Project Title

Semester project

Session 2023-2027

BS in Software Engineering

Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

FALL 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	Machine Learning

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
1	BSAIM-F23-093	Jawad Ali	j.ali861877@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Table of Contents

Figure's

Chapter 1

Introduction

Write Introduction about your project

Introduction to Glasses for Blind People Project

The "Glasses for Blind People" project is an advanced assistive technology designed to empower visually impaired individuals by leveraging artificial intelligence, computer vision, and image processing. These smart glasses provide real-time audio feedback to help users navigate their surroundings and identify objects, making everyday tasks more accessible and independent.

Key Features:

1. **Object Recognition:**
   Detects and identifies common objects, such as furniture, doors, or personal belongings, using cutting-edge object detection models like SSD (Single Shot Multi_box Detector) and ImageNet datasets.
2. **Image Processing Capabilities:**
   - Enhances object identification through real-time image analysis.
   - Processes complex scenes to distinguish between objects and background efficiently.
   - Adapts to varying lighting conditions for consistent performance.
3. **Audio Feedback System:**
   Converts visual data into clear, descriptive voice prompts to guide the user effectively.

This project exemplifies how AI and image processing can create meaningful impact by enhancing the quality of life for visually impaired individuals, fostering independence, and enabling safe interaction with their environment.

Chapter 2

Tool & Technology

Software and AI Models:

- **Programming Languages:**
  - **Python**: The primary language for AI, image processing
- **AI and Machine Learning Frameworks:**

  1. **Image processing** (OpenCV, NumPy, Matplotlib)
  2. **Deep learning** (PyTorch, torchvision)
  3. **Speech synthesis** (gTTS)
  4. **Operating system interaction** (os)

- **Pre-trained Models:**

- **SSD (Single Shot Multibox Detector)**: For object detection.

Integration Technologies:

- **Text-to-Speech (TTS):**
  - Libraries like **gTTS(Google Text-to-Speech.)**, services to convert object descriptions into audio feedback.

Development Tools:

- **IDE and Editors:**
  - Google Colab (ipynb) are utilized for coding and debugging.

Write tool and technology which you have used

Device name DESKTOP-KVOVANT

Processor Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz

Installed RAM 4.00 GB (3.88 GB usable)

Device ID	F12F5124-C0DB-4644-86AA-44A96037819E
Product ID	00330-80000-00000-AA850
System type	64-bit operating system, x64-based processor

Edition	Windows 10 Pro
Version	22H2
Installed on	4/29/2024
OS build	19045.5131
Experience	Windows Feature Experience Pack 1000.19060.1000.0

Chapter 3

Implementation Code

Library Utilized:

```
]  import torch
   import torchvision
   import cv2
   import numpy as np
   from gtts import gTTS
   import os
   import matplotlib.pyplot as plt
   import cv2
   from torchvision import transforms
   import matplotlib.pyplot as plt
```

Figure 1.

Image preprocessing:

```python
def img_processing(image_path):
    image = cv2.imread(image_path)
    if image is None:
        raise FileNotFoundError(f"Image not found: {image_path}")

    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    image_resized = cv2.resize(image_rgb, (300, 300))

    transform = transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    ])
    image_tensor = transform(image_resized).unsqueeze(0)

    return image_tensor, image_resized

image_path = "/content/2007_002266.jpg"
image_tensor, resized_image = img_processing(image_path)

plt.imshow(resized_image)
plt.title(" IMage.")
plt.axis("on")
plt.show()
```

Figure 2

SSD Model Train:

```
coco_labls = [
    'N/A', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat',
    'traffic light', 'fire hydrant', 'N/A', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog',
    'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella',
    'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat',
    'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass', 'cup', 'fork',
    'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog',
    'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table', 'N/A',
    'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven',
    'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier',
    'toothbrush'
]

def model(image_tensor, original_image):
    model = torchvision.models.detection.ssd300_vgg16(pretrained=True)
    model.eval()

    with torch.no_grad():
        predictions = model(image_tensor)

    bboxes = predictions[0]['boxes']
    scores = predictions[0]['scores']
    labels = predictions[0]['labels']

    confidence_score = 0.5
    feedback_data = []
```

