Searching for Efficient Multi-Scale Architectures for Dense Image Prediction

Authors: Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou,
Barret Zoph, Florian Schroff, Hartwig Adam, Jonathon Shlens

Presented by **Paras Jain**
AISys 2019

Background

Paper overview

Search space

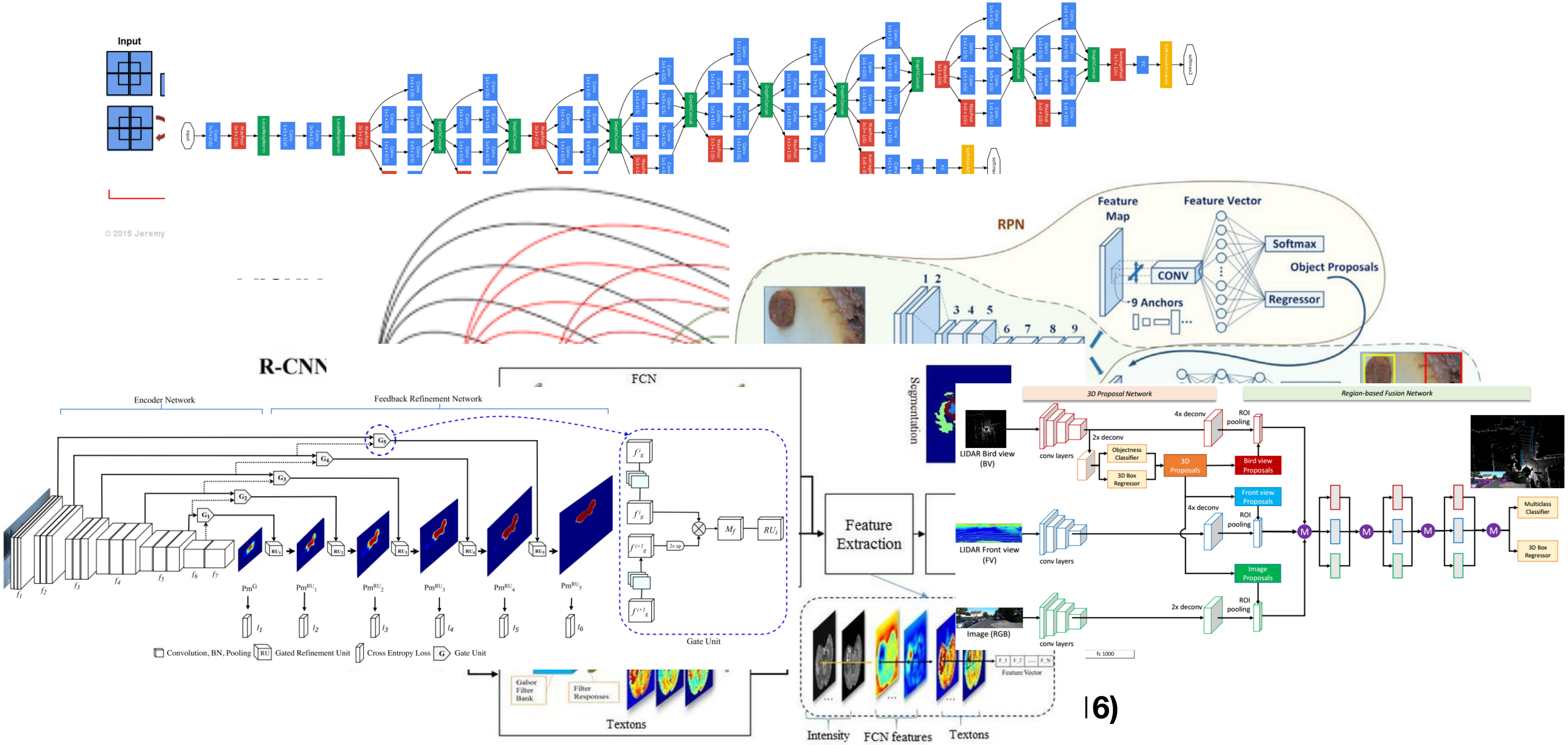
Sampling strategy

Performance estimation

Results

DNNs... now ubiquitous!

