
Overview

TODOs:

1. Reading documentation on Detectron2; general familiarization – what are the implementation requirements? What are the benchmark datasets (coco, pascal)? How is accuracy measured, and how good it is usually at these different benchmarks?
2. Reading documentation on custom datasets with detectron2 and fine-tuning pipelines — would we need to get segmentation masks or just bounding boxes?
3. Looking through SAYCam gold set frames — what instructions would we want to give to turkers so that they annotated the frames correctly ? (see Clerkin et al 2017)
4. Try out a few SAYCam frames with Detectron2 in the Colab notebook. When does the model do OK? When does it do very different things from what you would do?

General Familiarization:

1. Implementation requirements
    a. Detectron2 has its own input format and requires data to be registered.
    b. They have a builtin function to transform data from coco or lvis format to detectron2 format
    c. This tutorial uses Roboflow to preprocess labelled data in any format (including csv) and download in COCO JSON format
        i. Then, the builtin function register_coco_instances registers the custom data with Detectron2 and can be used for training/validation/testing.
2. Benchmark datasets
    a. COCO
    b. PASCAL VOC
    c. LVIS
    d. Backbone models pretrained on ImageNet; default weights are ImageNet
3. Accuracy measure and performance on benchmarks

Custom datasets and fine-tuning pipelines:

1. Segmentation Masks and/or bounding boxes
    a. If we're doing instance detection and segmentation, then we require both segmentation masks and bounding boxes
    b. If we're doing panoptic segmentation, then we only require segmentation masks
        i. TODO: see if panoptic segmentation inputs can have (and ignore) bounding boxes, or they require only segmentation masks

On instructions for turkers:

1. We might need to make a dictionary of baby items:
    a. [play] gym vs playpen vs crib vs play mat
        i. Maybe linking them to a list of words with pictures might be helpful
        ii. We can also restrict it to folks who have lived with an infant in the past 5 years
    b. Is knowing what the high chair is important?
    c.
2. Parents' clothing feels important, i.e. when the parent is wearing a jacket, this indicates that they are going outside; jacket seems like an important object, but none of these models classify clothing. Is clothing outside the scope of this project?
3. Instruct turkers to generalize as much as possible (don't specify type of table). See 1.b.v in Performance on SAYCam frames FMI

Performance on SAYCam frames:

1. Instance detection and segmentation
    a. Model does ok with…
        i. People (any part of body) -- often over classifies things as people
        ii. With darker views and tinted hues
        iii. Chairs: tries to classify a lot of furniture as chairs
    b. Model does not do well with…
        i. Baby toys/items
        ii. Piano vs keyboard
        iii. Outside objects (stick, grass, etc)
        iv. Cats
        v. Nontraditional furniture (stool, coffee table/side table)
            1. Intuitively, these items could be classified with more traditional items/categories, (side tables, coffee tables, nightstands could all be labelled as tables)
        vi. Windows and computers: all TVs
        vii. Awkward angles (inside playpen, aerial view of drawer)
        viii. Identifying things as pictures/art
        ix. Sink vs tub
2. Panoptic segmentation
    a. Model does ok with…
        i. People (any part of body) -- classifies many things as people

