Detecting and Blurring Picture Objects Using YOLO and Streamlit

An AI-Powered Solution for Privacy Protection

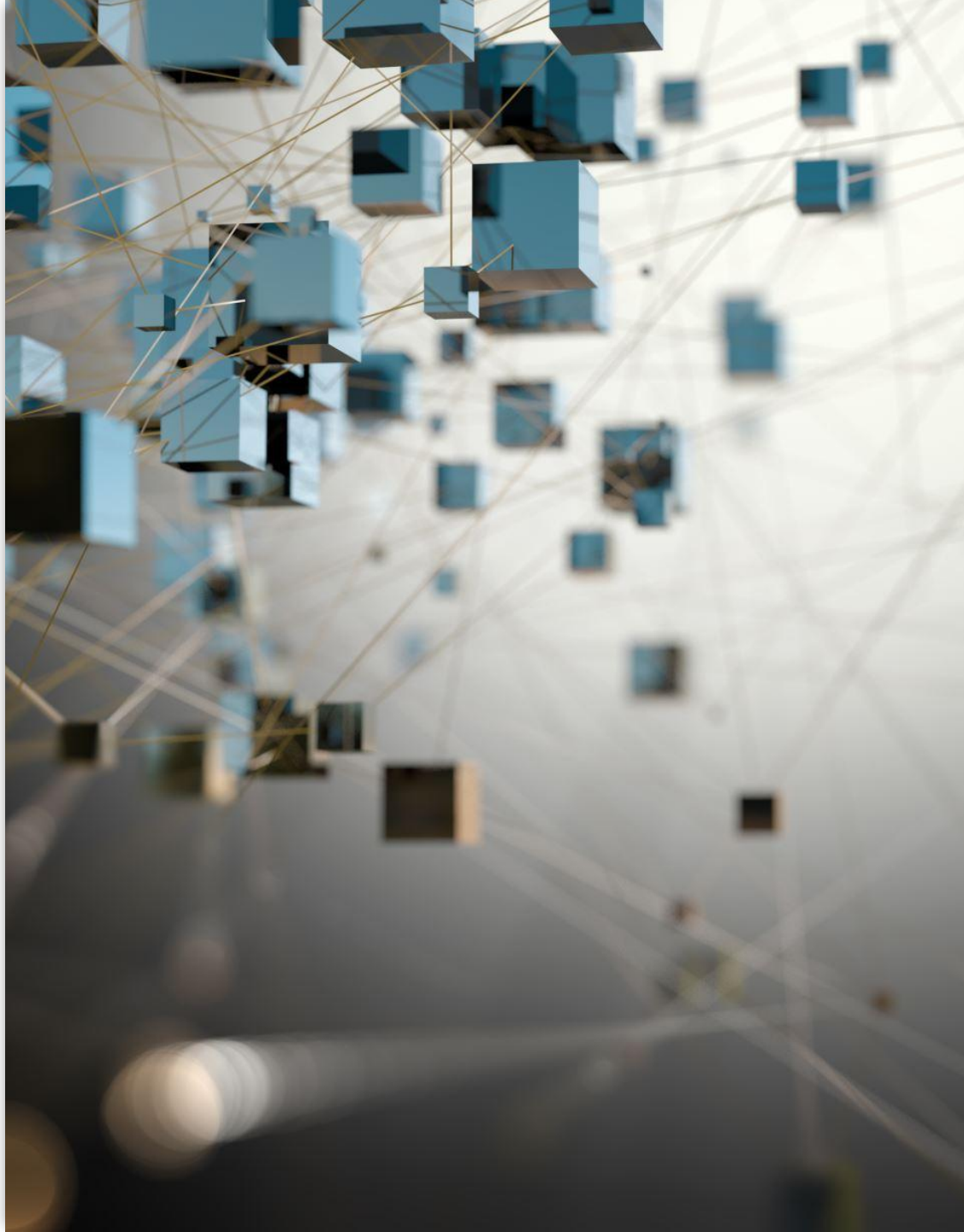
Authors: Silas Phillips, Keegan Nohavec, and Christoph Guenther

Introduction

Problem: Personal photos, art, and other picture objects in images and videos raise privacy concerns.