But DNN design is getting more complex



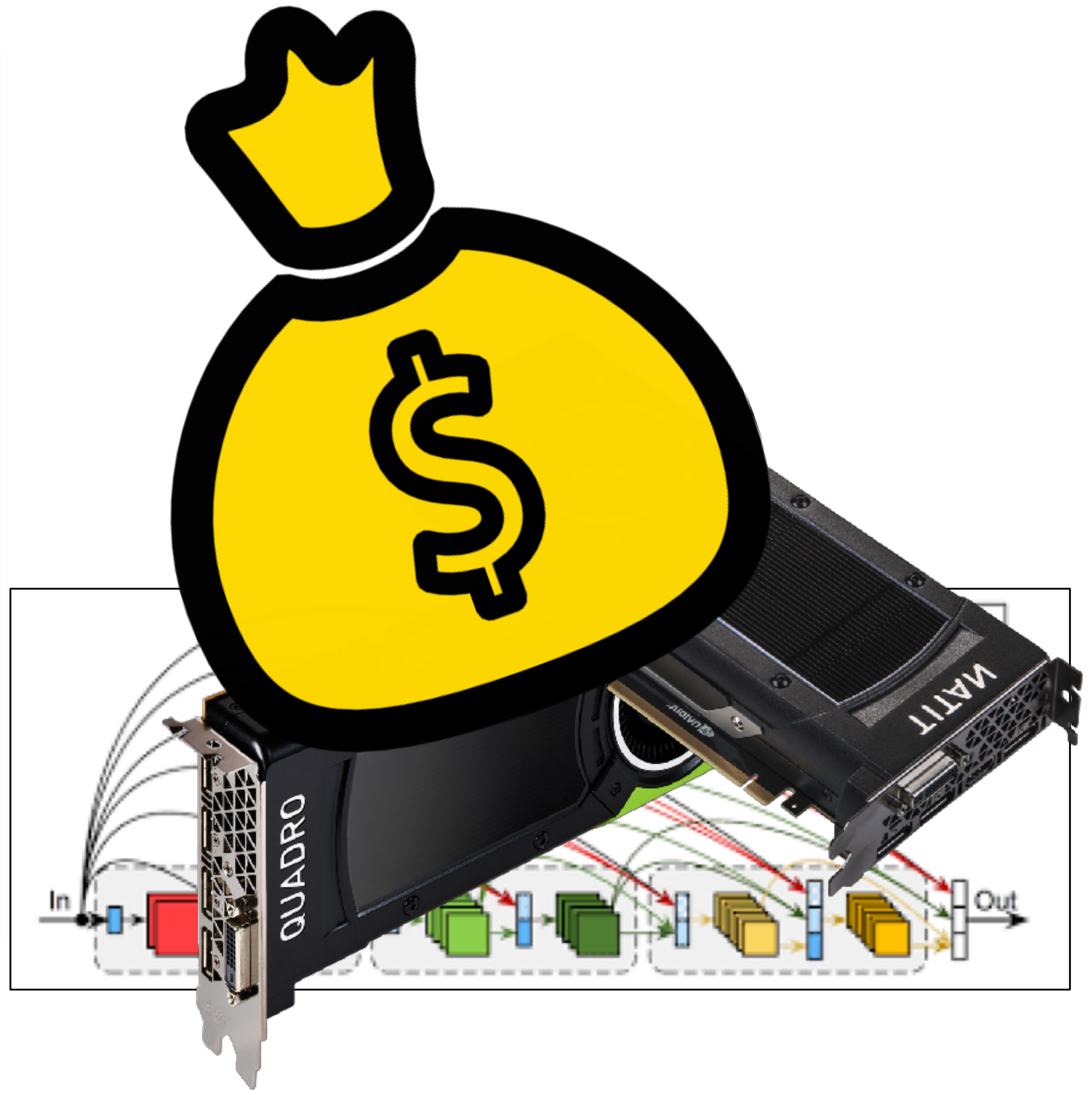
# of applications >> # of AI experts

Growing design space of DNNs

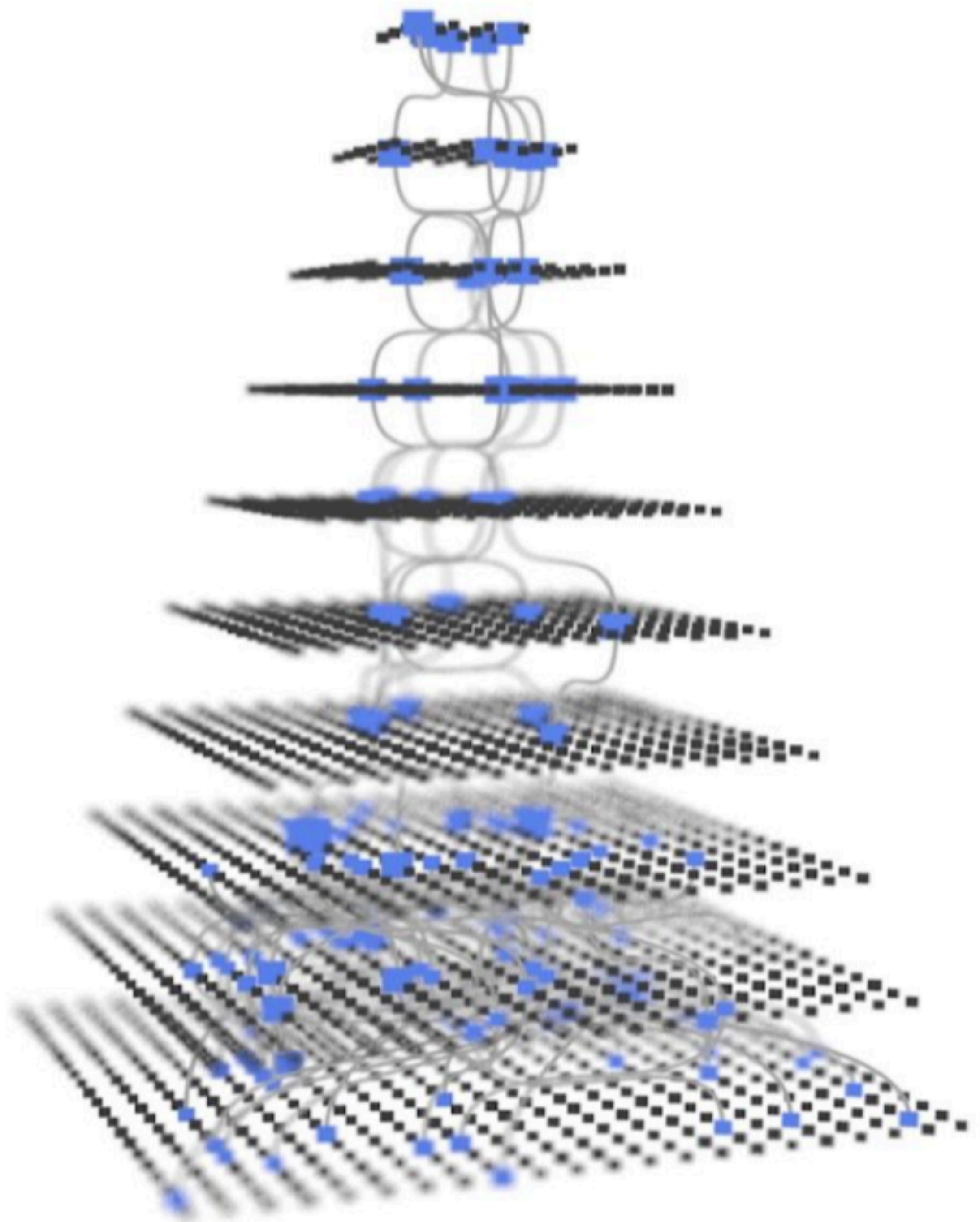
Falling price per FLOP

What is the **Design Automation stack for DNNs?**

AutoML tries to automatically generate high-accuracy models (*subject to constraints*)

