

Evaluating Object Detection Models for Small Object Recognition

MSc Research Project
Artificial Intelligence

Gonzalo Bugueño
Student ID: x23408731

School of Computing
National College of Ireland

Supervisor: Kislay Raj

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Gonzalo Bugueño
Student ID:	x23408731
Programme:	Artificial Intelligence
Year:	2025
Module:	MSc Research Project
Supervisor:	Kislay Raj
Submission Due Date:	11/08/2025
Project Title:	Evaluating Object Detection Models for Small Object Recognition
Word Count:	373
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating Object Detection Models for Small Object Recognition

Gonzalo Bugueño
x23408731

1 Deploying the experiment

This section explains how to set up the environment, download the required code, and prepare the dataset for processing.

1.1 Requirements

- A Linux machine with `git` installed.
- Python 3.9 or newer.
- (Optional) NVIDIA GPU with CUDA support.
- Blender 4

1.2 Clone the Repository

First, download the required code and real-world dataset:

```
git clone https://github.com/gonzalobugueno/od-eval
```

1.3 Install Python

On Arch-based distributions:

```
sudo pacman -S python
```

1.4 Create a Virtual Environment

This may vary depending on your distribution.

```
cd od-eval
python3 -m venv .venv
. .venv/bin/activate
```

1.5 Install Python Packages

```
pip install numpy pandas pillow opencv-python ultralytics notebook
```

1.6 Install Blender 4

Install via package manager or from the official website: <https://www.blender.org/download/>

```
sudo pacman -S blender
```

1.7 Optional: Install CUDA Runtime

If you have an NVIDIA GPU:

```
sudo pacman -S nvidia-open-dkms
sudo pacman -S cuda cudnn
```

1.8 Open a Notebook Client

You can use PyCharm, VS Code, or run Jupyter directly:

```
jupyter notebook
```

1.9 Modify Blender Render Script

The file `blender/render-cvat.py` runs Blender in an automated fashion, iterating over a Cartesian product of parameters defined in a dictionary of arrays.

- Two parameter grids are defined, only one is used
- The active grid is set by the variable `pg`
- Modify the parameter grid or leave it as default. The experiment runs on both parameter grids.

The output will be a CVAT for images 1.1 dataset in the directory specified by `output_dir`.

1.10 Run the Blender Script

```
cd blender
blender --python render-cvat.py
```

1.11 Augment the Dataset

Run the notebook `bg_augmentations` to apply background changes and augmentations. By default:

- Input: `datasets/synthetic`
- Backgrounds: `datasets/backgrounds`
- Output: `datasets/synthetic_bg`

Repeat this for each dataset you want to augment.

1.12 Evaluate the First Run

Use the `eval-ultralytics` notebook: - The first cell defines grid search parameters and dataset composition. - The function `convert_cvat_to_yolo` converts CVAT datasets to YOLO format (via symlinks). - Metrics are saved as JSON files for offline analysis.

1.13 Evaluate the Second Run

Use the `eval-ultralytics-2` notebook for a second evaluation phase.

2 Figures

All the plotting and figures are contained within the notebooks "plot", and "analytics"