# VISIONEXTRACT

## ISOLATION FROM IMAGES USING IMAGE SEGMENTATION

SUBMITTED BY:ILAKIYA EMILY JOSEPH IGNATIUS

# PROBLEM STATEMENT:

- Manual extraction of the main subject from images is slow and inconsistent.
- Required in applications like photography, digital art, AR/VR, virtual conferencing, and background replacement.

# GOAL:

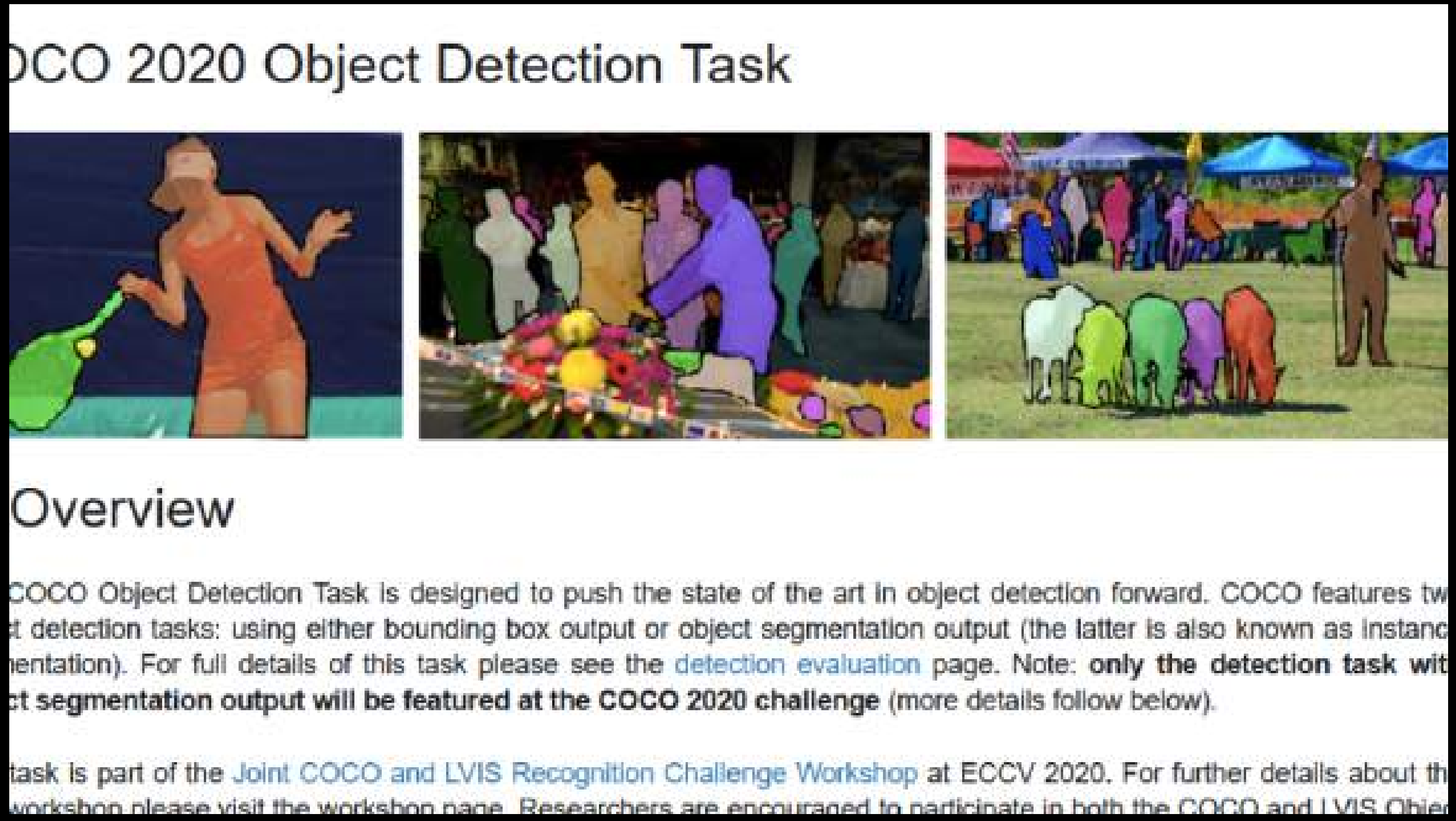
- Automatically detect and extract the main subject.
- Output an image where only the subject is visible; background is completely black.

# IMPACT / MOTIVATION:

- Saves time in media editing pipelines.
- Provides consistent and high-quality subject isolation.
- Enables automation for diverse imaging applications.