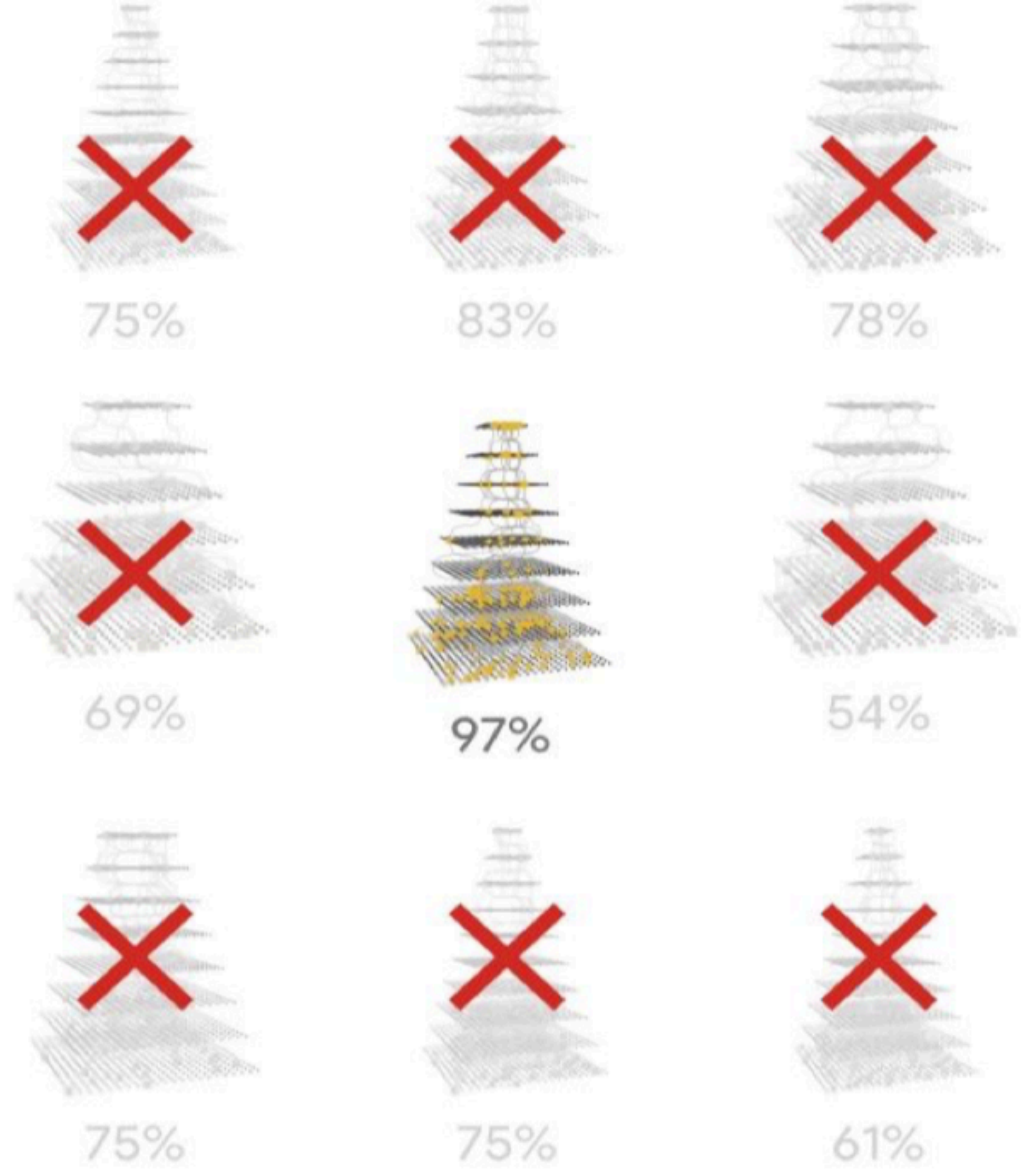


Controller: proposes ML models



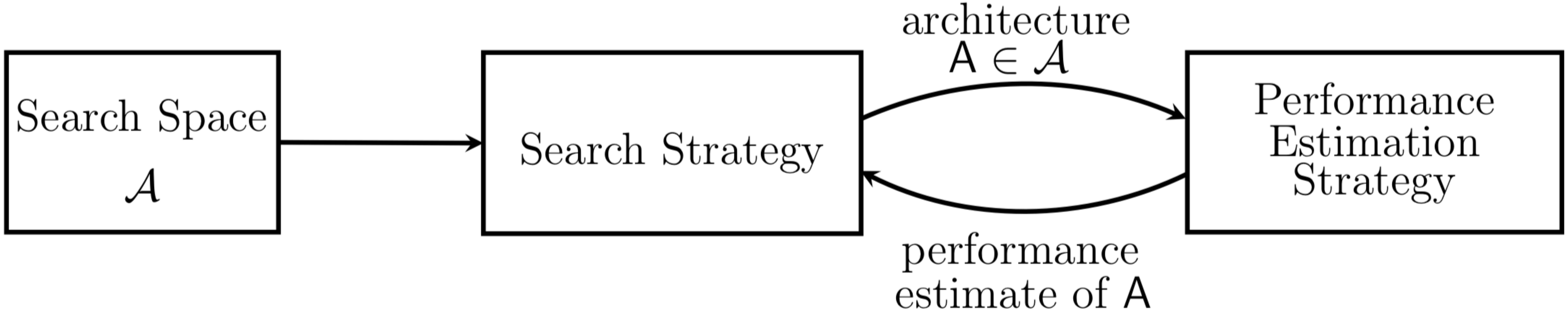
Iterate to find the most accurate model

Train & evaluate models



Blueprint image: https://arxiv.org/pdf/1808.05377.pdf
Loop image courtesy Barret Zoph, Quoc Le

Blueprint for an AutoML paper

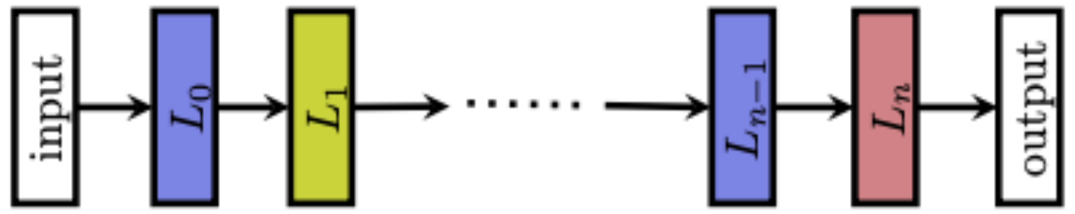


Blueprint image: https://arxiv.org/pdf/1808.05377.pdf
Loop image courtesy Barret Zoph, Quoc Le

DESIGNING NEURAL NETWORK ARCHITECTURES USING REINFORCEMENT LEARNING

Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar
Media Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
{bowen, otkrist, naik, raskar}@mit.edu

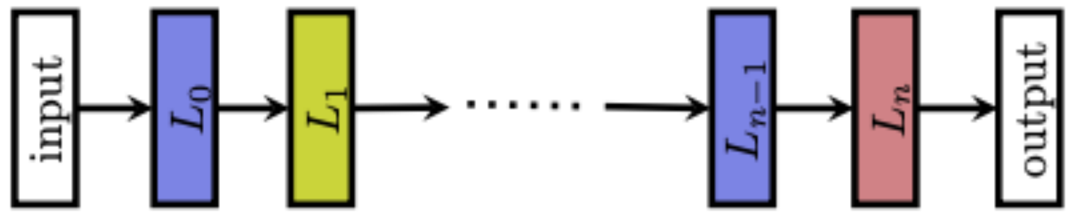
Learning straight-line DNNs (simple data)



Designing Neural Network Architectures using Reinforcement Learning

Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar
Media Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
{bowen, otkrist, naik, raskar}@mit.edu

