CAPSTONE PROJECT

Image Segmentation and Object Detection

PRESENTED BY

STUDENT NAME: FAKRUDDIN

COLLEGE NAME: REVA UNIVERSITY

DEPARTMENT: COMPUTER APPLICATIONS

EMAIL ID: FAKRUDDIN4121@GMAIL.COM

AICTE STUDENT ID:
STU6763031ae16ef1734542106

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Manual object detection is time-consuming, especially in large image datasets. A lightweight system is needed to automatically segment and identify regions of interest in images using simple machine learning methods.

PROPOSED SOLUTION

Proposed Solution:

The proposed system provides a user-interactive web application for image segmentation and object detection using K-Means clustering and HSV thresholding. It automates the detection of prominent visual elements by isolating a chosen color cluster and identifying the largest object through contour detection.

- **Image Acquisition:**
  - Users upload any image from their local system.
  - The uploaded image is resized and converted into a compatible color format for consistent processing.
- **Image Segmentation:**
  - The image is reshaped into pixel data and processed using K-Means clustering.
  - Users control the number of clusters (K) via a sidebar slider.
  - Segmented output visually differentiates color-based regions.
- **Cluster Masking:**
  - The user selects a target cluster to highlight.
  - Pixels belonging to this cluster are replaced with red to isolate the region.

PROPOSED SOLUTION

- Color Space Conversion:
  - The masked RGB image is transformed into HSV color space.
  - This conversion simplifies color filtering using HSV ranges.
- Object Detection:
  - HSV thresholds are user-controlled to filter specific color ranges.
  - The app identifies contours from the filtered mask.
  - Bounding boxes are drawn only around the largest detected object (filtered by area).
- Deployment:
  - Built with Streamlit for real-time web-based interaction.
  - Sidebar controls allow users to tune K-means and HSV parameters.
  - Results are organized into intuitive tabs (Segmented, Masked, HSV, Threshold, Detected).
- Evaluation:
  - Visual output includes six stages: Input, Segmentation, Masking, HSV View, Thresholded Mask, and Final Detection.
  - Detection success is determined by accuracy in isolating and bounding the intended region.

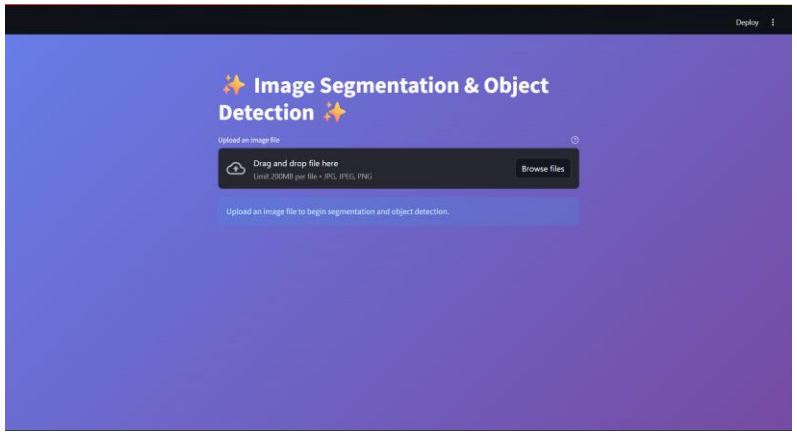
SYSTEM APPROACH

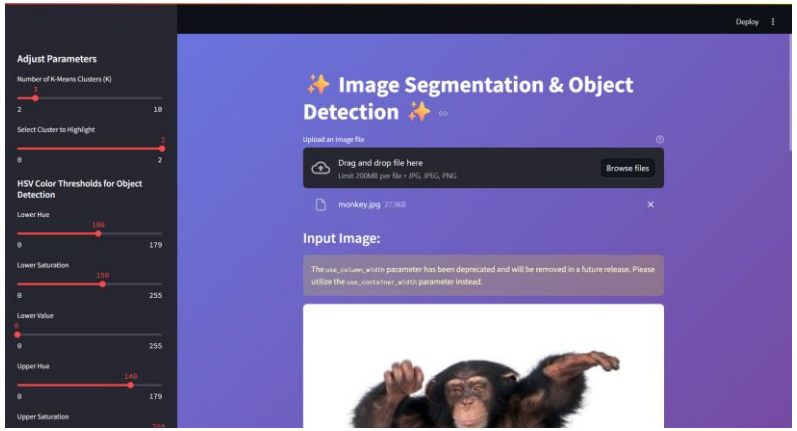
- **Language:** Python
- **Framework:** Streamlit
- **Libraries:** OpenCV, NumPy, Pillow
- **GUI:** Custom CSS for enhanced visuals
- **Input:** Local image file upload
- **Output:** Clustered, masked, HSV, thresholded, and annotated object views

