**PROJECT ABSTRACT**

# Real-Time Background Removal using U2-Net

*A U2-Net background remover exported to ONNX for CPU-speed photo cutouts, delivered as a built-to-order student project with code, model files, report, PPT and viva kit.*

## 01. Title

Real-Time Background Removal using U2-Net

## 02. Introduction / Background

Cutting a subject out of a photo used to mean painstaking manual masking; a single neural network now does it in a fraction of a second. U2-Net (Qin et al., arXiv:2005.09007, Pattern Recognition 2020) is the salient-object-detection architecture known for crisp edges on hair and fine detail: its nested U-structure stacks U-blocks inside U-blocks, capturing fine local boundaries and broad global context at once, with deep supervision at multiple side outputs. Trained on DUTS-TR and exported to ONNX, it runs near-real-time inference on an ordinary CPU. This project implements that pipeline end to end with a Flask demo that returns transparent PNGs and background replacements.

## 03. Problem Statement

Manual background removal is slow and tedious, while existing automatic tools are either paid services or black boxes with no inspectable model. The project addresses this by building an open, reproducible U2-Net pipeline — training on a public dataset, exporting to ONNX for CPU inference, and wrapping it in a demo app whose segmentation quality is measured with standard metrics on a public test split.

## 04. Objectives

- Train U2-Net on DUTS-TR for salient-object segmentation with deep supervision.
- Export the trained model to ONNX for portable near-real-time CPU inference, with near-real-time on typical photos as the design target.
- Compute MAE and max F-measure on the DUTS-TE split with the evaluation notebook during the build.
- Build a Flask demo app with photo upload, adjustable binarization threshold, transparent-PNG download and background replacement.

- Deliver the complete student kit: source code, model files, report, PPT and viva Q&A document.

## 05. Proposed System / Working

The system has three stages. (1) Training: DUTS-TR images with saliency masks train the nested U-structure of U2-Net, with deep supervision applied at multiple side outputs. (2) Export: the trained PyTorch model is converted to ONNX for optimized CPU inference. (3) Demo application: each uploaded photo is resized and normalized, passed through the ONNX model to produce a saliency probability map, binarized at a configurable threshold, smoothed, and used as the alpha channel; the Flask app composites the subject onto transparency or a replacement background and serves the PNG for download.

## 06. Technologies / Components Used

- Python 3.10, PyTorch (U2-Net training)
- ONNX Runtime (CPU inference in the demo app)
- OpenCV, Pillow (masking, compositing, PNG export)
- NumPy (probability maps, thresholding)
- Matplotlib (saliency visualizations, metric plots)
- Flask demo web app with photo-upload UI
- DUTS-TR dataset (salient-object-detection training set)

## 07. Key Features

- U2-Net salient-object segmentation with crisp edge output
- ONNX export for fast CPU inference — no GPU needed at demo time
- Flask demo with transparent-PNG download and background replacement
- Background blur effect and batch mode for multiple images
- Adjustable binarization threshold with live preview
- Evaluation procedure computing MAE and F-measure on DUTS-TE

## 08. Applications / Use Cases

- Profile-picture and product-photo cutout demos (prototype scope)
- Teaching material for semantic segmentation, nested architectures and ONNX deployment
- Reference build for other saliency and matting tasks
- Baseline for comparing newer segmentation models

## 09. Results & Scope

The deliverable is a design-target-driven build, not a pre-measured product. The evaluation notebook computes mean absolute error and max F-measure between predicted masks and ground truth on the DUTS-TE split during the build; the report presents these as the build's measured results and shows honest failure cases on cluttered scenes. Near-real-time inference on typical photos via ONNX is the design target. Scope is still-image background removal — no video matting — and this is an educational prototype rather than a production photo-editing service.

## 10. Conclusion

Real-Time Background Removal using U2-Net delivers a complete applied segmentation project: a real editing problem, the public DUTS benchmark pair, a U2-Net model trained through a fully logged experiment, ONNX export for CPU-speed inference, and a demo app that makes the result downloadable in seconds. It gives the student genuine material on modern segmentation architectures and practical model deployment.

## 11. References


- Qin, X. et al. — U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection (arXiv:2005.09007, Pattern Recognition 2020), the model implemented in this build.
- Wang, L. et al. — Learning to Detect Salient Objects with Image-Level Supervision (CVPR 2017), the paper introducing the DUTS dataset.
- ONNX Runtime documentation (onnxruntime.ai) — the CPU inference engine used in the demo app.
- PyTorch documentation (pytorch.org) — training tooling used in the build.