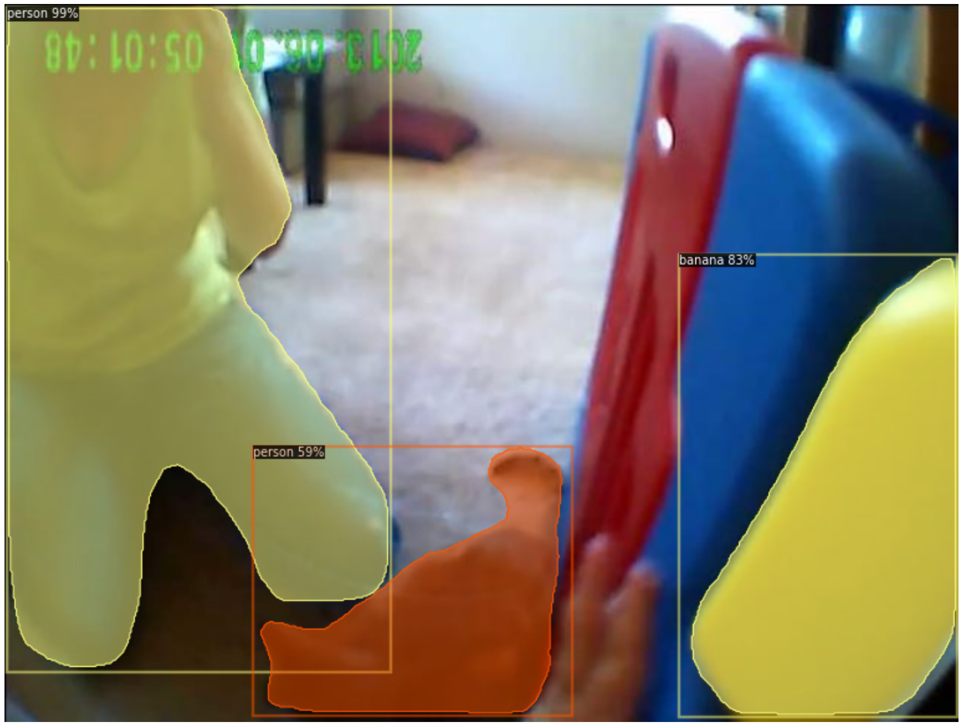
        ii. With darker views and tinted hues
        iii. Wall/floor/ceilings
        iv. Most furniture (not superb, but better than instance segmentation)
        v. Outside objects: detects trees and grass
        vi. Windows, but still sometimes classifies them as TVs
    b. Model does not do well with…
        i. Baby toys/items
        ii. Cats
        iii. Piano
        iv. Awkward angles (inside playpen, aerial view of drawer)
        v. Desktop computer screens (classifies as TVs)
        vi. Awkward angles (inside playpen, aerial view of drawer)
        vii. Natural objects outside of natural context (person holding stick or plant)
        viii. Identifying things as pictures/art
        ix. Sink vs tub

3. Overall
    a. All the models prioritize people, and I’m not sure if infants often identify people as people
    b. All ignore clothes worn
    c. Not very good at recognizing uncommon household items: art projects in one house are often mislabeled (easel is chair, concept of photo not recognized)
        i. Do the infants learn this as art or do they learn the objects shown in the art more?
            1. This might be a point of interest in parent surveys

