Working with Roboflow

A step-by-step guide to building your own computer vision model

What is this guide?

This guide walks you through the process of using Roboflow to build a computer vision model. Computer vision is a technology that lets a computer "see" and recognise objects in photos or video, just like a person would.

We originally wrote this guide for a sousveillance project where we trained a model to detect surveillance cameras in public spaces. But the same steps work for any kind of object you want to detect: fire hydrants, birds, potholes, street signs, trees, you name it!

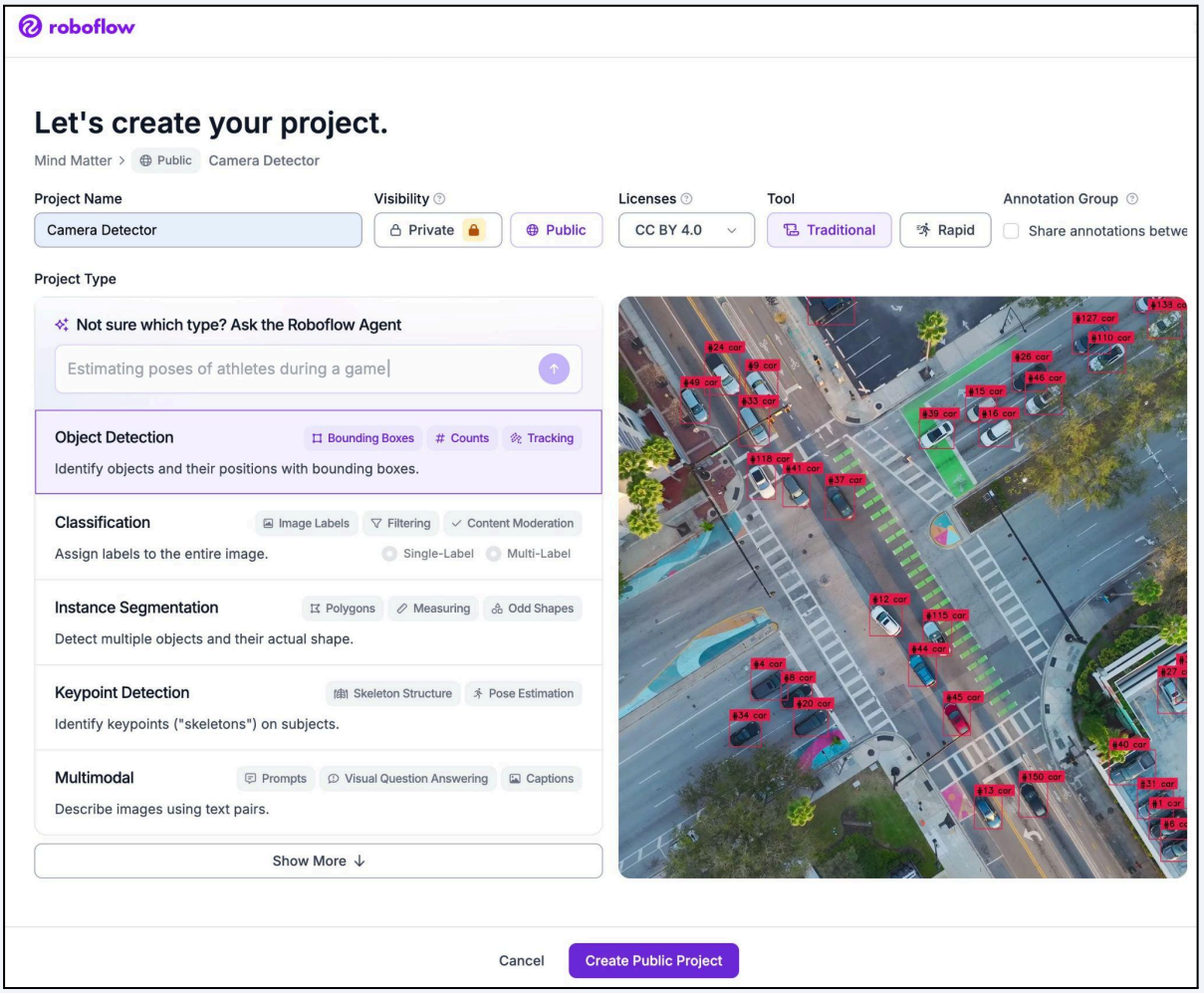
You do not need to be a programmer to follow this guide. Everything happens in your web browser. We will take it step by step

1. Starting your first Project

1. Before you can do anything, you need a free Roboflow account:
    a. Go to **roboflow.com** and click **"Get Started"** in the top right corner.
    b. Sign up with a Google or GitHub account, or your email address.
    c. Once signed in, you will see the Roboflow dashboard. This is where all your projects will live.
2. Click **“Projects”** in the left sidebar, and click **"+ New Project"** on the top right.
3. Add a project name. Keep **“Tool”** set to “**Traditional**”. For **"Project Type"**, choose **"Object Detection"**.
4. Click **“Create Public Project”**.