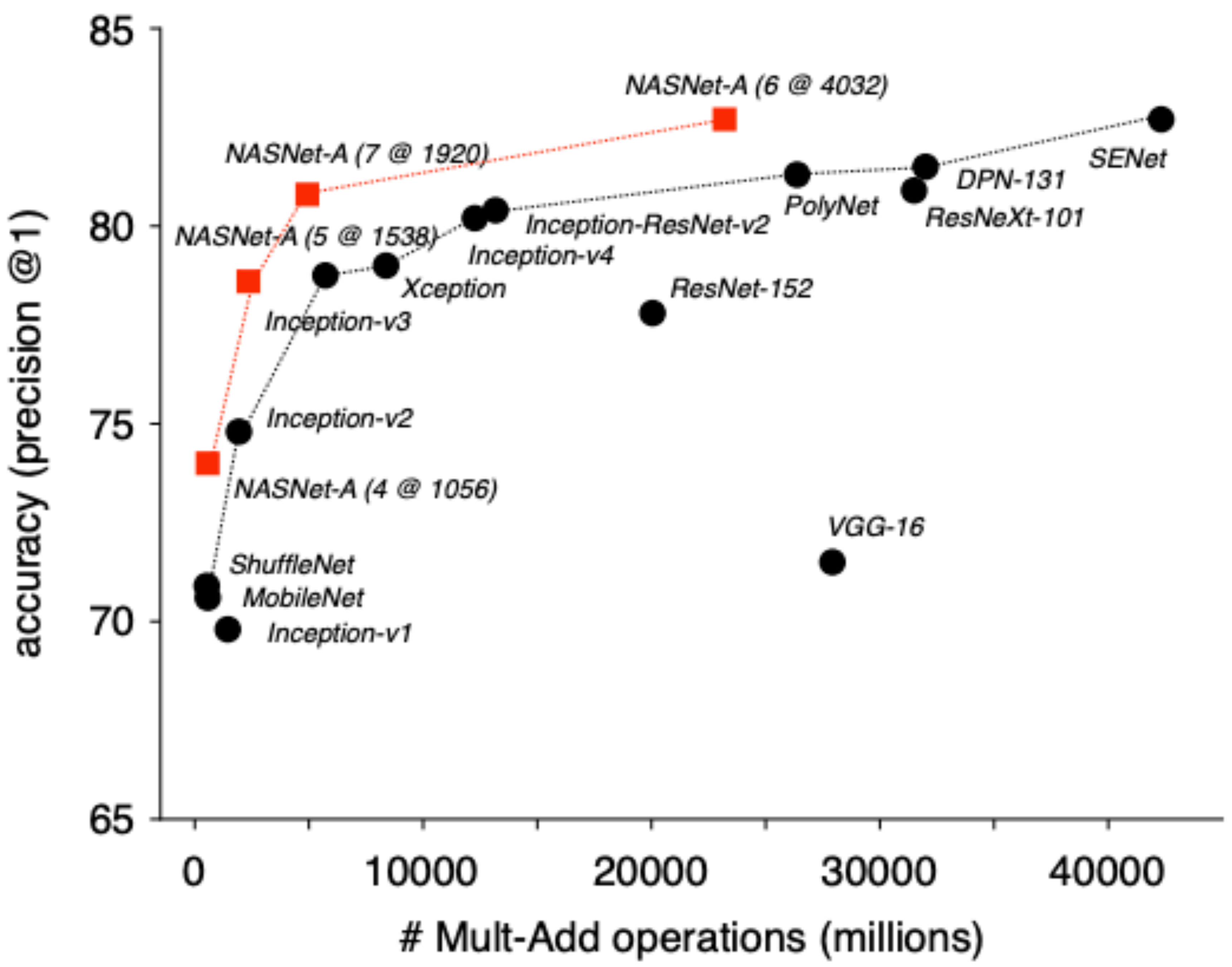
Learning straight-line DNNs (simple data)



Learning Transferable Architectures for Scalable Image Recognition

Barret Zoph, Google Brain, barretzoph@google.com
Vijay Vasudevan, Google Brain, vrv@google.com
Jonathon Shlens, Google Brain, shlens@google.com
Quoc V. Le, Google Brain, qvl@google.com

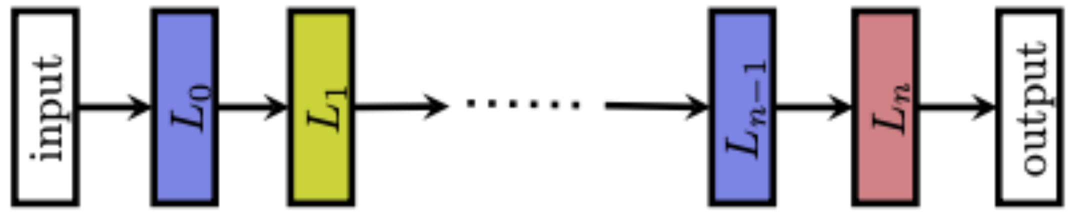
NASNet exceeded human performance on CIFAR and COCO



DESIGNING NEURAL NETWORK ARCHITECTURES USING REINFORCEMENT LEARNING

Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar
Media Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
{bowen, otkrist, naik, raskar}@mit.edu

Learning straight-line DNNs (simple data)



Learning Transferable Architectures for Scalable Image Recognition

Barret Zoph
Google Brain
barretzoph@google.com

Vijay Vasudevan
Google Brain
vrv@google.com

Jonathon Shlens
Google Brain
shlens@google.com

Quoc V. Le
Google Brain
qvl@google.com

NASNet exceeded human performance on CIFAR and COCO (classification, object detection)

