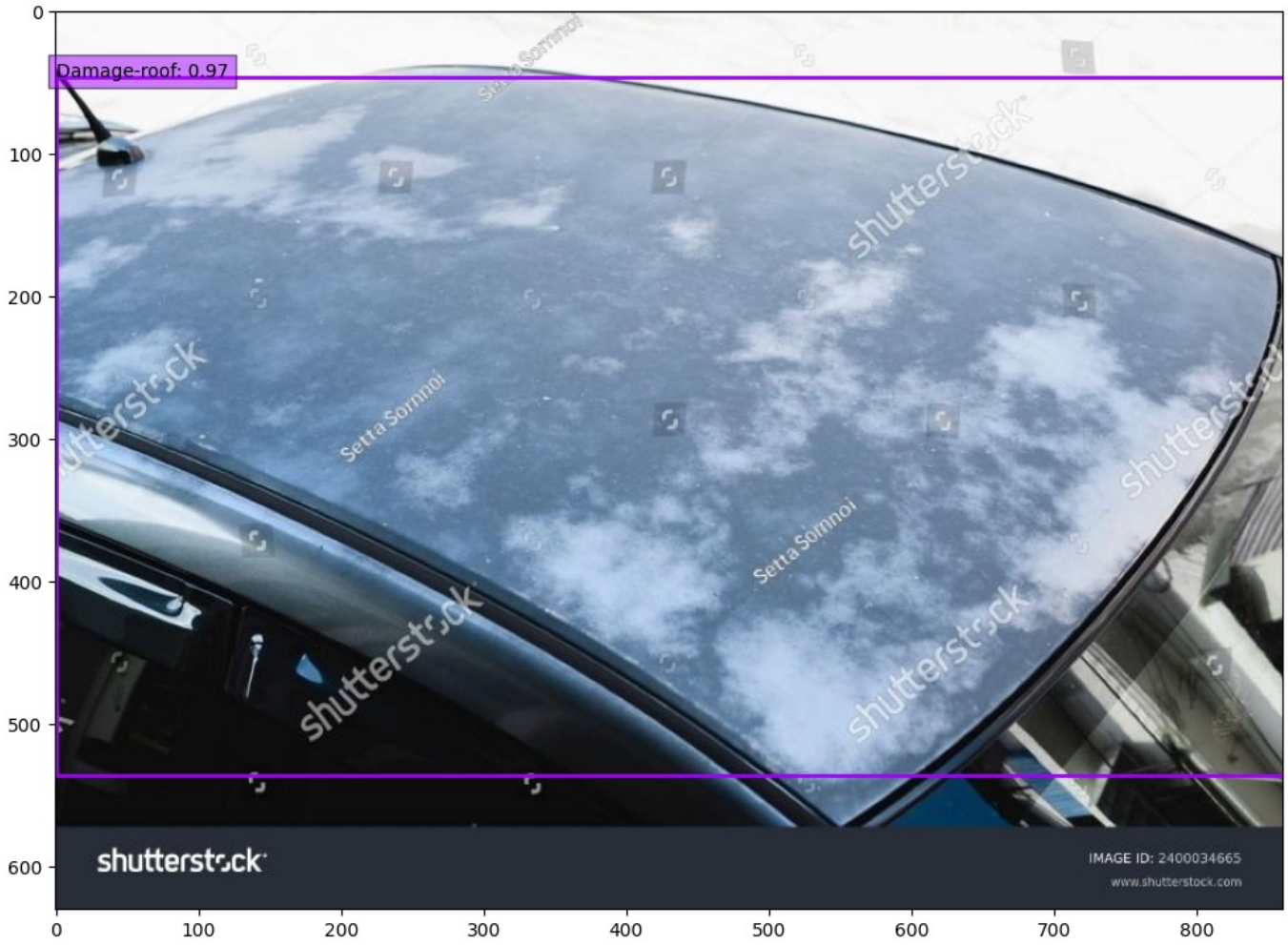

Damage-rear-bumper: 0.97
0
100
200
300
400
500
600
0
100
200
300
400
500
600

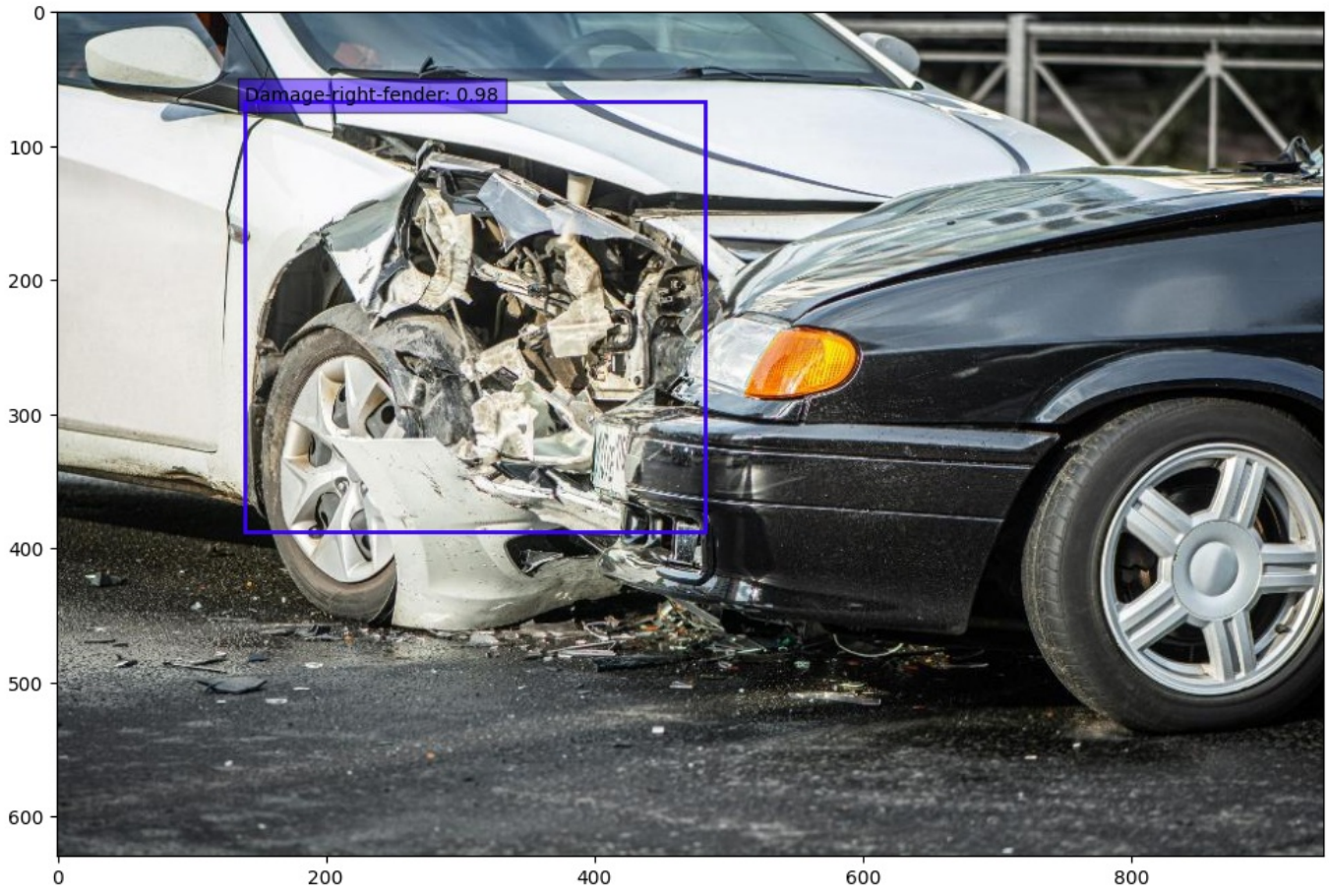

Damage-front-bumper: 0.93
0
100
200
300
400
500
600

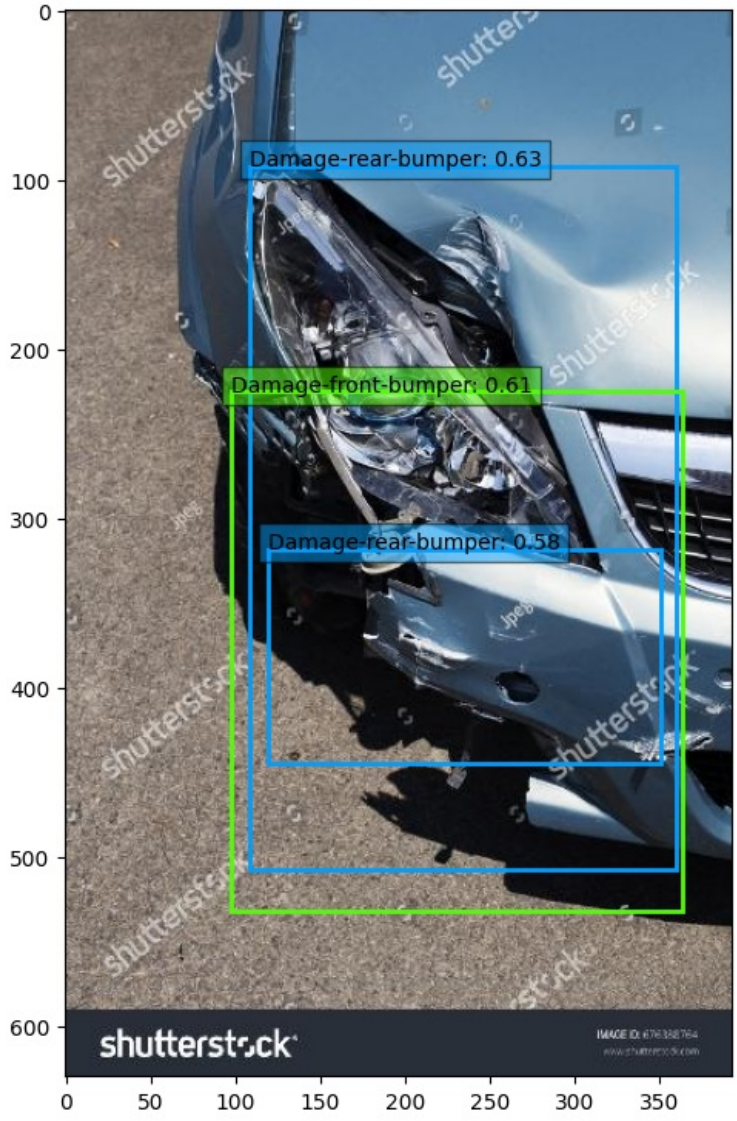

