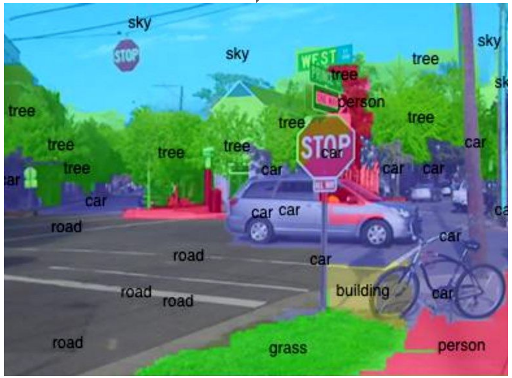
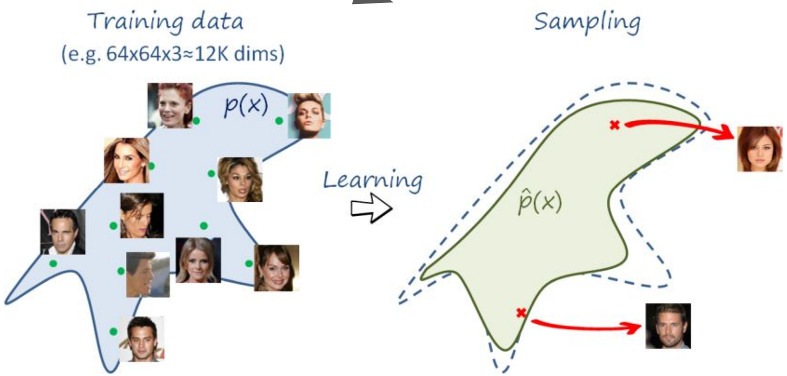
Stanford Artificial Intelligence Lab.

Machine Learning
for Sustainability



Computer Vision



Generative Models

Layout

- Large Scale Pre-training Using Image to Text Matching
- Learning When and Where to Zoom Using Deep Reinforcement Learning
- Efficient Object Detection in Large Images Using Deep Reinforcement Learning

Learning to Interpret Satellite Images using Wikipedia Articles

IJCAI - 2019

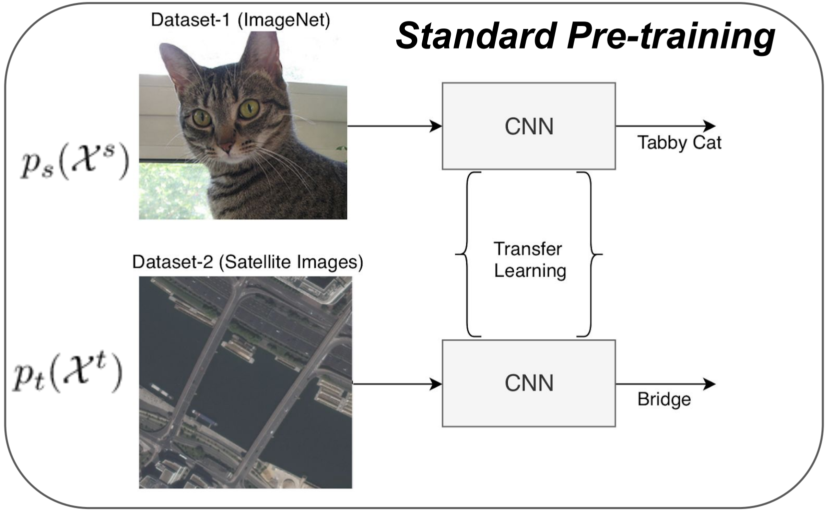
*Burak Uzkent, *Evan Sheehan, *Chenlin Meng, **David Lobell, **Marshall Burke, *Stefano Ermon

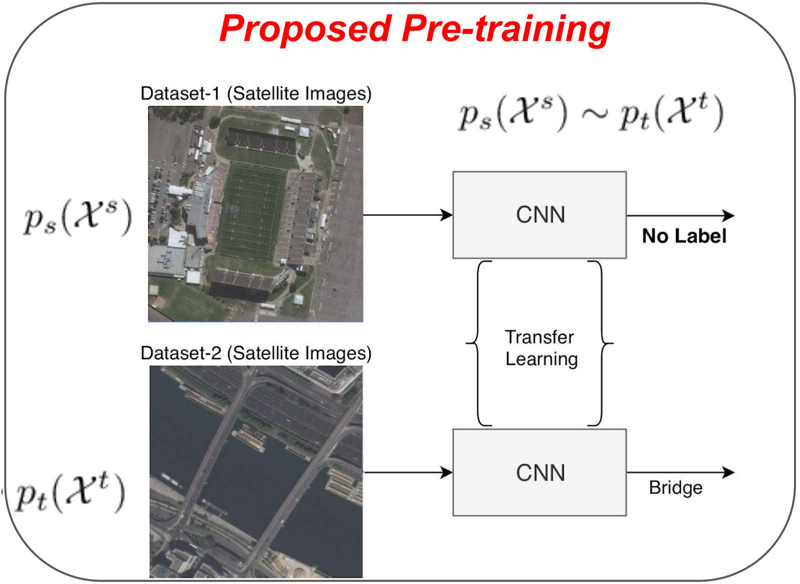
*Department of Computer Science, Stanford University

*Department of Earth Science, Stanford University

Introduction

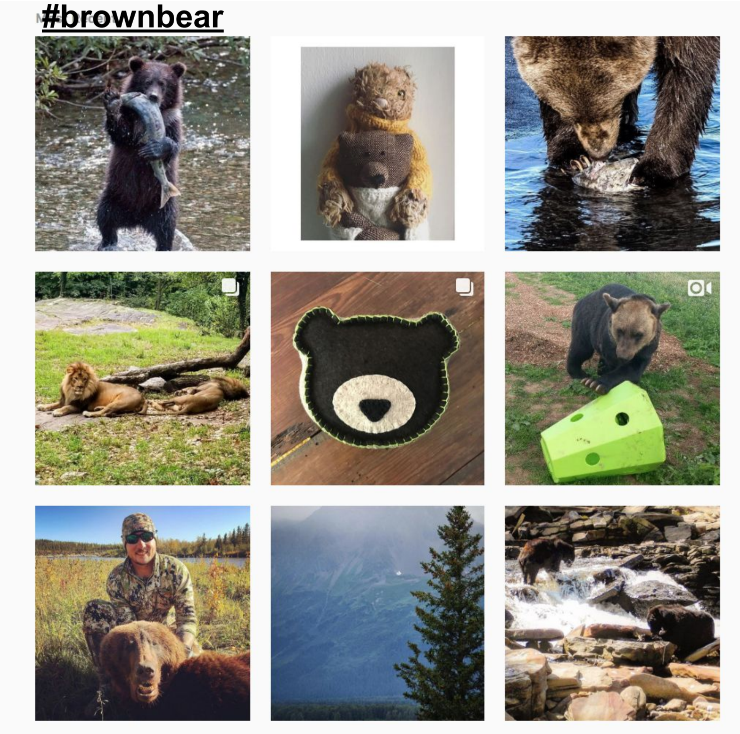
- Deep Neural Networks rely on the following framework:
    - *Pre-train on **ImageNet Dataset**.*
    - *Fine-tune on the **Target Dataset**.*





Related Work - Learning from Instagram Images

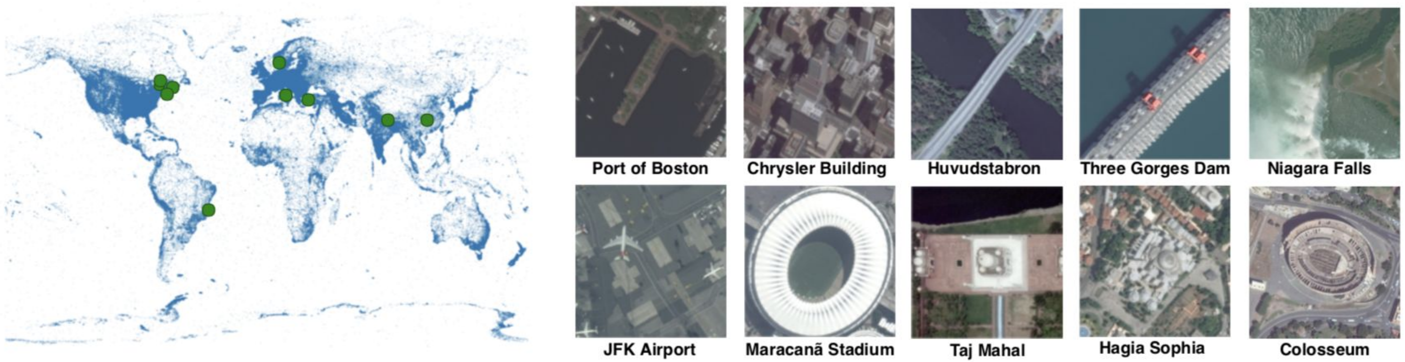
- Mahajan et al. builds an image dataset consisting of **3 billion images** from Instagram.
- They label the images using the hashtags given by the users.
- Pre-training improves recognition accuracy on **ImageNet by *%5*.**



Mahajan, Dhruv, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. "Exploring the limits of weakly supervised pretraining." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 181-196. 2018.

Learning from Satellite Images using Wikipedia Articles

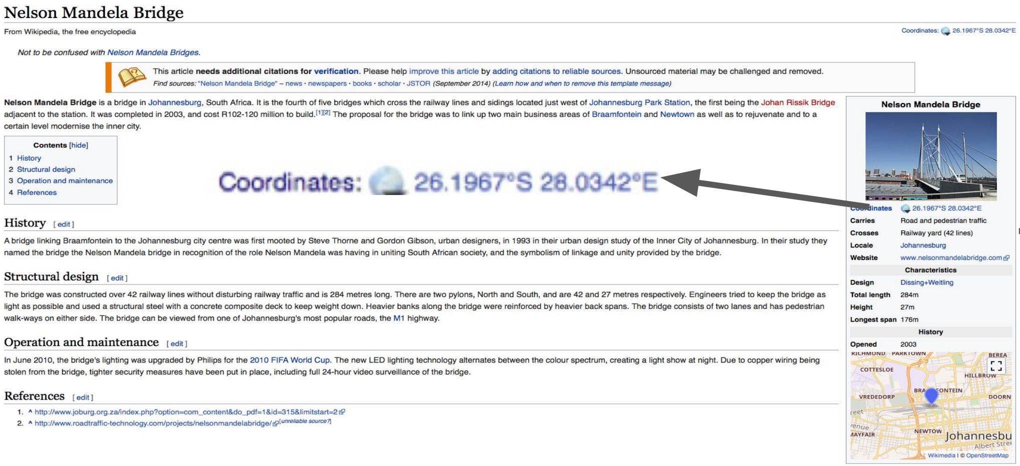
- In its latest dump, Wikipedia contains **~5 million articles (English)** and **~1 million articles** are geo-referenced.



Scatter plot of the distribution of geo-tagged Wikipedia articles together with corresponding high resolution images.

Pairing Articles to Satellite Images - WikiSatNet

$$\mathcal{D} = \{(c_1, x_1, y_1), (c_2, x_2, y_2), \cdots, (c_N, x_N, y_N)\}$$



Nelson Mandela Bridge

From Wikipedia, the free encyclopedia

Coordinates: 26.1967°S 28.0342°E

Not to be confused with Nelson Mandela Bridges.

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "Nelson Mandela Bridge" – news · newspapers · books · scholar · JSTOR (September 2014) (Learn how and when to remove this template message)

Nelson Mandela Bridge is a bridge in Johannesburg, South Africa. It is the fourth of five bridges which cross the railway lines and sidings located just west of Johannesburg Park Station, the first being the Johan Rissik Bridge adjacent to the station. It was completed in 2003, and cost R102-120 million to build.[1][2] The proposal for the bridge was to link up two main business areas of Braamfontein and Newtown as well as to rejuvenate and to a certain level modernise the inner city.

Contents [hide]

1 History
2 Structural design
3 Operation and maintenance
4 References

History [edit]

A bridge linking Braamfontein to the Johannesburg city centre was first mooted by Steve Thorne and Gordon Gibson, urban designers, in 1993 in their urban design study of the Inner City of Johannesburg. In their study they named the bridge the Nelson Mandela bridge in recognition of the role Nelson Mandela was having in uniting South African society, and the symbolism of linkage and unity provided by the bridge.

Structural design [edit]

The bridge was constructed over 42 railway lines without disturbing railway traffic and is 284 metres long. There are two pylons, North and South, and are 42 and 27 metres respectively. Engineers tried to keep the bridge as light as possible and used a structural steel with a concrete composite deck to keep weight down. Heavier banks along the bridge were reinforced by heavier back spans. The bridge consists of two lanes and has pedestrian walk-ways on either side. The bridge can be viewed from one of Johannesburg's most popular roads, the M1 highway.

Operation and maintenance [edit]

In June 2010, the bridge's lighting was upgraded by Philips for the 2010 FIFA World Cup. The new LED lighting technology alternates between the colour spectrum, creating a light show at night. Due to copper wiring being stolen from the bridge, tighter security measures have been put in place, including full 24-hour video surveillance of the bridge.

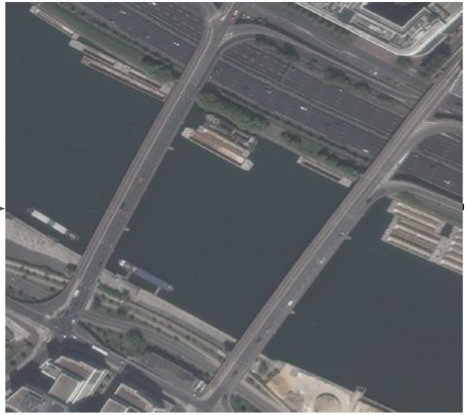
References [edit]

1. ^ http://www.joburg.org.za/index.php?option=com_content&do_pdf=1&id=315&limitstart=2
2. ^ http://www.roadtraffic-technology.com/projects/nelsonmandelabridge/[unreliable source?]

Nelson Mandela Bridge	
Coordinates	26.1967°S 28.0342°E
Carries	Road and pedestrian traffic
Crosses	Railway yard (42 lines)
Locale	Johannesburg
Website	www.nelsonmandelabridge.com
Characteristics	
Design	Dissing+Weitling
Total length	284m
Height	27m
Longest span	176m
History	
Opened	2003

Coordinates: 26.1967°S 28.0342°E

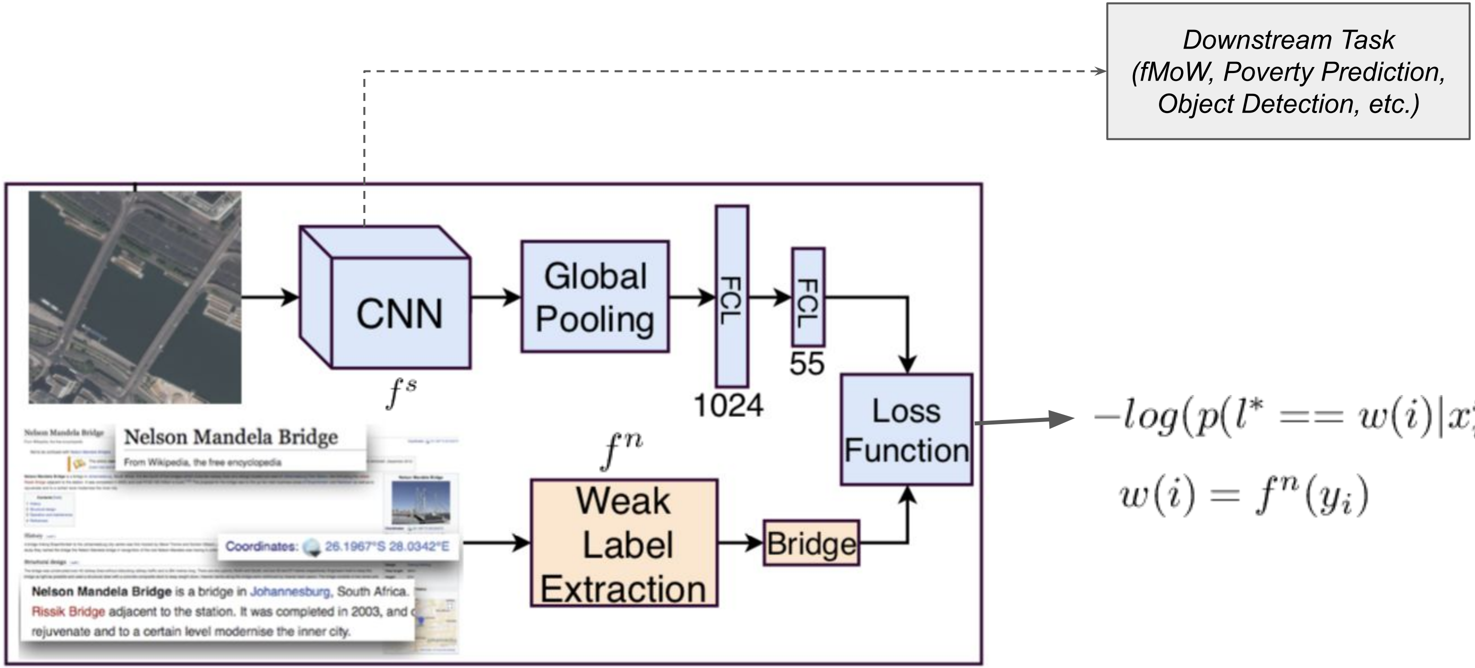
Pair to an overhead image



Coordinates: 26.1967°S 28.0342°E

Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D. and Jawahar, C.V., 2017. Self-supervised learning of visual features through embedding images into text topic spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4230-4239).

Representation Learning with Weak Labels

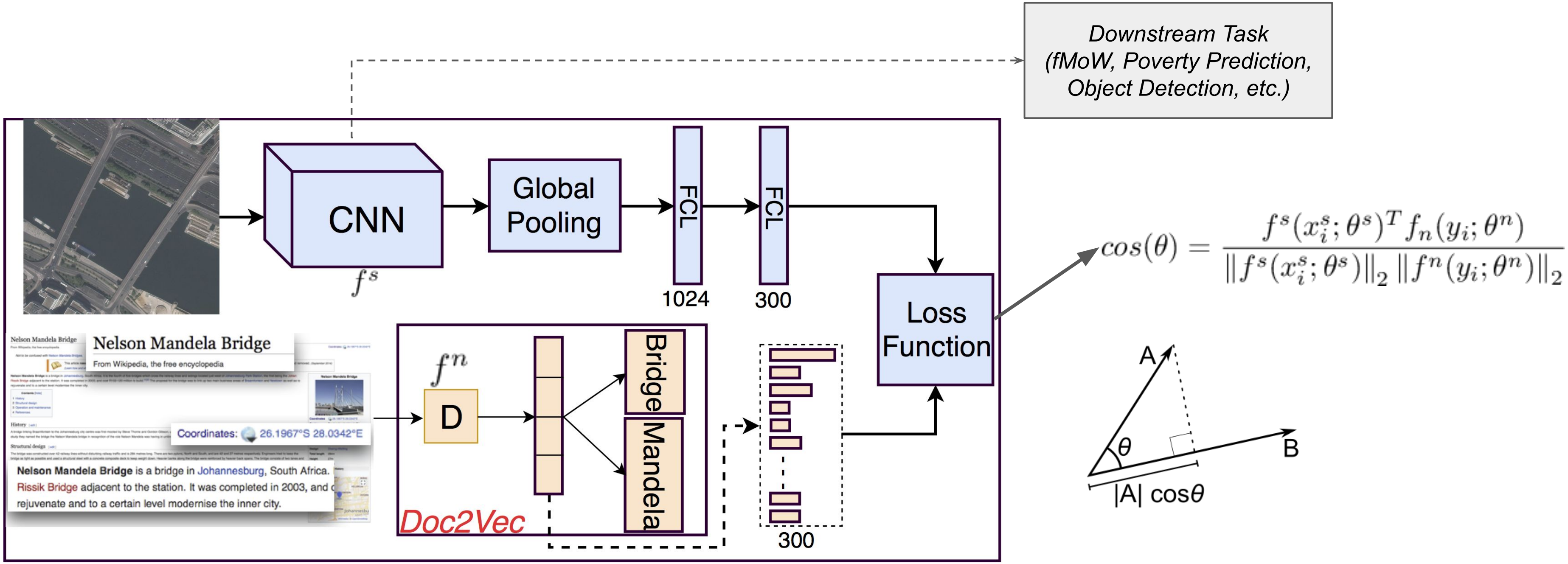


$$-log(p(l^* == w(i)|x_i^s;\theta^s))$$

$$w(i) = f^n(y_i)$$

***Requires human intervention and heuristics.**

Representation Learning with Image2Text Matching



***A more automatic approach.**

Flipped Label Noise

Tagged as *‘INCIDENT’*

Iserbrook (ship)

Iserbrook was a general cargo and passenger brig built in 1853 at Hamburg (Germany) for *Joh. Ces. Godeffroy & Sohn*. It spent over twenty years as an immigrant and general cargo vessel, transporting passengers from Hamburg to South Africa, Australia and Chile, as well as servicing its owner’s business in the Pacific. Later on, the vessel came into Australian possession and continued sailing for the Pacific trade. In 1878 it caught fire and was sunk the same year. At last, it was re-floated and used as a transport barge and hulk in Sydney until it sunk again and finally was blown up.

Construction and Description

The 240 ton Brig *Cesar & Helene* was built in 1855/56 in the Godeffroy shipyard at the Reiherstieg wharf. This vessel was just 30 tones larger and built one year after the *Iserbrook* for the same owners

The vessel was built for the Hamburg trading company *Joh. Ces. Godeffroy & Sohn*. At the time, the enterprise was operated by Johan César VI. Godeffroy who had large trading interests in the Pacific, focussing mainly on Copra, Coconut oil and luxuries like pearlshell. In the 1850s and 60s, the company was also strongly associated with emigration from Germany to Australia, especially to Adelaide and Brisbane.

In its original Hamburg registration (Bielbrief),

*The word “*Water*” is mentioned 10 times in the article.
*The word “*Sea*” is mentioned 11 times in the article
*The word “*Port*” is mentioned 11 times in the article

Adversarial Label Noise

Town

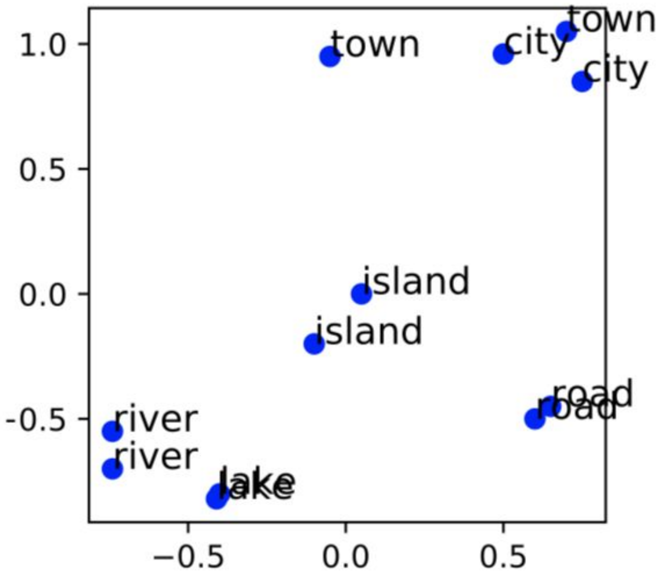
Country

City

Town

***It is hard to come up with a single label when some labels are sampled from similar distribution.**

Analyzing Doc2Vec Model

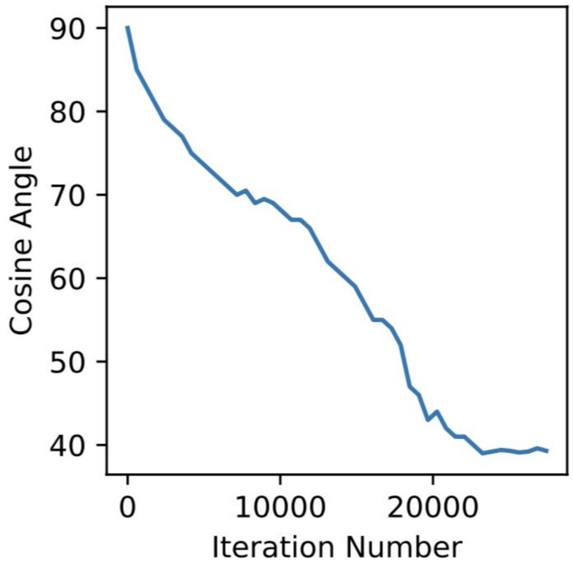


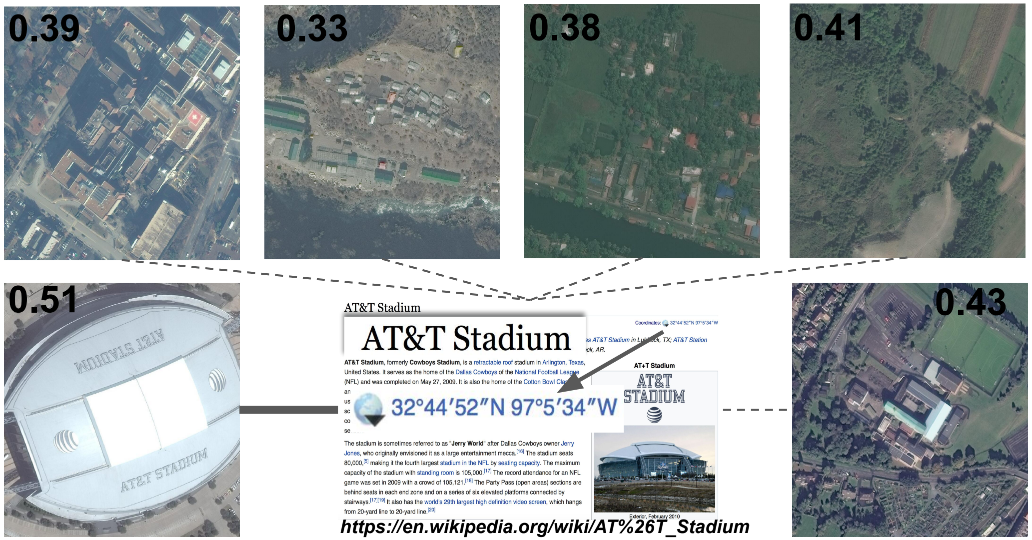
City - Middletown, Connecticut
City - Milton, Georgia
Lake - Timothy Lake
Lake - Tinquilco Lake
Town - Mingona Township, Kansas
Town - Moon Township, Pennsylvania
Road - Morehampton Road, Dublin
Road - Motorway M10 Pakistan
River - Motru River
River - Mousam River
Island - Aupaluktok Island
Island - Avatanak Island

***Articles with similar content are projected to the similar latent space.**

Image2Text Matching Pre-training Experiments

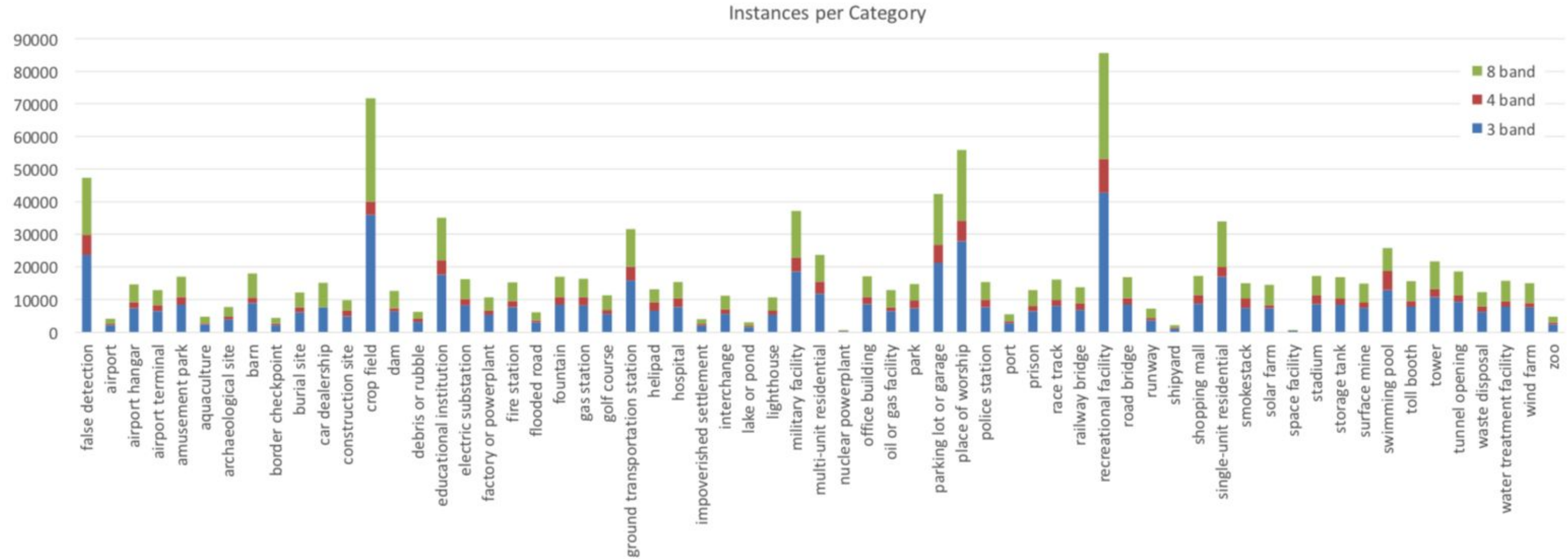
- We use DenseNet with 121 layers to parameterize the CNN.





Target Task- functional Map of the World (fMoW)

- It includes 350k, 50k, 50k samples across 62 classes from the training, validation, and test sets.



Christie, Gordon, Neil Fendley, James Wilson, and Ryan Mukherjee. "Functional map of the world." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6172-6180. 2018.

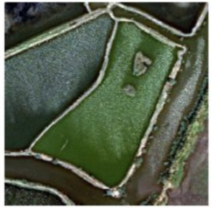
Examples

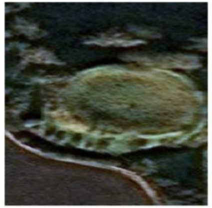

airport


airport hangar


airport terminal


amusement park


aquaculture


archaeological site


barn


border checkpoint


burial site


car dealership


construction site


crop field


dam


debris or rubble

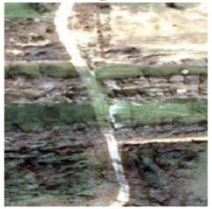

educational institution


electric substation


factory or powerplant


false detection


fire station


flooded road

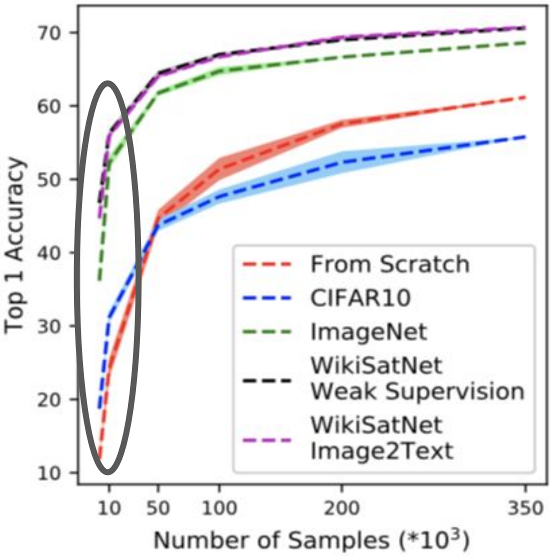

fountain


gas station


golf course


ground transportation station

Image Classification on fMoW

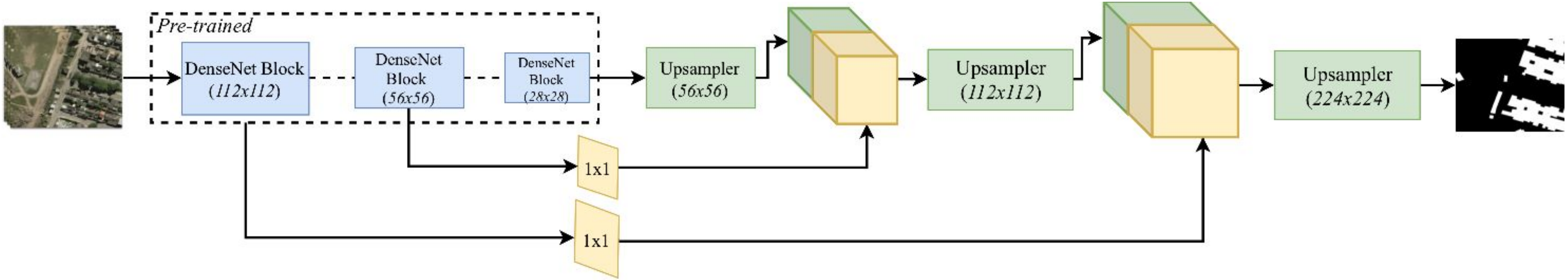


*Gap decreases w.r.t sample complexity

Model	CIFAR10	ImageNet	WikiSatNet *Weak Labels*	WikiSatNet *Image2Text*
F1 Score (*Single View*)	55.34 (%)	64.71 (%)	66.17 (%)	**67.12** (%)
F1 Score (*Temporal Views*)	60.45 (%)	68.73 (%)	71.31 (%)	**73.02** (%)

Table 1: F1 scores of pre-training methods on fMoW's test set.

Building Segmentation on SpaceNet



Model	From Scratch	ImageNet	WikiSatNet *Image2Text*
200 Samples	42.11 (%)	50.75 (%)	**51.70** (%)
500 Samples	48.98 (%)	54.63 (%)	**55.41** (%)
5000 Samples	57.21 (%)	59.63 (%)	**59.74** (%)

Table 2: mIoU scores of pre-training methods on SpaceNet test set.

***Pre-training works best when we consider the same level tasks. (He et. al CVPR 2019)**

Learning Where and When to Zoom using Deep Reinforcement Learning

CVPR - 2020

Burak Uzkent, Stefano Ermon

Department of Computer Science, Stanford University

Motivation

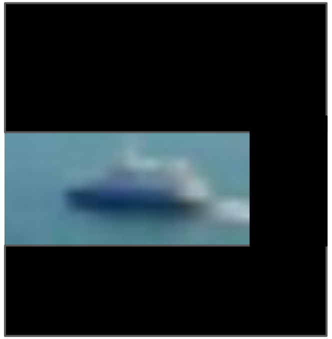
- Understanding the salient parts of an image is an important research field in computer vision.
- In our study, we pose it as a Reinforcement Learning task and train an RL agent to learn *patch dropping policies*.

*Do we need the full image to be able to classify this image as ship?

*Can we just process small part of this image and identify that it is ship?

Motivation

- Understanding the salient parts of an image is an important research field in computer vision.
- In our study, we pose it as a Reinforcement Learning task and train an RL agent to learn *patch dropping policies*.

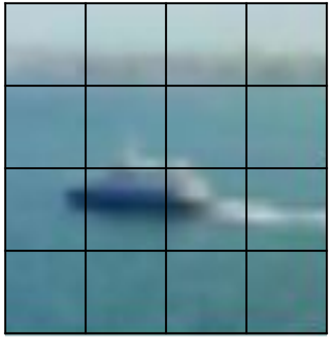


<u>*If we process less number of pixels, we can build more efficient models.</u>

How Robust is CNNs to Patch Dropping?

Do we need all the patches in an image to infer correct decisions?

We train a ResNet32 on CIFAR10 and test it with random patch drop policy.

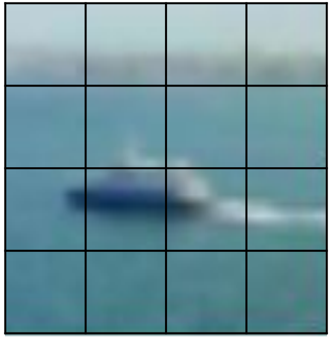


92.3%

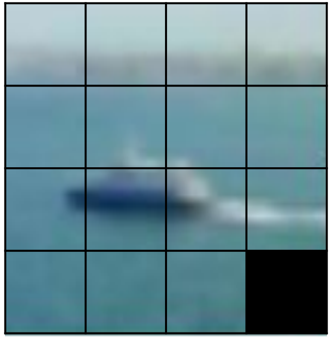
How Robust is CNNs to Patch Dropping?

Do we need all the patches in an image to infer correct decisions?

We train a ResNet32 on CIFAR10 and test it with random patch drop policy.



92.3%

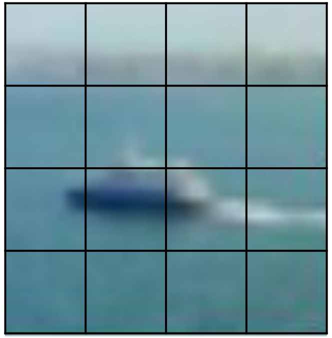


91.1%

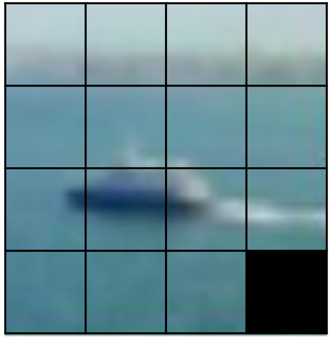
How Robust is CNNs to Patch Dropping?

Do we need all the patches in an image to infer correct decisions?

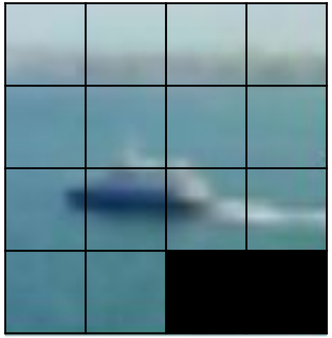
We train a ResNet32 on CIFAR10 and test it with random patch drop policy.



92.3%



91.1%

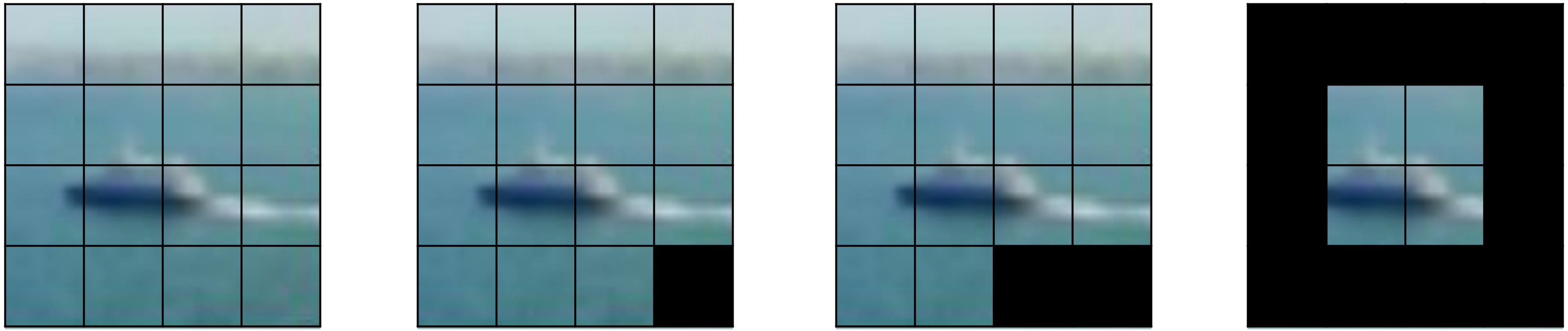


88.4%

How Robust is CNNs to Patch Dropping?

Do we need all the patches in an image to infer correct decisions?

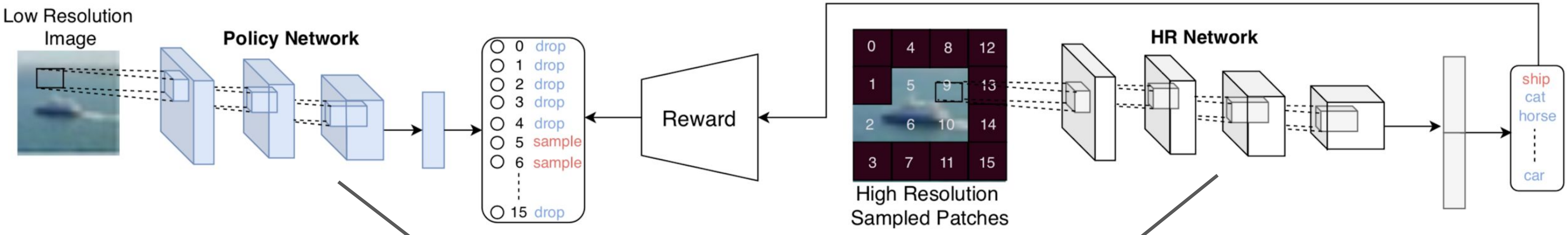
We train a ResNet32 on CIFAR10 and test it with random patch drop policy.



92.3% 91.1% 88.4% 46.3%

Can we design a conditional patch dropping strategy?

PatchDrop - Proposed Solution



Policy Network

Classifier

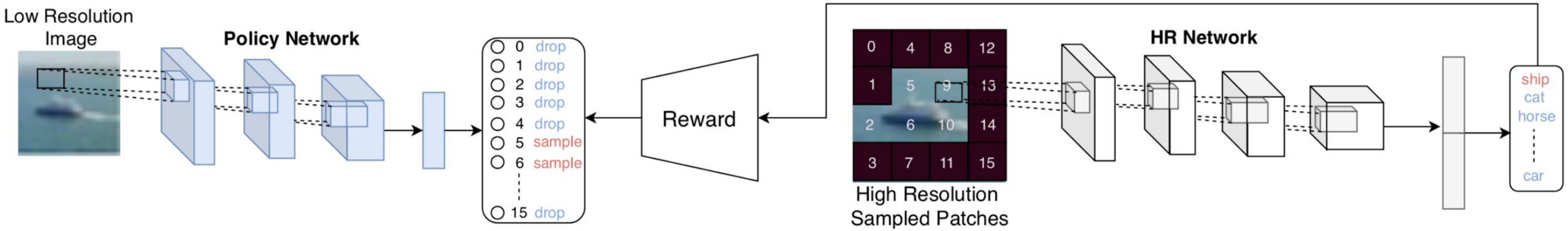
Policies -> $\pi_1(\mathbf{a_1}|x_l;\theta_p) = p(\mathbf{a_1}|x_l;\theta_p)$ $\pi_2(\mathbf{a_2}|x_h^m;\theta_{cl}) = p(\mathbf{a_2}|x_h^m;\theta_{cl})$

Actions -> $\mathbf{a}_1 \in \{0, 1\}^P$ $\mathbf{a}_2 \in \{0, 1, \ldots N\}$

*Conditioning the Policy Network on LR images introduces minimal computational overhead.

*In some domains, i.e. remote sensing, LR images are more affordable than HR images.

Modeling the Policy Network and Classifier



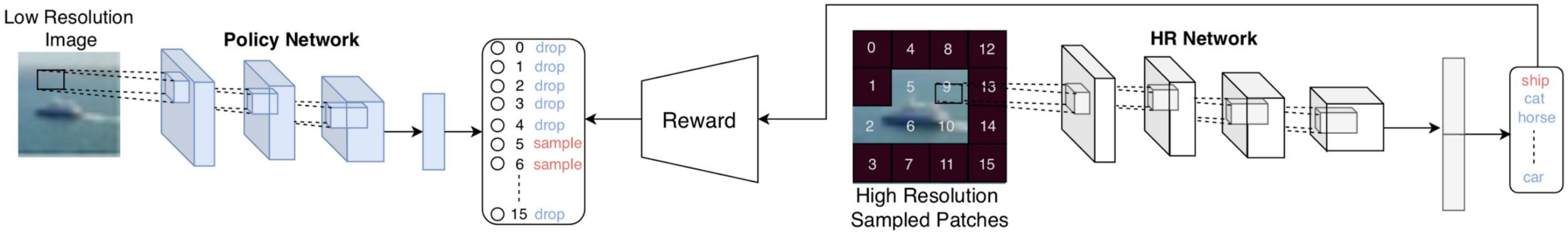
Patch Sampling Policy-> $$\pi_1(\mathbf{a}_1|x_l,\theta_p) = \prod_{p=1}^{P} s_p^{\mathbf{a}_1^{\mathbf{p}}}(1-s_p)^{(1-\mathbf{a}_1^{\mathbf{p}})}$$

Policy Network Predictions-> $$s_p = f_p(x_l;\theta_p) \qquad s_p \in [0,1]$$

Classifier Predictions-> $$s_{cl} = f_c(x_h^m;\theta_{cl})$$

Cost Function-> $$\boxed{\max_{\theta_p} J(\theta_p,\theta_{cl}) = \mathbb{E}_p[R(\mathbf{a_1},\mathbf{a_2},y)]}$$

Modeling the Reward Function



Reward Function->

$$R(\mathbf{a}_1, \mathbf{a}_2, y) = \begin{cases} 1 - \left(\frac{|\mathbf{a}_1|_1}{P}\right)^2 & \text{if } y = \hat{y}(\mathbf{a}_2) \\ -\sigma & \text{Otherwise.} \end{cases}$$

Cost Function->

$$\boxed{\max_{\theta_p} J(\theta_p, \theta_{cl}) = \mathbb{E}_p[R(\mathbf{a}_1, \mathbf{a}_2, y)]}$$

NOT Differentiable!

Optimizing the Policy Network

- We train the Policy Network using the Policy Gradient Algorithm.

Cost Function->

$$\nabla_{\theta_p} J = \mathbb{E}[R(\mathbf{a_1}, \mathbf{a_2}, y)\nabla_{\theta_p} \boxed{\log \pi_{\theta_p}(\mathbf{a_1}|x_l)}]$$

Differentiable!

$$\nabla_{\theta_p} J = \mathbb{E}[A \sum_{p=1}^{P} \nabla_{\theta_p} \log(s_p \mathbf{a_1^p} + (1 - s_p)(1 - \mathbf{a_1^p}))]$$

Advantage Function ->

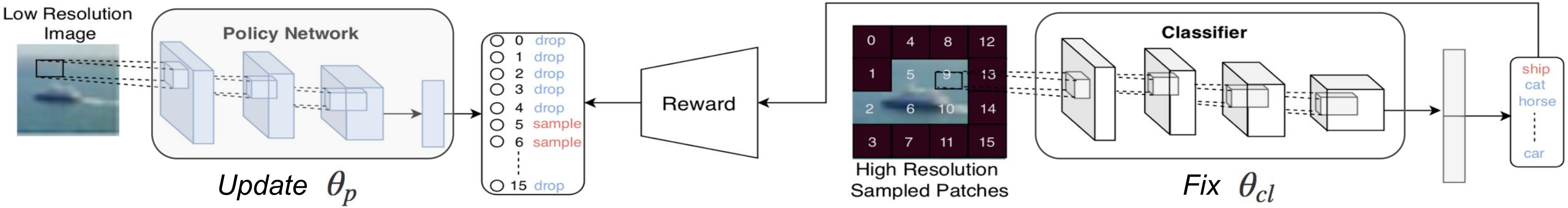
$$A(\mathbf{a_1}, \hat{\mathbf{a}}_1, \mathbf{a_2}, \hat{\mathbf{a}}_2) = R(\mathbf{a_1}, \mathbf{a_2}, y) - R(\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, y)$$

Temperature Scaling ->

$$s_p = \alpha s_p + (1 - \alpha)(1 - s_p)$$

Pre-training Stage

- First, we train the classifier using original images.
- Next, we fix the classifier's weights and train the policy network.

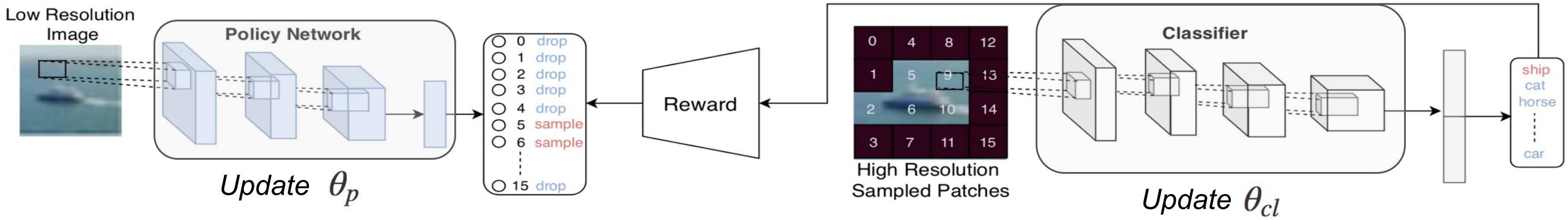


Pre-training Stage

- The policy network learns *informative* patches however the accuracy is reduced since the classifier is not trained on masked images.

Jointly Fine-tuning the Policy Network and Classifier

- We fine-tune the classifier jointly with the policy network.
- The classifier updates itself to adapt to the learned masked images and policy network updates the learned policies.



Joint Fine-tuning Stage

- In this step, we learn to drop more patches while increasing the accuracy w.r.t to the pre-training stage.

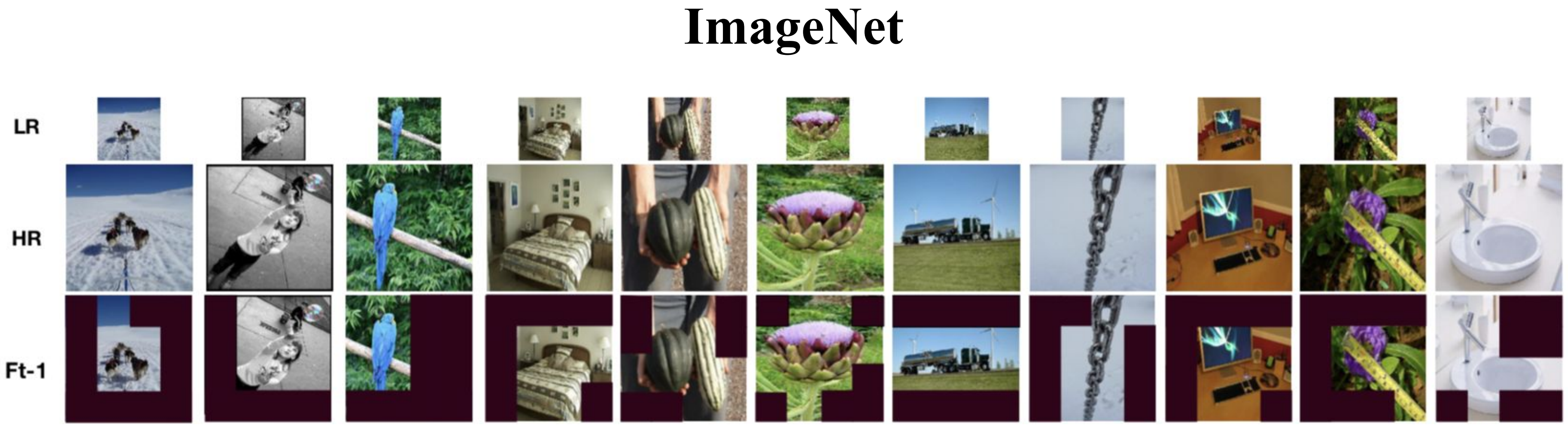
Experiments on CIFAR10/CIFAR100/ImageNet

- For CIFAR10/100, we use 45k, 5k, and 10k training, validation and test samples and for ImageNet, we use 1.2 million, 50k, and 150k training, validation and test images.

	CIFAR10				CIFAR100				ImageNet			
	Acc. (%) (Pt)	Acc. (%) (Ft-1)	Acc. (%) (Ft-2)	S (Pt,Ft-1,Ft-2)	Acc. (%) (Pt)	Acc. (%) (Ft-1)	Acc. (%) (Ft-2)	S (Pt,Ft-1,Ft-2)	Acc. (%) (Pt)	Acc. (%) (Ft-1)	Acc. (%) (Ft-2)	S (Pt,Ft-1,Ft-2)
LR-CNN	75.8	75.8	75.8	0,0,0	55.1	55.1	55.1	0,0,0	58.1	58.1	58.1	0,0,0
SRGAN [19]	78.8	78.8	78.8	0,0,0	56.1	56.1	56.1	0,0,0	63.1	63.1	63.1	0,0,0
KD [37]	81.8	81.8	81.8	0,0,0	61.1	61.1	61.1	0,0,0	62.4	62.4	62.4	0,0,0
PCN [37]	83.3	83.3	83.3	0,0,0	62.6	62.6	62.6	0,0,0	63.9	63.9	63.9	0,0,0
HR-CNN	92.3	92.3	92.3	16,16,16	69.3	69.3	69.3	16,16,16	76.5	76.5	76.5	16,16,16
Fixed-H	71.2	83.8	85.2	9,8,7	48.5	65.8	67.0	9,10,10	48.8	68.6	70.4	10,9,8
Fixed-V	64.7	83.4	85.1	9,8,7	46.2	65.5	67.2	9,10,10	48.4	68.4	70.8	10,9,8
Stochastic	40.6	82.1	83.7	9,8,7	27.6	63.2	64.8	9,10,10	38.6	66.2	68.4	10,9,8
STN [31]	66.9	85.2	87.1	9,8,7	41.1	64.3	66.4	9,10,10	58.6	69.4	71.4	10,9,8
PatchDrop	**80.6**	**91.9**	**91.5**	**8.5,7.9,6.9**	**57.3**	**69.3**	**70.4**	**9,9.9,9.1**	**60.2**	**74.9**	**76.0**	**10.1,9.1,7.9**

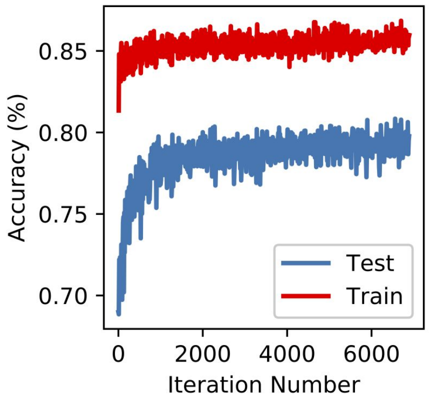
The proposed framework drops about %40-%60 of the patches while maintaining the classification accuracy of the model using original HR images.

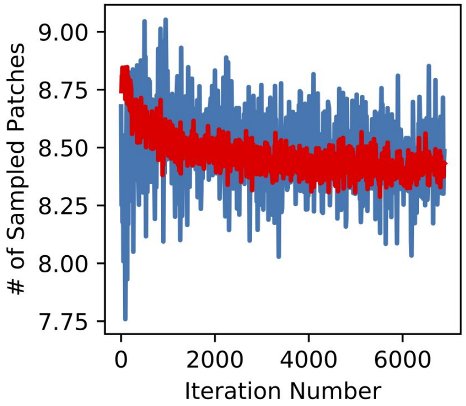
Learned Patch Sampling Policies



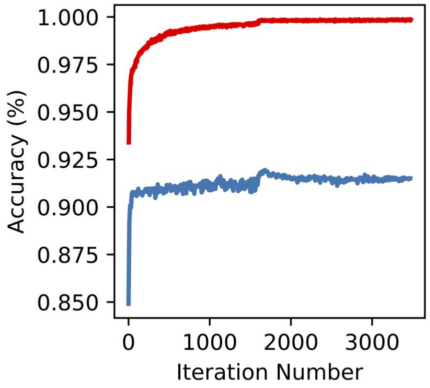
Impact of Joint Fine-tuning

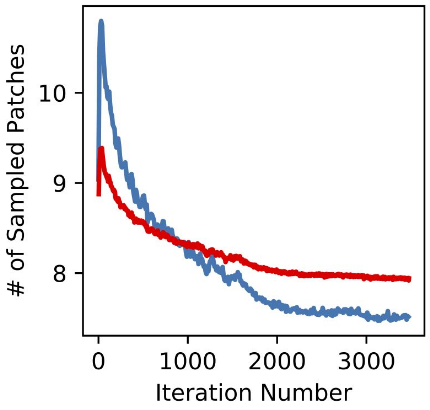
Pre-training



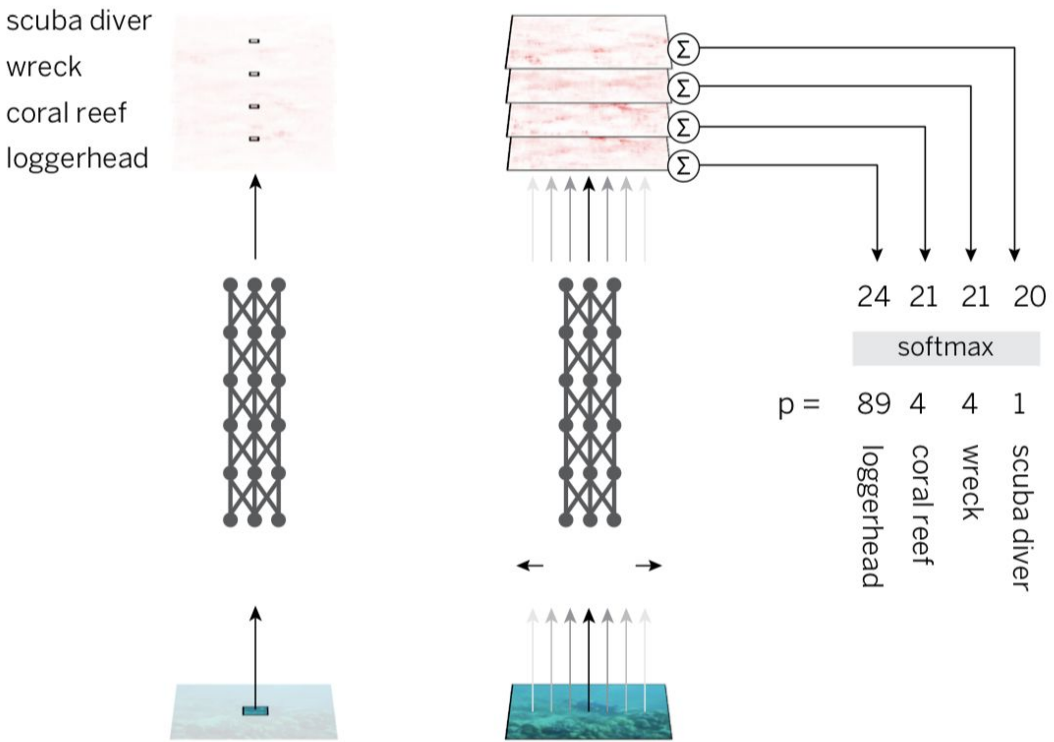


Joint Fine-tuning



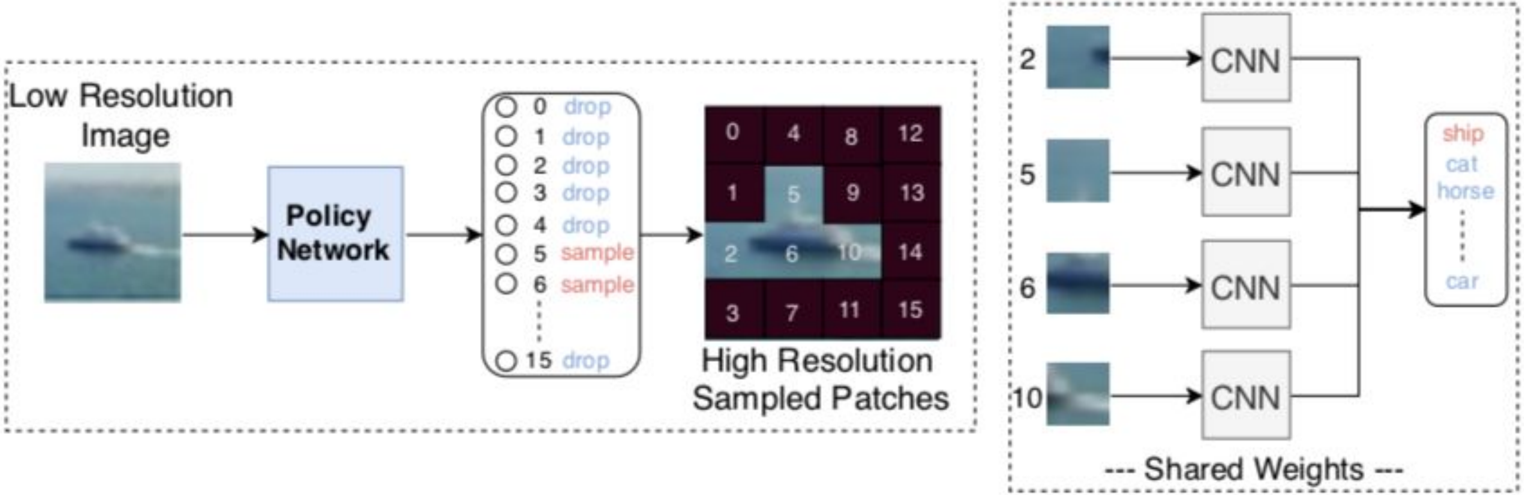


BagNets (Brenden et al. ICLR 2019)



Brendel, Wieland, and Matthias Bethge. "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet." *arXiv preprint arXiv:1904.00760* (2019).

Conditional BagNets - Experiments on CIFAR10



Conditional BagNets - Experiments on CIFAR10

	Acc. (%) (Pt)	S	Acc. (%) (Ft-1)	S	Run-time. (%) (ms)
BagNet (No Patch Drop)	85.6	16	85.6	16	192
CNN (No Patch Drop)	92.3	16	92.3	16	77
Fixed-H	67.7	10	86.3	9	98
Fixed-V	68.3	10	86.2	9	98
Stochastic	49.1	10	83.1	9	98
STN	67.5	10	86.8	9	112
BagNet (PatchDrop)	**77.4**	**9.5**	**92.7**	**8.5**	98

Table 1: Results on the CIFAR10 dataset. S represents the number of sampled patches.

Brendel, Wieland, and Matthias Bethge. "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet." *arXiv preprint arXiv:1904.00760* (2019).

Conditional Hard Positive Generation

	CIFAR10 (%) (ResNet32)	CIFAR100 (%) (ResNet32)	ImageNet (%) (ResNet50)	fMoW (%) (ResNet34)
No Augment.	92.3	69.3	76.5	67.3
CutOut	93.5	70.4	76.5	67.6
PatchDrop	**93.9**	**71.0**	**78.1**	**69.6**

Table 2: Accuracies on different benchmark after adversarial training.

DeVries, Terrance, and Graham W. Taylor. "Improved regularization of convolutional neural networks with cutout." *arXiv preprint arXiv:1708.04552* (2017).

Efficient Object Detection in Large Images Using Deep Reinforcement Learning

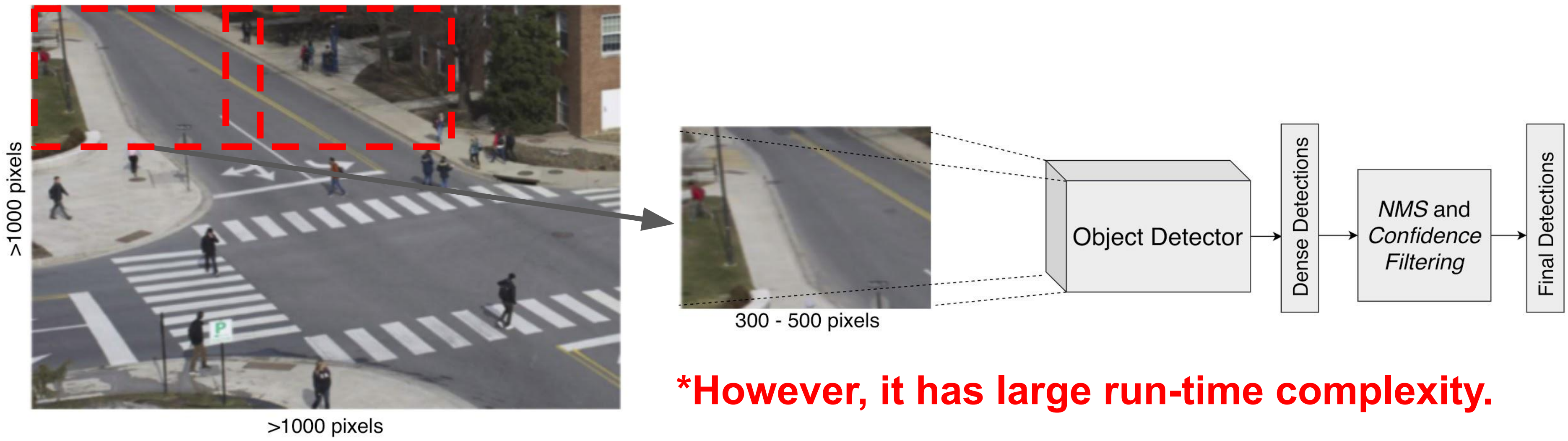
WACV - 2020

Burak Uzkent, Christopher Yeh, Stefano Ermon

Department of Computer Science, Stanford University

Detection in Large Images - Sliding Window

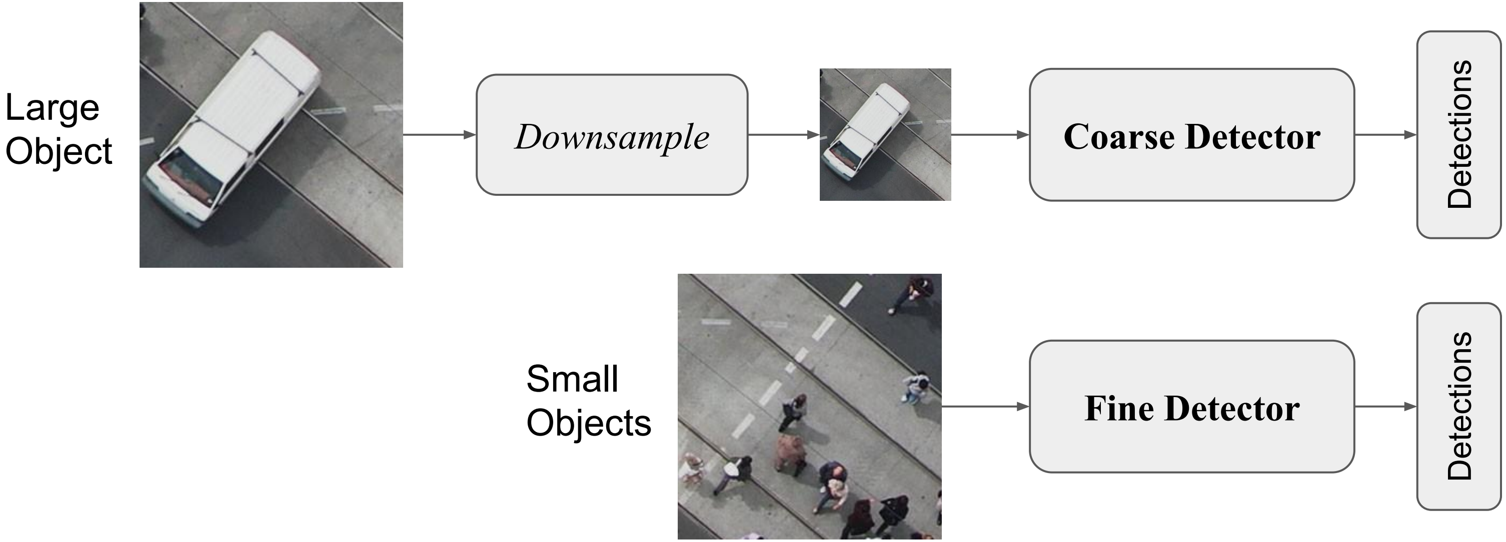
- Large images are processed with sliding window approach since
  - We do not need to downsample
  - It has low memory requirement

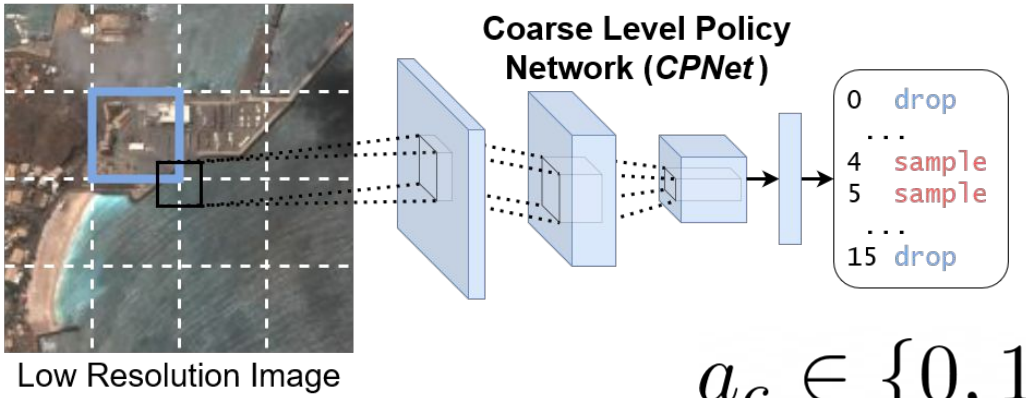


***However, it has large run-time complexity.**

Proposed Solution - Adaptive Sliding Window

- Small objects requires fine-level information whereas large objects can be detected at coarse-level.



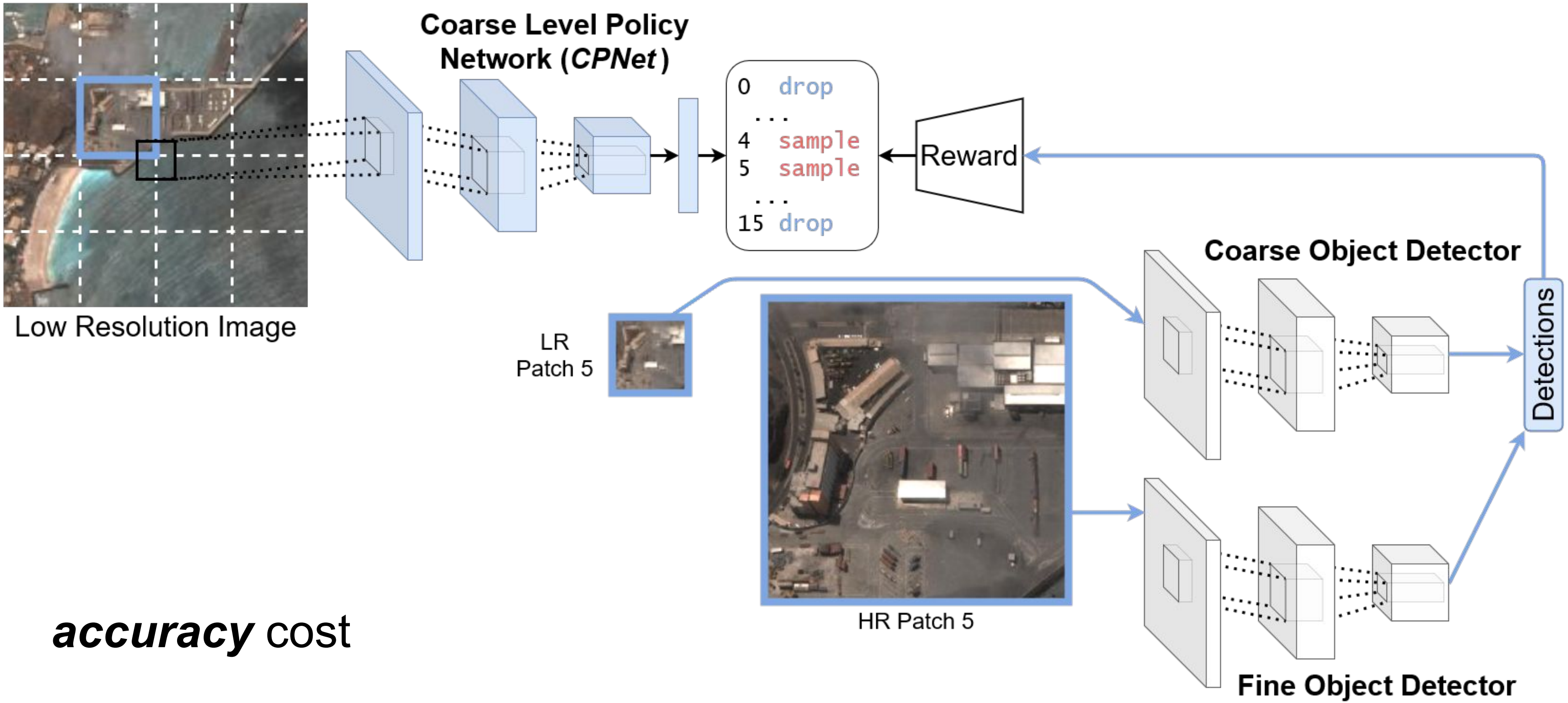


$$a_c \in \{0,1\}^{P_c}$$

$$s_c = f_p^c(x_L; \theta_p^c)$$

$$\pi_c(a_c | x_L; \theta_p^c) = p(a_c | x_L; \theta_p^c)$$

*The goal is to learn zooming-in policies.



accuracy cost

$$R_{acc} = \sum_{i=1}^{P_c} \left(Recall\left(\hat{Y}_i^f, Y_i\right) - \left(Recall\left(\hat{Y}_i^c, Y_i\right) + \beta \right) \right) \cdot N_i$$

Modeling the Policy Networks

Policy network treats sampling each image patch as a Bernoulli variable.

$$\pi_c(a_c|x_L, \theta_p^c) = \prod_{i=1}^{P_c} (s_c^i)^{a_c^i}(1 - s_c^i)^{(1-a_c^i)}$$

$$s_c = f_p^c(x_L; \theta_p^c)$$

Policy network is trained with policy gradient method, with advantage function.

$$J_c = \mathbb{E}\left[R_c(a_c, a_d, Y)\right]$$

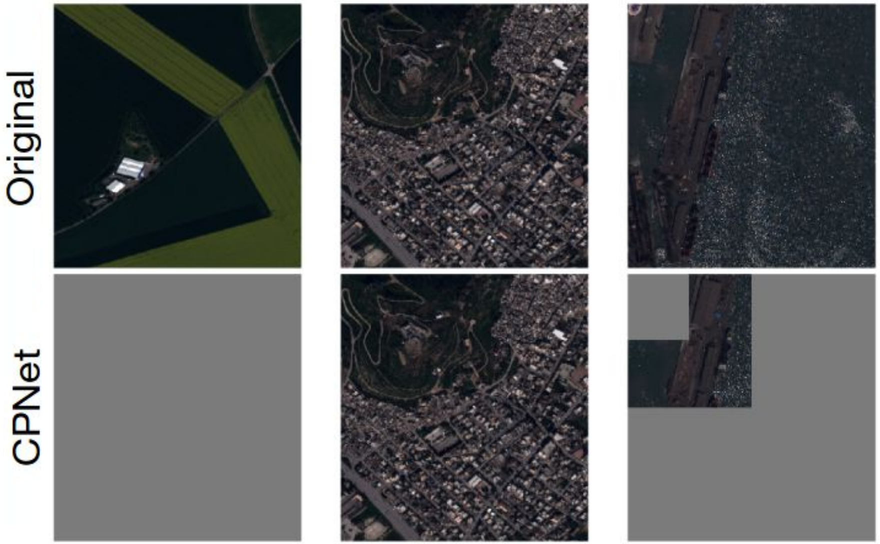
$$\nabla_{\theta_p^c} J_c = \mathbb{E}\left[R_c \cdot \nabla_{\theta_p^c} \log \pi_{\theta_p^c}(a_c|x_L)\right]$$

Experiments - xView

- Experiments on the xView dataset, consisting of 847 very large images (***>3000*** x ***>3000*** px).

Model/Metric	HR	AP	AR	Run-time	HR
Random (5×)	50	24.1	47.1	1408	31
Entropy (5×)	50	25.4	47.2	1415	31
Sliding Window-L (5×)	0	26.3	39.8	640	0
Sliding Window-H	100	39.0	60.9	3200	100
Gao et al. [7] (5×)	35.4	35.2	55.5	1551	31.6
Ours (5×)	35.5	38.1	59.7	1484	31.5

Table 1 : Results on the xView test set.



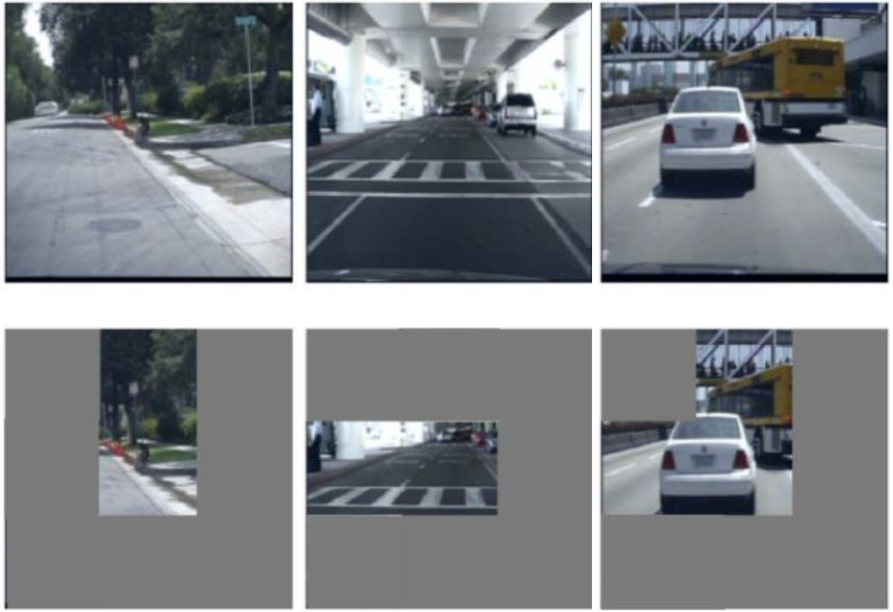
Gao, Mingfei, Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. "Dynamic zoom-in network for fast object detection in large images." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6926-6935. 2018.

Experiments - Caltech Pedestrian

- Experiments on the Caltech Pedestrian dataset (***>800*** x ***>800*** px).

Model/Metric	AP	AR	Run-time	HR
Random (×5)	30.9	62.1	248	44.4
Entropy (×5)	34.0	63.9	250	44.4
Sliding Window-L (×5)	21.2	46.3	90	0
Sliding Window-H	64.7	74.7	450	100
Gao et al. [7] (×2)	64.5	73.1	295	7.1
Gao et al. [7] (×5)	57.3	70.7	309	43.3
CPNet (×2)	64.4	74.5	267	6.6
CPNet (×5)	61.7	74.1	270	44.4

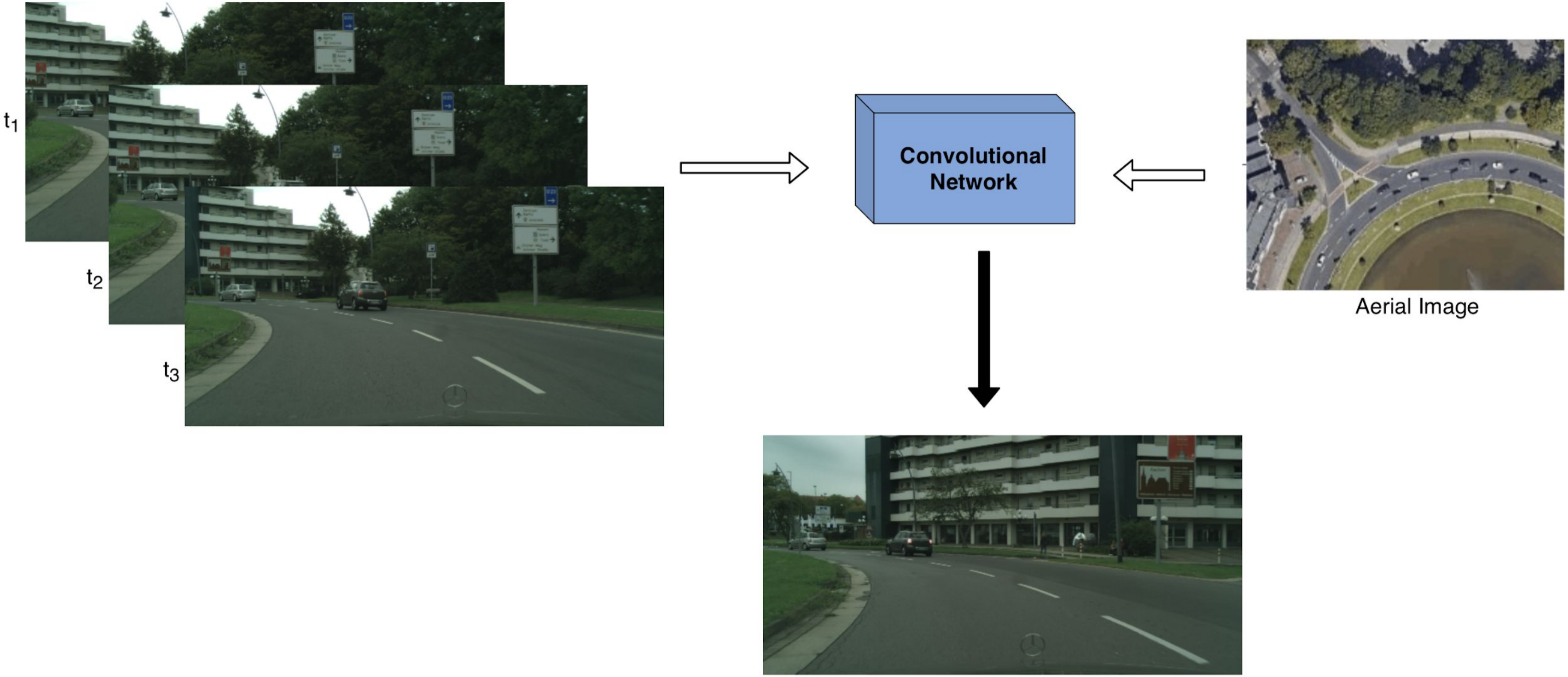
Table 2 : Results on the Caltech Pedestrian test set.



Thanks!

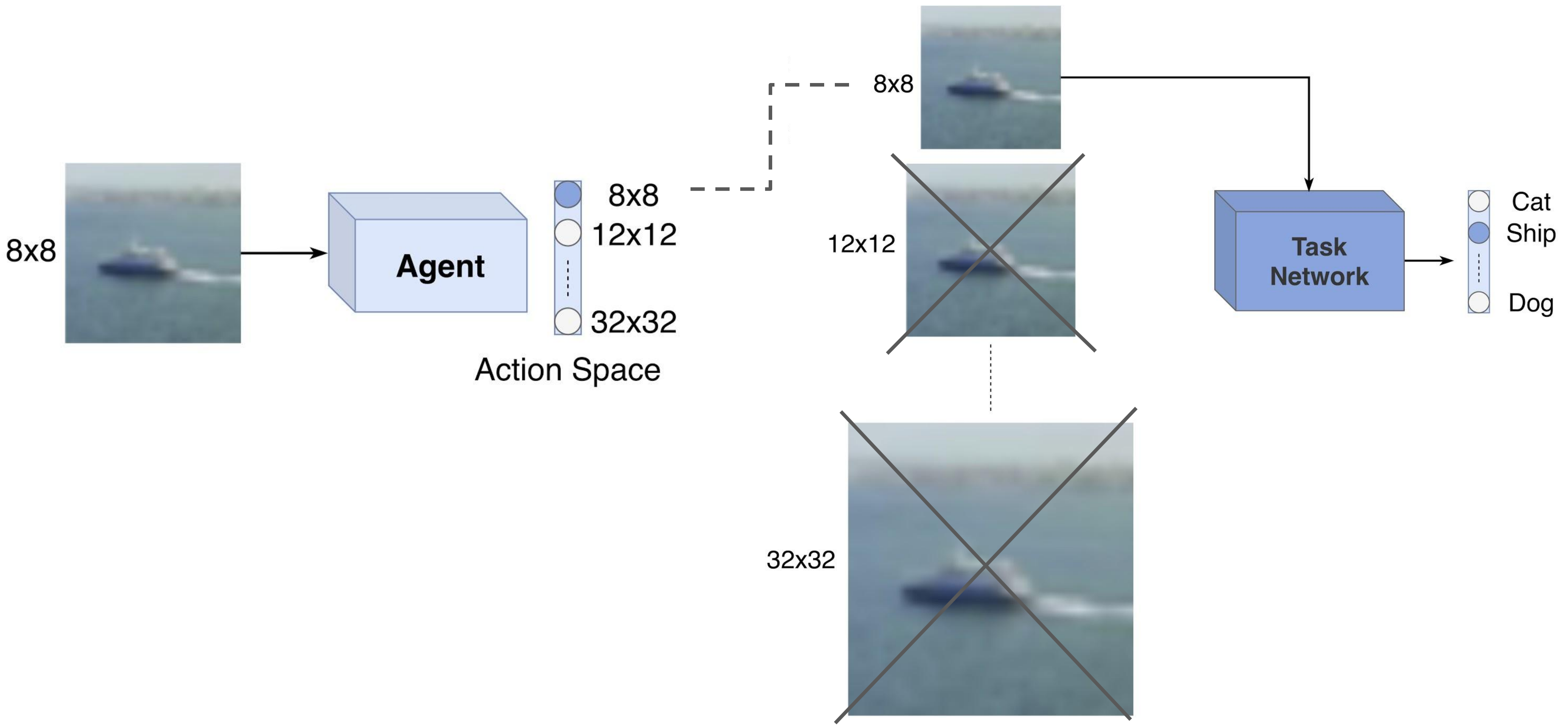
Questions?

Current Projects - Future Frame Prediction

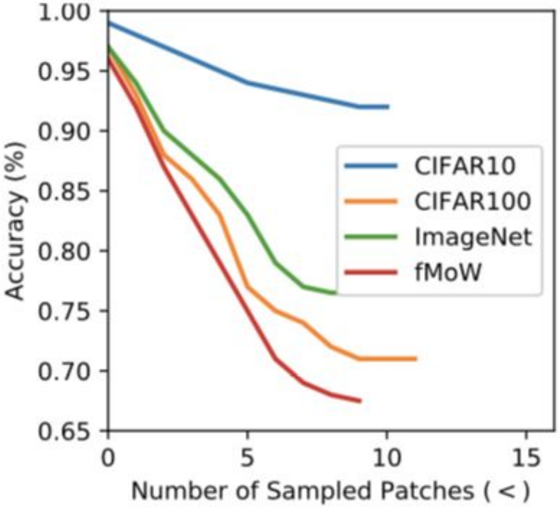


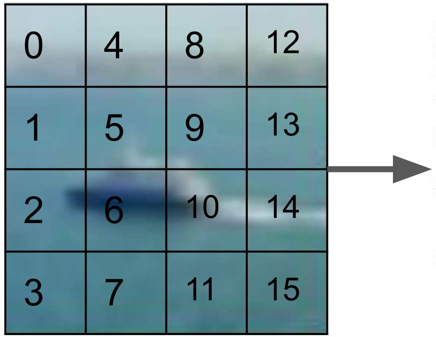
Predicting 29th Frame

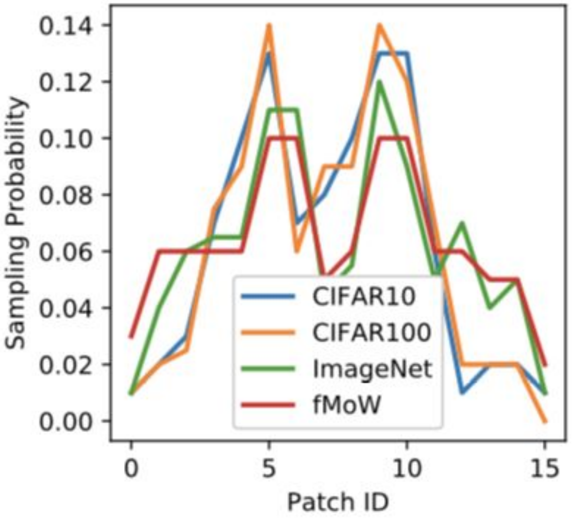
Current Projects - Modality Selection with RL



Analyzing Policy Network’s Actions







***Policy Network samples more patches when there is more ambiguity.**

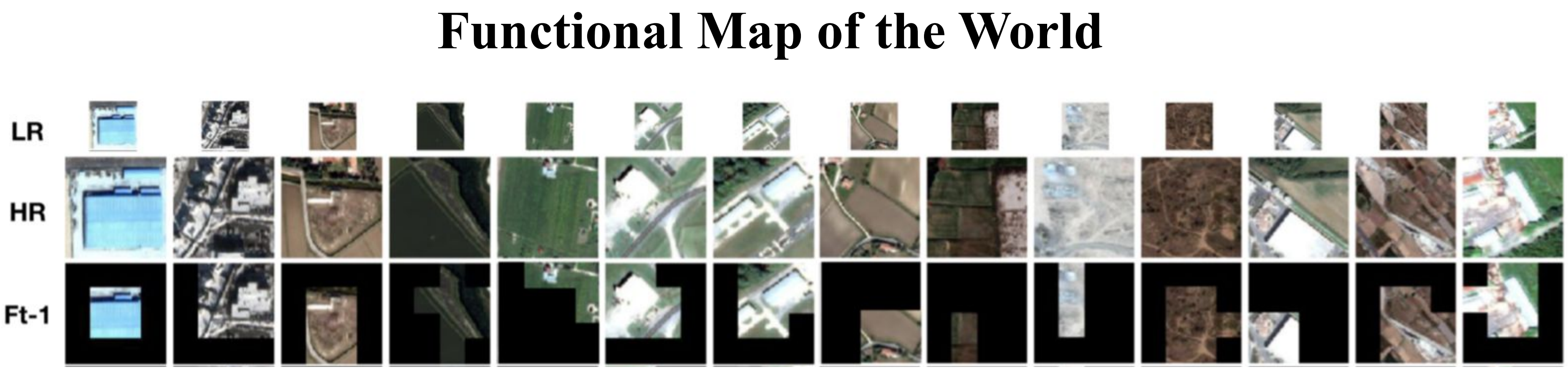
***Policy Network focuses more on the central patches.**

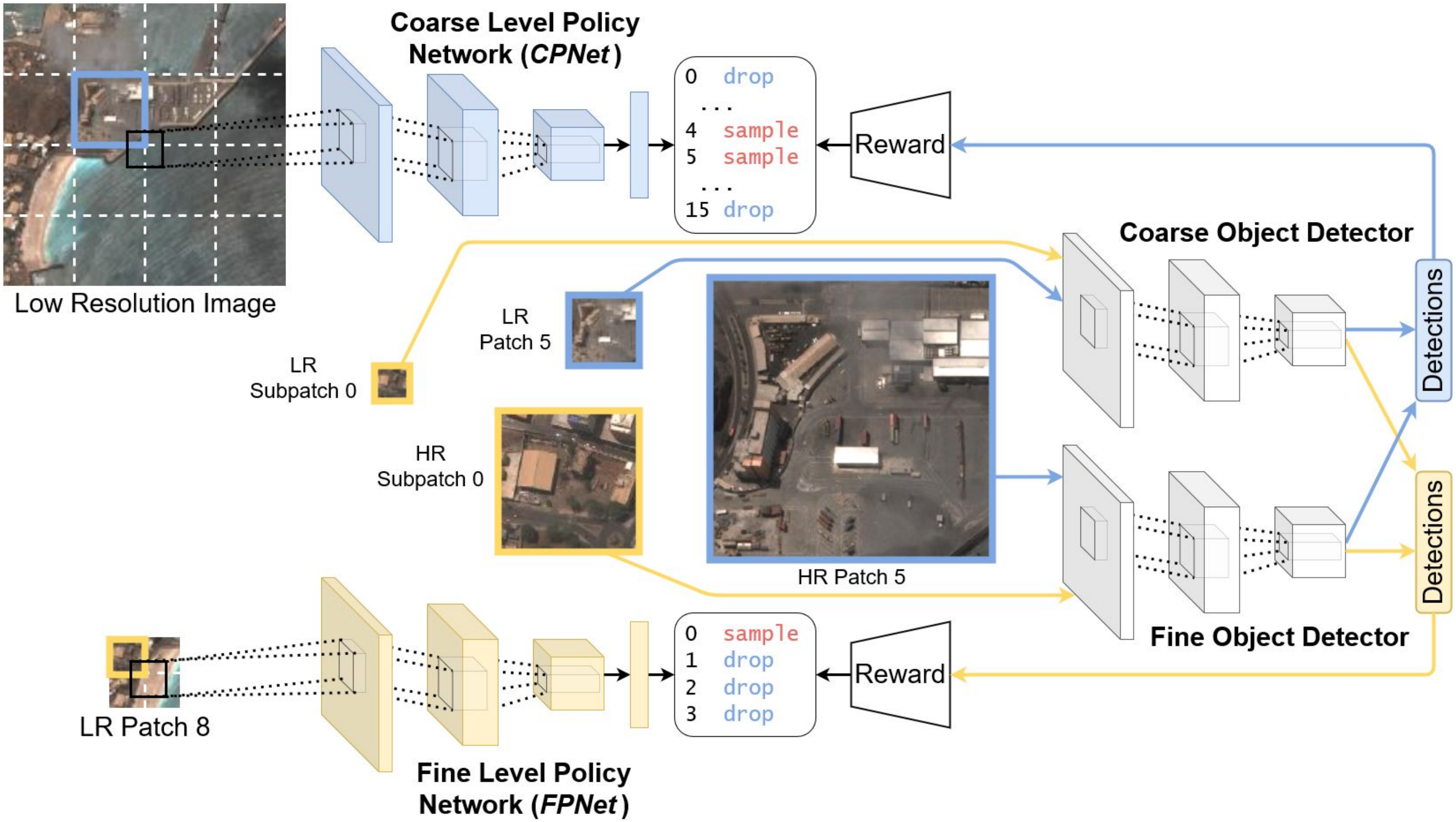
Experiments on fMoW

- We use 350k, 50k, and 50k training, validation and test samples.

	Acc. (%) (Pt)	S	Acc. (%) (Ft-1)	S	Acc. (%) (Ft-2)	S
LR-CNN	61.4	0	61.4	0	61.4	0
SRGAN [19]	62.3	0	62.3	0	62.3	0
KD [37]	63.1	0	63.1	0	63.1	0
PCN [45]	63.5	0	63.5	0	63.5	0
HR-CNN	67.3	16	67.3	16	67.3	16
Fixed-H	47.7	7	63.3	6	64.9	6
Fixed-V	48.3	7	63.2	6	64.7	6
Stochastic	29.1	7	57.1	6	63.6	6
STN [31]	46.5	7	61.8	6	64.8	6
PatchDrop	**53.4**	**7**	**67.1**	**5.9**	**68.3**	**5.2**

Learned Patch Sampling Policies

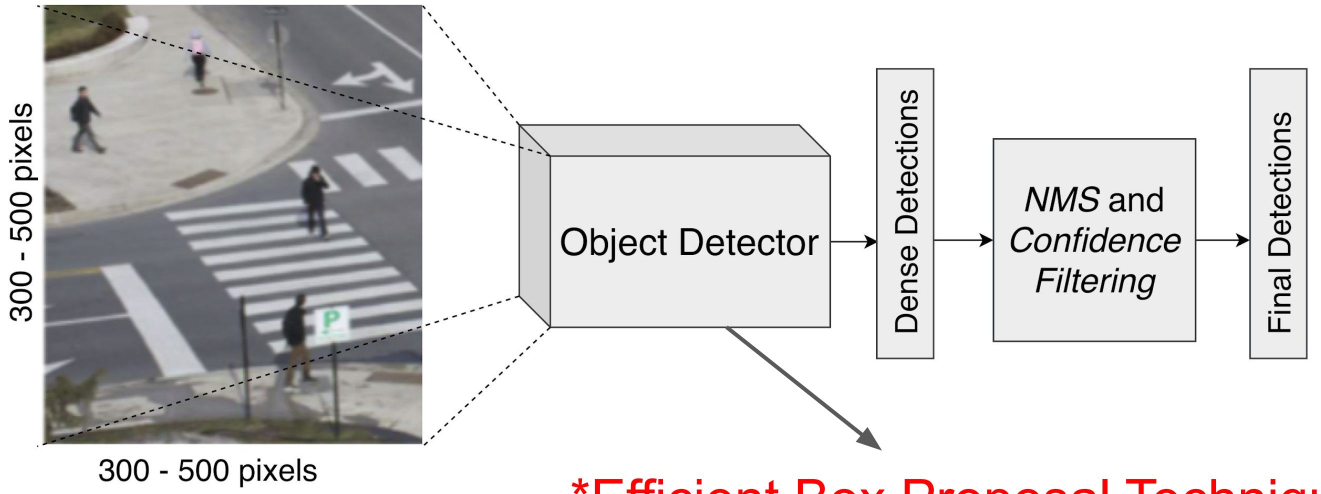




Coarse Level Policy Network (CPNet)
Low Resolution Image
0 drop
...
4 sample
5 sample
...
15 drop
Reward
Coarse Object Detector
Detections
LR Patch 5
LR Subpatch 0
HR Subpatch 0
HR Patch 5
Detections
Fine Object Detector
LR Patch 8
0 sample
1 drop
2 drop
3 drop
Reward
Fine Level Policy Network (FPNet)

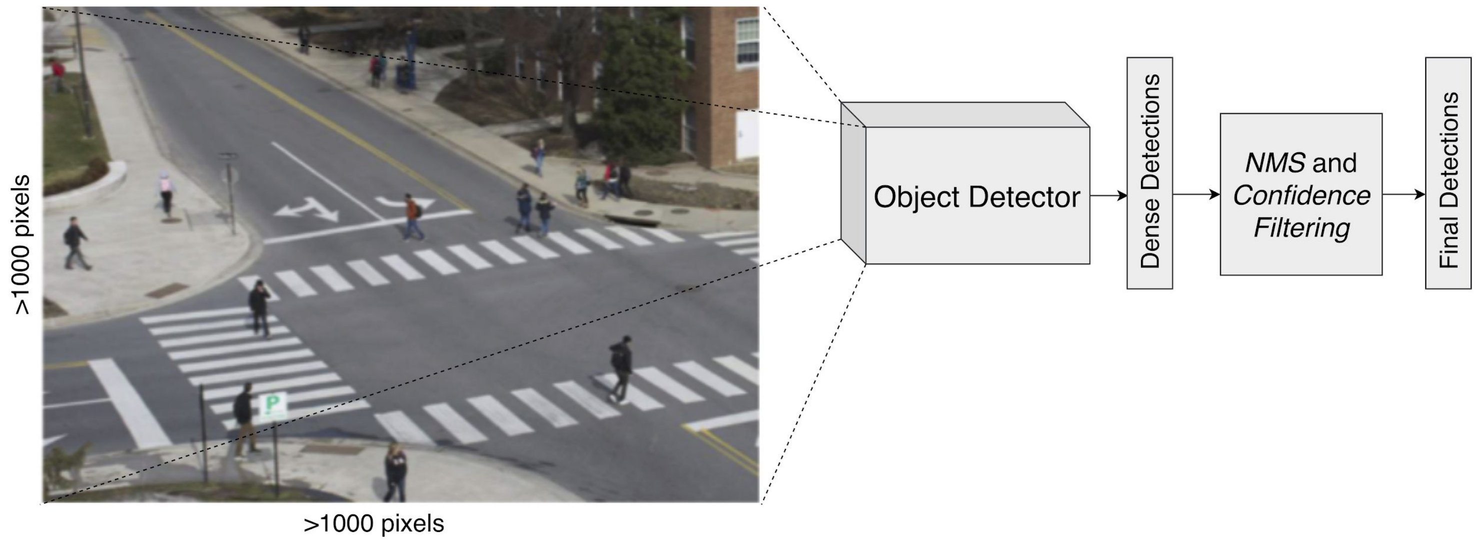
Introduction to Efficient Object Detection

Most of the literature focuses on *efficient box proposal techniques* and *backbone architectures*.



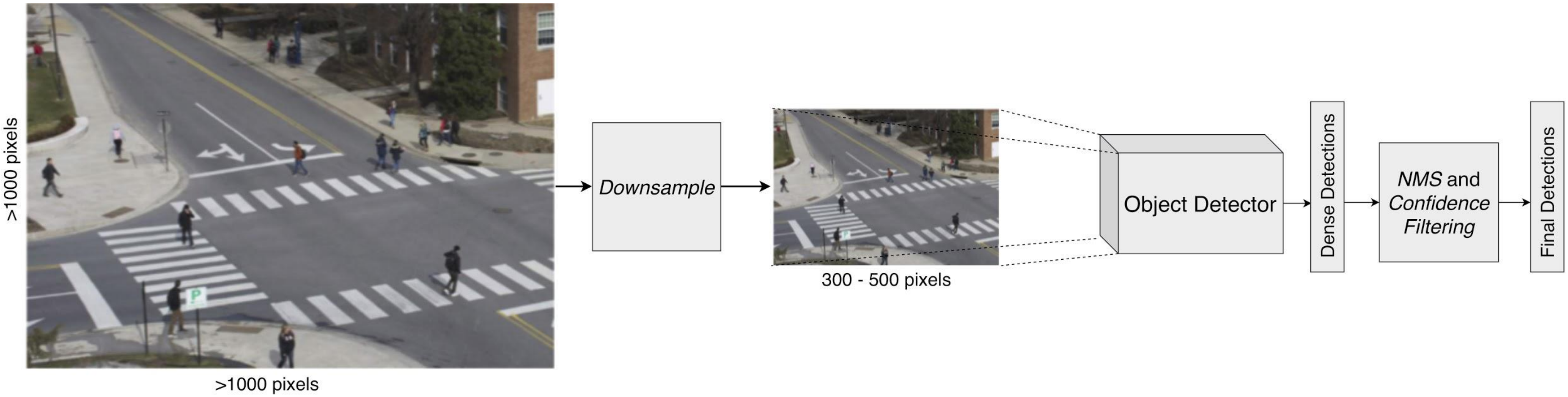
*Efficient Box Proposal Techniques
*Efficient Backbone Architectures

Detection in Large Images - Passing Full Image



Needs large amount of memory to store large size feature maps.

Detection in Large Images - Using LR Image



Downsampling loses spatial information → lower mAP and mAR