Paper Discussion
RILOD: Near Real-Time Incremental Learning for Object Detection at the Edge

Jyotirmaya Shivottam

National Institute of Science Education and Research

January 8, 2021

1. Introduction
2. Methodology
3. Evaluation
4. Enhancements & Conclusion

1 Introduction

2 Methodology

3 Evaluation

4 Enhancements & Conclusion

Key Terms

- Incremental Learning (IL): Ability to learn new information, while keeping old information intact - ability to "memorize"

Key Terms

- Incremental Learning (IL): Ability to learn new information, while keeping old information intact - ability to "memorize"
- Two major challenges: *Plasticity* and (ensuring) *Stability*

Key Terms

- Incremental Learning (IL): Ability to learn new information, while keeping old information intact - ability to "memorize"
- Two major challenges: *Plasticity* and (ensuring) *Stability*
- Plasticity is the inability to retain old knowledge

Key Terms

- Incremental Learning (IL): Ability to learn new information, while keeping old information intact - ability to "memorize"
- Two major challenges: *Plasticity* and (ensuring) *Stability*
- Plasticity is the inability to retain old knowledge
- Stability - IL models need to be able to take in data continually, which can make the model unstable.

Key Terms

- Incremental Learning (IL): Ability to learn new information, while keeping old information intact - ability to "memorize"
- Two major challenges: *Plasticity* and (ensuring) *Stability*
- Plasticity is the inability to retain old knowledge
- Stability - IL models need to be able to take in data continually, which can make the model unstable.
- Catastrophic Forgetting: Loss of old information, that occurs when a standard NN is trained on new data

Problem Statement

- We want a Real-Time Object Detection IL model, that is:

Problem Statement

- We want a Real-Time Object Detection IL model, that is:
  - One-Stage

Problem Statement

- We want a Real-Time Object Detection IL model, that is:
  - One-Stage
  - Stable

Problem Statement

- We want a Real-Time Object Detection IL model, that is:
  - One-Stage
  - Stable
  - Scalable

Problem Statement

- We want a Real-Time Object Detection IL model, that is:
  - One-Stage
  - Stable
  - Scalable
- Additionally, we want a pipeline, that acquires training data for new objects on-the-fly.

Problem Statement

- We want a Real-Time Object Detection IL model, that is:
  - One-Stage
  - Stable
  - Scalable
- Additionally, we want a pipeline, that acquires training data for new objects on-the-fly.
- All of this should be able to work on the "Edge" - on devices with far lesser computational power, than the machines, these models train on.