That is it! You now have a Roboflow workspace with a project ready to go.

Created by Thomas Jongstra (thomas@mindmatter.nl)



The Roboflow dashboard to create a new project.

What is a workspace?

Think of a workspace like a folder on your computer. It holds all your projects.
You can invite other people to your workspace if you want to collaborate.
The free plan is enough to follow this entire guide.

2. Using an Existing Dataset

A "dataset" is simply a collection of images that have been labelled. Labels tell the model what is in the picture and where it is. For example, in our project, every surveillance camera in a photo has a box drawn around it with the label "camera".

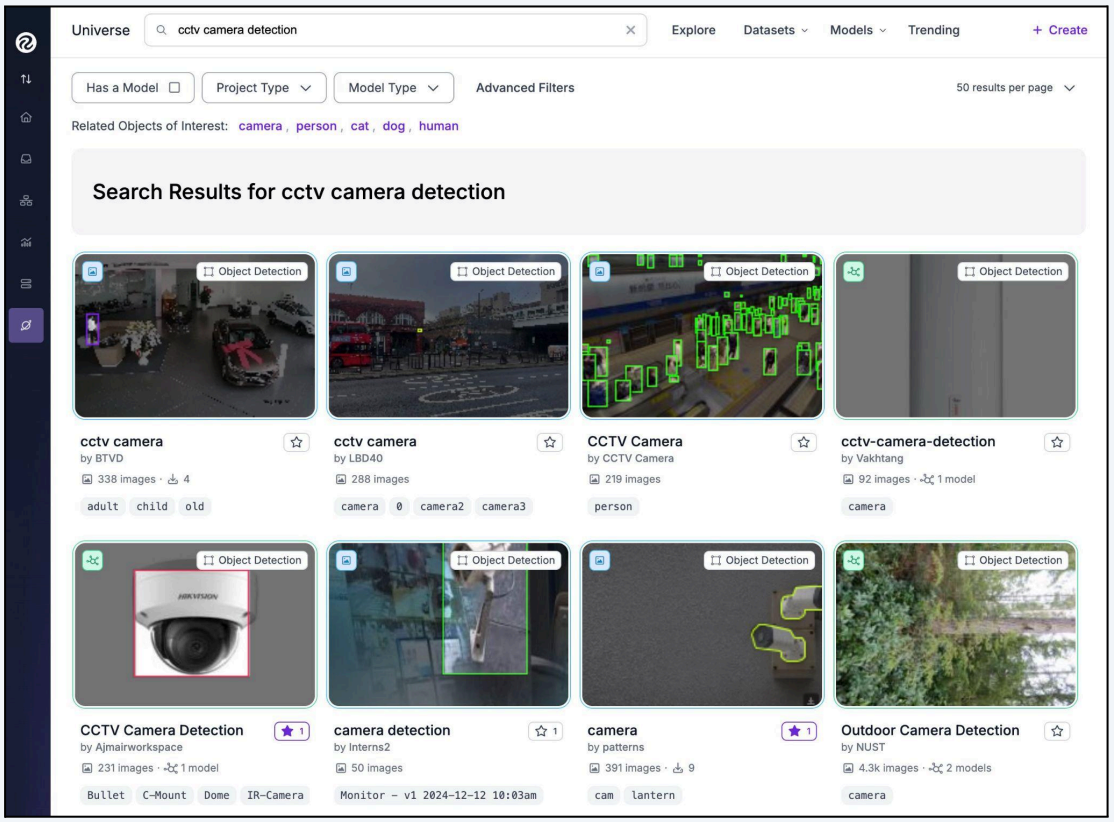
The good news is that you do not always have to start from scratch. Roboflow has a public library called Universe, where thousands of people have shared their labelled datasets. There is a good chance someone has already collected images of whatever you are looking for.

How to find and use a public dataset

1. Go to the Roboflow Universe and use the search bar. Select the **“Object Detection”** project type, and type in what kind of objects you would like to detect (for example, "surveillance camera", "traffic light", “tree"). Try some analogous terms, you might find better datasets!

Created by Thomas Jongstra (thomas@mindmatter.nl)

2. Browse the results. Look at the number of images and labels. More is usually better.
3. Open a dataset that looks promising. You will see example images so you can check if the labels look accurate.
4. Click **"Use This Dataset"** and then **"Fork"** to copy the dataset directly into your Roboflow workspace/project. You can also select individual images and **clone** them into your workspace/project. This way you can combine (parts of) multiple existing datasets. You can even mix in your own images later if you like.



