Learning When and Where to Zoom with Deep Reinforcement Learning

Burak Uzkent
Department of Computer Science
Stanford University
buzkent@cs.stanford.edu

Stefano Ermon
Department of Computer Science
Stanford University
ermon@cs.stanford.edu

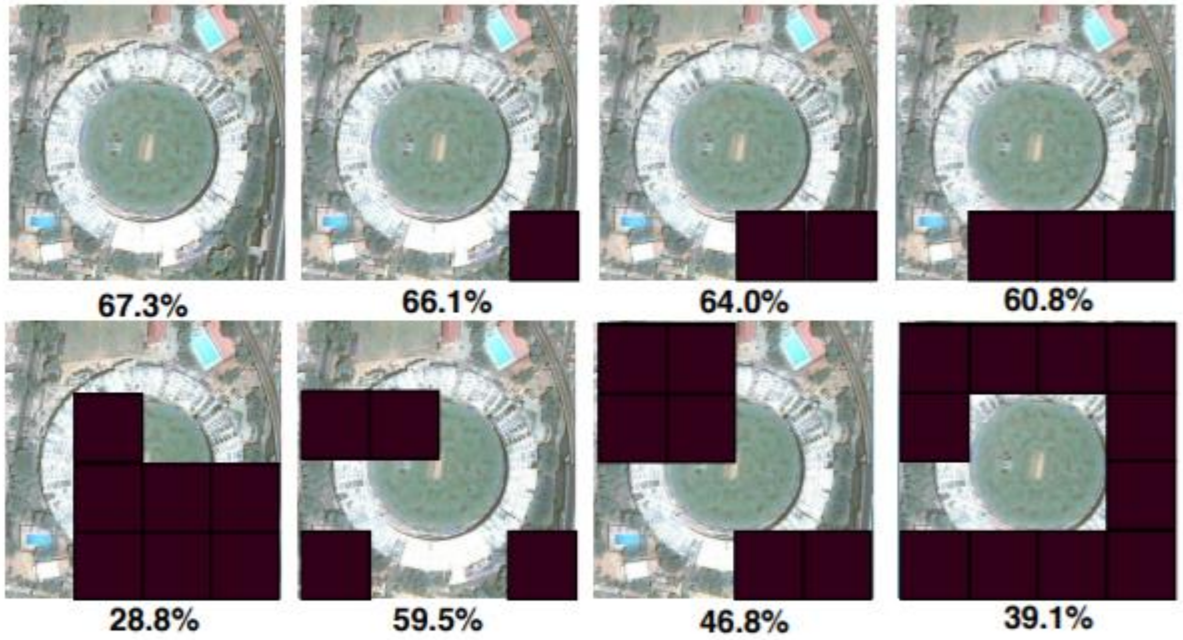
Outline

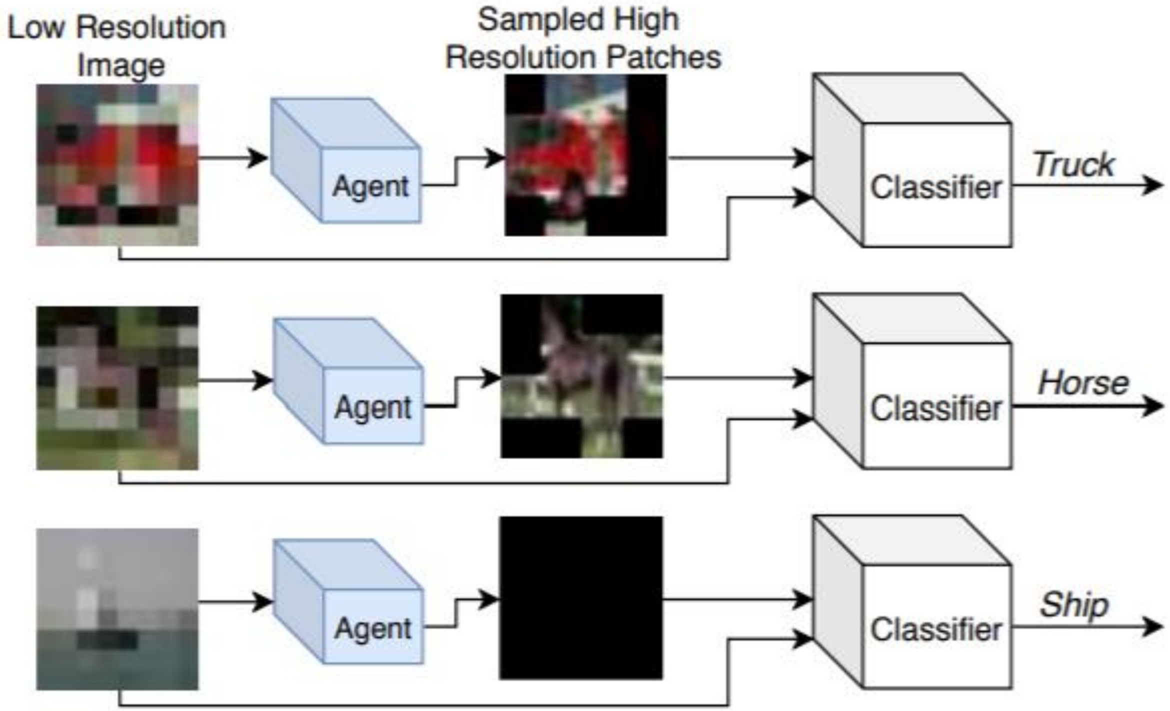
- Challenge
  - ➢ need high quality input data to perform well, and the performance drops significantly on degraded
  - ➢ down sampling is often performed for computational and statistical reasons
  - ➢ an adaptive framework can also benefit application domains where acquiring high resolution data is particularly expensive.
  - ➢ how to perform this selection automatically and efficiently

