

# NEURAL SEGMENTELLO

REFINING USER INPUT FOR  
HIGH-PRECISION OBJECT  
SEGMENTATION

«Allab: Computer Vision and NLP»  
Project Presentation  
11/06/2025



A PROJECT BY:  
ANDREA GENTILINI (2043590)  
MICHELE MAGRINI (2066913)  
LEO PETRARCA (2087113)  
IACOPO SCANDALE (2085989)



# INTRODUCTION & CONTEXT

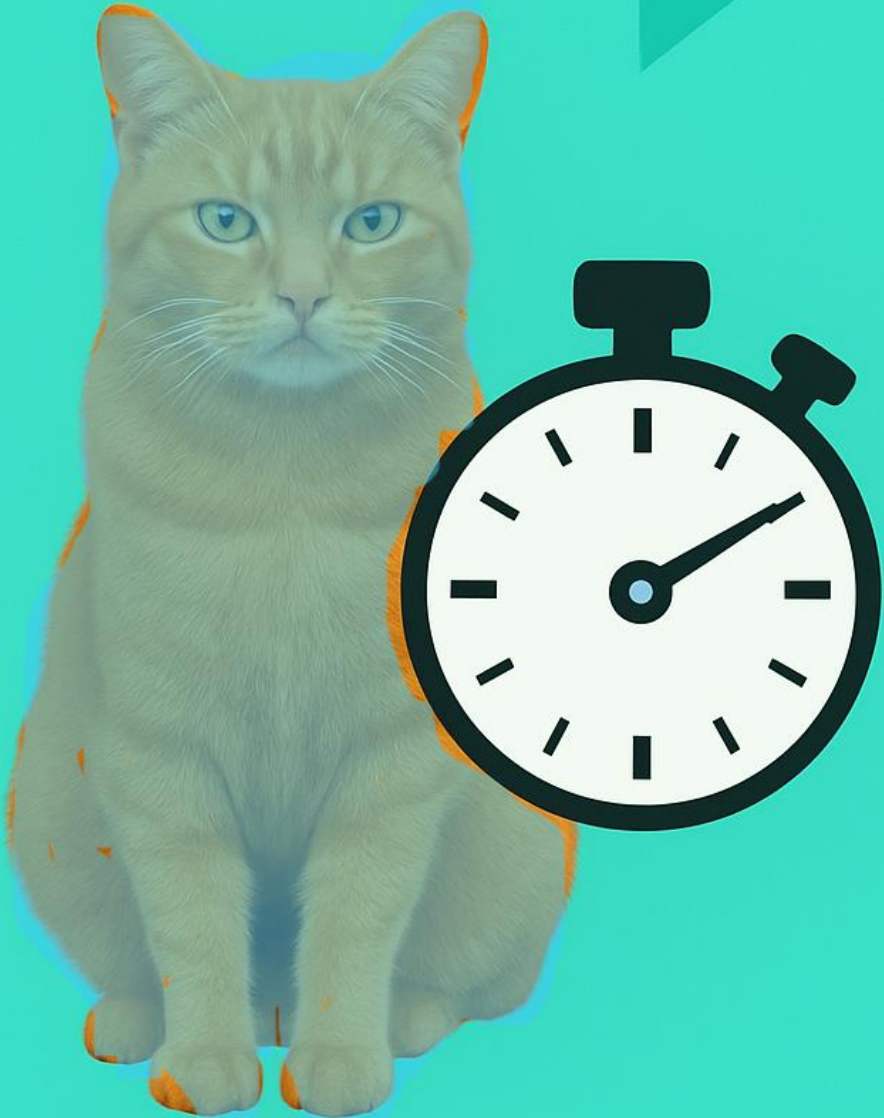
- Interactive object segmentation aims to accurately isolate objects within an image using minimal human input.
- This project focuses on refining a **coarse binary mask**, mimicking a user-drawn brush stroke, into a **high-precision segmentation mask**.
- Our approach bridges **human intuition** and **deep learning capabilities**, reducing annotation effort without sacrificing accuracy.

# **FAST VS ACCURATE-WHY NOT BOTH?**

Precise segmentation is time-consuming and demands expert knowledge

A single brush stroke is fast, intuitive – but too rough to be used directly

What if neural networks could refine it into pixel-perfect masks?