Roboflow Universe search results page

If you decide to simply fork a full existing dataset, all the images and their labels will appear in your project. From there, you can add more images, fix labels, or jump straight to training a model.

> **When should you use an existing dataset?**
> When you want to get started quickly without labelling hundreds of images yourself.
> When someone has already done a great job collecting the exact type of images you need.
> You can always add your own images on top of a forked dataset to make it better.

3. Creating a New Dataset (optional)

If you cannot find what you need on the http://universe.roboflow.com, or if you want to build something from scratch, you can create your own dataset by uploading images.

Collecting images

Created by Thomas Jongstra (thomas@mindmatter.nl)

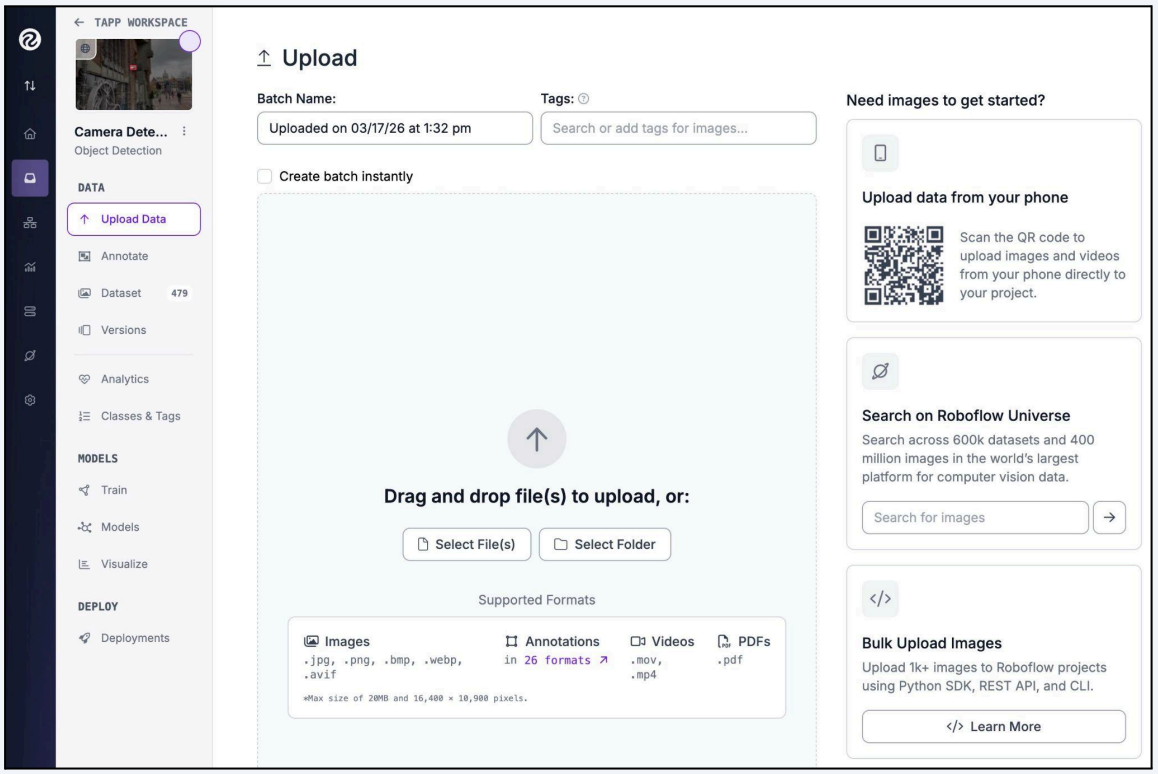
First, you need images. The more variety in your images, the better your model will perform. Here are some ideas for where to get them:

- Take some photos yourself, with your phone or camera.
- Use screenshots from Google Street View (great for things you find in streets or on buildings).
- Search for Creative Commons images online.
- Record a short video and let Roboflow split it into individual frames automatically.

Try to include images taken in different lighting conditions, from different angles, and at different distances. For example, for our camera project, we collected photos taken during the day, at dusk, close up, and from across the street. **This variety helps the model learn to recognise the object in all sorts of situations.**

Uploading images to Roboflow

1. Open your project on the Roboflow dashboard.
2. Click "**Upload Data**" in the left sidebar.
3. Drag and drop your images (or a folder containing images). Roboflow accepts common formats like JPEG and PNG.
4. You can also upload a video file: Roboflow will automatically extract individual frames for you.
5. After uploading, your images will appear in the "Unassigned" queue. Roboflow will automatically split them into training, validation, and test groups for you (you can adjust this if you want, but the defaults work well).