MnasNet: Platform-Aware Neural Architecture Search for Mobile

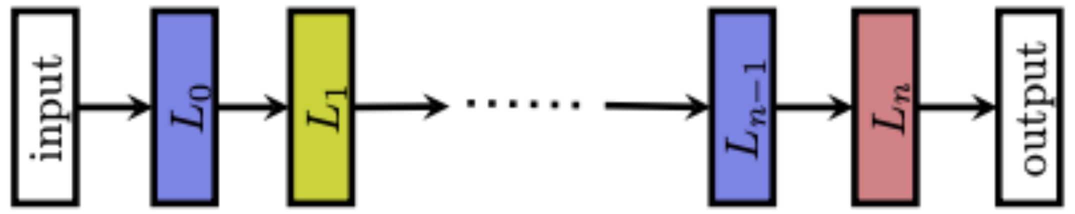
Mingxing Tan[1], Bo Chen[2], Ruoming Pang[1], Vijay Vasudevan[1], Quoc V. Le[1]
[1]Google Brain, [2]Google Inc.
{tanmingxing, bochen, rpang, vrv, qvl}@google.com

Constrained optimization objective for mobile inference latency

DESIGNING NEURAL NETWORK ARCHITECTURES USING REINFORCEMENT LEARNING

Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar
Media Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
{bowen, otkrist, naik, raskar}@mit.edu

Learning straight-line DNNs (simple data)



Learning Transferable Architectures for Scalable Image Recognition

Barret Zoph, Google Brain, barretzoph@google.com
Vijay Vasudevan, Google Brain, vrv@google.com
Jonathon Shlens, Google Brain, shlens@google.com
Quoc V. Le, Google Brain, qvl@google.com

NASNet exceeded human performance on CIFAR and COCO (classification, object detection)

MnasNet: Platform-Aware Neural Architecture Search for Mobile

Mingxing Tan[1], Bo Chen[2], Ruoming Pang[1], Vijay Vasudevan[1], Quoc V. Le[1]
[1]Google Brain, [2]Google Inc.
{tanmingxing, bochen, rpang, vrv, qvl}@google.com

Constrained optimization objective for mobile inference latency

---

DARTS: Differentiable Architecture Search

Hanxiao Liu, CMU, hanxiaol@cs.cmu.edu
Karen Simonyan, DeepMind, simonyan@google.com
Yiming Yang, CMU, yiming@cs.cmu.edu

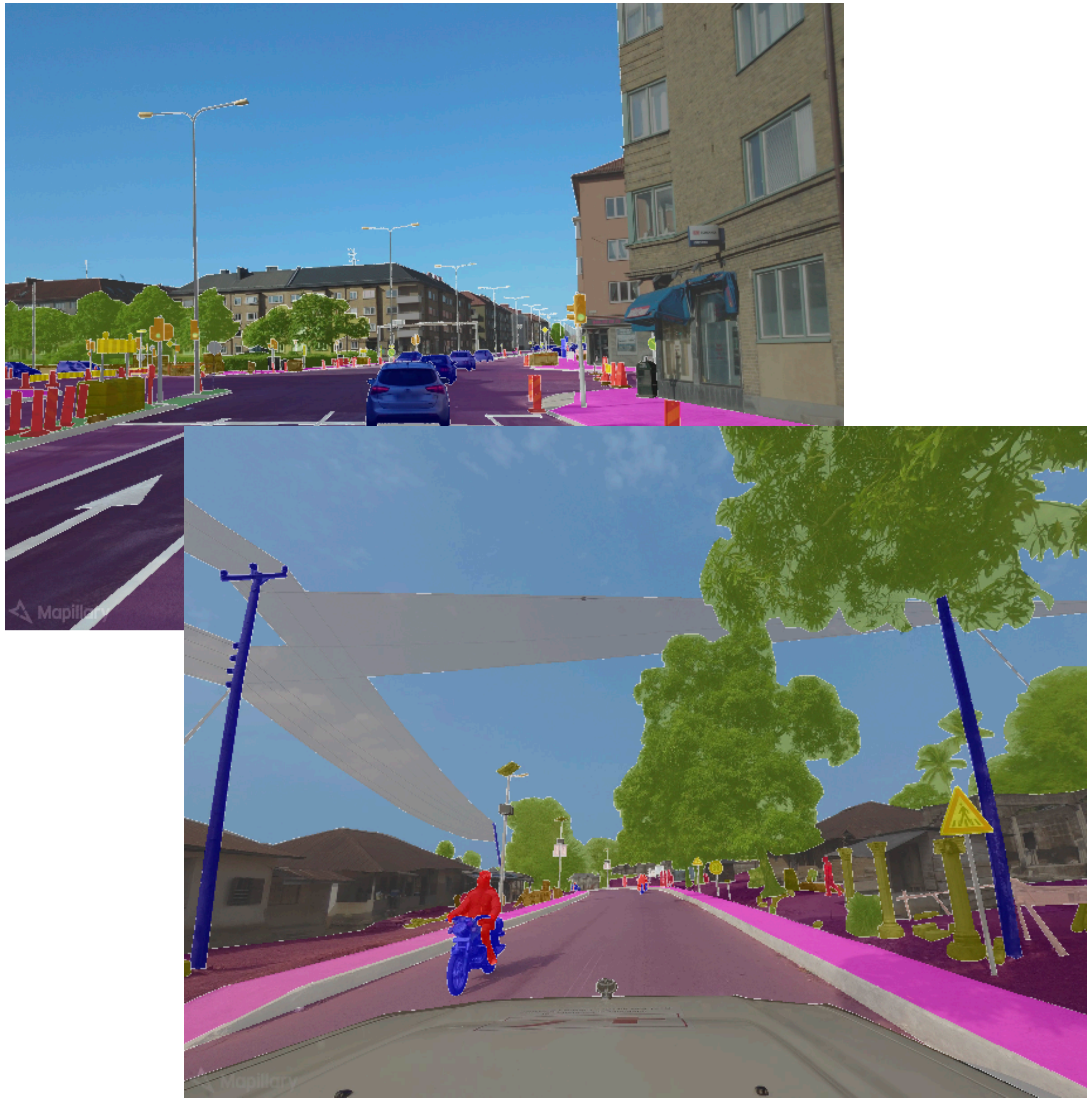
Low-cost architecture search via backprop into architecture