# WHAT NEURALSEGMENTELLO AIMS TO ACHIEVE

1. Design and train lightweight U-Net architectures for refining coarse user masks.
2. Experiment with architectural variants: → residual connections, attention-based decoding.
3. Evaluate using segmentation-specific losses: → Binary Cross-Entropy, Dice, Boundary Loss.
4. Ensure reproducibility via:
  - Fixed seeds and config files
  - Reduced dataset and public codebase

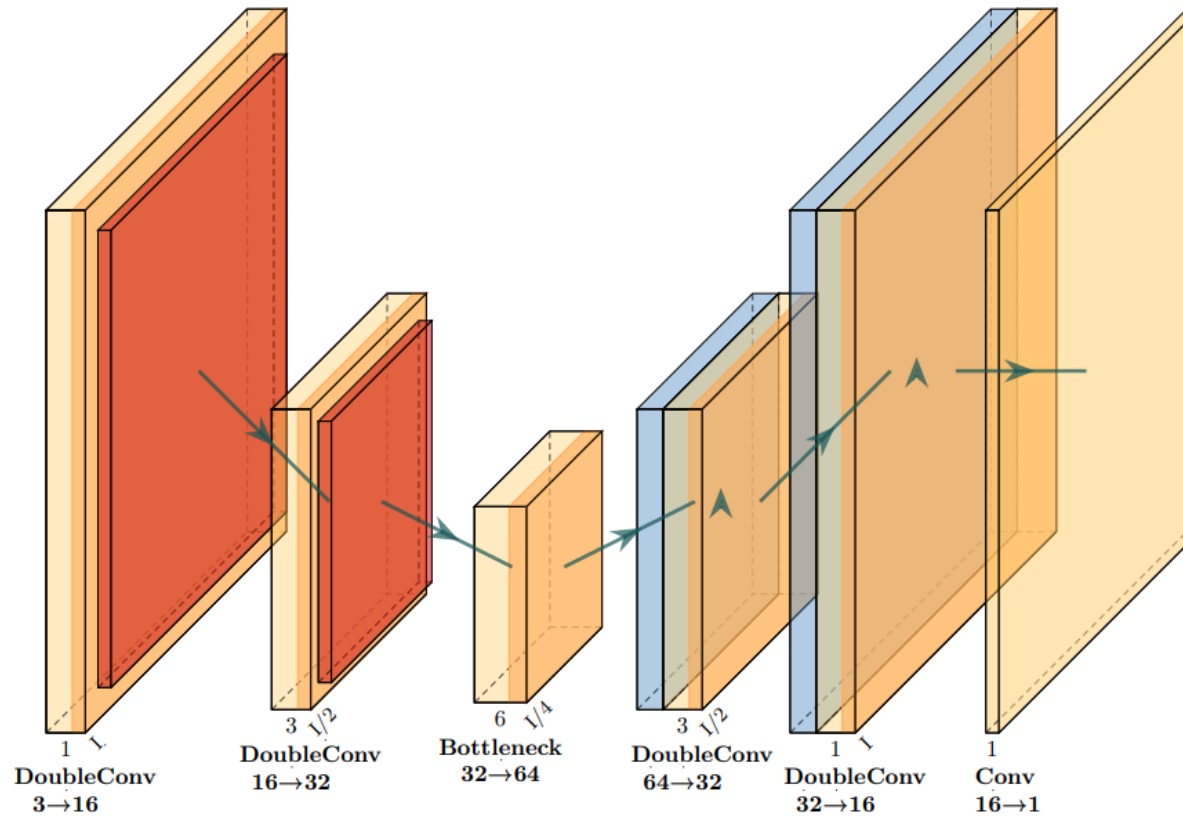


# DATASET & USER INPUT SIMULATION

- WE USE A SUBSET OF THE MICROSOFT  **COCO**  
Common Objects in Context  
DATASET A LARGE-SCALE BENCHMARK WITH OVER 200,000  
ANNOTATED IMAGES.
- TO SIMULATE REALISTIC USER INPUT, WE GENERATED  
SYNTHETIC BRUSH STROKES BY INTRODUCING RANDOM  
CIRCULAR OCCLUSIONS AND GAUSSIAN BLURRING.