Damage-Windshield: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


206
0
100
200
300
400
500
0
200
400
600
800

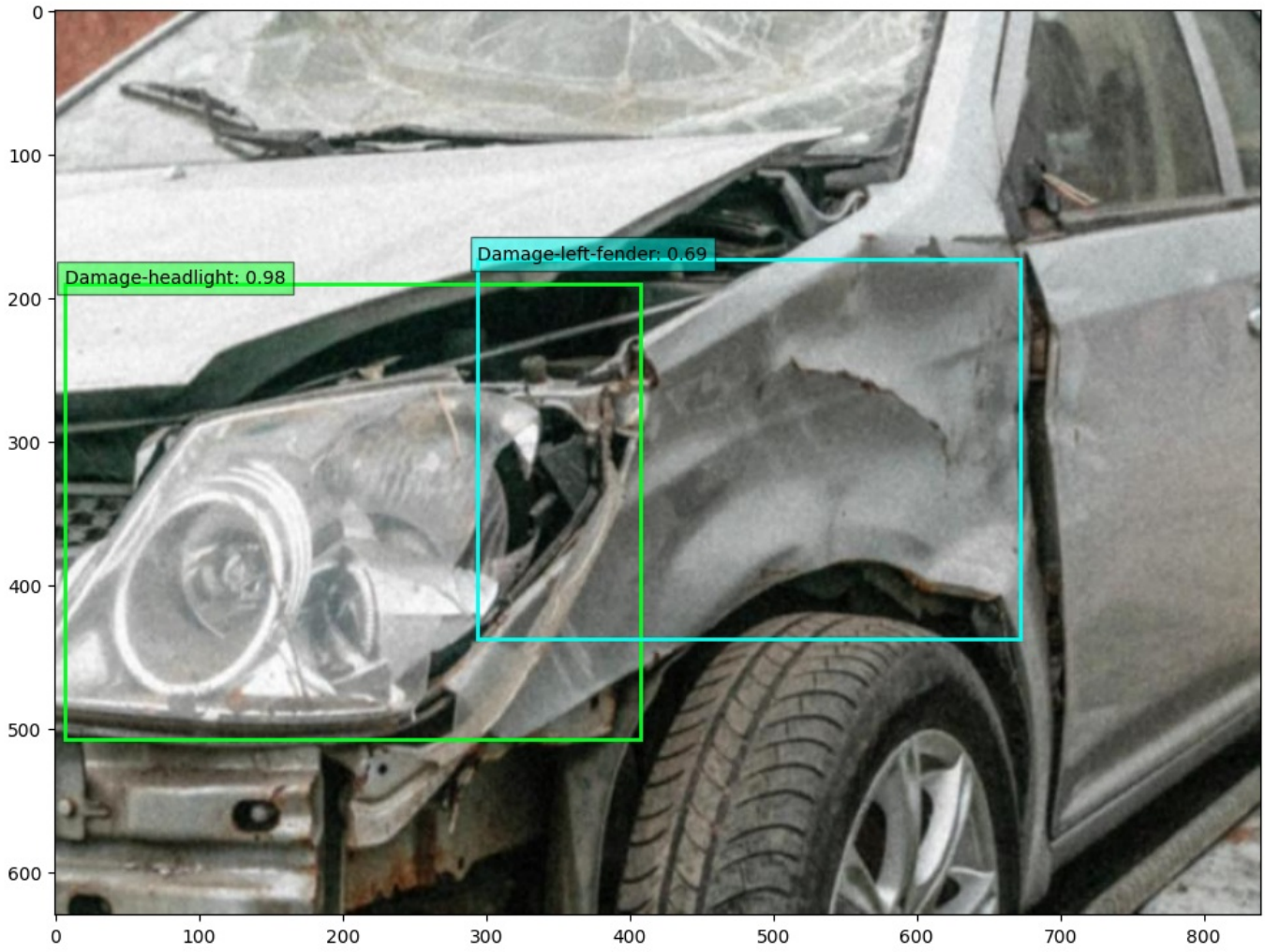

depositphotos
0
100
200
300
400
500
600
0
200
400
600
800


Damage-roof: 0.97
shutterstock