Outline

- Contribution
  - we show that we can use only about 40% of full HR images without any significant loss of accuracy——save in the order of 100,000 dollars.
  - PatchDrop performs well on traditional computer vision benchmarks.
  - generate hard positive training examples to boost the accuracy of CNNs on ImageNet and fMoW by 2-3%.
- Problem statement and Proposed solution
- Experiments

Background and Motivation





Problem statement

not observed by the agent

$$x_h = (x_h^1, x_h^2, \cdots, x_h^P) \longleftrightarrow x_l$$

$$y \in \{1, \cdots, N\}$$

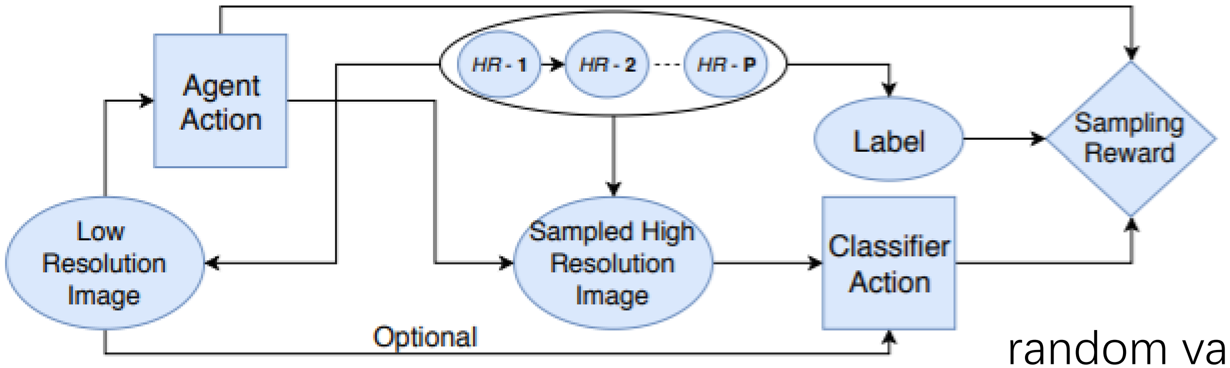
$$\mathbf{a_1} \in \{0,1\}^P, \longrightarrow \mathbf{a_1^P} = 1 \longrightarrow x_h^p.$$