The upload screen with the drag-and-drop area

What are training, validation, and test splits?

Training images: The model studies these to learn what your object looks like.

Validation images: The model checks its own progress against these during training.
Test images: You use these at the end to see how well the model performs on images it has never seen.
Think of it like studying for an exam: textbook (training), practice test (validation), real exam (test).

4. Annotating Your Dataset

Annotation (or "**labelling**") is the process of telling Roboflow exactly where your objects are in each image. You do this by drawing boxes around them. This is the most hands-on part of the process, but Roboflow makes it as painless as possible.

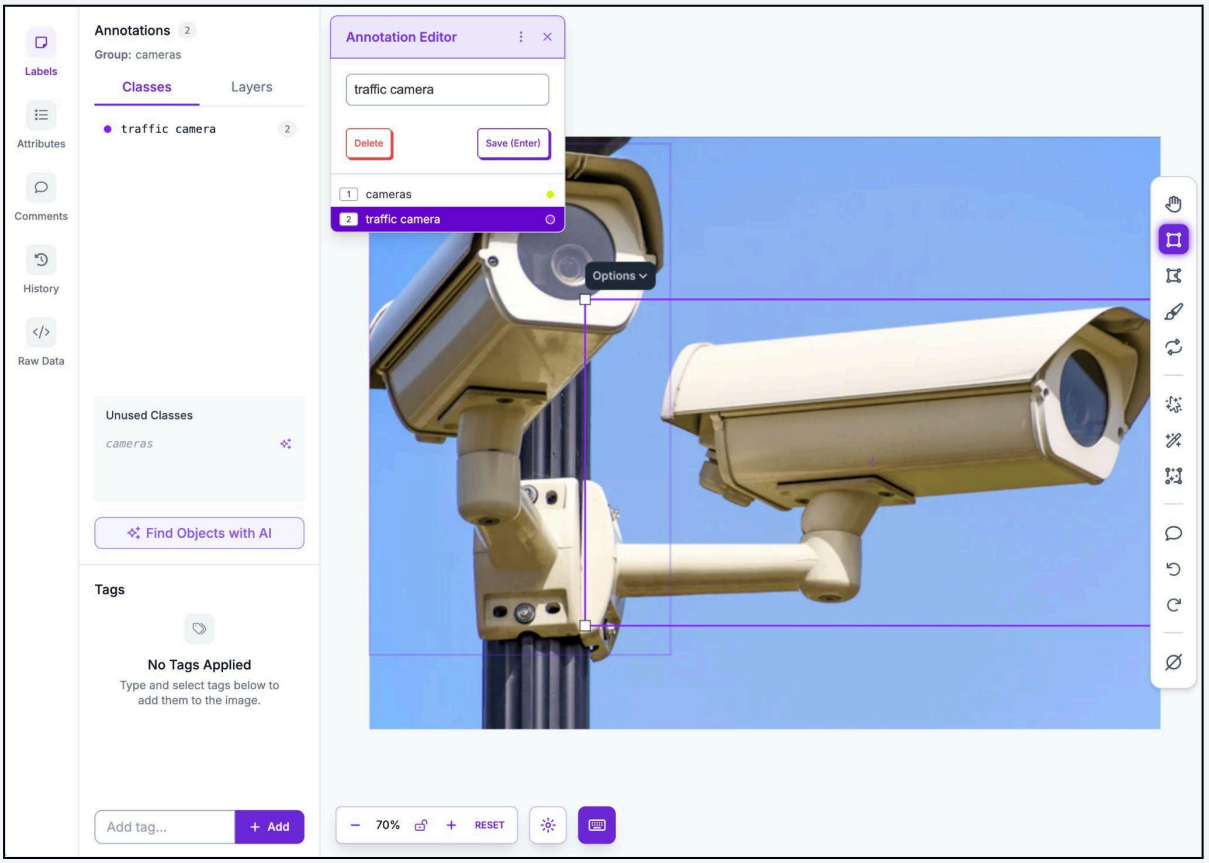
Note: you probably only need to annotate images that you uploaded yourself. Images from the Roboflow Universe are usually already annotated and directly usable for training (unless you want to add different labels/annotations to them).