IMAGE ID: 2400034665
www.shutterstock.com


Damage-right-fender: 0.98


Damage-rear-bumper: 0.63
Damage-front-bumper: 0.61
Damage-rear-bumper: 0.58
shutterstock
0
100
200
300
400
500
600
0
50
100
150
200
250
300
350


Damage-right-fender: 0.97
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

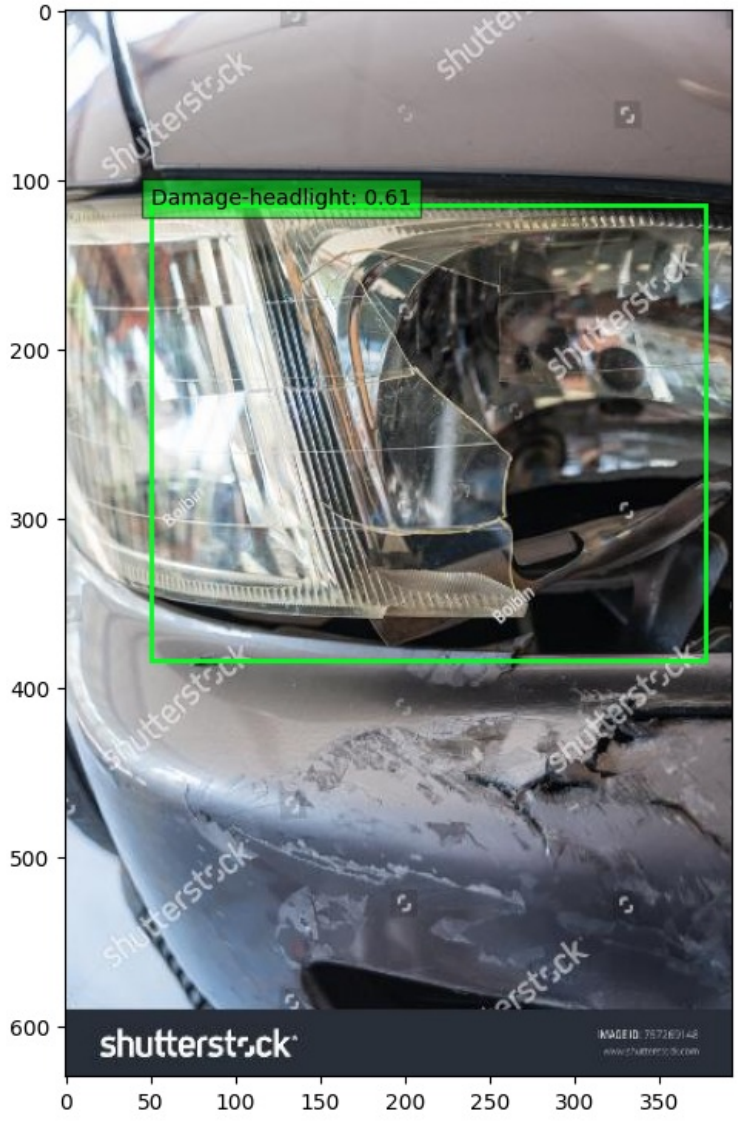

Damage-headlight: 0.98
Damage-left-fender: 0.69
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-hood: 0.56
0
100
200
300
400
500
600
0
200
400
600
800