# EXAMPLE:

# DATASET OVERVIEW



DCO 2020 Object Detection Task

Overview

COCO Object Detection Task is designed to push the state of the art in object detection forward. COCO features tw t detection tasks: using either bounding box output or object segmentation output (the latter is also known as instanc entation). For full details of this task please see the detection evaluation page. Note: **only the detection task wit ct segmentation output will be featured at the COCO 2020 challenge** (more details follow below).

task is part of the Joint COCO and LVIS Recognition Challenge Workshop at ECCV 2020. For further details about th workshop please visit the workshop page. Researchers are encouraged to participate in both the COCO and LVIS Objec

**Dataset Name: COCO 2017**

Source: Kaggle COCO 2017 Dataset

**Key Details:**

- Size: 118,000+ training images with pixel-wise mask annotations
- Mask Annotations: Binary masks marking the main subject for each image
- Diversity: Covers 80+ object categories, varied backgrounds, lighting, and perspectives

**Why This Dataset:**

- Well-labeled for semantic segmentation tasks
- Diverse scenarios help the model generalize across real-world images

# MILESTONE PLAN

**Milestone 1 (Week 1–2):**

- Project Initialization & Dataset Acquisition
- Data Preprocessing & Augmentation

**Milestone 2 (Week 3–4):**

- Initial Model Training
- Validation & Fine-Tuning

**Milestone 3 (Week 5–6):**

- Improve Data Preprocessing & Experiment with Architectures
- Model Inference on Unseen Images

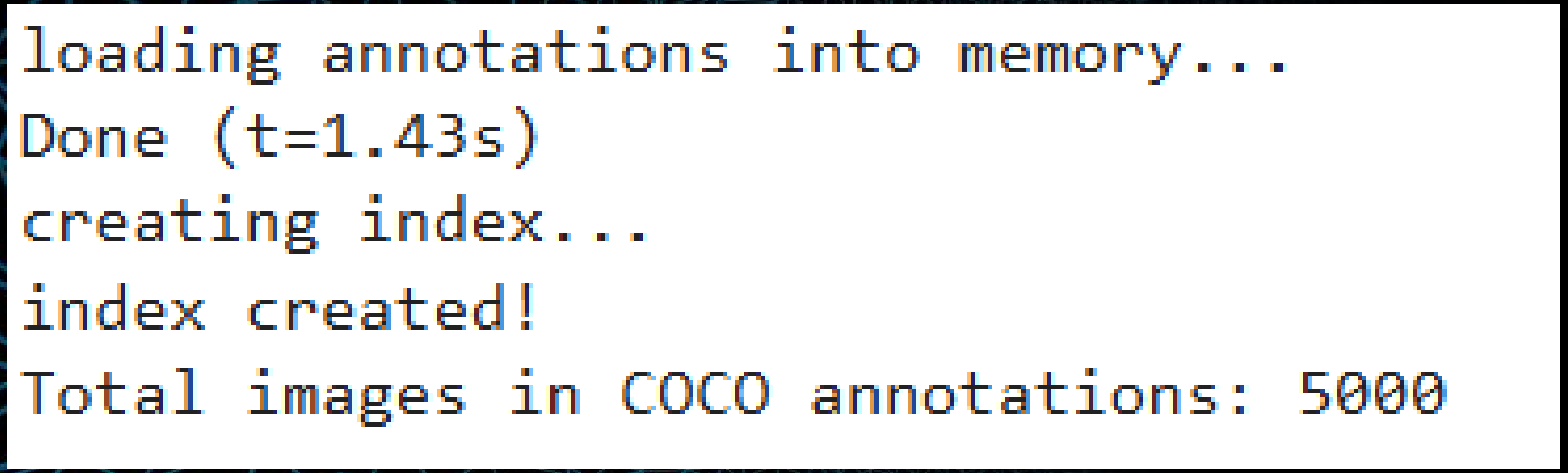
**Milestone 4 (Week 7–8):**

- Full Pipeline Integration & Web Interface
- Documentation, Presentation, and Final Demo

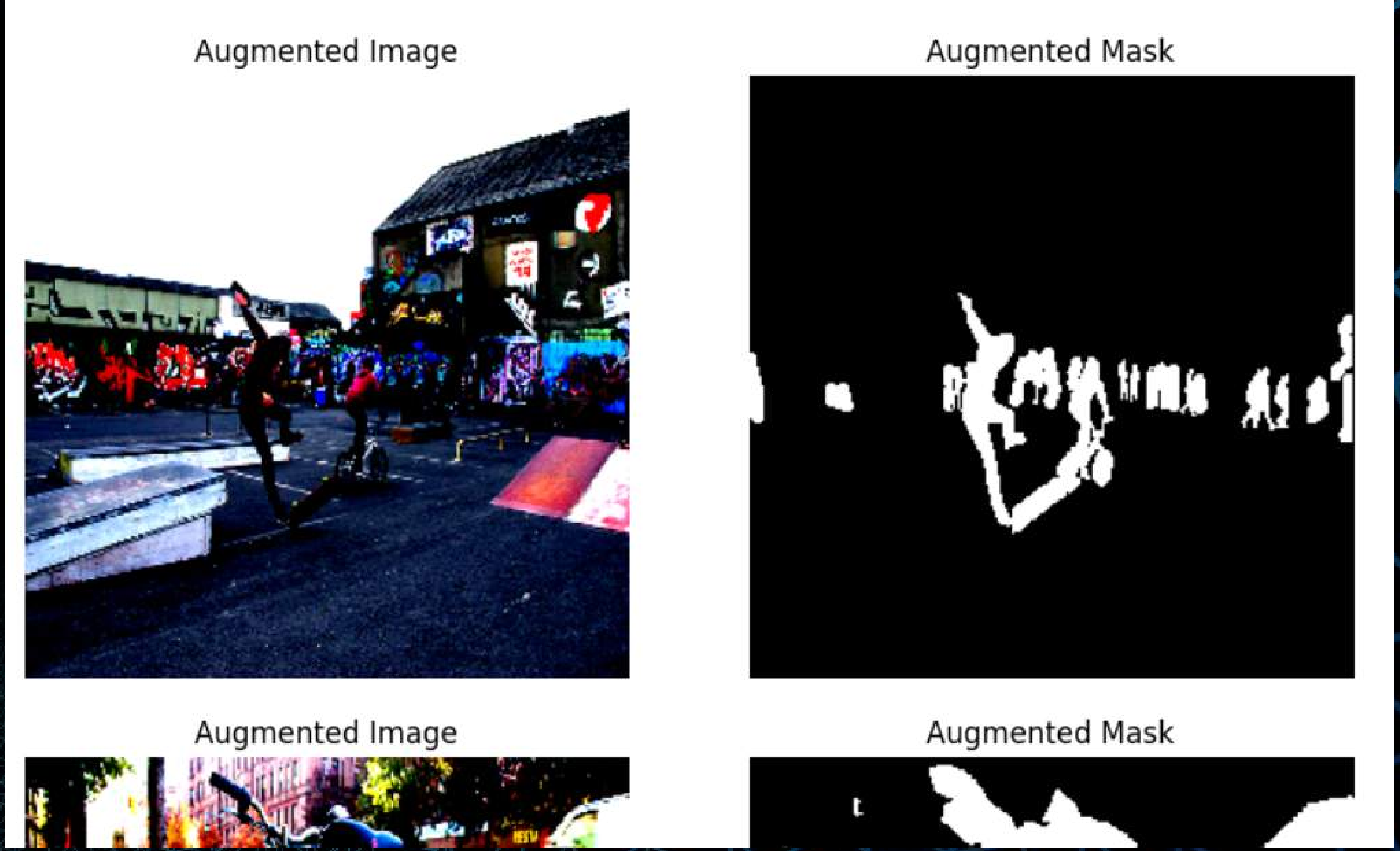
# MILESTONE 1 – DATA PREPROCESSING & AUGMENTATION

- Objective: Prepare dataset for training a subject-isolation model.
- Dataset: COCO 2017 – 5000 validation images with mask annotations.
- Preprocessing Steps:
  - Resized images to 256×256
  - Normalized pixel values for model input
  - Converted multi-class masks into binary masks for main subject
- Augmentation Techniques:
  - Horizontal/vertical flips, brightness & contrast adjustments
  - Rotation, shifting, and scaling
- How it was achieved:
  - Used PyCOCOtools to load images & masks
  - Applied Albumentations library for augmentation & preprocessing
  - Verified results visually with before/after images

# MILESTONE 1 – DATA PREPROCESSING & AUGMENTATION



```
loading annotations into memory...
Done (t=1.43s)
creating index...
index created!
Total images in COCO annotations: 5000
```



# MILESTONE 2 — MODEL TRAINING & FINE-TUNING

Phase 1:

- Implemented UNet (ResNet34) as baseline using Segmentation Models PyTorch.

Phase 2:

- Upgraded to DeepLabV3 (ResNet50) for better context & boundary precision.

Training:

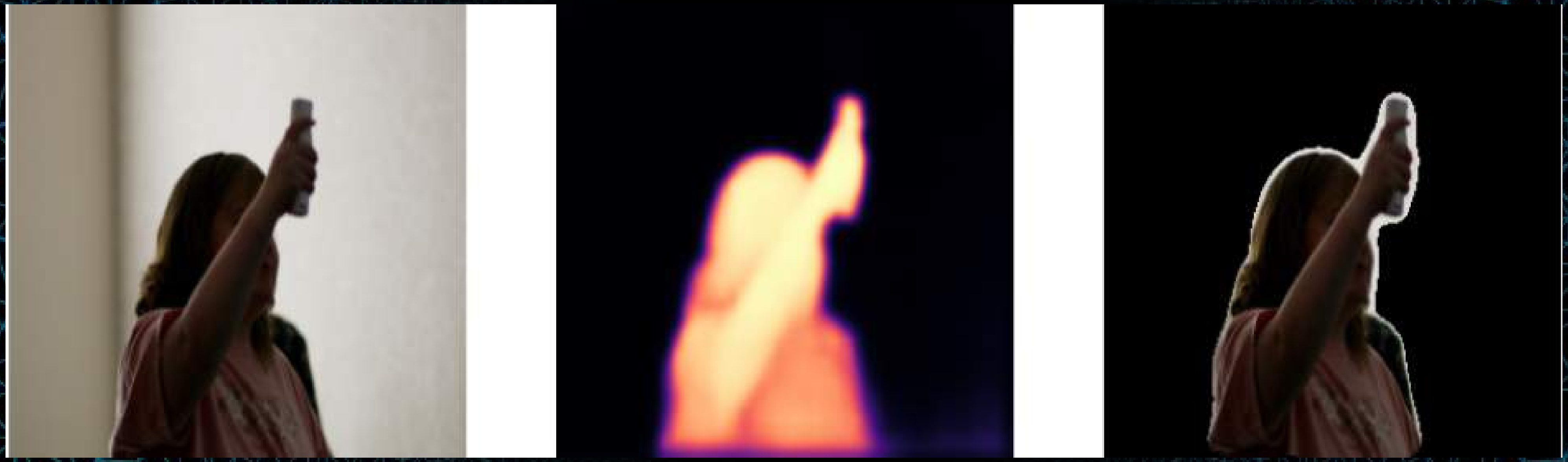
- Dataset: COCO 2017 with custom binary masks
- Augmentations: flips, color jitter, resize (Albumentations)
- Used Mixed Precision (AMP) for memory efficiency

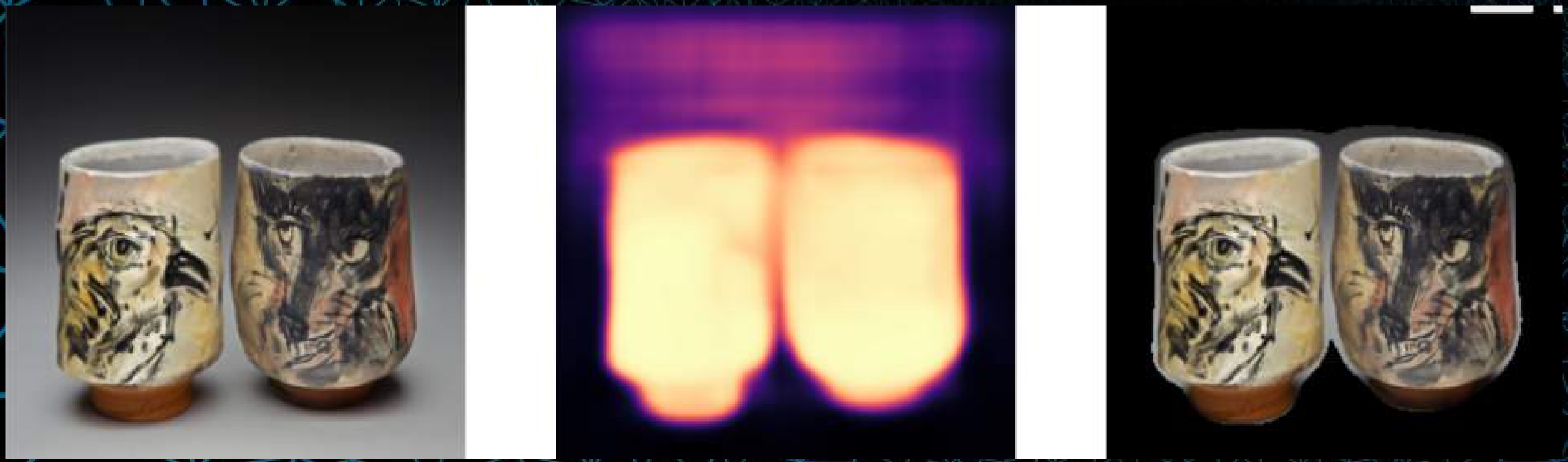
Optimization & Validation:

- Metrics: IoU, Dice, Pixel Accuracy
- Fine-tuned with backbone unfreezing + TTA
- Visualized results — consistent subject extraction

Tools: PyTorch · Albumentations · COCO API · Matplotlib

# MILESTONE 2 – MODEL TRAINING & FINE-TUNING





**Results:**

**Average Pixel Accuracy: 🟩 95.0%**

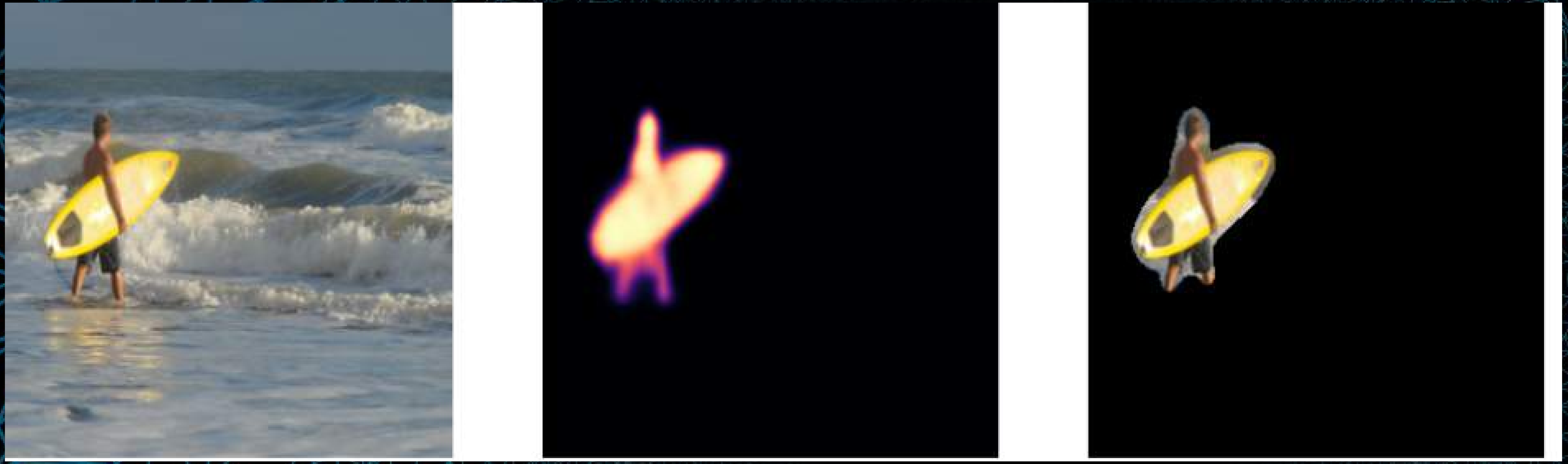
# MILESTONE 3: REFINEMENT & INFERENCE DEPLOYMENT

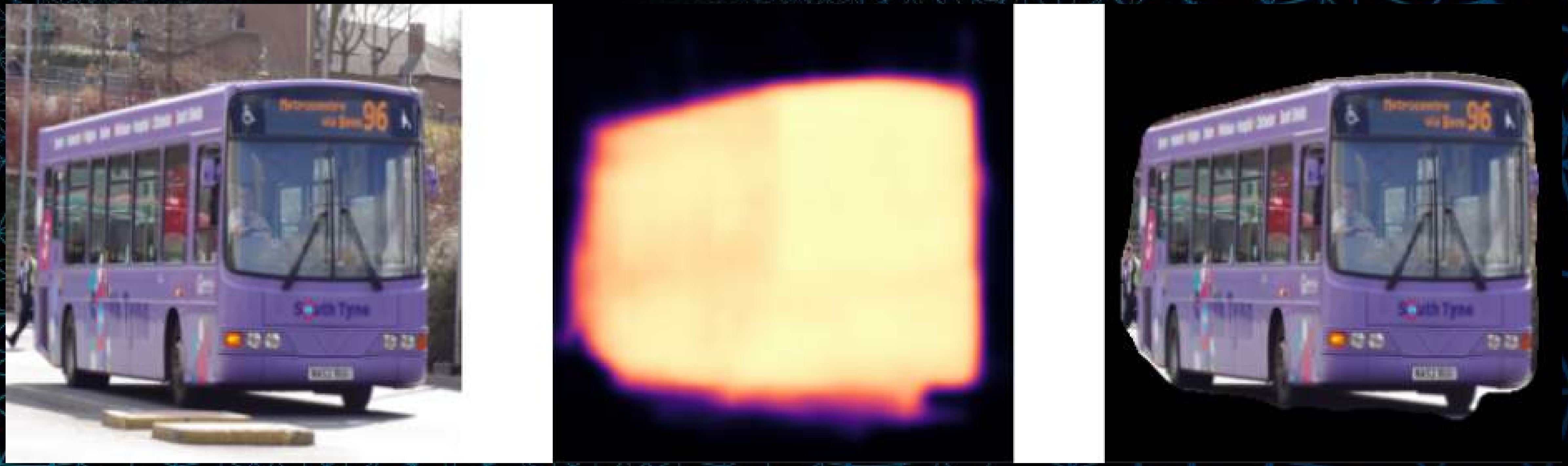
Milestone 3: Refinement & Inference Deployment (Weeks 5–6)
Phase 1: Refinement & Experiments
Enhanced preprocessing with mask smoothing, adaptive thresholding, and morphological post-processing.
Experimented with TTA (Test-Time Augmentation): multi-scale, flip & rotation-based inference.
Compared architectures (UNet → DeepLabV3-ResNet50).
Evaluated metrics (IoU, Dice, Pixel Accuracy) for model robustness.

Phase 2: Inference & Deployment
Automated full inference pipeline using PyTorch & Streamlit.
Supported real-world testing on random web images for generalization.
Applied mask refinement using Gaussian smoothing + Otsu threshold + morphological cleanup.
Saved & visualized segmented results for 25+ test samples.

Tools: PyTorch · torchvision · Albumentations · Matplotlib · skimage · Streamlit

# MILESTONE 3: REFINEMENT & INFERENCE DEPLOYMENT





# MILESTONE 4: FULL PIPELINE & WEB UI

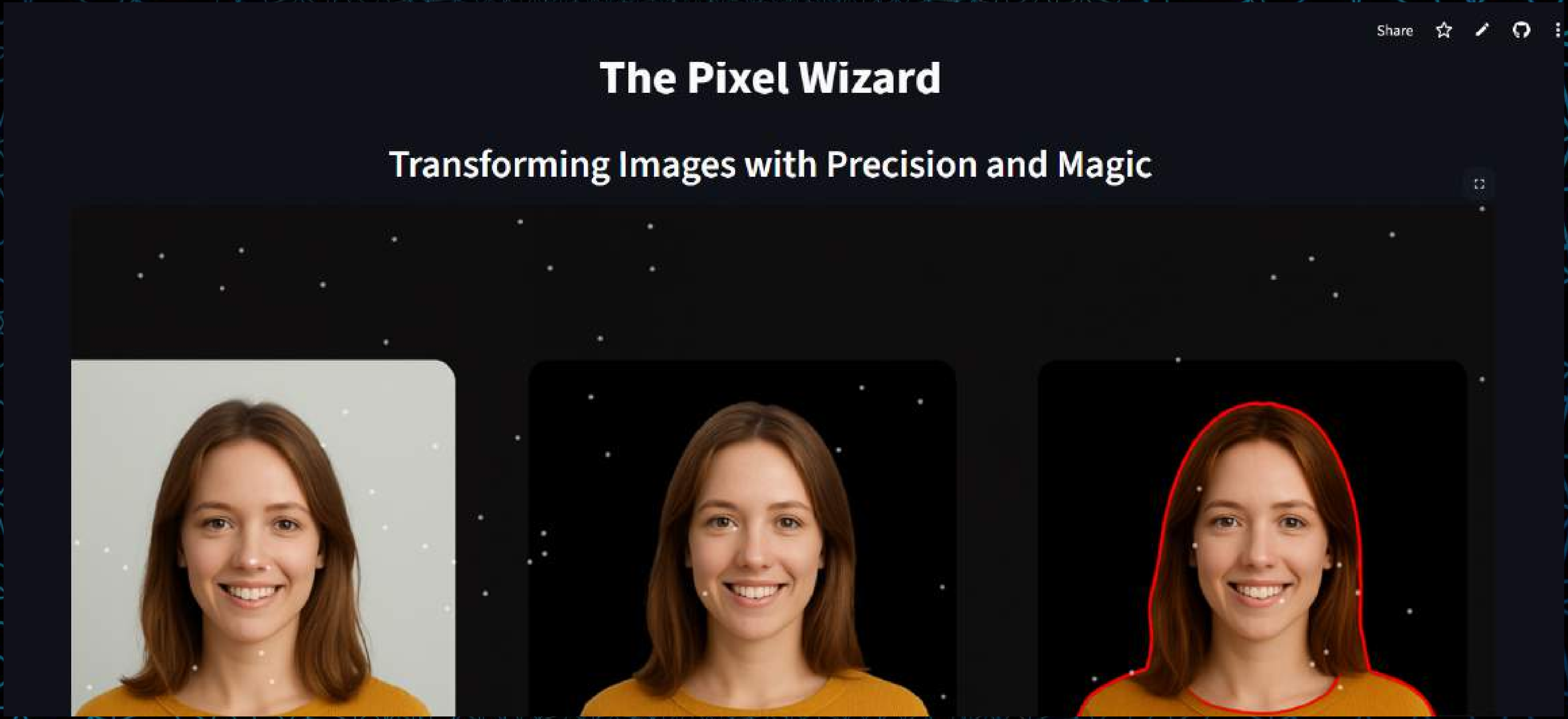
Objectives:
Build a user-friendly web application for image segmentation.
Integrate preprocessing, model inference, mask refinement, and output generation.
Enable background removal, custom edge overlays, and downloadable results.

Key Features:
TTA Inference: Multi-scale + flip-based predictions for robust masks.
Mask Morphology: Gaussian smoothing, Otsu threshold, small object removal, dilation & closing.
UI Controls:
Slider for min object size & dilation
Edge color & thickness
Background selection (transparent, black, white, custom)

Outputs:
Original image
Segmented / background-removed image
Edge overlay visualization
Download options for all outputs

Technologies Used:
PyTorch, torchvision, Streamlit, PIL, NumPy, skimage, SciPy

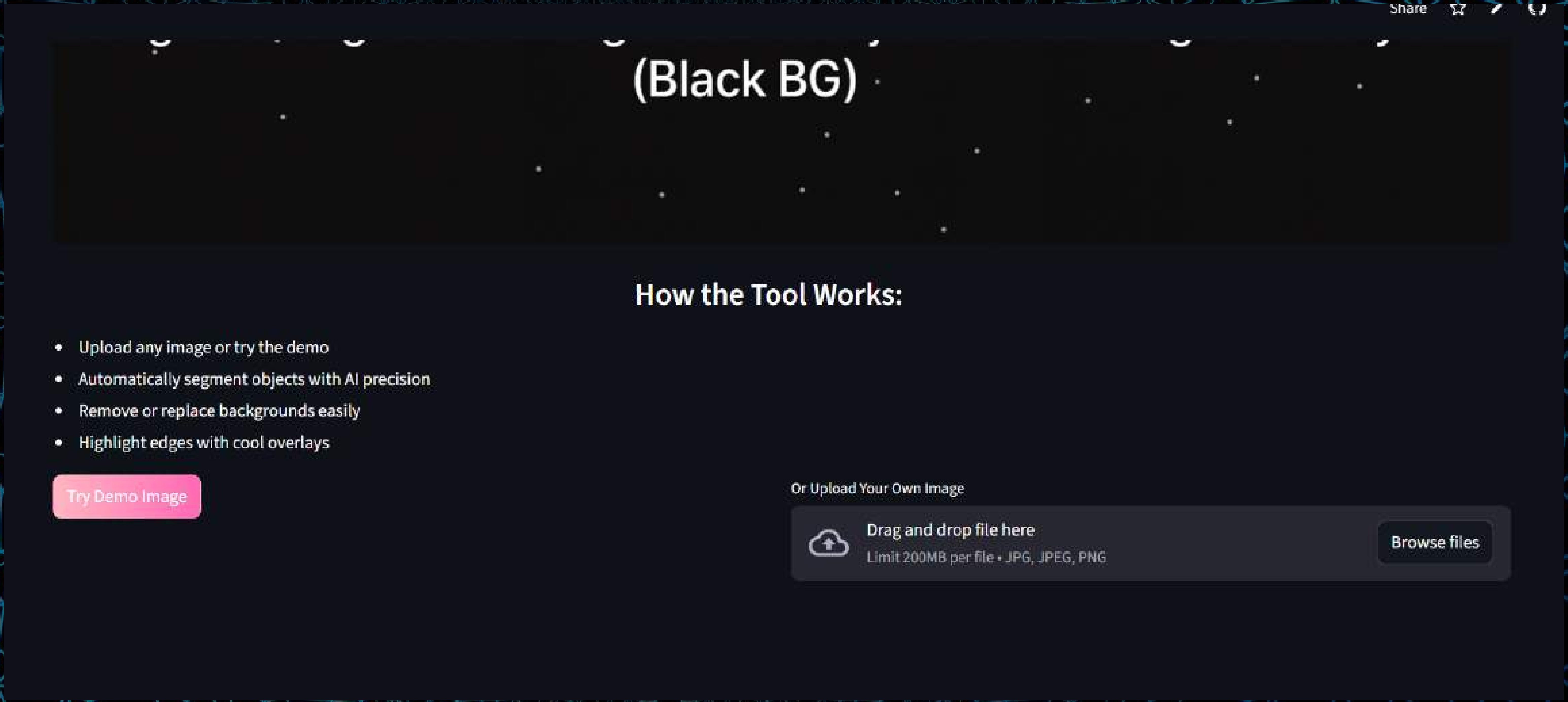
# MILESTONE 4: FULL PIPELINE & WEB UI



The Pixel Wizard

Transforming Images with Precision and Magic

# MILESTONE 4: FULL PIPELINE & WEB UI



(Black BG)

### How the Tool Works:

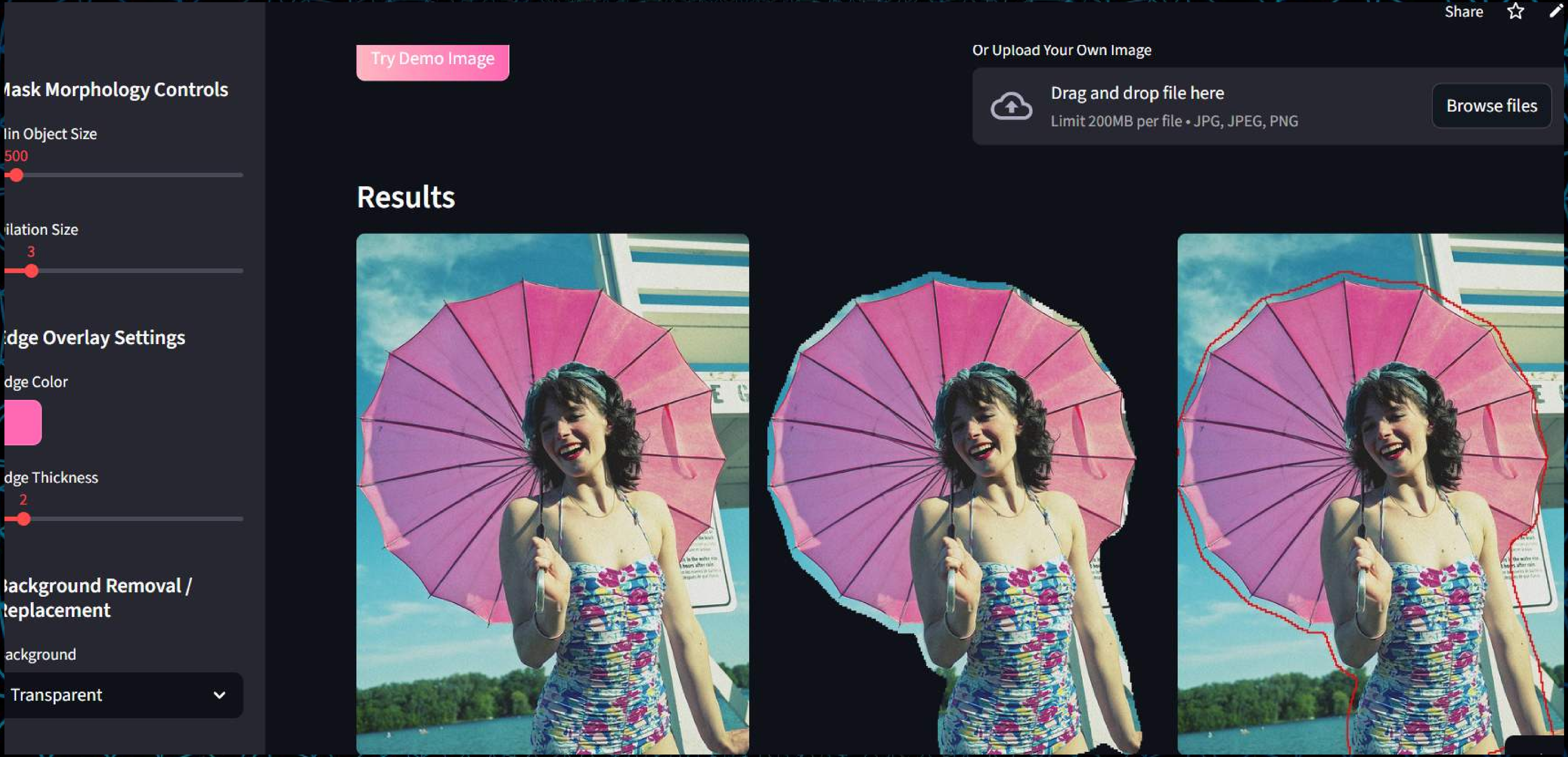
- Upload any image or try the demo
- Automatically segment objects with AI precision
- Remove or replace backgrounds easily
- Highlight edges with cool overlays

Or Upload Your Own Image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

# MILESTONE 4: FULL PIPELINE & WEB UI



lask Morphology Controls

lin Object Size
500

ilation Size
3

dge Overlay Settings

dge Color

dge Thickness
2

ackground Removal / eplacement

ackground
Transparent

Or Upload Your Own Image

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Results

# CHALLENGES & LEARNINGS

**Challenges:**

- Handling small objects & noisy masks → needed morphology refinements.
- Balancing accuracy vs. GPU memory → mixed precision & TTA optimization.
- Adapting model to unseen images → robust generalization beyond COCO dataset.
- Managing multi-step pipeline → integration of preprocessing, inference, and UI.

**Learnings:**

- DeepLabV3 + ResNet50 improved boundary precision vs UNet baseline.
- TTA + morphology drastically reduces artifacts & smooths masks.
- Streamlit allows rapid prototyping for interactive AI tools.
- Visualization is key → seeing masks & overlays early prevents pipeline errors.

# CONCLUSION & NEXT STEPS / FUTURE WORK

**Impact Recap:**

- Fully automated end-to-end segmentation tool.
- Works on any uploaded image, not just dataset samples.
- Interactive UI allows custom background & edge overlays.

**Future Work / Improvements:**

- Integrate multi-class segmentation (beyond binary masks).
- Add real-time webcam or video support.
- Implement AI-powered background replacement (e.g., stylized, scene-aware).
- Optimize inference further → faster TTA or quantized models for low-end devices.

# THANK YOU