1 Introduction

2 Methodology
- System Overview
- Algorithm

3 Evaluation

4 Enhancements & Conclusion

1 Introduction

2 Methodology
- System Overview
- Algorithm

3 Evaluation

4 Enhancements & Conclusion

RILOD

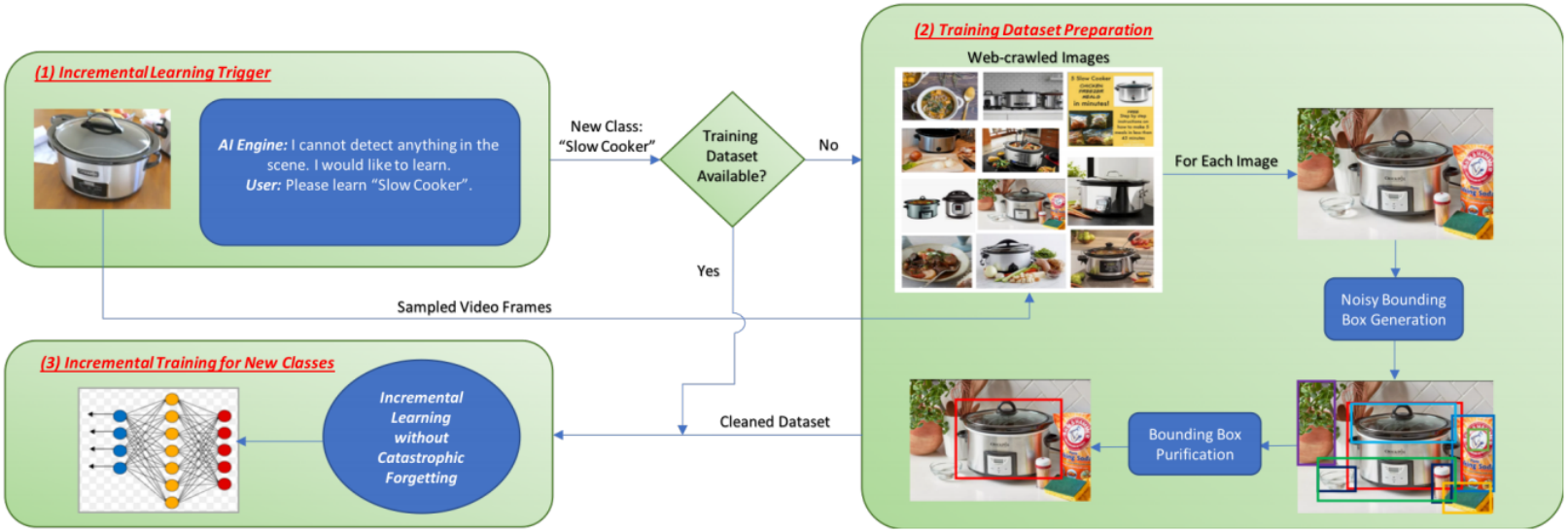

Figure 1. System overview of RILOD. It includes 3 main components: (1) Incremental learning trigger which incorporates the logic on when the incremental learning will start and what classes will be learned (one or more new classes can be learned at a time), (2) Training dataset preparation which downloads, labels and purifies the training images for the new classes in case that there's no available training dataset, and (3) Incremental training for new classes which applies our novel and efficient incremental learning algorithm to incrementally train the model with only the training data of the new classes while maintaining its knowledge on the old classes. Components (2) and (3) may run on a remote cloud server or locally on a mobile GPU for a fully on-device system, and both have been implemented for prototype system evaluation.

1 Introduction

2 Methodology
- System Overview
- Algorithm

3 Evaluation

4 Enhancements & Conclusion

One-stage Object Detection

- Usually, OD algorithms first obtain Region Proposals and then use CNN to tune the bounding box and to classify the objects in those regions
- These are very accurate but far too slow for real-time inferencing
- Recently, architectures, based on grid-based prediction (YOLO), anchor boxes (SSD), feature pyramids and focal loss (RetinaNet) have come up, that combine the two stages and are much faster
- RILOD uses the RetinaNet model, which has 3 subnets:
  - Feature Net (F) or Backbone Net
  - Class Subnet (C)
  - Box Subnet (B)