0
100
200
300
400
500
0
200
400
600
800
GLOUCESTERSHIRE POLICE


0
100
200
300
400
500
600
0
200
400
600
800
Damage-Windshield: 0.98

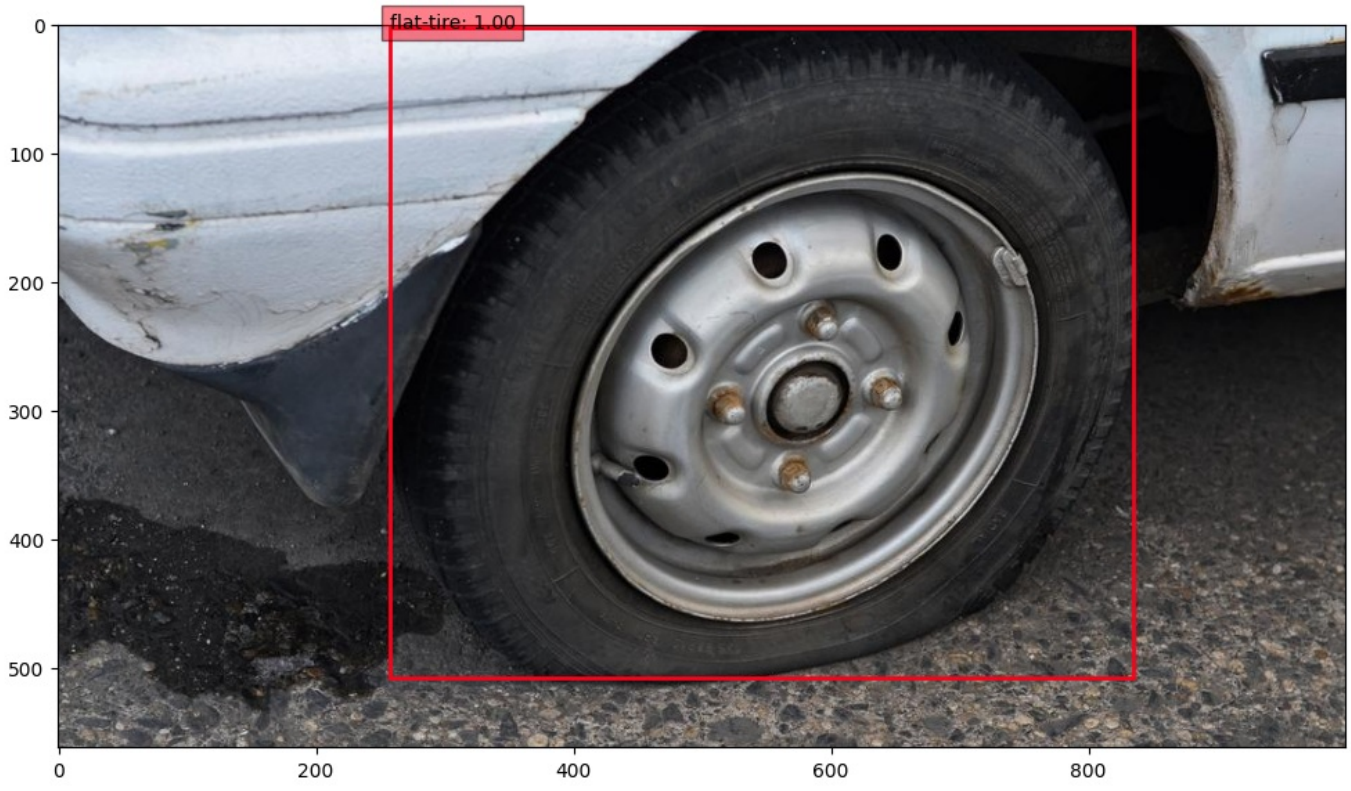

Damage-window-glass: 0.90
0
100
200
300
400
500
600
0
200
400
600
800


Damage-headlight: 0.61
shutterstock
0
100
200
300
400
500
600
0
50
100
150
200
250
300
350