$$\pi_1(\mathbf{a_1}|x_l;\theta_p) = p(\mathbf{a_1}|x_l;\theta_p),$$

masked HR image

$$x_h^m = x_h \bigodot \mathbf{a_1}$$

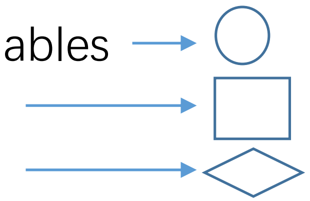
two step episodic Markov Decision Process



random variables

actions

utilities



The first step of the MDP

$$p(x_h, x_h^m, x_l, y, \mathbf{a_1}) = p(x_h)p(y|x_h)p(x_l|x_h) \cdot p(\mathbf{a_1}|x_l;\theta_p)p(x_h^m|\mathbf{a_1}, x_h).$$

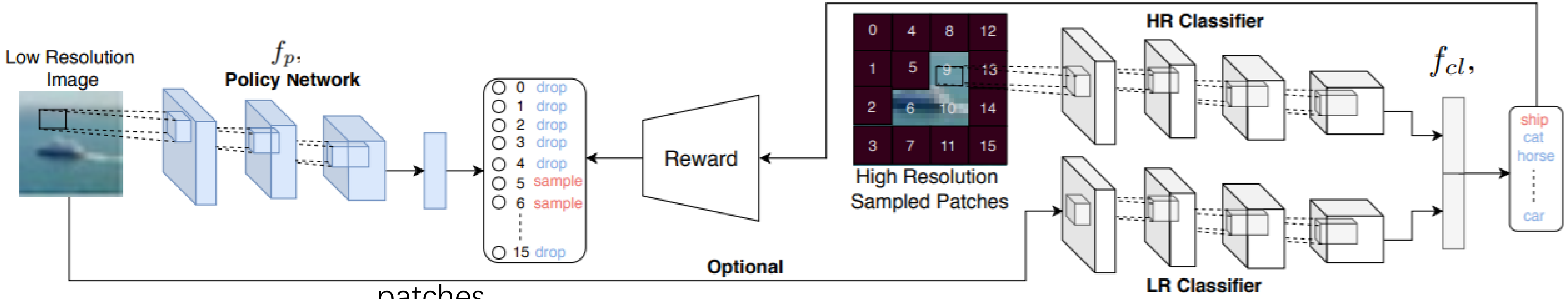
The second step of the MDP $\mathbf{a_2} \in \{1, \cdots, N\}.$

$$\pi_2(\mathbf{a_2}|x_h^m, x_l;\theta_{cl}) = p(\mathbf{a_2}|x_h^m, x_l;\theta_{cl}),$$

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

maximizing the expected utility

Proposed Solution



patches

$\mathbf{a_1}$

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

prediction vector

$$s_p = f_p(x_l;\theta_p).$$

$, s_p \in [0,1],$ sigmoid function

$\mathbf{a_2}$

$$s_{cl} = (S/P)s_{cl}^{h^m} + (1-S/P)s_{cl}^{l},$$

number of sampled patches

Training

$\theta_p \;\; \theta_{cl}$

REINFORCE method

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

$R(\mathbf{a_1}, \mathbf{a_2}, y)$ advantage function

reduce the variance

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

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

Training

Input: Input($\mathcal{X}_l, \mathcal{Y}, \mathcal{C}$) $\quad \mathcal{X}_l = \{x_l^1, x_l^2, ..., x_l^N\}$
for $k \leftarrow 0$ *to* K_1 **do**
  $s_p \leftarrow f_p(x_l; \theta_p)$
  $s_p \leftarrow \alpha + (1 - s_p)(1 - \alpha)$
  $\mathbf{a_1} \sim \pi_1(a_1 | s_p)$
  $x_h^m = x_h \bigodot \mathbf{a_1}$
  $\mathbf{a_2} \leftarrow f_{cl}^h(x_h^m; \theta_{cl}^h)$
  Evaluate Reward $R(\mathbf{a_1}, \mathbf{a_2}, y)$
  $\theta_p \leftarrow \theta_p + \nabla\theta_p$
end
for $k \leftarrow 0$ *to* K_2 **do**
  Jointly Finetune θ_{cl}^h and θ_p using f_{cl}^h
end
for $k \leftarrow 0$ *to* K_3 **do**
  Jointly Finetune θ_{cl}^h and θ_p using f_{cl}^h and f_{cl}^l
end

Pre-training the Classifier:

Pre-training the Policy Network:

Finetuning the Agent and HR Classifier（Ft-1）：

Finetuning the Agent and HR Classifier（Ft-2）：

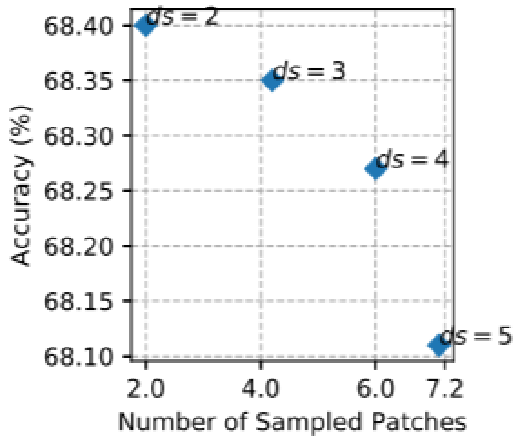
Experiment

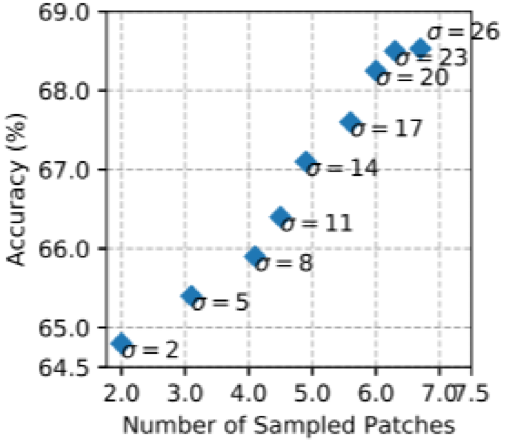
Figure 4: Policies learned on the fMoW dataset. In columns 5 and 8, Ft-2 model does not sample any HR patches and the LR classifier is used. Ft-1 model samples more patches as it does not utilize LR classifier.

Experiment

average number of sampled patches

	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 [40]	63.1	0	63.1	0	63.1	0
PCN [55]	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**





Reward Function

a large value

$y = \hat{y}(\mathbf{a_2})$ and $-\sigma$ otherwise as a reward.

$$R = 1 - \left(\frac{|\mathbf{a_1}|_1}{P}\right)^2$$

True class

predicted class

sample more patches to preserve accuracy

Experiment

	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 [40]	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 [40]	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**

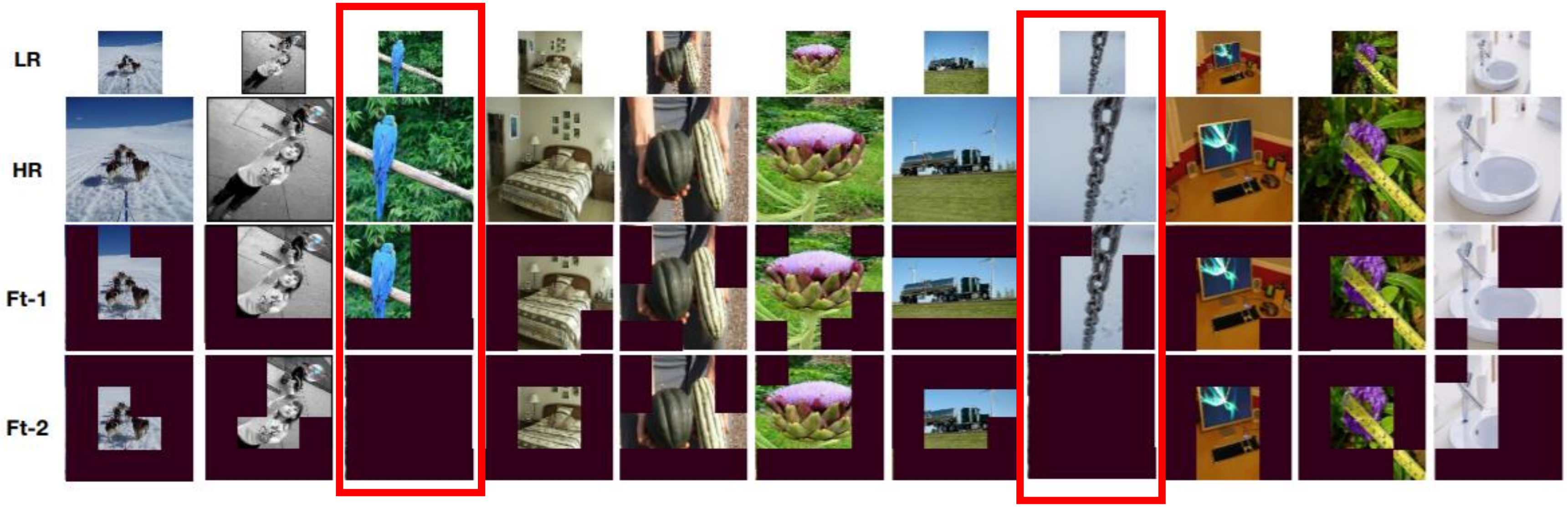
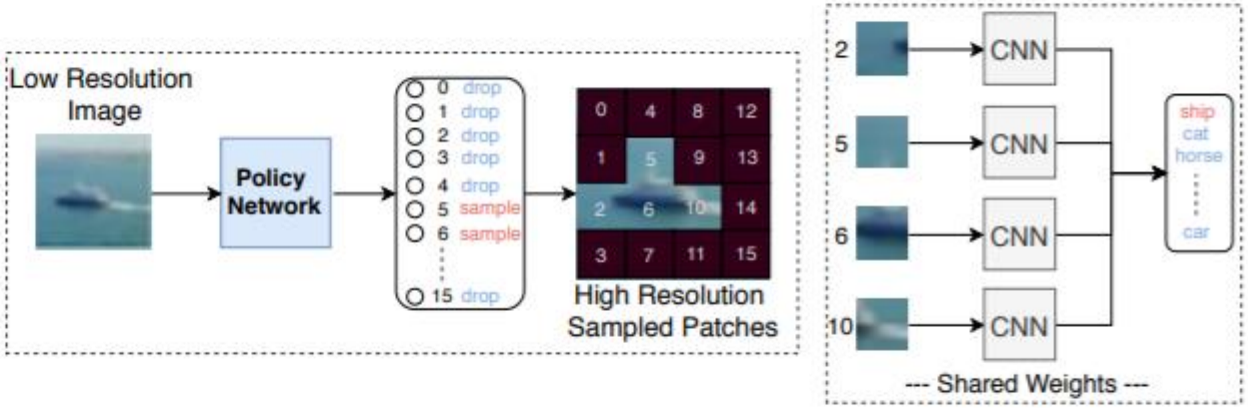


Figure 6: Policies learned on ImageNet. In columns 3 and 8, Ft-2 model does not sample any HR patches and the LR classifier is used. Ft-1 model samples more patches as it does not use the LR classifier.

Experiment decrease the run-time complexity of local CNNs(BagNet)



	Acc. (%) (Pt)	S	Acc. (%) (Ft-1)	S	Run-time. (%) (ms)
BagNet (No Patch Drop) [1]	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 [19]	67.5	10	86.8	9	112
BagNet (PatchDrop)	**77.4**	**9.5**	**92.7**	**8.5**	98

	CIFAR10 (%) (ResNet32)	CIFAR100 (%) (ResNet32)	ImageNet (%) (ResNet50)	fMoW (%) (ResNet34)
No Augment.	92.3	69.3	76.5	67.3
CutOut [5]	93.5	70.4	76.5	67.6
PatchDrop	**93.9**	**71.0**	**78.1**	**69.6**