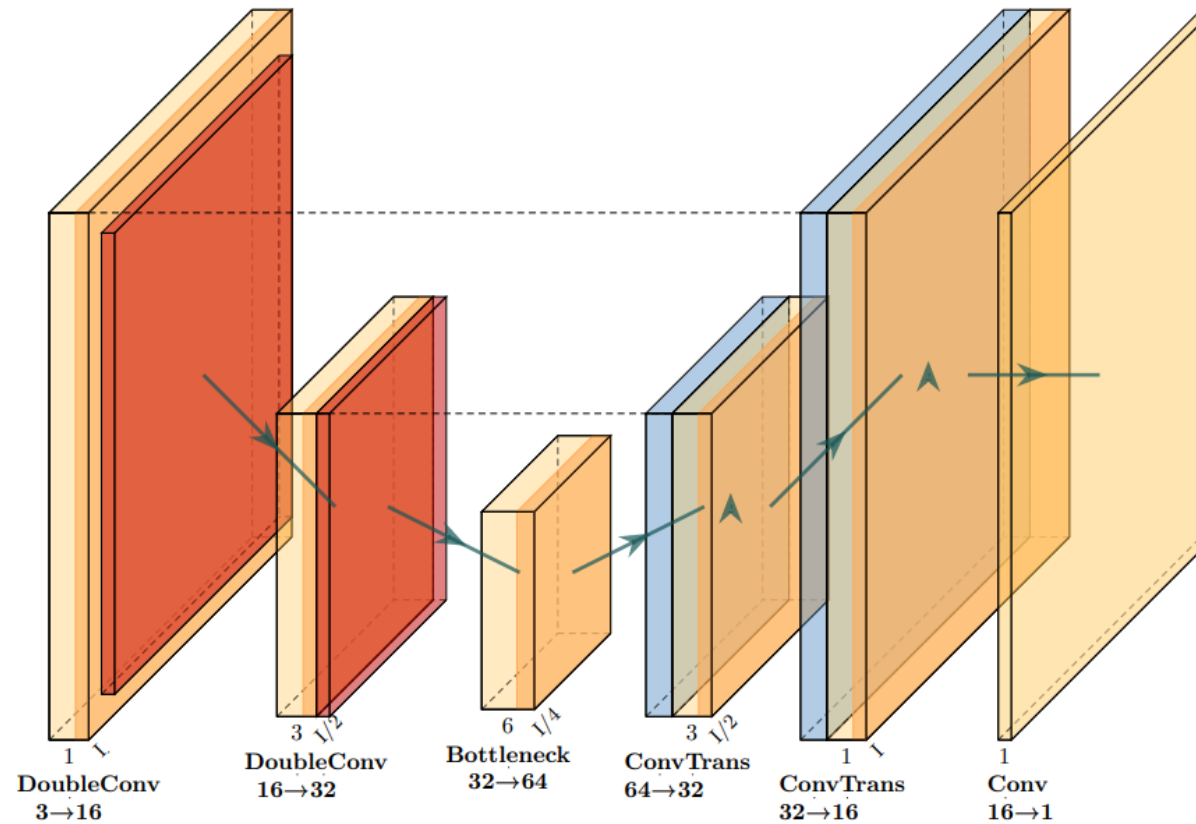


# OUR MODELS



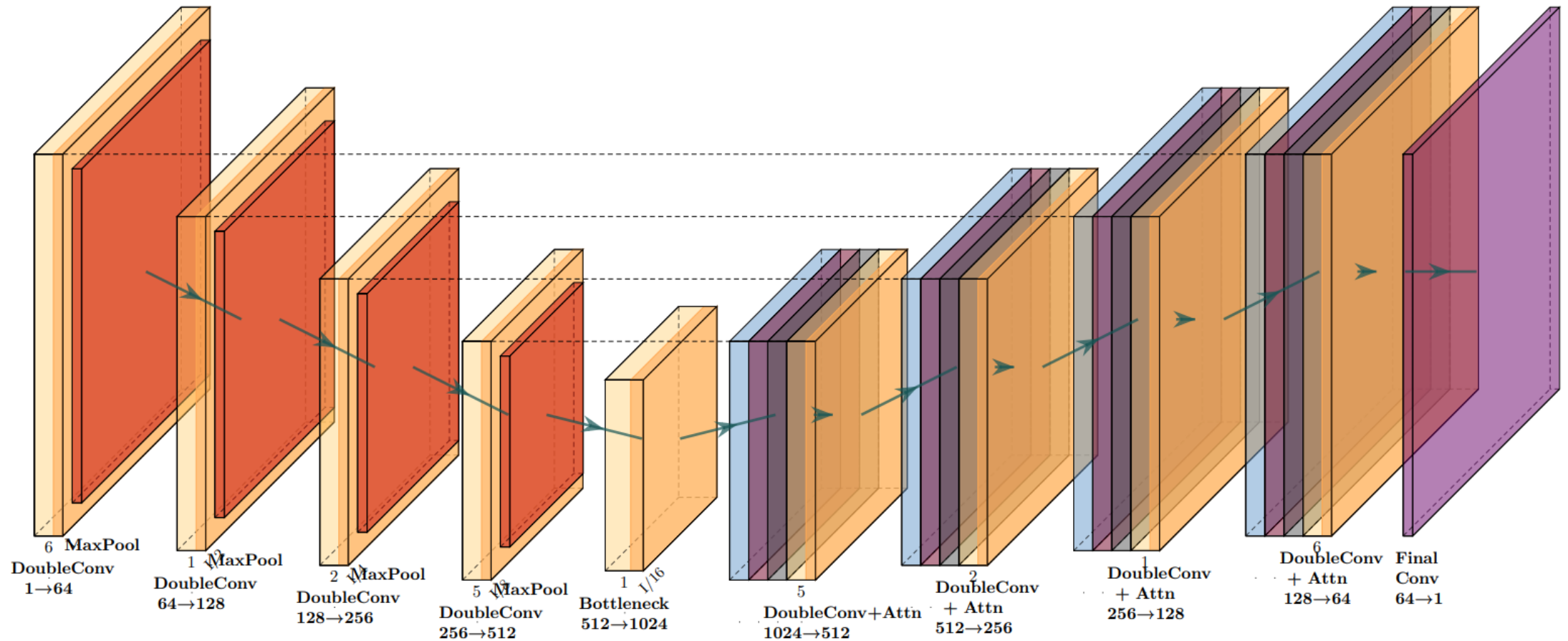
**U-Net Tiny:** A compact version of the U-Net model with reduced feature dimensions, designed to balance performance and computational efficiency. Explored in tiny16-32, tiny16-128.

# OUR MODELS



U-Net Tiny with Residual Connections: Incorporates residual connections within both encoder and decoder blocks to improve gradient flow and facilitate faster convergence.  
Explored in res16-32, res16-128, res32-256.

# OUR MODELS

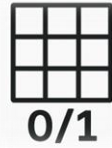


U-Net with Attention Mechanisms: Integrates attention modules within the decoder to enhance focus on semantically relevant regions guided by the coarse input. Explored in attn32-256.



# UNDERSTANDING OUR LOSS FUNCTIONS

## **BINARY CROSS ENTROPY**



- Treats segmentation as pixel-wise classification
- Penalizes incorrect predictions at the individual pixel level
- Sensitive to class imbalance

$$L_{BCE} = -[y \log(p) + (1 - y) \log(1 - p)]$$

## **DICE LOSS**



- Measures overlap between prediction and ground truth
- Optimizes global object-level accuracy
- Especially useful for imbalanced classes

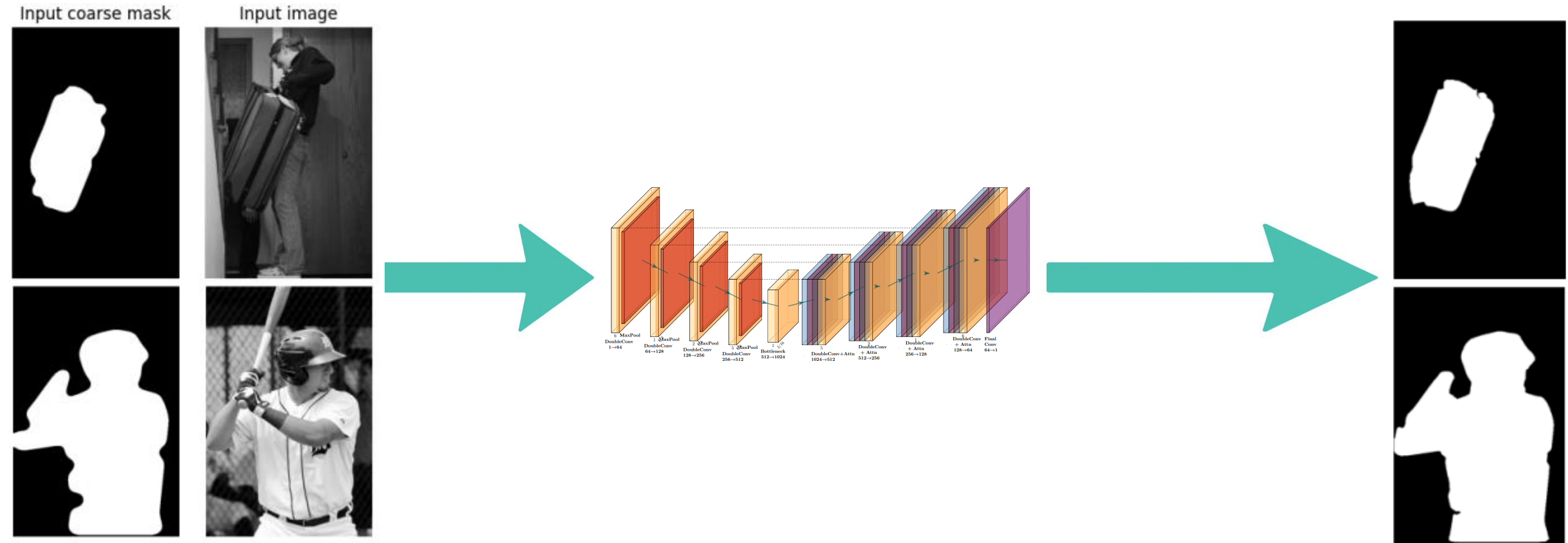
$$L_{Dice} = 1 - \frac{2|P \cap G|}{|P| + |G|}$$

## **BOUNDARY LOSS**



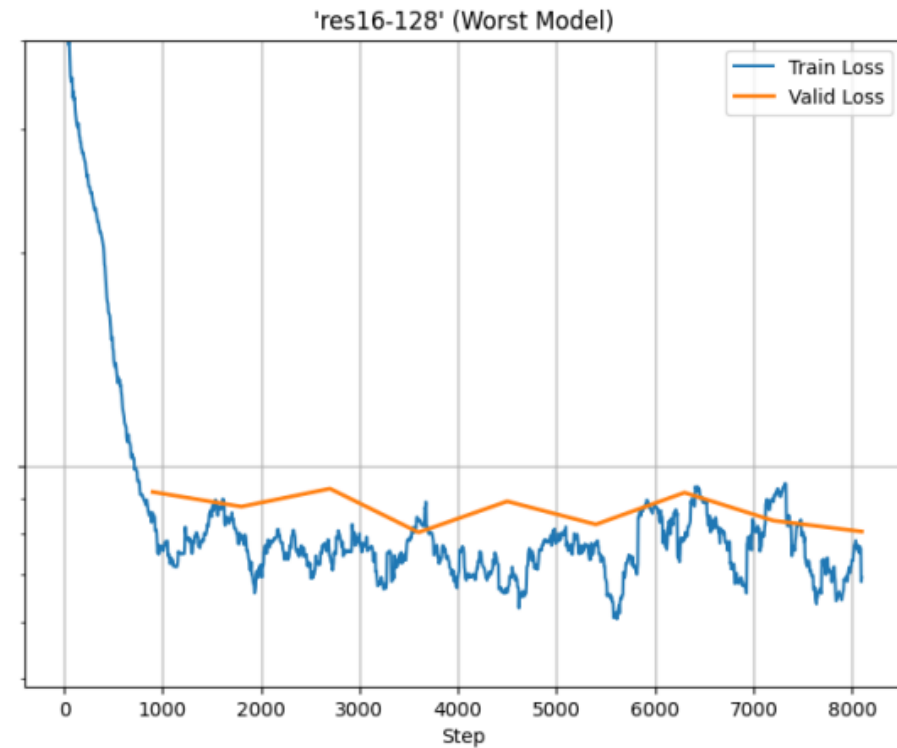
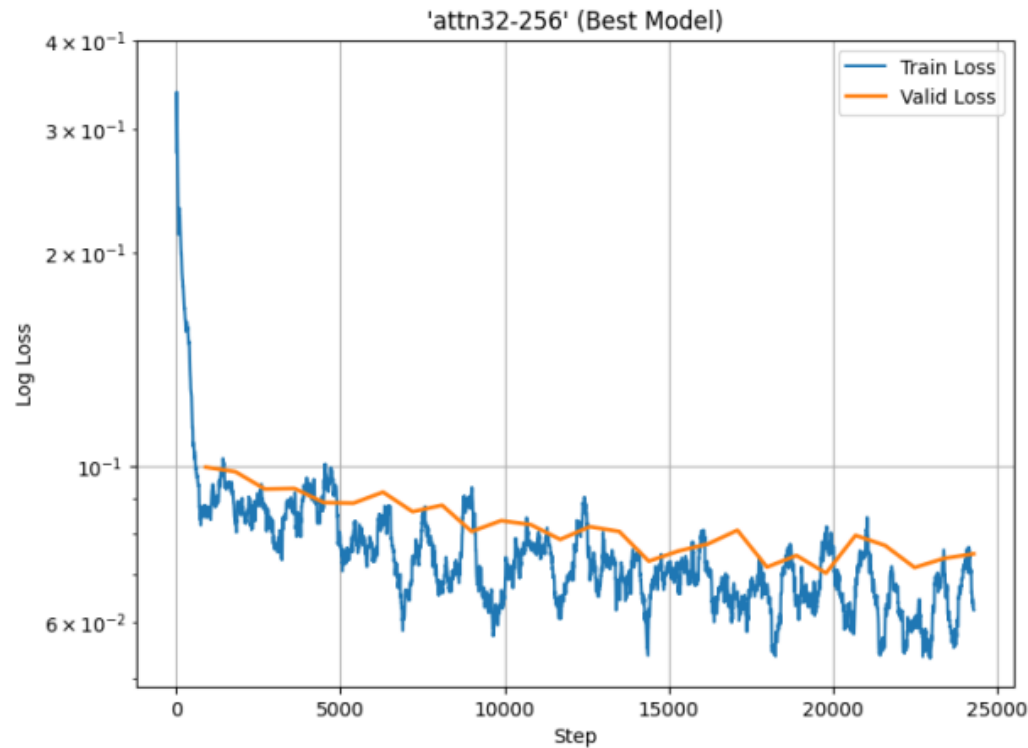
- Focuses on the alignment of object boundaries
- Penalizes misalignment even if global mask is mostly correct
- Useful for fine detail segmentation (edges, contours)

# LET'S SEE HOW OUR MODEL DOES QUALITATIVELY



This is how our attention-based U-Net inspired model performs on these two images from the MS COCO Dataset.

# AND NOW SOME NUMBERS...



Model Name	Pixel Accuracy	IoU	Boundary IoU	Hausdorff Distance
tiny16-32_bce-dice	0.9839	0.8609	0.1566	43.10
res16-32_bce-dice	0.9840	0.8611	0.1581	42.73
tiny16-128_bce-dice	0.9833	0.8528	0.1485	46.29
res16-128_bce-dice	0.9830	0.8534	0.1416	43.64
tiny16-128_bce-dice-bound	0.9864	0.8796	0.2207	40.66
res16-128_bce-dice-bound	0.9857	0.8749	0.2082	42.17
res32-256_bce-dice-bound	0.9859	0.8654	0.2040	43.03
attn32-256_bce-dice-bound	0.9866	0.8792	0.2210	40.77

- Showcases refined masks from **coarse** user input
- Attention-based and deep residual models yield the **sharpest contours**
- All models drastically improve over initial brush stroke



**THANK YOU FOR  
YOUR ATTENTION!**