flat-tire: 1.00
0
100
200
300
400
500
600
0
200
400
600
800


Damage-door: 0.97
0
100
200
300
400
500
600
0
200
400
600
800

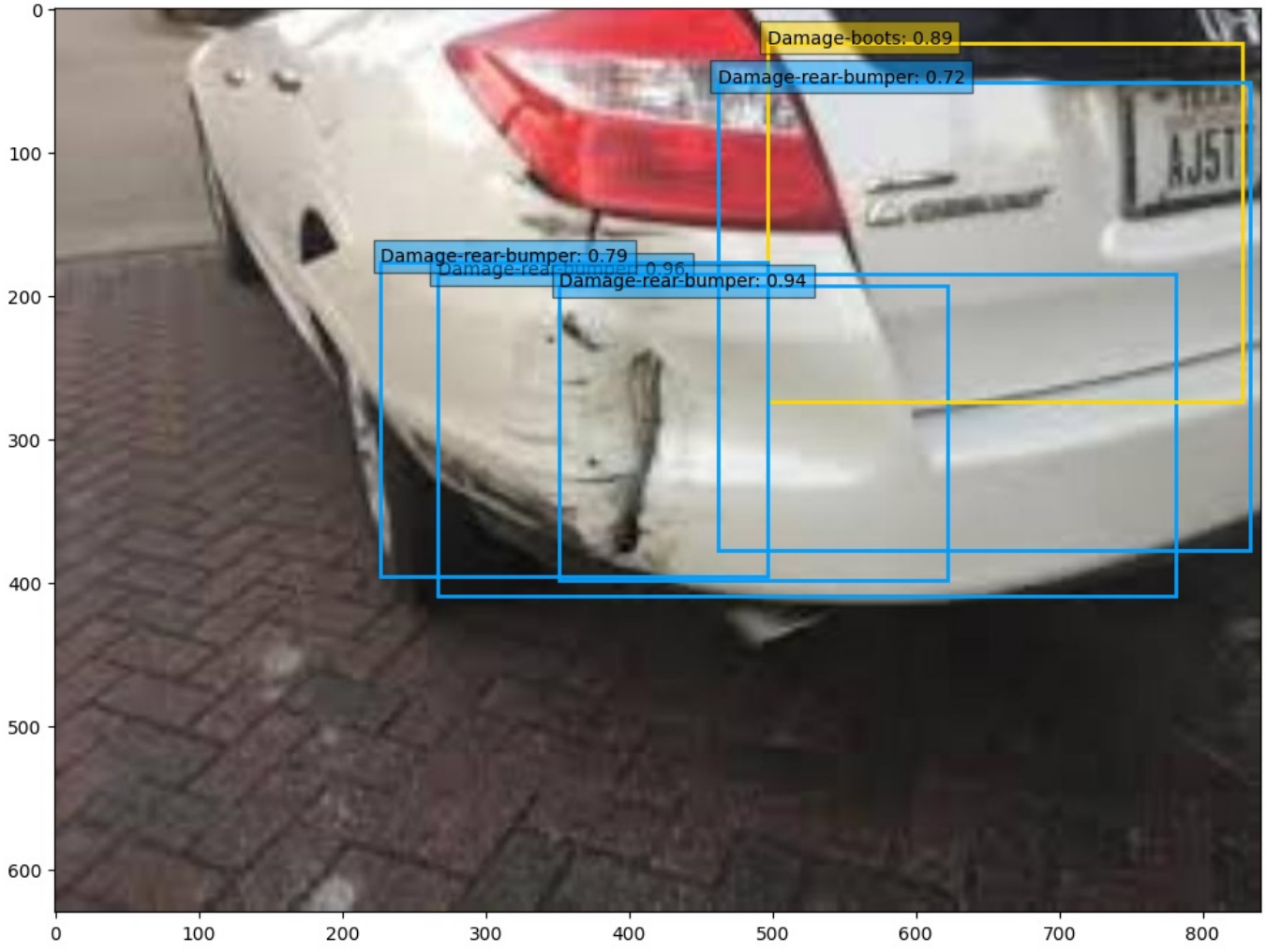

Damage-Windshield: 1.00
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


