Project Report: Intelligent Image Captioning and Segmentation Web App

Project Overview

Project Name:

Intelligent Image Captioning and Segmentation Web App

Organization:

ZIDIO DEVELOPMENT (Internship)

Project Duration:

1 Month

Supervisor:

Chandan Mishra

Date of Submission:

27/07/2025

1. Introduction

The Intelligent Image Captioning and Segmentation Web App is designed to harness the power of deep learning to enhance the understanding and processing of visual data. This project aims to create a web application that empowers users to upload images, which the system then analyzes to produce descriptive captions and segment key objects visually. Utilizing state-of-the-art pretrained models, this application was developed with a user-friendly interface built using Streamlit.

2. Research Overview

Image Captioning – What & Why?

Image captioning is crucial for automating the generation of textual descriptions that accurately represent the content of an image. This task necessitates a blend of visual understanding—such as recognizing objects and their locations—and linguistic capabilities to formulate coherent sentences.

- **Model Used:** Salesforce BLIP (Bootstrapped Language-Image Pretraining)
- **Why BLIP?:** The BLIP model incorporates a visual encoder and a language decoder, making it proficient in understanding both image and text. Its training on diverse datasets like COCO and Conceptual Captions enhances its effectiveness in real-world applications.

Image Segmentation – What & Why?

Image segmentation is the process of classifying each pixel within an image to identify and differentiate various objects or segments. This detailed analysis is vital for applications in fields such as autonomous driving, healthcare imaging, and more.

- **Model Used:** Mask R-CNN (with ResNet-50 FPN backbone)
- **Why Mask R-CNN?:** Mask R-CNN stands out for its ability to not only detect objects but also create pixel-wise masks, providing an accurate representation of object boundaries.

3. Objectives

The primary objectives of this project included:

- Developing an image captioning system capable of describing uploaded images in natural language.
- Implementing object segmentation to highlight significant regions within images.
- Integrating both functionalities into a seamless, interactive web interface.
- Ensuring compatibility with real-world images, including those with complex backgrounds and varying object types.
- Structuring the codebase to be modular, readable, and easily deployable.

4. Tools and Technologies Used

Category	Tools / Frameworks
Programming	Python
Web UI	Streamlit
Image Captioning	Salesforce/blip-image-captioning-base (via Hugging Face Transformers)
Image Segmentation	Mask R-CNN (ResNet-50 FPN pretrained on COCO dataset)
Libraries	PyTorch, TorchVision, Transformers, PIL, OpenCV
Version Control	Git + GitHub

5. Project Structure

The project is organized as follows:

```
ZIDIO_Task1/│ ├── app/ │   ├── app.py            # Streamlit frontend │   ├──
utils.py          # Functions for captioning & segmentation │   └──
download_models.py # Model downloader script │ ├── requirements.txt      # Python
dependencies ├── README.md             # Project documentation
```

6. Methodology

Image Captioning

Leveraging the BLIP model from Hugging Face, the image captioning component processes an uploaded image to generate a relevant and coherent textual description. This is achieved through a transformer-based architecture that understands both visual inputs and language semantics.

Image Segmentation

The segmentation aspect utilizes a pretrained Mask R-CNN model with a ResNet-50 backbone, which detects and delineates objects within the image. The system highlights the primary object by overlaying bounding boxes and pixel-wise masks.

Frontend Integration

A user interface was created using Streamlit, allowing users to upload images seamlessly. The application presents the original image, the generated caption, and the segmented image side by side, providing an intuitive user experience.

7. Features

- Supports a variety of image formats (JPG, PNG).
- Generates descriptive captions, e.g., “A man riding a horse on a beach”.
- Capable of segmenting main objects such as people, animals, and other items.
- Provides real-time output and visual feedback in the web browser.
- Automatically handles model downloads on first use.
- Simple installation and setup for local running.

8. How to Run

To run the application, follow these steps:

1. Clone the repository:
2. `git clone https://github.com/Rishikesh4089/ZIDIO_Task1.gitcd ZIDIO_Task1/app`
3. Install dependencies:
4. `pip install -r ../requirements.txt`
5. Download models:
6. `python download_models.py`
7. Run the app:
8. `streamlit run app.py`

9. Challenges Faced

The models employed were substantial in size, necessitating careful management of download and load times, particularly involving GPU resources. Ensuring compatibility across various image types and dimensions required thorough preprocessing and validation. Effective collaboration through Git version control necessitated careful management of commits and resolution of conflicts among team members.

10. Outcome

The project culminated in a well-functioning web application with a fully operational frontend that proficiently handles image captioning and segmentation. The application demonstrates the potential to effectively process real-world images and is designed for further extension or deployment on cloud platforms such as Hugging Face or Streamlit Cloud.

Conclusion

The Intelligent Image Captioning and Segmentation Web App signifies a substantial achievement in automating the interpretation and segmentation of visual content. Through the application of advanced deep learning techniques, this project not only meets its objectives but also opens avenues for future enhancements and real-world applications.