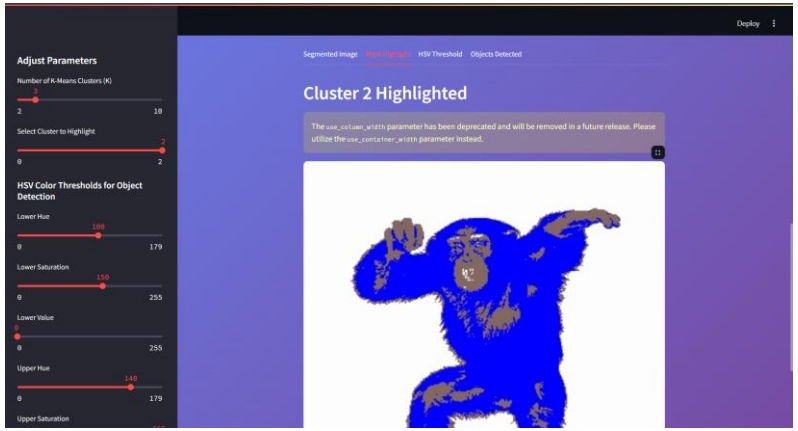
ALGORITHM & DEPLOYMENT

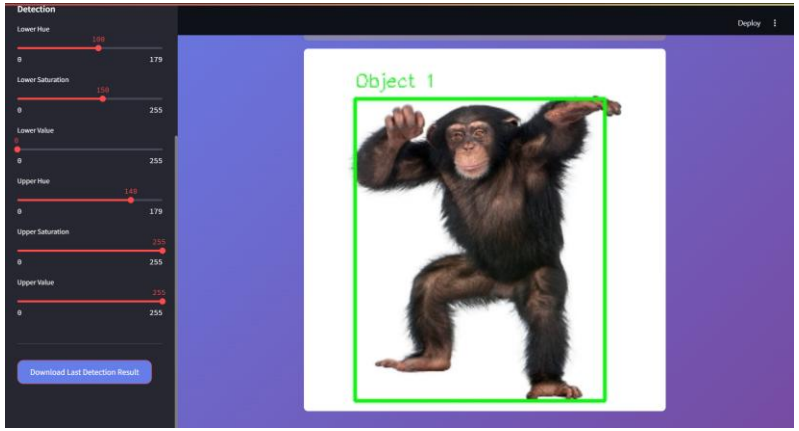
- K-Means clustering segments the image (k configurable via sidebar).
- Selected cluster is colored and converted to HSV.
- HSV thresholds (customizable) isolate object.
- Contours are extracted and bounding boxes are drawn.
- App is deployed locally via Streamlit interface with download support.

RESULT


Deploy
✨ Image Segmentation & Object Detection ✨
Upload an image file
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
Upload an image file to begin segmentation and object detection.


Deploy
Adjust Parameters
Number of K-Means Clusters (K)
Select Cluster to Highlight
HSV Color Thresholds for Object Detection
Lower Hue
Lower Saturation
Lower Value
Upper Hue
Upper Saturation
✨ Image Segmentation & Object Detection ✨
Upload an image file
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
monkey.jpg
Input Image:


Deploy
Adjust Parameters
Number of K-Means Clusters (K)
Select Cluster to Highlight
HSV Color Thresholds for Object Detection
Lower Hue
Lower Saturation
Lower Value
Upper Hue
Upper Saturation
Segmented Image
Mask Highlight
HSV Threshold
Objects Detected
Cluster 2 Highlighted


Deploy
Detection
Lower Hue
Lower Saturation
Lower Value
Upper Hue
Upper Saturation
Upper Value
Download Last Detection Result
Object 1

CONCLUSION

The project successfully demonstrates object detection through K-Means segmentation and HSV filtering. It provides a real-time, interactive platform for users to adjust parameters and view image processing results.

FUTURE SCOPE

- Integrate webcam or live video support
- Add multi-object labeling with classification
- Enable image batch processing
- Export object coordinates or cropped ROIs
- Use advanced techniques like YOLO or DeepLab

REFERENCES

- OpenCV Documentation
- Streamlit Docs
- NumPy, PIL Documentation

- GitHub : https://github.com/Fakruddin002/Image-Segmentation-and-Object-Detection.git

Thank you