flat-tire: 1.00
0
100
200
300
400
500
0
200
400
600
800

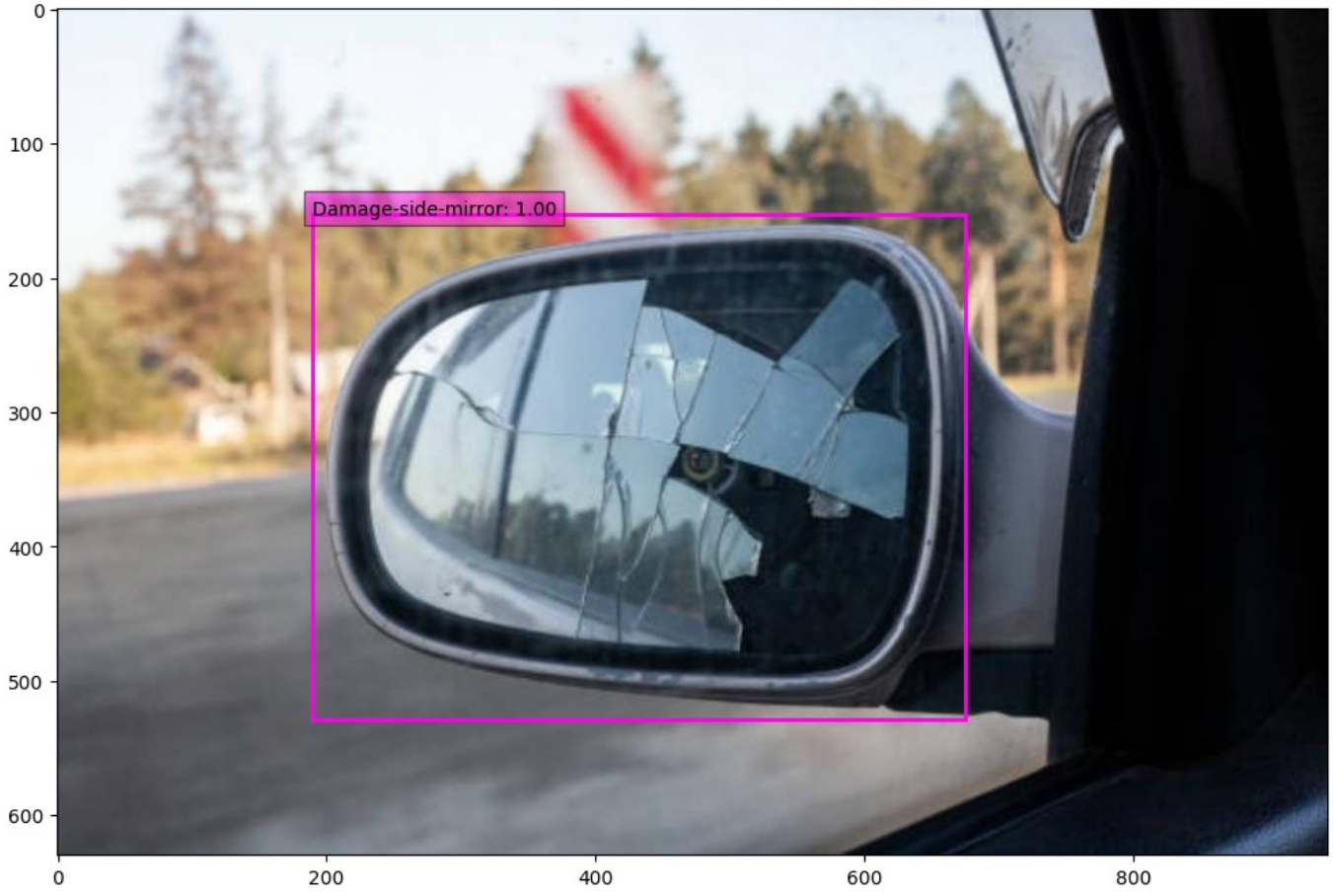

Damage-hood: 0.62
Damage-left-fender: 0.80
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