Incremental Learning

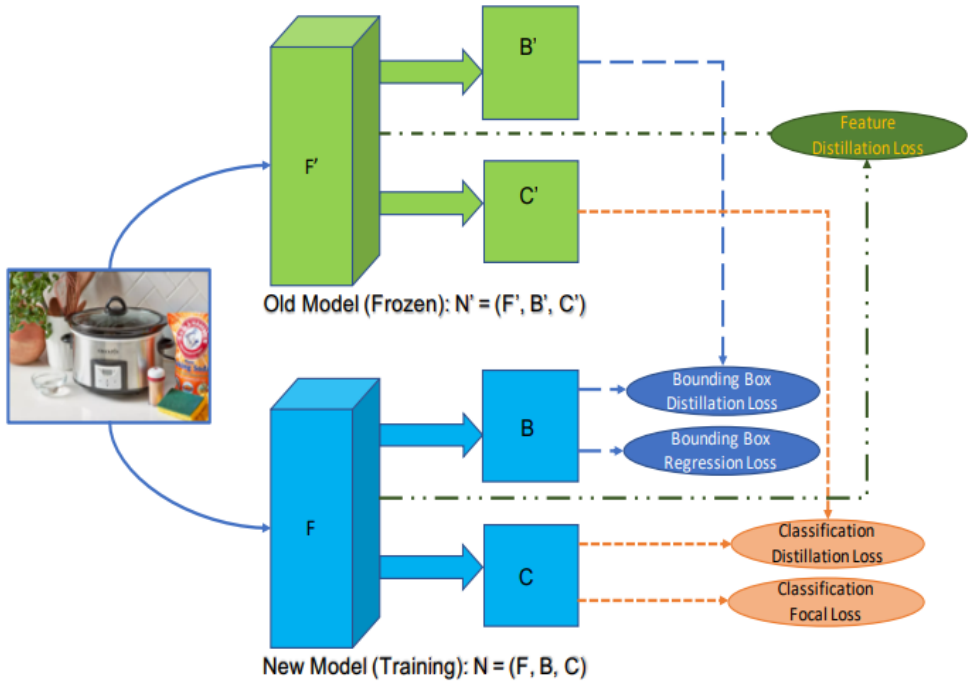

Figure 3. The proposed incremental learning method using RetinaNet as an example.

Learning without Forgetting (LwF)

- LwF is a Knowledge Distillation Technique
- It makes use of a modified Cross-Entropy Loss, that uses the output from the old model on new images, as a "soft ground-truth".
- This is done for both C and B
- Additionally, a distillation loss term is also added for output from intermediate layers of F.

Learning without Forgetting (LwF)

Algorithm 1 Learning without forgetting

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

X_n, Y_n: training data and ground truth on the new task

Initialize:

$Y_o \leftarrow CNN(X_n, \theta_s, \theta_o)$ // compute output of old tasks for new data

$\theta_n \leftarrow RANDINIT(|\theta_n|)$ // randomly initialize new parameters

Train:

Define $\hat{Y}_o \equiv CNN(X_n, \theta_s, \theta_o)$ // old task output

Define $\hat{Y}_n \equiv CNN(X_n, \theta_s, \theta_n)$ // new task output

$$\theta_s*, \theta_o*, \theta_n* \equiv \underset{\theta_s*, \theta_o*, \theta_n*}{argmin} (\lambda_o L_{old}(Y_o, \hat{Y}_o) + L_{new}(Y_n, \hat{Y}_n) + R(\hat{\theta}_s + \hat{\theta}_o + \hat{\theta}_n)$$

Details of the loss function

- We want to discourage changes to output from C and B, for old classes and prevent large changes to output from intermediate layers of F. So, a corresponding loss function can be:

$$
\begin{aligned}
loss = & L_{focal}(Y_n, \hat{Y}_n) \\
& + \lambda_1 L_{regr}(B_n, \hat{B}_n) \\
& + \lambda_2 L_{dist\_clas}(Y_o, \hat{Y}_o) \\
& + \lambda_3 L_{dist\_bbox}(B_o, \hat{B}_o) \\
& + \lambda_4 L_{dist\_feat}(T, \hat{T})
\end{aligned} \tag{1}
$$

Mixing examples from old classes

- To further reduce the forgetting on old classes, RILOD performs a clustering for each old class and picks a random example to add to the new training set
- As an improvement, number of examples to be added can be fixed to prevent the linear increase in dataset size, as new classes are added

Automated dataset collection and labeling

Algorithm 1 Automatic Dataset Construction

Require: the given new class name l_g
Require: the downloaded images D using l_g as query
Require: the large-scale classification model M_{cls}
Require: the Word2vec model M_{w2v}

Voting for "credible labels" set S_{cl}:
initialize a counter Ct for each label in M_{cls}
for each image in D **do**
  produce a set of noisy bounding boxes B
  for each b in B that $size(b) > thr_b$ **do**
    predict top k labels L_k using M_{cls}
    for each label l in L_k **do**
      $Ct[l] += 1$
sort Ct_l and append the top1 label l_1 to S_{cl}
for each l_i in $sorted(Ct_l)$ **do**
  if $cos\_sim(l_i, l_1) + cos\_sim(l_i, l_g) > thr_d$ **then**
    $S_{cl}.append(l_i)$
return S_{cl}

