International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IMAGE SEGMENTATION

SELVENDRAN B[1], SARANKUMAR S[2] , SABTHARISHI S[3] , UTHANDAM A[4]

UG STUDENTS , SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY , COIMBATORE.

ABSTRACT:

This project presents an interactive web-based application for image segmentation using deep learning, specifically the Mask R-CNN model implemented with PyTorch. The goal of this application is to detect and segment objects in uploaded images in real-time. The system leverages the power of pretrained models from the COCO dataset to identify over 80 different object categories such as people, vehicles, animals, and daily-use items. Users can upload an image, and the app automatically performs instance segmentation—drawing bounding boxes, labels, and applying colored masks over detected objects. The frontend is developed using Streamlit, which offers a clean and responsive user interface, making the app easy to use even for those without a technical background. Behind the scenes, the model performs inference on the uploaded image and returns the detected regions with a confidence threshold adjustable by the user. Additional features like dynamic resizing, real-time visualization, and customizable score filtering enhance the functionality and performance of the application. This project demonstrates how cutting-edge deep learning models can be integrated with modern web frameworks to create efficient and user-friendly computer vision tools. The result is a compact, intuitive, and powerful system that serves as a practical introduction to image segmentation and deep learning deployment.

Keywords: Image Segmentation, Mask R-CNN, Deep Learning, PyTorch, Streamlit, COCO Dataset, Instance Segmentation, Computer Vision, Object Detection, Web Application.

Introduction:

Image segmentation is one of the core challenges in computer vision, aimed at identifying and separating objects within an image. With the rise of deep learning, modern models like Mask R-CNN have made it possible to not only detect objects but also generate precise segmentation masks that highlight the exact shape and position of each object. This project focuses on building a user-friendly web application that applies Mask R-CNN to perform instance segmentation on uploaded images.

The application is built using Streamlit, which allows quick deployment of interactive machine learning tools with minimal setup. By combining PyTorch's pretrained Mask R-CNN model with a clean web interface, users can upload any image and instantly view segmented outputs, where objects are identified with labels, bounding boxes, and color-coded masks.

The model used in this project is pretrained on the COCO (Common Objects in Context) dataset, which means it can recognize and segment a wide variety of common objects such as people, animals, vehicles, tools, and food items. Users can also adjust the confidence threshold to control which predictions are shown, adding flexibility for different use cases.

This project bridges the gap between complex deep learning models and real-world usability by offering an end-to-end solution for object detection and segmentation that works right in the browser—no advanced technical knowledge required.

What is IMAGE SEGMENTATION ?

This project is a web-based image segmentation tool that allows users to upload an image and view detected objects with colored masks and labels. It uses the Mask R-CNN deep learning model, which performs instance segmentation, meaning it can identify and separate multiple objects even if they belong to the same class. The backend is built with PyTorch, and the interface is created using Streamlit, making it easy to use and accessible from any device with a browser.

What is use of Image Segmentation project?

This project helps users understand and visualize how deep learning models can detect and segment objects in images. It is useful for:

- Educational purposes – to learn how segmentation models work.
- Developers and researchers – as a prototype to test on various images.
- Practical applications – like identifying objects in surveillance images, analyzing visual data, or preprocessing images for further AI tasks. It makes advanced computer vision technology available in a simplified and interactive format for real-time use.

Methodology:

The core of this project revolves around the use of a pretrained Mask R-CNN model for instance segmentation. The methodology can be broken down into several stages, from model setup to final visualization through a web interface. Each step has been carefully designed to ensure a smooth and real-time experience for the user.

1. Model Selection and Setup:

We used the Mask R-CNN model with a ResNet-50 backbone and Feature Pyramid Network (FPN), provided by PyTorch's torchvision.models.detection module. This model is pretrained on the COCO dataset, which allows it to recognize and segment 80+ common object categories without additional training .The model is loaded with default weights and set to evaluation mode using:

```
model = torchvision.models.detection.maskrcnn_resnet50_fpn(weights=weights)

model.eval()
```

2. User Interface Development:

To create an accessible and responsive frontend, we used Streamlit, a lightweight Python library for building interactive web apps. Users can:

- Upload .jpg, .jpeg, or .png files
- View uploaded images
- Adjust a score threshold slider to control the sensitivity of detection
- Click a button to run the segmentation process
- Streamlit handles file uploads and displays both the input and output images in the browser, with no setup needed on the user's end.

3. Image Preprocessing:

Once the image is uploaded:

- It is converted to RGB using the PIL library
- If the image is too large, it is resized to a maximum dimension (e.g., 1024 px) to prevent memory issues
- The image is then converted into a PyTorch tensor using torchvision.transforms.functional.to_tensor() and passed to the model

4. Inference and Prediction:

- With the image tensor prepared, we run inference through the model:

  ```
  with torch.no_grad():

  prediction = model(image_tensor)[0]
  ```

- The output includes:

1. Bounding boxes
2. Segmentation masks
3. Class labels
4. Confidence scores
5. A confidence threshold (set by the user) filters out weak predictions.

5. Post-processing and Visualization:

- For each prediction above the threshold:
- The object's segmentation mask is extracted and thresholded to create a binary mask
- A random color is assigned to each detected object
- The mask is blended over the original image using NumPy
- A bounding box and label are drawn using OpenCV functions
- This creates a final segmented output where each object is clearly highlighted with a label and a color-coded mask.

6. Output Display:

The processed image with segmented outputs is displayed on the Streamlit interface. Users can see the results instantly and re-run detection with different thresholds to explore how the model responds.

Objective:

The main aim of this project is to develop a web-based image segmentation tool using deep learning that is accurate, interactive, and easy to use. The specific objectives of the project are:

- To perform instance segmentation on uploaded images using the Mask R-CNN model pretrained on the COCO dataset.
- To identify and highlight multiple objects within a single image by drawing masks, labels, and bounding boxes.
- To build a user-friendly interface using Streamlit that allows users to upload images, set detection thresholds, and view results in real-time.
- To visualize the segmentation output clearly with minimal lag, making it accessible for both technical and non-technical users.
- To provide a customizable and educational tool that demonstrates how computer vision models work in practice.
- To ensure efficient performance by handling large image sizes through resizing and optimizing model inference using PyTorch.

Results:

1. Accurate Object Detection: Mask R-CNN is one of the most accurate models for instance segmentation, providing both object detection (bounding boxes) and pixel-wise segmentation (masks).

2. Real-time Segmentation: When optimized, it can provide real-time segmentation, making it suitable for applications like autonomous vehicles, surveillance, or even real-time video analysis.

3. Versatility: This system can detect and segment various objects (like people, animals, furniture, etc.) from images, making it useful in numerous fields, including healthcare, robotics, and entertainment.

4. Ease of Use: The project is implemented in Streamlit, which provides a user-friendly interface for interaction with the model. Users can easily upload images and adjust the score threshold for detecting objects.

5. Customizable Output: The segmentation masks can be modified, allowing users to define the level of confidence (via score threshold) and control how objects are detected and displayed.

6. Pre-trained Model: By using a pre-trained Mask R-CNN model, this project saves time and computational resources. It leverages a powerful, ready-to-use model without needing to train from scratch.

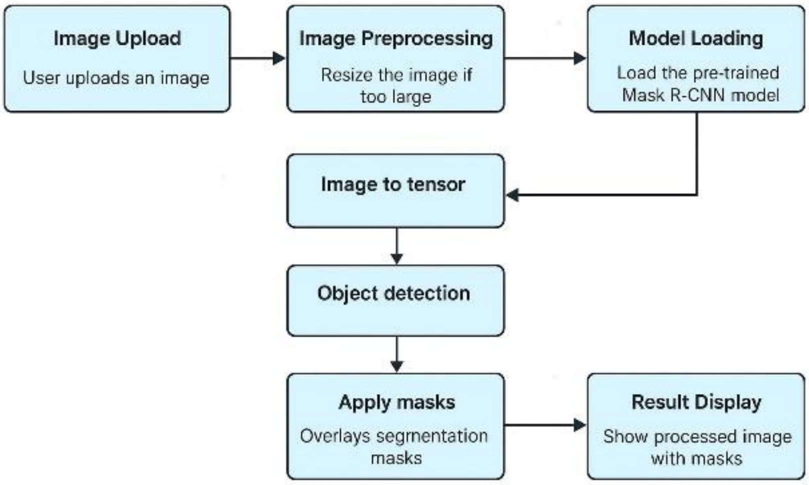
Image Upload
User uploads an image
Image Preprocessing
Resize the image if too large
Model Loading
Load the pre-trained Mask R-CNN model
Image to tensor
Object detection
Apply masks
Overlays segmentation masks
Result Display
Show processed image with masks

Fig 1 Block Diagram

Conclusion:

This project successfully demonstrates the power of deep learning in real-world computer vision tasks through image segmentation using Mask R-CNN. By integrating PyTorch, Torchvision, and a clean Streamlit UI, the application enables users to detect and segment multiple objects in an image with high accuracy.The tool is efficient, easy use, and visually intuitive, making it suitable for both educational and practical purposes. It highlights how pre-trained models can be leveraged for advanced tasks without requiring extensive training or computational resources. Overall, this project serves as a strong foundation for further enhancements in the field of object detection and image segmentation.

REFERENCES:

List all the material used from various sources for making this project proposal
Research Papers:

1. **PyTorch Documentation:**
   https://pytorch.org/docs/stable/index.html
2. **Torchvision Models:**
   https://pytorch.org/vision/stable/models.html
3. **Mask R-CNN Paper:**
   He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 2961–2969).
   https://arxiv.org/abs/1703.06870
4. **COCO Dataset:**
   https://cocodataset.org/#home
5. **Streamlit Documentation:**
   https://docs.streamlit.io/
6. **OpenCV Documentation:**
   https://docs.opencv.org/
7. **PIL (Python Imaging Library):**
   https://pillow.readthedocs.io/en/stable/