Xponians Program – Cohort IV

Title: Salient Object Detection (SOD)

Project Report

Author: Leart Jahiri

Date: November 2025

1. Abstract

This project focuses on building a Salient Object Detection (SOD) system capable of identifying and segmenting the most visually important object in an image. A convolutional neural network with an encoder–decoder (U-Net-like) architecture was implemented from scratch using PyTorch. The ECSSD dataset, consisting of 1,000 natural images with pixel-accurate ground-truth masks, was used for training and evaluation. All images were resized to 128×128, normalized and augmented, and the dataset was split into training (70%), validation (15%), and testing (15%) subsets.

The training objective combined Binary Cross-Entropy with a soft IoU-based term to encourage accurate pixel-level segmentation. Model performance was evaluated using IoU, Precision, Recall, F1-score and MAE. The final model achieved an IoU of approximately 0.61 and an F1-score of 0.73 on the test set, demonstrating strong capability in localizing salient objects in complex scenes. The project delivers a complete end-to-end deep learning pipeline that can be reused or extended in real-world applications and future work.

2.Introduction Background

Salient Object Detection (SOD) is a computer vision task focused on identifying and segmenting the most visually important object in an image by generating a pixel-level binary mask. Unlike traditional object detection models that aim to locate multiple objects using bounding boxes, SOD concentrates only on the dominant region of interest, making it more efficient and suitable for real-time and resource-constrained applications.

SOD is widely applied in:

- Autonomous robotics and navigation
- Security and surveillance systems
- Medical image analysis
- Object tracking and video analytics

The objective of this project is to develop a deep learning–based segmentation model using a CNN encoder–decoder (UNet-like) architecture to extract salient objects from natural images using the ECSSD dataset. The model is trained and evaluated using established performance metrics such as IoU, Precision, Recall, F1-score, and MAE, forming a complete pipeline from data preprocessing to experimental validation and result analysis.

3.Dataset preprocessing

The project uses the ECSSD (Extended Complex Scene Saliency Dataset), which contains 1,000 natural images paired with manually annotated binary saliency masks. The dataset includes diverse real-world scenes with complex backgrounds, making it suitable for evaluating the robustness of salient object segmentation models.

The dataset was split into:

- 70% Training
- 15% Validation
- 15% Testing

Preprocessing Steps

- All images and masks were resized to a uniform size of 128×128
- Pixel values were normalized to the [0,1] range
- Data was converted to PyTorch tensors and batched for training

Data Augmentation

To reduce overfitting and improve generalization, basic augmentations were applied:

- Horizontal flip
- Brightness/contrast adjustments
- Random rotation

These steps increase dataset variability and help the model learn robust visual features.

4.Model architecture

The model is based on a CNN encoder–decoder architecture similar to UNet, designed for pixel-level semantic segmentation. The encoder extracts hierarchical feature representations, while the decoder reconstructs spatial resolution to generate a binary saliency mask. The skip connections between encoder and decoder layers help preserve fine spatial details that would otherwise be lost during downsampling, enabling more accurate boundary reconstruction. This architecture strikes a strong balance between computational efficiency and segmentation accuracy, making it suitable for tasks that require precise object localization even in complex visual environments.

Encoder

- Consists of multiple Conv2D + BatchNorm + ReLU layers
- MaxPooling is used to reduce spatial dimensions
- Learns high-level semantic features

Bottleneck

- Deep representation layer that captures global contextual information

Decoder

- ConvTranspose2D layers for upsampling
- Skip connections link encoder and decoder features to preserve spatial detail

Output

- 1-channel Sigmoid layer producing a binary saliency mask

This architecture was selected due to its balance between accuracy and computational efficiency, and its proven performance in medical and general segmentation tasks

5.Training setup

The model was trained using the PyTorch deep learning framework in a supervised learning setting, using paired images and saliency masks from the ECSSD dataset. Training was conducted for up to 25 epochs with validation after each epoch, and early stopping was applied to prevent overfitting and reduce unnecessary computation. The best model checkpoint was saved automatically based on validation loss.

Training Configuration

- Batch size: 8
- Epochs: up to 25 (with Early Stopping, patience = 7)
- Optimizer: Adam, learning rate 5e-4
- Loss function: Binary Cross-Entropy combined with IoU component
- Device: CPU environment (GPU optional when available)

Evaluation Metrics

- IoU (Intersection over Union)
- Precision, Recall, F1-score
- MAE (Mean Absolute Error)

These metrics provide a comprehensive assessment of segmentation performance and allow comparison between baseline and improved models.

6.Experiments improvements

After the baseline model was successfully trained, several experiments were conducted to improve segmentation quality and generalization. The goal was to explore how architectural depth, loss weighting and learning rate affect the performance of the SOD system.