✓ Connected to Python 3 Google Compute Engine backend

Figure 3

```
    h, w, _ = original_image.shape

    for i in range(len(scores)):
        if scores[i] > confidence_score:
            x1, y1, x2, y2 = map(int, bboxes[i])
            x1, x2 = int(x1 * w / 300), int(x2 * w / 300)
            y1, y2 = int(y1 * h / 300), int(y2 * h / 300)

            cv2.rectangle(original_image, (x1, y1), (x2, y2), (0, 255, 0), 2)

            label_name = coco_labls[int(labels[i])]

            label = f"{label_name}: {scores[i]:.2f}"
            cv2.putText(original_image, label, (x1, y1 + 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

            feedback_data.append(f"{label_name} detected with confidence {scores[i]:.2f}")

    if feedback_data:
        print(". ".join(feedback_data))
    else:
        print("No objects detected above the confidence threshold.")

    plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
    plt.title("Object Detection Results")
    plt.axis("off")
    plt.show()
model (image_tensor, resized_image)
```

Figure 4

Chapter 4

Result

Image processing Out Come:

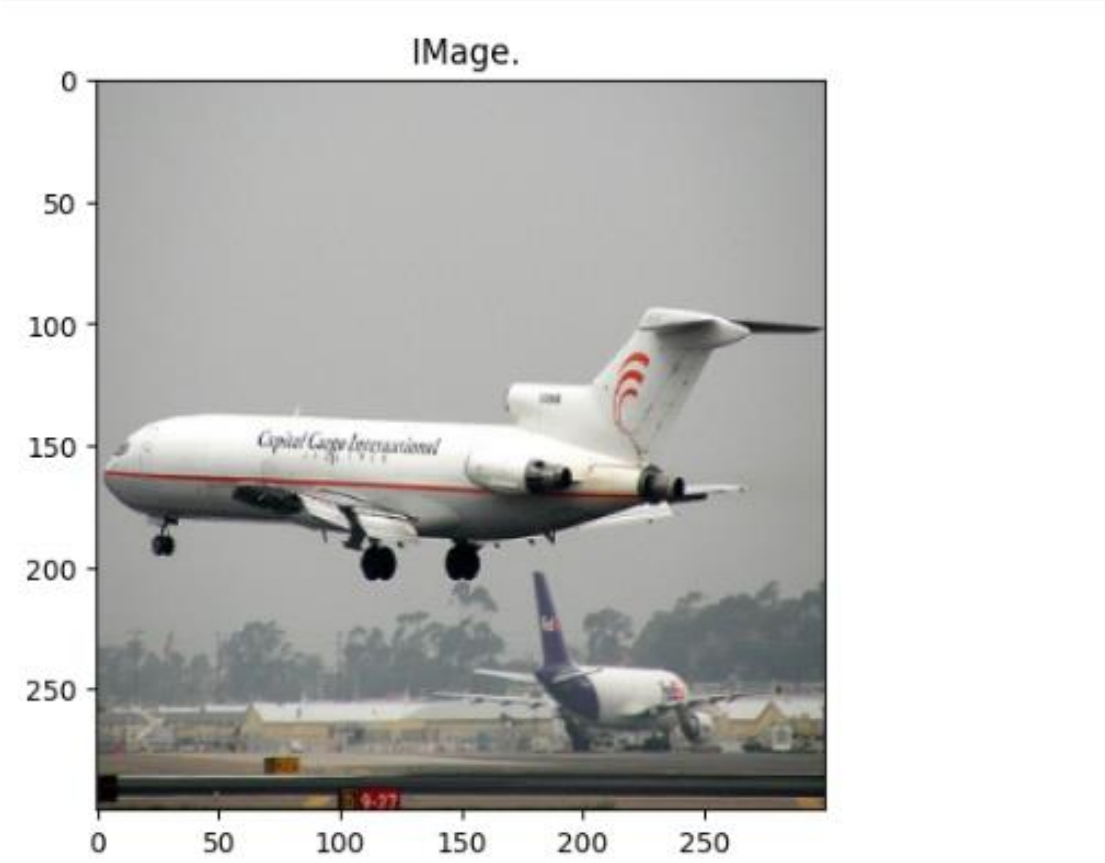

Figure 5

Machine Train Model Out Comes:

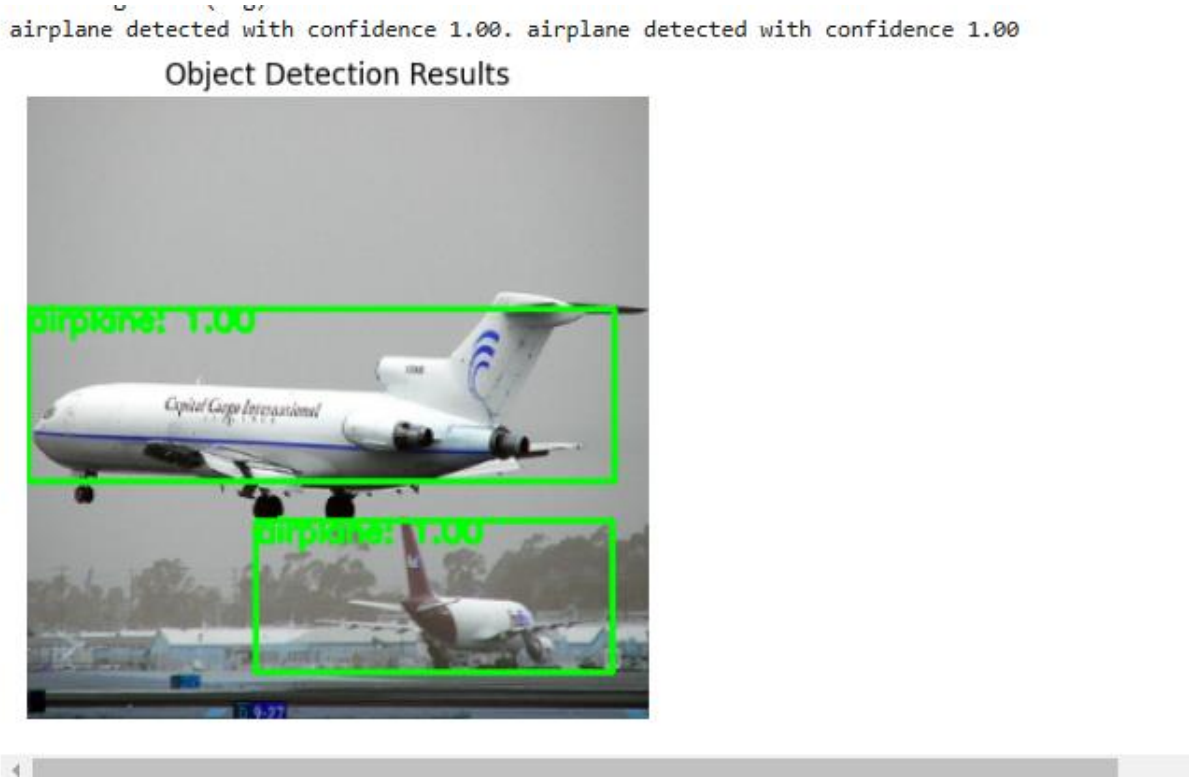

Figure 6

What I Learned from This Project:

This project provided valuable insights and hands-on experience in several key areas, particularly in **machine learning** and **image processing**. Here are the main learnings from the project:

1. **Training Machine Learning Models:**
    - I learned how to train and fine-tune machine learning models for **object detection**. This experience gave me a deeper understanding of how models like **SSD** or **TensorFlow Object Detection** work to identify objects in real-time.

- I also explored the importance of **model evaluation** using metrics like **accuracy**, **precision**, and **recall**, helping me improve the model's performance.

2. **Image Processing Techniques:**
   - I gained hands-on experience with **image preprocessing** techniques, which are crucial in preparing images for machine learning models. I learned how to handle various aspects of image quality, such as **Resize**, **Converting the image to a tensor for use in models, normalization**.
   - The project introduced me to the importance of techniques to improve model robustness, making sure the system works effectively even under varying conditions.
3. **Real-time Processing and Object Detection:**
   - I learned how to capture video frames and process them in **real-time**, applying object detection algorithms to identify and label objects efficiently.
   - This experience taught me how to integrate **text-to-speech** technology to provide audio feedback, turning image data into actionable information for visually impaired users.

How I Will Use These Skills in Future Projects:

1. **Handling and Preprocessing Images:**
   - In future projects, I will apply the skills I've learned in handling **image Resize**, **normalization**, and **other processing** adjustments to ensure the images are properly prepared for analysis.

- I will also focus on techniques such as **Converting the image to a tensor for use in models,** to make the images clearer for both human interpretation and machine learning algorithms.

Developing Advanced Assistive Technologies:

I could expand the project to include **AI-based personal assistants**, **smart glasses**, or **robotic guidance systems** for visually impaired individuals, leveraging the knowledge gained about **image analysis**

.