How to annotate

1. Make sure you are in the **“Annotate”** view (or go there using the left sidebar). If you uploaded images, the **“Unassigned”** list should show them.
2. Click **“Annotate Images”** to open the annotation editor.
3. Unless you are working in a team, choose the **“Label Myself”** option.
4. Click **“Start Annotating”**.
5. Select the **”Bounding Box”** tool (or press the "B" key on your keyboard). This lets you draw rectangles around the object you want to annotate.[1]
6. Click and drag to draw a box tightly around your object. Try to make the box as snug as possible.
7. A pop-up will ask you for a label. **Type the name of the object** (for example, "traffic camera"). **After the first time, you can just select it from a dropdown.**
8. Create a bounding box for every object in the image.
9. When done, **click the purple checkmark** at the top of the screen to go to the next image.
10. Tip: You can use Ctrl+Z ((or Cmd+Z on Mac) to undo your bounding box selections.

[1] Optionally you can use the **“Smart Select”** tool, but for object detection using a bounding box is often good enough and simpler to use/understand. This is why we prefer using it in the guide. When you get a better feel and understanding for computer vision and want better performance (or do other tasks like image segmentation), you may want to use the polygon selections that the “Smart Select” tool supports.

Created by Thomas Jongstra (thomas@mindmatter.nl)



The annotation editor with a drawn bounding box

Tips for good annotations

- Be consistent. If you include the mounting bracket of a camera in one image, include it in all.
- Draw boxes tightly. Loose, sloppy boxes confuse the model.
- Label every instance. If there are three cameras, draw three boxes. Even if they overlap.

5. Training a Model

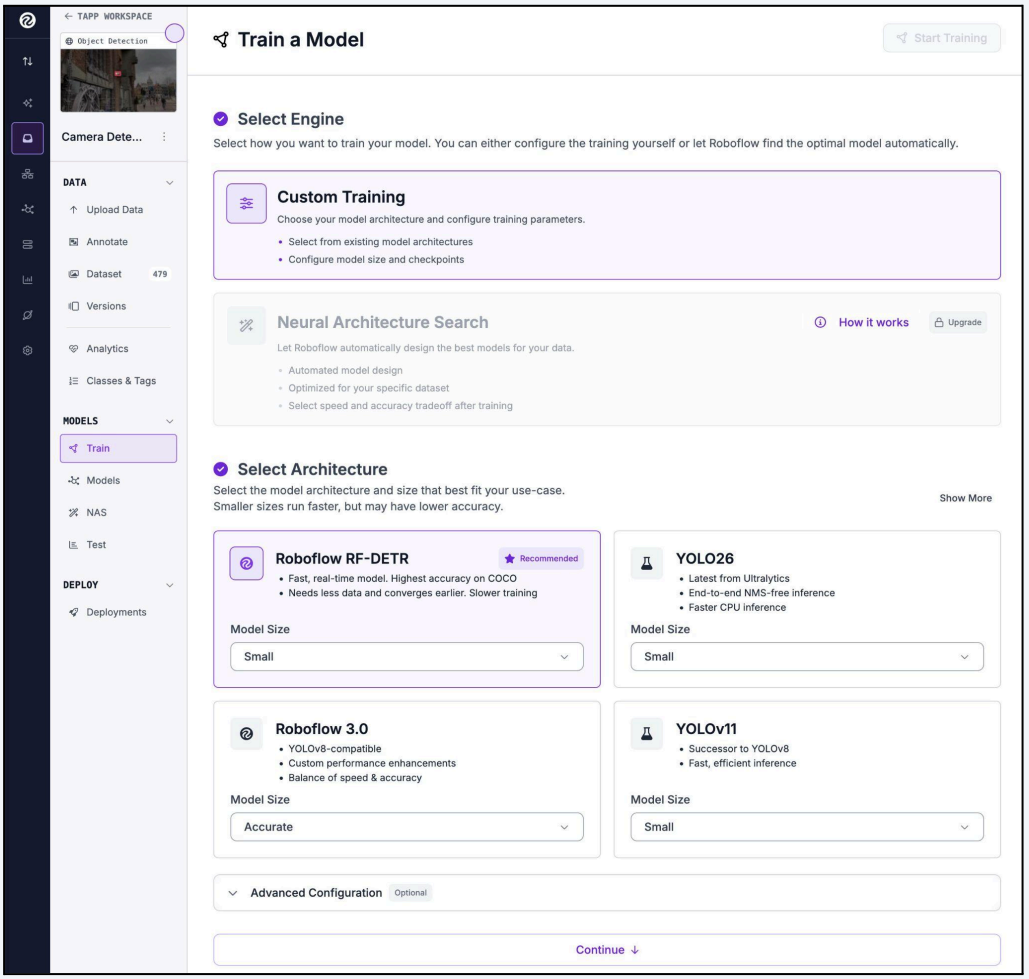
This is where the magic happens. You are going to take your labelled images and use them to teach a computer to recognise your objects on its own. Roboflow handles all the heavy lifting; you just click a few buttons.