Background
Paper overview
Search space
Sampling strategy
Performance estimation
Results

Motivation

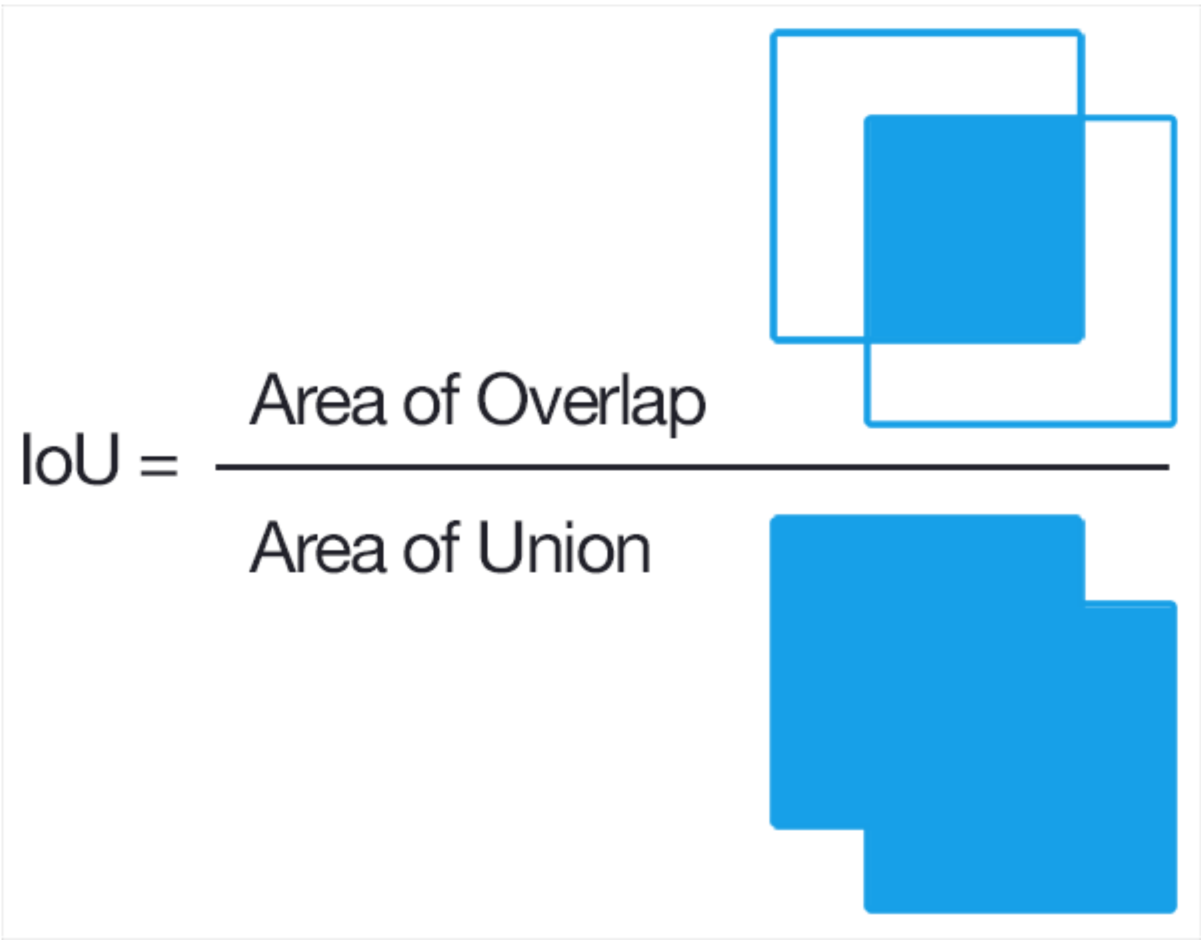
- AutoML has exceeded human performance on classification
- Can we apply search to a new vision task (semantic segmentation)?

Semantic Segmentation task



Images: Mapillary Vistas

- **What is segmentation?** Label each pixel of an image with an class
- **Key application:** Autonomous driving, cancer detection, deforestation detection
- **Metric:** Intersection-over-Union aka Jaccard index



Searching for Efficient Multi-Scale Architectures for Dense Image Prediction

Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou
Barret Zoph, Florian Schroff, Hartwig Adam, Jonathon Shlens
Google Inc.

