

Ensemble Learning via Graph Inference Cliques

Eion Blanchard

Convolutional neural networks have vitalized computer vision (CV) tasks such as object detection, but mostly so for visible-light imagery. Object detection on X-ray imagery suffers from relatively meager datasets and crowded object contexts. Adversarial subjects and inevitable blind-spots in any CV model further impede application to non-intrusive inspection (NII) environments, such as X-ray scanning hand-carry baggage. We craft an ensemble method to process object detections from disparate CV models as a graph encoding object localizations. By viewing cliques in this inference graph as object detection events, our ensemble simplifies cluttered and often contradictory inferences. When applied to its development dataset, our ensemble performs more accurately and precisely in gestalt than any of its component models.

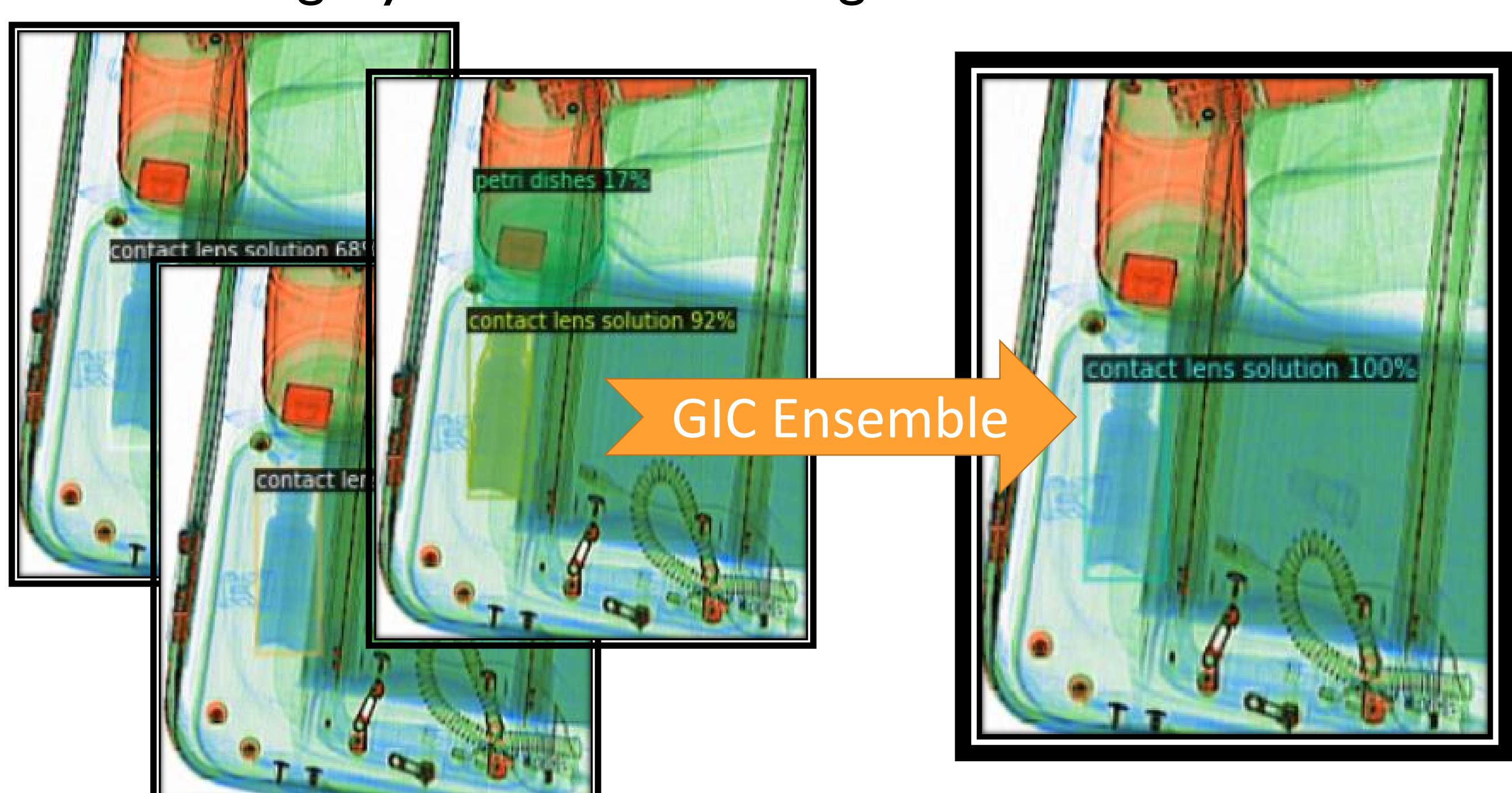
Data

For ensemble development, we used the Scans_V3.1 dataset, a repository of X-ray scans operated by Sandia. Included categories of objects are:

- orange
- apple
- potato
- vials
- sausage
- petri dishes
- banana
- contact lens solution

Images contain an annotation for each present object. We converted annotations from VGG into COCO format:

- category
- bounding box
- mask

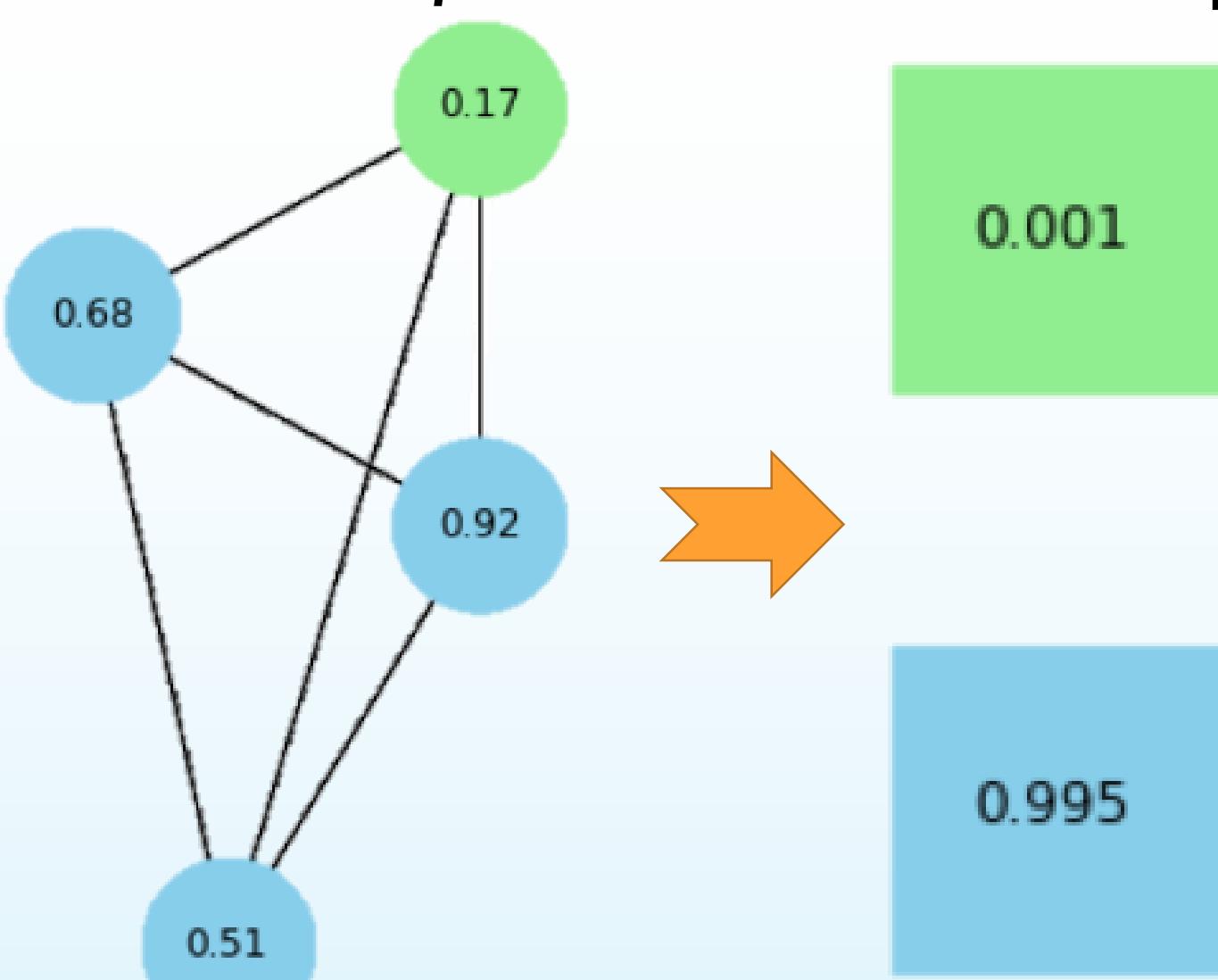


Baseline Models

Detectron2, an object detection and segmentation model library from Facebook AI Research (FAIR), implemented the convolutional neural network (CNN) models comprising our ensemble:

- RetinaNet
- Faster R-CNN
- Mask R-CNN

We trained and employed three distinct versions of each model. For testing and validation, each model separately infers image annotations on unannotated images and includes confidence scores for its predictions.



Graph Inference Clique Ensemble

1. Construct a graph in which vertices represent baseline inferences, and edges represent intersections of their bounding boxes
2. Greedily select heaviest valid clique
 - No two vertices from same baseline model
 - Vertex count at least threshold parameter
 - Weighed by sum of edge IoU values
3. Repeat 2. while possible
4. In each clique, compute categorical confidence scores:
 - Scale inference confidence scores by known baseline model performance on category
 - Compute confidence for each category C
 - $$\left(1 - \prod_{s \in C} (1 - s)\right) \prod_{s \notin C} (1 - s)$$
 - Normalize up to clique detection confidence
 - $$1 - \prod_s (1 - s)$$
5. For each clique, assign most confident category
6. Take weighted average of bounding boxes for each clique's selected category

Results

model	mAP
RetinaNet	0.705
Faster R-CNN	0.670
Mask R-CNN	0.671
GIC Ensemble	0.767

(left) Mean average precision for baseline representative and ensemble models
(below) Average ensemble performance metrics with IoU threshold 0.50

precision	recall	F1
0.915	0.958	0.935

Future Work

- Prepare data pipeline for industry-standard DICOS image format
- Input better baseline models, like those being worked on by Sandia's Lynceus group
- Train and test on variety of available datasets within Lynceus
- Optimize ensemble parameters via learning
- Implement optimal clique selection algorithm, rather than greedy
- Consider object mask inferences in ensemble
- Export ensemble method to other computer vision applications