Starting a training run

1) Select the **“Train”** view in the **“Models”** section of the left sidebar.
2) Click “**Custom Training**” and select “**Roboflow RF-DETR**” with model size “**Small**”. Small models can be trained faster, so they can quickly give you an idea of dataset quality/performance. This makes them ideal to start your project with.
3) Click **“Continue”**.
4) **“Train/Test Split”**: we recommend keeping this untouched.
5) **“Preprocessing”**: we recommend keeping this untouched unless there is a good reason to change things. You probably always want to keep **“Auto-Orient”** and **“Resize”** on with their default values.
6) **“Augmentation”**: We recommend adding a few image augmentations. These are automatic variations of your images (flipped, brightened, slightly rotated) that help the model learn better.

Think of it as showing the model the same object from slightly different perspectives. Play around with this, but certainly consider adding flip, rotate, brightness, blur, crop. Bounding box autementations are also very much recommended to use. It may be worth asking for some input from an expert when selecting these options, but experimenting (and checking the results) is often a good idea. Pick the 3x for the best performance (this will increase training time), or 1x for a quick test.

7) Click "**Start Training**". Roboflow will train the model on its own servers. You should get an email when it is done (the training can take a long time for large datasets).



The Custom Train model selection window

We recommend experimenting with the RF-DETR in different model sizes. YOLOv12 is also a very good model that can train quickly and give good results. However, after recent changes in Roboflow only the YOLOv12 Nano size model can be used by free users. So unless you pay for Roboflow, we recommend exclusively using RF-DETR.

RF-DETR (best accuracy)

RF-DETR is a newer model built by the Roboflow team. It uses a different approach that can be more accurate, especially when objects are small or partially hidden.

- For more technical people: RF-DETR can also be trained using a free Google Colab notebook (a free online coding environment) or on your own computer (if you have a good GPU).

Roboflow provides a ready-made notebook with all the steps filled in. You just click "Run" on each step.
- For the Google Colab route, you will need to export your dataset in a specific format from the Versions page before opening the notebook.
- **Link:** RF-DETR training guide

YOLOv12 (faster training)

YOLOv12 is a fast and reliable model that works well for most projects. "YOLO" stands for "You Only Look Once", which refers to how quickly the model can scan an image.

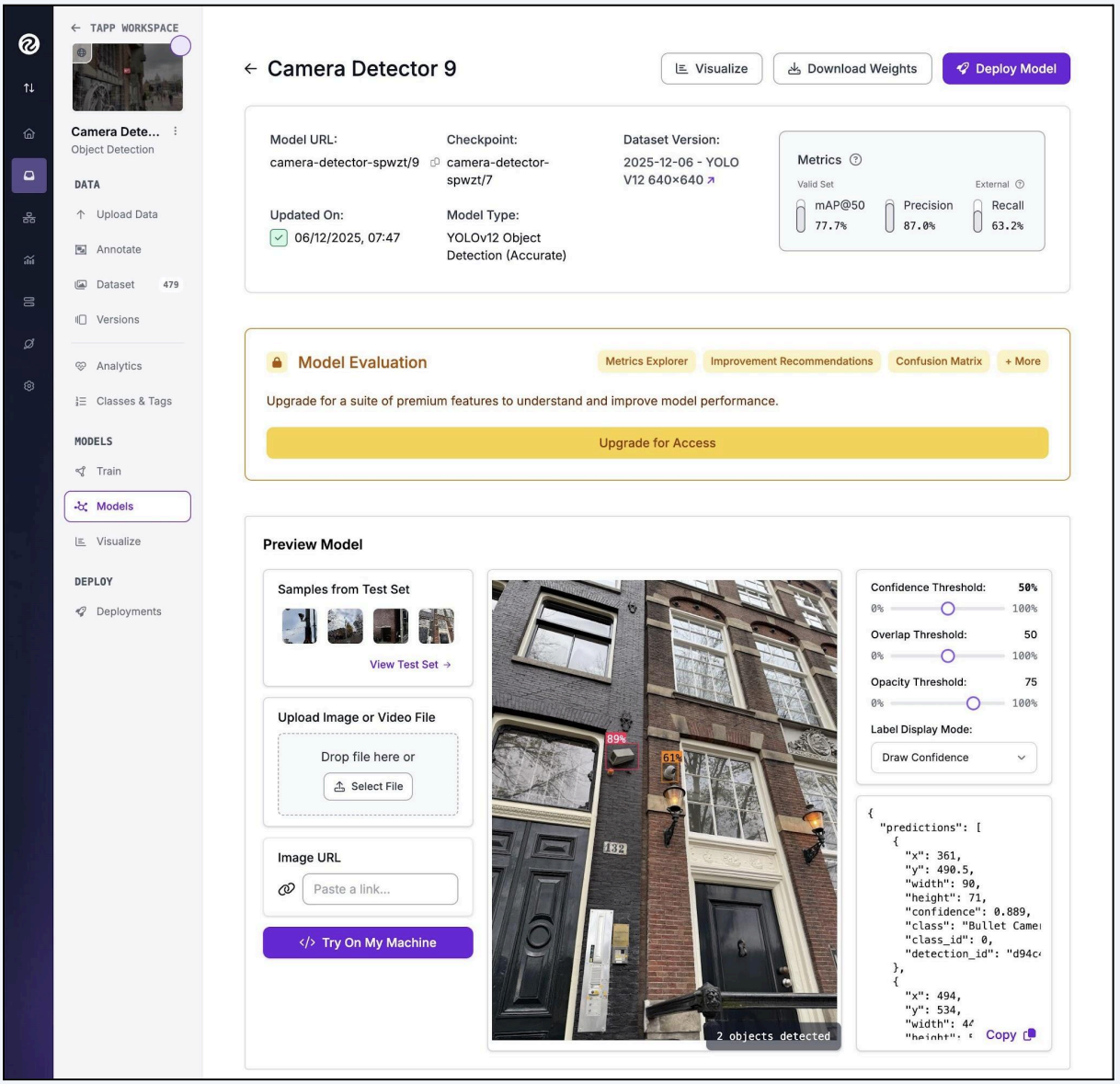
- The model starts with knowledge it already gained from millions of general images, then it fine-tunes itself on your specific images. This is why you do not need thousands of photos to get good results.
- **Link:** More about YOLOv12 on Roboflow