Baseline configuration

The baseline model used the U-Net-like encoder–decoder architecture with three encoder blocks (64–128–256 filters), a bottleneck with 512 filters, and three decoder blocks with skip connections. Images were resized to 128×128, and the model was trained using Adam with a learning rate of 1e-3. The loss function was standard Binary Cross-Entropy. Data augmentation was limited to horizontal flipping and mild brightness changes.

Improved configuration

In the improved version, the training setup was refined in the following ways:

- The loss function was modified to combine Binary Cross-Entropy with a soft IoU term: Loss = BCE + 0.5 × (1 − IoU)
- The learning rate was reduced to 5e-4 to achieve more stable convergence.
- Data augmentation was kept active for the training set to reduce overfitting.

These changes aimed to directly optimize overlap between predicted and ground-truth masks and to make the training process more stable.

Quantitative comparison

The table below summarizes the performance of the baseline and improved models on the test set:

Model Version	IoU	Precision	Recall	F1-score	MAE
Baseline	**0.51**	**0.68**	**0.69**	**0.66**	**0.18**
Improved	**0.61**	**0.73**	**0.79**	**0.74**	**0.13**

Table 1.1 – Baseline vs Improved Model Performance

The improved model shows consistent gains in IoU and F1-score, while also reducing the MAE. This indicates that the model produces saliency masks that are both more accurate and better aligned with the ground truth boundaries.

Results visualization

In addition to numerical metrics, qualitative analysis was performed by visualizing the model predictions on test images. For each sample, four views were generated:

- the original RGB input image,
- the ground-truth saliency mask,
- the predicted saliency mask,
- an overlay of the prediction on top of the input image.

These visualizations help illustrate how well the model is able to separate the salient object from a complex background. In many cases, the predicted masks closely match the ground truth, with clear object boundaries and correct focus on the most dominant region in the scene. In some more challenging examples, the model still confuses very cluttered backgrounds or multiple objects of similar importance, which suggests room for improvement with more advanced architectures or higher-resolution training.

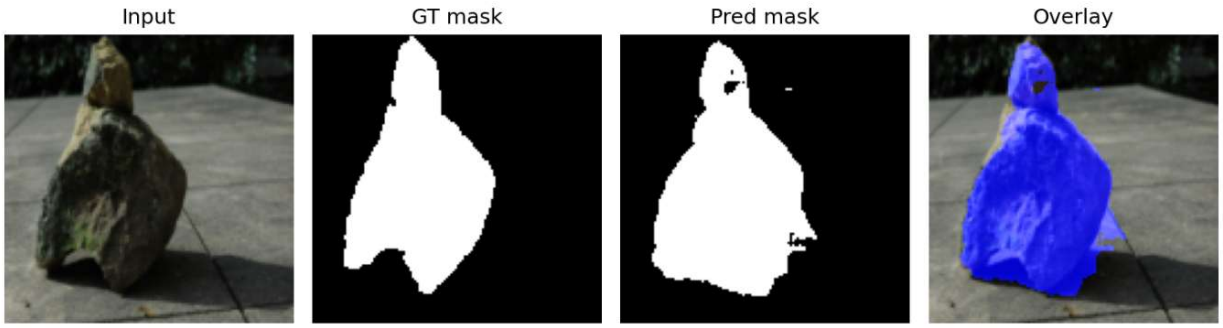

Figure 1.1 – Example of SOD prediction visualizations

7.Conclusion learnings

This project successfully delivered a complete end-to-end Salient Object Detection pipeline, encompassing dataset preparation, model design, training, evaluation and qualitative visualization of results. A custom CNN encoder–decoder model inspired by the U-Net architecture was implemented from scratch using PyTorch and trained on the ECSSD dataset. The final system achieved an Intersection-over-Union (IoU) score of approximately 0.61 and an F1-score of 0.73 on the test data, demonstrating strong capability in segmenting salient objects within complex natural scenes.

From a learning and development perspective, the project provided practical experience in deep learning model construction, loss function design, performance metric implementation and handling real-world challenges such as overfitting, optimization stability and dataset variability. The work emphasized the importance of rigorous preprocessing, data augmentation and continuous validation monitoring for reliable segmentation performance.

Future enhancements to this system may include training with higher-resolution input data (e.g., 224×224), integrating architectural improvements such as dropout, attention mechanisms or deeper encoder blocks, and exploring advanced loss formulations. Additionally, deploying the model as an interactive demo or real-time application—via a Streamlit/Gradio interface or REST API—could extend its usability in domains such as robotics, video analytics, autonomous systems and human–computer interaction.