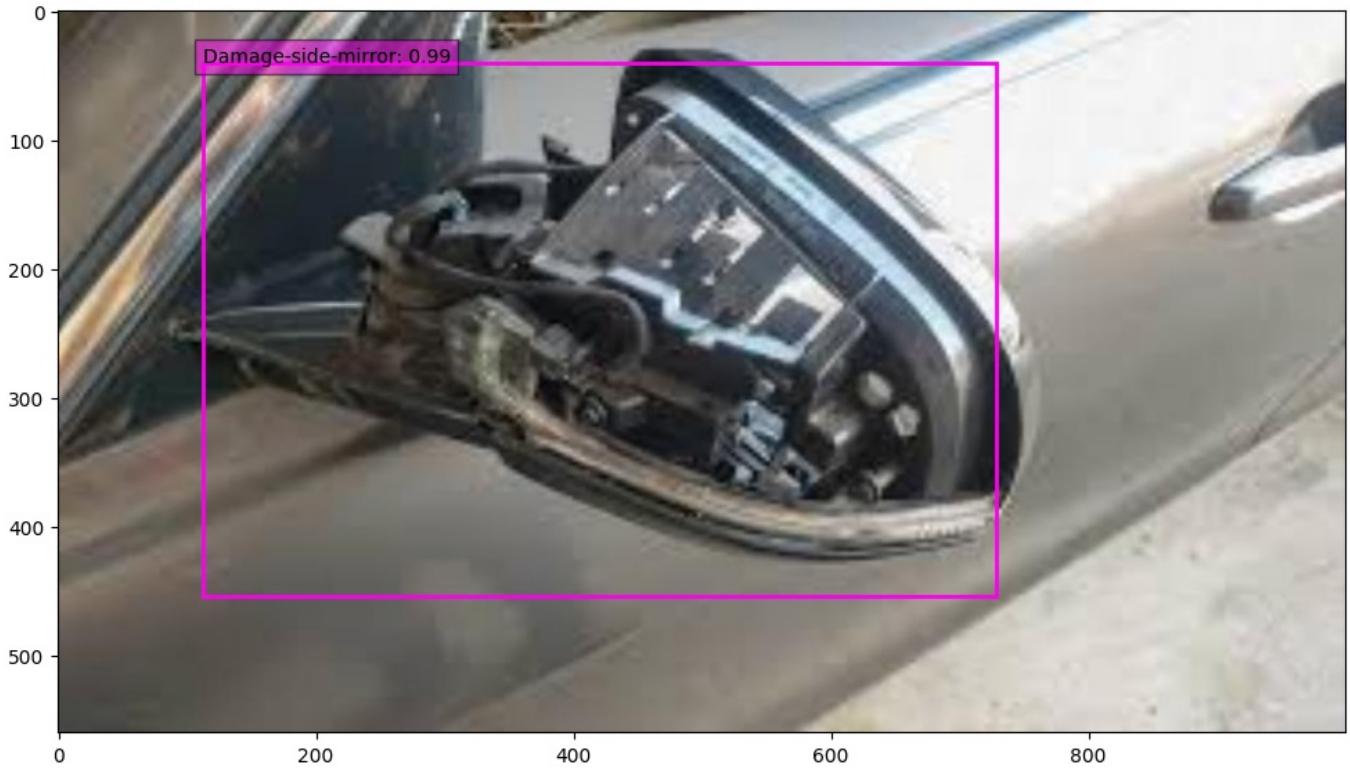

Damage-door: 0.89
Damage-left-fender: 0.51
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-side-mirror: 0.54
Damage-window-glass: 0.62
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-boots: 0.89
Damage-rear-bumper: 0.72
Damage-rear-bumper: 0.79
Damage-rear-bumper: 0.96
Damage-rear-bumper: 0.94
AJ5T
0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800


Damage-side-mirror: 1.00
0
100
200
300
400
500
600
0
200
400
600
800




```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[23], line 16
     13                 visualize_predictions(image_pil, output, train_label_map,color_map)  # Pass the label ma
p
     15 # Correct function call with train label map
---> 16 run_inference(model, test_loader, train_label_map, device)

Cell In[23], line 7, in run_inference(model, data_loader, train_label_map, device)
      5 for images, _ in data_loader:
      6     images = [img.to(device) for img in images]
----> 7     outputs = model(images)
      9     for i, output in enumerate(outputs):
     10         image = images[i].cpu().permute(1, 2, 0).numpy()

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
```

```
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torchvision\models\detection\generalized_rcnn.py:101, in GeneralizedRCNN.forward(self, images
, targets)
     94             degen_bb: List[float] = boxes[bb_idx].tolist()
     95             torch._assert(
     96                 False,
     97                 "All bounding boxes should have positive height and width."
     98                 f" Found invalid box {degen_bb} for target at index {target_idx}.",
     99             )
--> 101 features = self.backbone(images.tensors)
    102 if isinstance(features, torch.Tensor):
    103     features = OrderedDict([("0", features)])

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torchvision\models\detection\backbone_utils.py:57, in BackboneWithFPN.forward(self, x)
     56 def forward(self, x: Tensor) -> Dict[str, Tensor]:
---> 57     x = self.body(x)
     58     x = self.fpn(x)
     59     return x

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torchvision\models\_utils.py:69, in IntermediateLayerGetter.forward(self, x)
     67 out = OrderedDict()
     68 for name, module in self.items():
---> 69     x = module(x)
     70     if name in self.return_layers:
     71         out_name = self.return_layers[name]

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\container.py:217, in Sequential.forward(self, input)
    215 def forward(self, input):
    216     for module in self:
--> 217         input = module(input)
    218     return input

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
```

```
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torchvision\models\resnet.py:150, in Bottleneck.forward(self, x)
    147 out = self.bn1(out)
    148 out = self.relu(out)
--> 150 out = self.conv2(out)
    151 out = self.bn2(out)
    152 out = self.relu(out)

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\module.py:1501, in Module._call_impl(self, *args, **kwargs)
   1496 # If we don't have any hooks, we want to skip the rest of the logic in
   1497 # this function, and just call forward.
   1498 if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks or self._forward_pre_hoo
ks
   1499         or _global_backward_pre_hooks or _global_backward_hooks
   1500         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1501     return forward_call(*args, **kwargs)
   1502 # Do not call functions when jit is used
   1503 full_backward_hooks, non_full_backward_hooks = [], []

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\conv.py:463, in Conv2d.forward(self, input)
    462 def forward(self, input: Tensor) -> Tensor:
--> 463     return self._conv_forward(input, self.weight, self.bias)

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Pytho
n311\site-packages\torch\nn\modules\conv.py:459, in Conv2d._conv_forward(self, input, weight, bias)
    455 if self.padding_mode != 'zeros':
    456     return F.conv2d(F.pad(input, self._reversed_padding_repeated_twice, mode=self.padding_mode),
    457                     weight, bias, self.stride,
    458                     pair(0), self.dilation, self.groups)
--> 459 return F.conv2d(input, weight, bias, self.stride,
    460                 self.padding, self.dilation, self.groups)

KeyboardInterrupt:
```

In [ ]: