Machine Learning: Dummy data as an example

**Your Dataset is Synthetic Dummy Data “**ciliasync\image\_core\checkpoints\dummy\_data”

Right now, you're using the generate\_dummy\_data() function, which creates:

* **Random red or green** shapes (ellipses and rectangles),
* **On a black background**, and
* Saves **RGB images** and **binary masks** for those shapes.

So what you're seeing — ovals, squares, rectangles — is **exactly what the model was trained to detect**. It has **never seen real cilia**, because no real cilia images were used during training.

**🔬 Why It Looks “Wrong” (But Isn't)**

Your model is doing its job *correctly* based on what it was trained on:

* You gave it artificial shapes.
* It learned to find and segment those same shapes.
* It is now predicting masks that match those.

So it's not a bug — it's a placeholder dataset for pipeline testing.

**✅ What You’ve Accomplished So Far:**

* You've built a full segmentation pipeline: data loading, training, evaluation, and visualization.
* You verified your model can learn **from scratch** on toy data.
* The predicted masks match the fake ground truth — success!

**🧭 What’s Next: Train on Real Data**

To make this pipeline detect actual **cilia**, you'll need:

1. **Real microscope images** of cilia.
2. **Ground truth masks** (annotated regions showing where cilia are).
3. Replace the dummy data generation with loading your real data into IMAGES\_DIR and MASKS\_DIR.

Would you like help updating your pipeline to work with real cilia data next?

4o

**You said:**

SO if i understand, the input image with the green rectangles, squares and red oval is the input image. The ground truth mask, is almost like a training image used to train the model that these are the shapes to expect. And finally the predicted mask is the model identify the shapes used to train it (from ground truth mask) in the input image ? Lemme know if I have missed anything

**ChatGPT said:**

You've got the **core idea exactly right** — great job! Here's a clearer breakdown to confirm your understanding and fill in any gaps:

**🖼️ 1. Input Image**

This is the synthetic image with:

* Black background,
* Red ovals and green rectangles.

It simulates what a microscope image might look like, but it’s completely artificial. These shapes are the "objects of interest" (standing in for cilia).

**🟩 2. Ground Truth Mask**

This is the **label** or **expected output** — a grayscale (black & white) image where:

* **White (255)** pixels mark where the object (shape) is.
* **Black (0)** pixels mark the background.

The ground truth tells the model: “Here is where the object you should learn to detect is located.”

✅ You're right — this is the image used during training as the **target** (mask) in the loss function (BCELoss).

**🧠 3. Predicted Mask**

After training, the model takes a new input image and outputs:

* A mask where it thinks the objects (the shapes it learned to recognize) are.
* Ideally, it should match the ground truth if the model learned correctly.

It’s a **probability map**, then thresholded (e.g., >0.5) into a binary prediction.

**✅ Summary of the Process**

| **Step** | **What’s Happening** |
| --- | --- |
| Train | Model sees (Input Image, Ground Truth Mask) pairs |
| Learn | Model adjusts weights to match its prediction to the ground truth |
| Predict | You give it a new Input Image → it predicts a mask |
| Visualize | You compare Input ⬅️ Ground Truth ⬅️ Prediction side-by-side |

So yes, your explanation is spot on — just remember the "ground truth mask" isn't used directly to find shapes, but to **teach** the model *where* shapes are during training.