Understanding your results

After training finishes, Roboflow shows you a results page. The most important number is called "**mAP**" (mean Average Precision). It tells you how accurate your model is on a scale from 0 to 100. Here is a rough guide:

- Above 80: very good, likely ready for real use.
- 60 to 80: decent, but you could improve it by adding more images or fixing inconsistent labels.
- Below 60: the model is struggling. Try adding more training images, especially in conditions where it makes mistakes.

Created by Thomas Jongstra (thomas@mindmatter.nl)



The training results page showing mAP score and example predictions

You will also see example predictions where the model shows what it detected. Look through these carefully to understand where the model does well and where it gets confused. When you find out that the model struggles with specific objects, you might want to add more (or more varied) images of these objects to your training data. If you find out that the model struggles with specific situations (noisy images, darker images, lens flare, water drops on the lens, etc.), you may have to take more complex steps. You might be able to use data augmentation to make your training images look more like the noisy/dark images, but this is not guaranteed to improve the model. For images with lens flare or water drops on the lens (this often happens for CCTV cameras, making the scene not visible at all), you may have to introduce new classes called ‘lens flare’ and ‘water drops’ and try to train the model on these ‘unwanted’ classes as well. You probably want to brainstorm with an expert or do some research on computer vision approaches here if you really want to improve your performance.

6. Testing Your Model with a Workflow

Now that your model is trained, you probably want to try it out! Roboflow has a feature called "**Workflows**" that lets you test and deploy your model without writing any code.