Purification for final dataset D_f
for each image I in D **do**
  for Each bounding box b_i in B **do**
    if Any label in predicted top $k \in S_{cl}$ **then**
      calculate $ACCS_i$
    else
      remove b_i from B
  if $size(B) > 1$ **then**
    for each box pair b_i, b_j **do**
      if $overlap(b_i, b_j) > thr_o$ **then**
        remove the box with lower $ACCS$
  if $size(B) \geq 1$ **then**
    add I with B to D_f
return D_f

Automated dataset collection and labeling

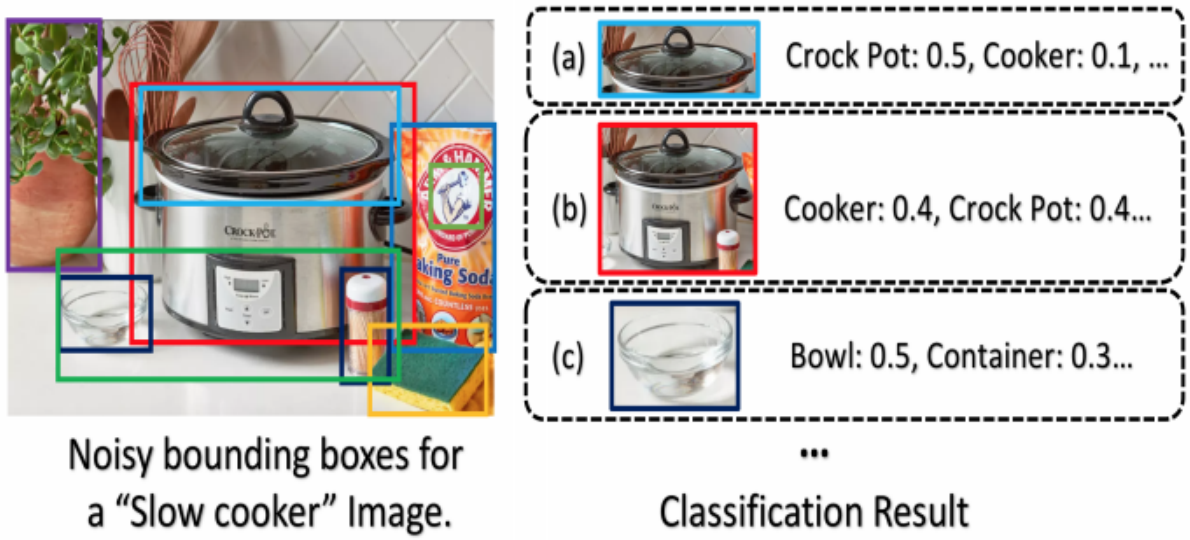

Figure 4. An example demonstrating the bounding box labeling problem. First, the labels in the classification model may not match the given new class name. Second, the noisy bounding boxes may overlap with each other (e.g., (a) and (b)) and the ideal box (b) may be eliminated by simply setting a low IoU threshold for NMS.

1 Introduction

2 Methodology

3 Evaluation

4 Enhancements & Conclusion

Measure, Datasets & Model Details

- Metrics:
  - Effectiveness of the proposed IL and the Automated Dataset Construction algorithms
  - Running time and bottlenecks
- Datasets:
  - Pascal VOC 2007
  - iKitchen
- Model Details:
  - Input Image Size: $(x, 1024)$
  - Initial Learning Rate: $1e-3$
  - Optimizer: Adam
  - Batch Size: 8
  - Trained on: Single NVidia Tesla M40

Setup for Pascal VOC 2007

- Model used: RetinaNet (ResNet-50 backbone)
- Trained for 100 epochs
- IL Experiments:
  - Scenarios:
    - $19 + 1$
    - $10 + 10$
  - Learning Schemes:
    - All Data
    - New data
    - New data (Without Distillation Loss)
    - New data (With Distillation Loss)

Setup for iKitchen

- Model used: RetinaNet (ResNet-18 backbone)
- Trained for 10 epochs on 8 base classes
- Excluded 2 classes: "Slow Cooker" (SC) and "Cocktail Shaker" (CS)
- IL Experiments:
  - 8 + SC
  - 8 + CS
  - 8 + SC + CS
  - 8 + CS + SC
- All experiments considered distillation loss