Solution: Use an Artificial Intelligence (AI) model to automatically detect and obfuscate these objects.

Technology Stack: Roboflow, YOLO, Streamlit, Python.



Goals

Main Goal: Develop an AI model to detect and blur picture objects in images and videos.

Sub-goals:

- High- accuracy detection of picture objects.
- Efficient blurring of detected objects maintaining image quality.
- Fast processing of images/videos without sacrificing accuracy.
- User interface for uploading and processing images/videos.

Our Process

- Data Sourcing and Preparation
- Model Selection
- Model Training
- Video Processing
- Web Application Development

Data Preparation

Data Sources: JPEG images of indoor spaces (Kaggle).

Image Preparation: Tool – Roboflow.

- 1 class (Pictures-on-Wall).
- Standard size: 224 x 224 pixels.
- Annotation – Identify, draw bounding boxes, and label picture objects.
- Create Train, Validation, and Test datasets.
- Augmentation – Crop, rotate, vary brightness, and blur images to increase training dataset.

Dataset Size: 3415 images split into

- Train: 2390 images (7170 after augmentation).
- Validation: 602 images.
- Test: 423 images

Model Selection

YOLO (You Only Look Once)

Why Yolo?

- Speed – Performs object detection and framing in one pass.
- Accuracy – Comparable or better compared to slower models (such as R-CNN based models).
- Flexibility – Variety of different models with different complexity.
- Convenience – Saves best parameters from each training run.

Training

Setup:

- Jupyter notebook on Google Colab.
- GPU.
- Multiple processors.

Process:

- Load datasets from Roboflow.
- Train model.
- Evaluate performance on
  - Precision.
  - Recall.
  - Mean Average Precision.
  - Confusion Matrix.
- Repeat, varying these parameters
  - YOLO version.
  - YOLO model.
  - Cache.
  - Batch size.
  - Number of epochs.
  - Patience.
  - Intersection over Union (IoU).

Model Performance

Runs	Model	Folder	Precision	Recall	MAP50	MAP50-95
6	yolov10m	V10M_8195im_300ep_05IoU_1cls_abandoned	72%	0.561872	0.608826	0.377785
0	yolov8m	Medium_8195im_300ep	73%	0.610124	0.650352	0.429412
2	yolov8m	Medium_8195im_900ep_abandoned	75%	0.578153	0.641539	0.430134
1	yolov8m	Medium_8195im_600ep	75%	0.587922	0.650033	0.426259
9	yolov10m	V10M_8195im_900ep_05IoU_1cls_abandoned	77%	0.57016	0.639074	0.424282
4	yolov10m	V10M_8195im_300ep	78%	0.563055	0.637978	0.428406
3	yolov8x	XL_8195im_300ep_abandoned	78%	0.538314	0.627686	0.411876
7	yolov10m	V10M_8195im_300ep_05IoU_1cls	78%	0.571936	0.636226	0.420036
8	**yolov10m**	V10M_8195im_600ep_05IoU_1cls	79%	0.551516	0.637381	0.424642
5	yolov10m	V10M_8195im_300ep_04IoU_abandoned	100%	0	0.320722	0.022025

Training Outcome

Model Chosen:

- YOLOv10m: 16,451,542 parameters.

Hyperparameters:

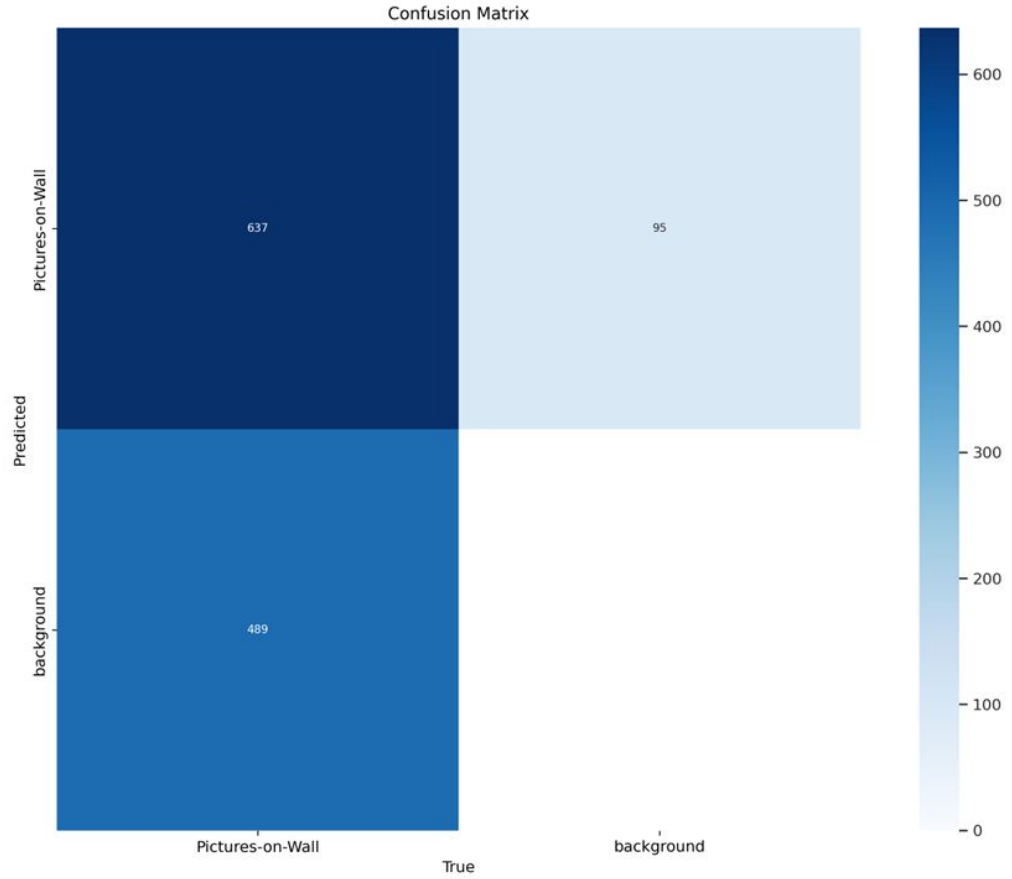
- Epochs = 600,
- Cache = True,
- Batch Size = -1,
- Patience = 100,
- IoU = 0.5.

Performance:

- Precision = 0.792,
- Recall = 0.552,
- mAP50 = 0.637,
- mAP50-95 = 0.425.

Performance - cont'd:

- Confusion Matrix



Video Processing

Improve processing speed by

- Applying object detection to a subset of video frames only.
- Use tracker to track objects in other frames.

Blurring:

- Apply Gaussian blur to bounding box.

Algorithm:

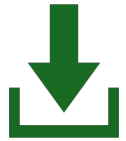
1. Instantiate chosen YOLO model.
2. Convert video to list of frames.
3. Every 5 frames (starting at frame 0)
    a. Detect picture objects with bounding boxes.
    b. Add new bounding boxes to tracker.
    c. Remove bounding boxes for objects no longer present from tracker.
    d. Apply Gaussian blur to each bounding box in tracker.
4. Reassemble list of processed frames into output video.

Streamlit User Interface

Why Streamlit?

Rapid development of web interfaces.

Easy integration with Python.



Features:

Upload images and videos.

Display original and processed images/videos.

Option to download processed videos.

UI Design:

Intuitive layout for user interaction.

Progress bar for processing status.


Streamlit Video Upload
Demo


reate your
ream room
or under
$680

Conclusion

- Successfully trained a YOLO model to detect picture objects with bounding boxes in images and videos.
- Applied model to video frames.
- Applied Gaussian blur to bounding boxes corresponding to detected objects.
- Built web application with Streamlit allowing users to submit videos for processing and download processed videos.

Future

Improve model accuracy

- More data, images and video, in a variety of formats and resolutions.
- Systematic hyperparameter tuning.
- Smarter video processing.

Improve processing speed

- More compute power.
- Use multiple processors.


Q&A