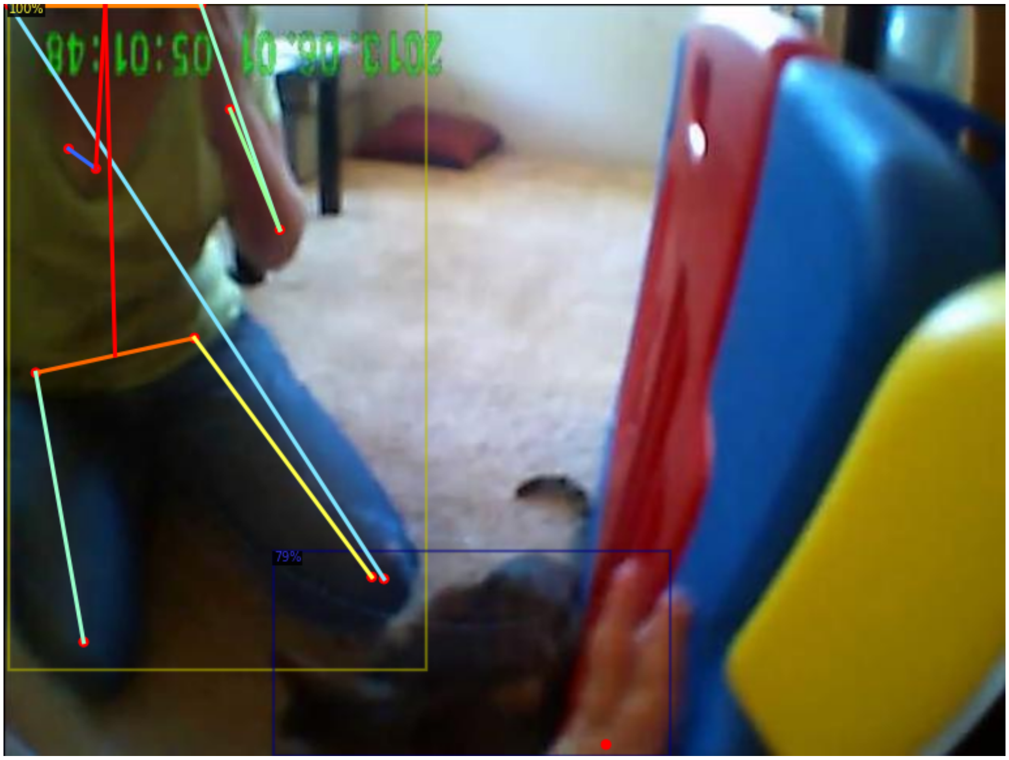
Baselines and Model Zoo

1. Three backbone combinations
    a. ResNet+FPN
        i. Best speed/accuracy tradeoff
    b. ResNet conv4 backbone with conv5 head
        i. Original Faster R-CNN
    c. ResNet conv5 with dilations in conv5
        i. Deformable ConvNet paper
    d. All trained with ~37 COCO epochs
    e. Pretrained Models on ImageNet-1k
        i. Different from Detectron models (no Batchnorm in affine layer)
        ii. ResNEt-50, ResNet-101, ResNeXt-101-32x8d

Detectron2 Beginner's Tutorial

1. Some general notes
    a. Cannot run locally on my mac: need CUDA compatible GPU (nvidia)
    b.
2. Using default predictor
    a. 

    b. Definitely some misrepresentation and errors to improve upon through training
3. Inference with keypoint detection model

a. 

b. No labels just recreating keypoints in key objects

4. Inference with panoptic segmentation

a. 

b. Quite robust to begin with and includes objects/features like wall and rug (haven't seen this in other models)
c. But, it confidently says the cat is a person, while the baseline model is more uncertain
d. Shades in instead of using a bounding box, just like what we were thinking for MTurk

[Training custom dataset]

1. Can register dataset and/or register metadata
2. We can use labelled dataset and convert csvs to dictionaries for the model to use
3. Uses dataset dict, which includes
    a. image filename
    b. height and width of image
    c. assigned image id
    d. list of dictionary of annotations of image
        i. Each dictionary contains
            1. Bounding box instance (list of 4 numbers), format of bounding box, list of polygons for segmentation mask of instance, keypoints (each point has x, y and visibility value)
    e. Filename of semantic segmentation ground truth: image with integer labels in place of pixel values
4. Register metadata for shared information across dataset, includes
    a. For instance detection/segmentation
        i. List of names for each instance category
        ii. List of rbg tuples corresponding to colors for each category (optional, if not included it'll randomly input!)
        iii. Dictionary mapping class ids to contiguous ids (numbers corresponding to classes)
    b. For semantic and panoptic segmentation
        i. List of names for each stuff category
        ii. List of rbg tuples corresponding to colors for each category (optional, if not included it'll randomly input!)
        iii. Panoptic root and json for panoptic evaluation
        iv. Dictionary mapping semantic segmentation ids to contiguous ids (number corresponding to categories)

Review of Pretrained Models in Different Images in Dataset

1. The models do not recognize baby objects: playpen, crib, etc
   a. We might need to build a dictionary of baby things
2. COCO knows people very well
   a. All the models frequently classify things as people
3. Panoptic segmentation consistently does better than the instance segmentation models
4. The LVIS pretrained model is trained on a 1x schedule, so its performance is subpar in comparison to the COCO models, which are trained on a 3x schedule.
5. FPN vs C4 vs DC5

Predicting on Videos

1. Detectron2 ~can~ take videos and outputs frame by frame predictions
   a. According to the demo docs, it has the ability to read in videos and output video visualizations. Still working on getting this functional.
   b. There are a few different tutorials for inputting videos, detectron2 demo does it and found a stackoverflow page that does it as well