Results

Table 1. Per-class accuracy of Pascal VOC (%) for 19 + 1 scenario.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Class 1-19	70.6	79.4	76.6	55.6	61.7	78.3	85.2	80.3	50.6	76.1	62.8	78.0	78.0	74.9	77.4	44.3	69.1	70.5	75.6	-	-
All Data	77.8	85.0	82.9	62.1	64.4	74.7	86.9	87.0	56.0	76.5	71.2	79.2	79.1	76.2	83.8	53.9	73.2	67.4	77.7	78.7	74.7
Catastrophic Forgetting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68.9	3.4
w/o Feat-Distill Loss	61.9	78.5	62.5	39.2	60.9	53.2	79.3	84.5	52.3	52.6	62.8	71.5	51.8	61.5	76.8	43.8	43.8	69.7	52.9	44.6	60.2
w Feat-Distill Loss	69.7	78.3	70.2	46.4	59.5	69.3	79.7	79.9	52.7	69.8	57.4	75.8	69.1	69.8	76.4	43.2	68.5	70.9	53.7	40.4	**65.0**

Table 2. Per-class accuracy of Pascal VOC (%) for 10 + 10 scenario.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Class 1-10	76.8	78.1	74.3	58.9	58.7	68.6	84.5	81.1	52.3	61.4	-	-	-	-	-	-	-	-	-	-	-
All Data	77.8	85.0	82.9	62.1	64.4	74.7	86.9	87.0	56.0	76.5	71.2	79.2	79.1	76.2	83.8	53.9	73.2	67.4	77.7	78.7	74.7
Catastrophic Forgetting	0	0	0	0	0	0	0	0	0	0	66.3	71.5	75.2	67.7	76.4	38.6	66.6	66.6	71.1	74.5	33.7
w/o Feat-Distill Loss	67.1	64.1	45.7	40.9	52.2	66.5	83.4	75.3	46.4	59.4	64.1	74.8	77.1	67.1	63.3	32.7	61.3	56.8	73.7	67.3	62.0
w/ Feat-Distill Loss	71.7	81.7	66.9	49.6	58.0	65.9	84.7	76.8	50.1	69.4	67.0	72.8	77.3	73.8	74.9	39.9	68.5	61.5	75.5	72.4	**67.9**

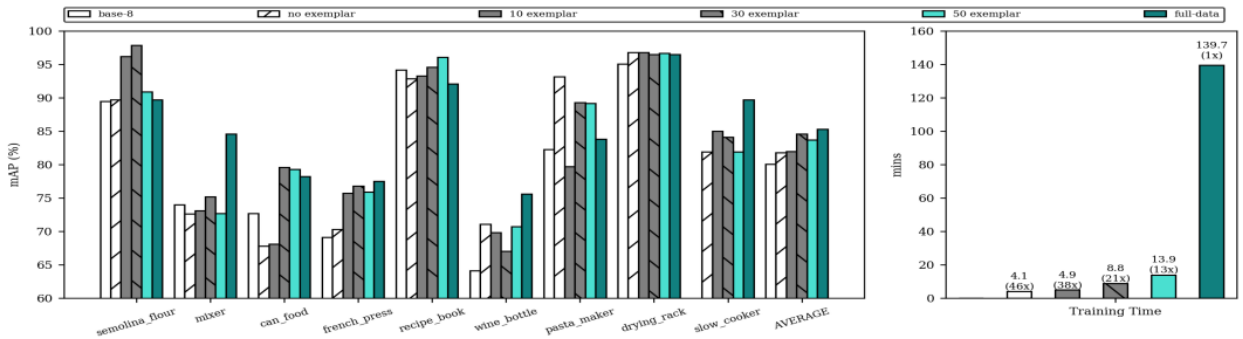

Figure 5. Result of adding "slow cooker" to the base 8 class model on iKitchen Dataset.

Results

Table 3. Credible Labels for PASCAL Dataset.

PASCAL Label	Top Returned Credible Labels
aeroplane	'airplane, aeroplane, plane', 'jet, jet plane, jet-propelled plane', 'jetliner', 'warplane, military plane'
bicycle	'bicycle, bike, wheel, cycle', 'safety bicycle, safety bike', 'push-bike', 'ordinary, ordinary bicycle'
bird	'bird', 'passerine, passeriform bird', 'dickeybird, dickey-bird, dickybird, dicky-bird', 'parrot'
boat	'boat', 'small boat', 'dinghy, dory, rowboat', 'sea boat', 'rowing boat', 'river boat', 'cockleshell'
bottle	'bottle', 'pop bottle, soda bottle', 'water bottle', 'jar', 'smelling bottle', 'flask', 'jug', 'carafe'
bus	'public transport', 'local', 'bus, autobus', 'express, limited', 'shuttle bus', 'trolleybus, trolley coach'
car	'motor vehicle, automotive vehicle', 'car, auto', 'coupe', 'sports car, sport car', 'sedan, saloon'
cat	'domestic cat, house cat', 'kitty, kitty-cat', 'tom, tomcat', 'mouser', 'Manx, Manx cat', 'tabby'
chair	'chair', 'seat', 'armchair', 'straight chair, side chair', 'rocking chair, rocker', 'swivel chair'
cow	'bull', 'cattle, cows', 'cow', 'bullock, steer', 'beef, beef cattle', 'cow, moo-cow', 'dairy cattle'
dining table	'dining-room table', 'dining table, board', 'dinner table', 'table', 'dining-room furniture'
dog	'sporting dog, gun dog', 'terrier', 'retriever', 'hunting dog', 'Labrador retriever', 'water dog'
horse	'horse, Equus caballus', 'equine, equid', 'gelding', 'mare, female horse', 'yearling', 'pony', 'dobbin'
motorbike	'motorcycle, bike', 'wheeled vehicle', 'trail bike, dirt bike, scrambler', 'motor scooter, scooter'
person	'person, individual', 'male, male person', 'face', 'oldster, old person', 'man', 'eccentric person'
potted plant	'pot, flowerpot', 'planter', 'houseplant', 'bucket, pail', 'vase', 'crock, earthenware jar', 'watering pot'
sheep	'sheep', 'domestic sheep, Ovis aries', 'ewe', 'ram, tup', 'black sheep', 'wild sheep', 'mountain sheep'
sofa	'seat', 'sofa, couch, lounge', 'love seat, loveseat', 'chesterfield', 'settee', 'easy chair, lounge chair'
train	'train, railroad train', 'passenger train', 'mail train', 'car train', 'freight train, rattler', 'commuter'
tv/monitor	'monitor', 'LCD', 'television monitor, tv monitor', 'OLED', 'digital display, alphanumeric display'

Table 4. Evaluation on the quality of dataset construction.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	Avg
Retention Rate (deep)	64.09	19.38	78.02	60.00	62.37	59.50	79.50	85.71	61.83	78.86	64.82	74.49	75.90	47.18	50.67	63.30	76.06	67.50	74.32	48.59	64.60
Retention Rate (ebox)	38.00	25.5	26.5	37.37	36.0	35.5	43.5	29.0	48.0	16.29	47.5	23.0	19.0	40.5	26.0	24.5	31.33	41.0	37.0	33.55	32.95
FP Rate (deep)	0.00	12.41	1.74	3.73	11.11	3.01	2.01	4.02	4.71	3.43	15.63	1.16	1.55	9.34	3.51	12.50	2.13	6.12	1.68	7.62	5.37
FP Rate (ebox)	4.5	11.5	19.79	11.16	9.64	2.5	3.51	9.54	6.63	9.55	13.5	14.19	13.0	12.0	14.06	27.71	15.64	11.16	6.03	18.86	11.73

Results

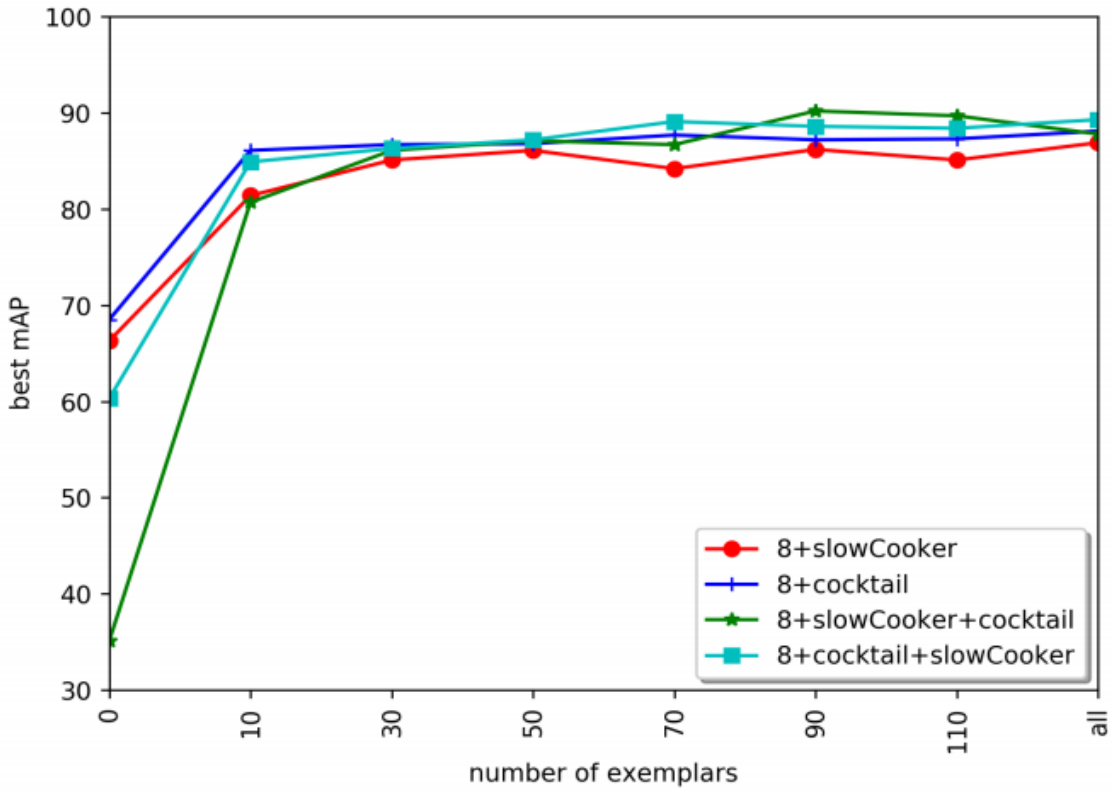

Figure 6. The average mAP over all classes with different number of exemplars per class.

Results

Table 5. iKitchen Accuracy on Automatically Constructed Dataset. ($Base_O$: average mAP for the base 8 classes before learning the new class; $Base_N$: average mAP for the base 8 classes after learning the new class; N: mAP for the new learned class; Avg: average mAP for 9 classes after learning the new class.)

	$Base_O$	$Base_N$	New	Avg
8+SC	80.1%	80.8%	85.4%	81.3%
8+CS	80.1%	79.2%	32.2%	74.0%

Results

Table 6. Training time of different input image size on NVIDIA Tesla M40 (seconds per epoch).

Input Size	**10 Exemplar**	**30 Exemplar**
Small (512)	8.3	14.1
Large (1024)	17.2	30.7

Table 7. System Running Time (s).

	Edge-Only	**Edge-Cloud**
Download image	16	10
Build dataset	44	21
Train model	233	83
Download model	N/A	5
Total	**293**	**119**

1 Introduction

2 Methodology

3 Evaluation

4 Enhancements & Conclusion

Enhancements

- Model optimization using MobileNet
- System enhancements
- Unsupervised bounding box generation
- Scaling the model using intelligent selection of examples of old classes

Conclusion

- The loss function provides insights into what is transferable between similar domains
- It also constrains the various facets that need to be accounted for, such as the intermediary feature maps
- The automated image annotation pipeline is first of its kind in Edge Computing
- Overall, this is an exciting direction of research into Incremental Learning.

Questions?