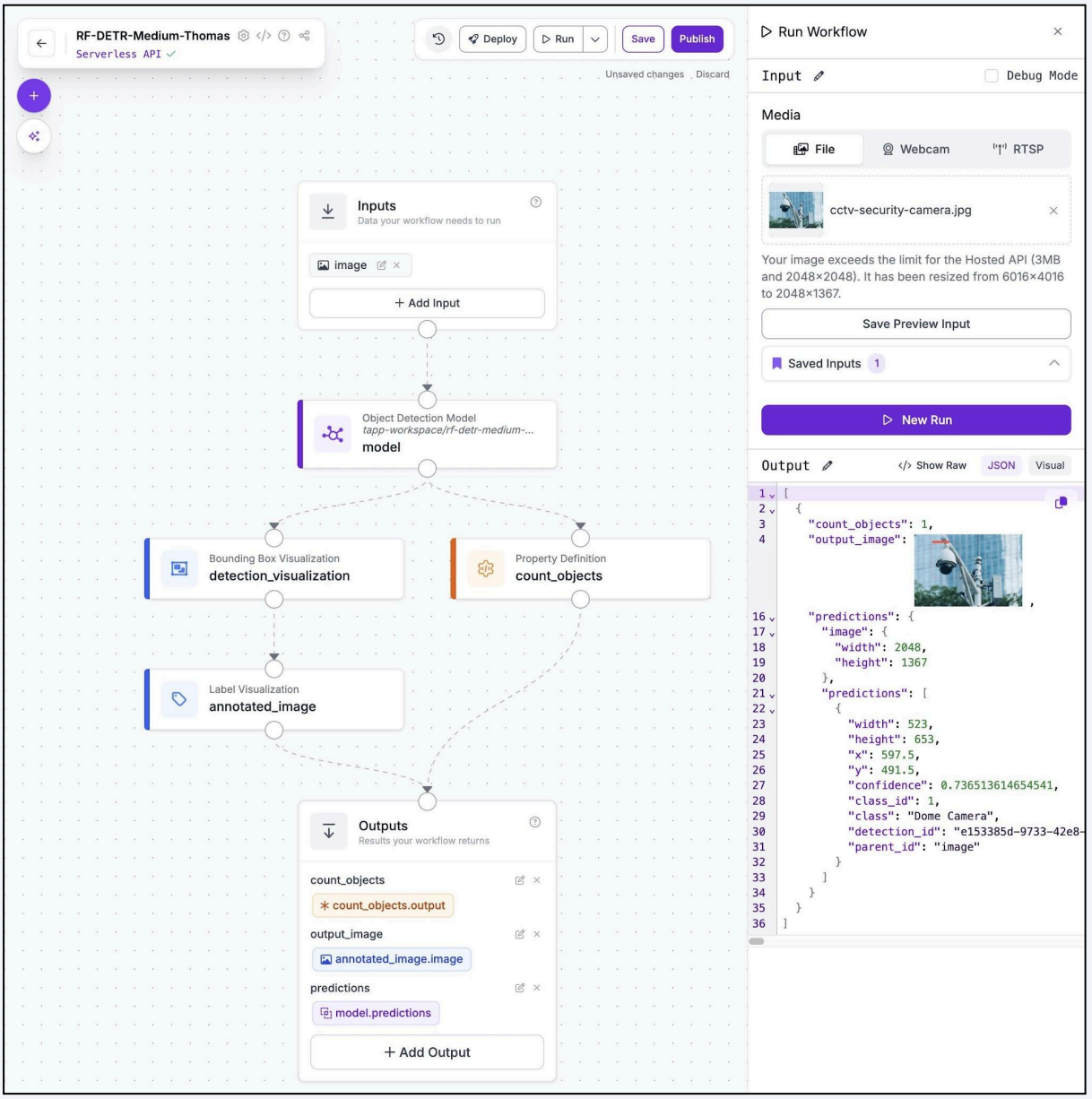
What is a Workflow?

A Workflow is like a recipe: you connect building blocks together to create a pipeline. The simplest pipeline takes an image, runs your model on it, and shows you the results with coloured boxes around the detected objects.

Creating and testing a Workflow

1. When in your workspace, click "**Workflows**" in the dark part of the left sidebar.
2. Click "**Explore Templates**".
3. Pick the “**Detect, Count, and Visualize**” template, and follow the tutorial instructions.
4. When you are done with the tutorial click the "▷**Run**" button in the top right corner.
5. Upload a test image (ideally one your model has never seen before).
6. Review the results. You should see coloured boxes around the detected objects, along with confidence scores (a percentage showing how sure the model is).

Created by Thomas Jongstra (thomas@mindmatter.nl)



The Workflow editor with detection and visualiser blocks connected, and

If the model detects too many false objects, increase the minimum confidence threshold in the detection block. If it misses real objects, try lowering it. Finding the right balance takes a little trial and error.

Going further: deploying your Workflow

Once you are happy with the results, you can use your Workflow beyond just manual testing:

- **API endpoint:** Every Workflow automatically gets a web address (an API) that you or a friend can send images to. The response comes back with all the detections.
- **Video input:** You can connect a video stream and have the model analyse frames in real time.
- **Extra blocks:** Roboflow offers over 50 different blocks you can add to a Workflow, including sending alerts, counting objects, filtering results, and more.

Link: Roboflow Workflows documentation

Quick Reference

Here is a summary of each step and what you need to do:

Step	What to do
Account setup	Sign up at **roboflow.com**, create a new project, choose **Object Detection**.
Existing dataset (optional)	Search the **Roboflow Universe** for labelled images. Fork a dataset into your project.
New dataset	**Collect and upload your own images.** Aim for variety in angles, lighting, and distance.
Annotation	**Draw bounding boxes around your objects in each image.**
Training	Start with **RF-DETR** in size nano or fast. Try larger sizes if you need more accuracy.
Testing	Create a "**Detect, Count, and Visualize**" workflow. Upload images to test.
Deployment	This is mainly intended for tech-savvy people who want to use their model in an app or website. **Fun to explore later.**

For more help, visit the Roboflow documentation at docs.roboflow.com or ask questions on the Roboflow community forum.

Created by Thomas Jongstra (thomas@mindmatter.nl)