Abstract

The design of neural network architectures is an important component for achieving state-of-the-art performance with machine learning systems across a broad array of tasks. Much work has endeavored to design and build architectures automatically through clever construction of a search space paired with simple learning algorithms. Recent progress has demonstrated that such meta-learning methods may exceed scalable human-invented architectures on image classification tasks. An open question is the degree to which such methods may generalize to new domains. In this work we explore the construction of meta-learning techniques for dense image prediction focused on the tasks of scene parsing, person-part segmentation, and semantic image segmentation. Constructing viable search spaces in this domain is challenging because of the multi-scale representation of visual information and the necessity to operate on high resolution imagery. Based on a survey of techniques in dense image prediction, we construct a recursive search space and demonstrate that even with efficient random search, we can identify architectures that outperform human-invented architectures and achieve state-of-the-art performance on three dense prediction tasks including 82.7% on Cityscapes (street scene parsing), 71.3% on PASCAL-Person-Part (person-part segmentation), and 87.9% on PASCAL VOC 2012 (semantic image segmentation). Additionally, the resulting architecture is more computationally efficient, requiring half the parameters and half the computational cost as previous state of the art systems.

1 Introduction

The resurgence of neural networks in machine learning has shifted the emphasis for building state-of-the-art systems in such tasks as image recognition [44, 84, 83, 34], speech recognition [36, 8], and machine translation [88, 82] towards the design of neural network architectures. Recent work has demonstrated successes in automatically designing network architectures, largely focused on single-label image classification tasks [100, 101, 52] (but see [100, 65] for language tasks). Importantly, in just the last year such meta-learning techniques have identified architectures that exceed the performance of human-invented architectures for large-scale image classification problems [101, 52, 68].

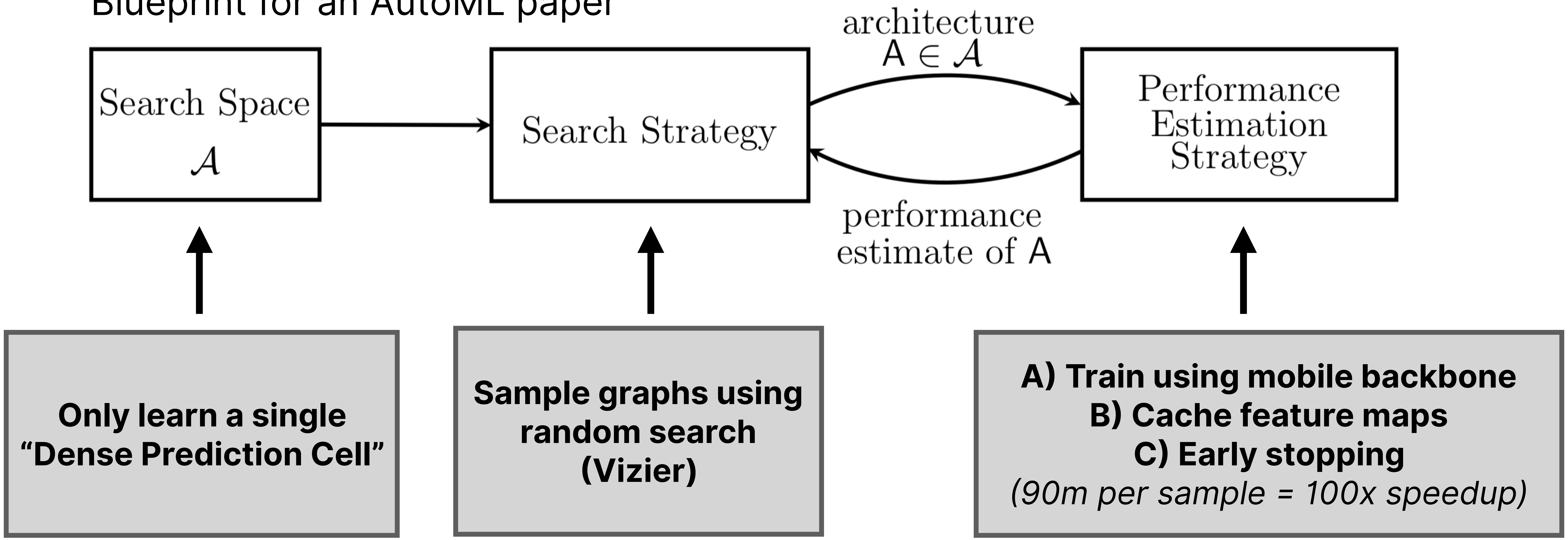
Image classification has provided a great starting point because much research effort has identified successful network motifs and operators that may be employed to construct search spaces for architectures [52, 68, 101]. Additionally, image classification is inherently multi-resolution whereby fully convolutional architectures [77, 58] may be trained on low resolution images (with minimal computational demand) and be transferred to high resolution images [101].

Although these results suggest opportunity, the real promise depends on the degree to which meta-learning may extend into domains beyond image classification. In particular, in the image domain, many important tasks such as semantic image segmentation [58, 11, 97], object detection [71, 21], and instance segmentation [20, 33, 9] rely on high resolution image inputs and multi-scale image

"Cheap AutoML" = 370 GPUs over one week

- Current state of the art in semantic segmentation
- Results generalize to scene parsing (above) and person-part matching
- Used AutoML to search space of 10^{11} models, sampled 28000 models

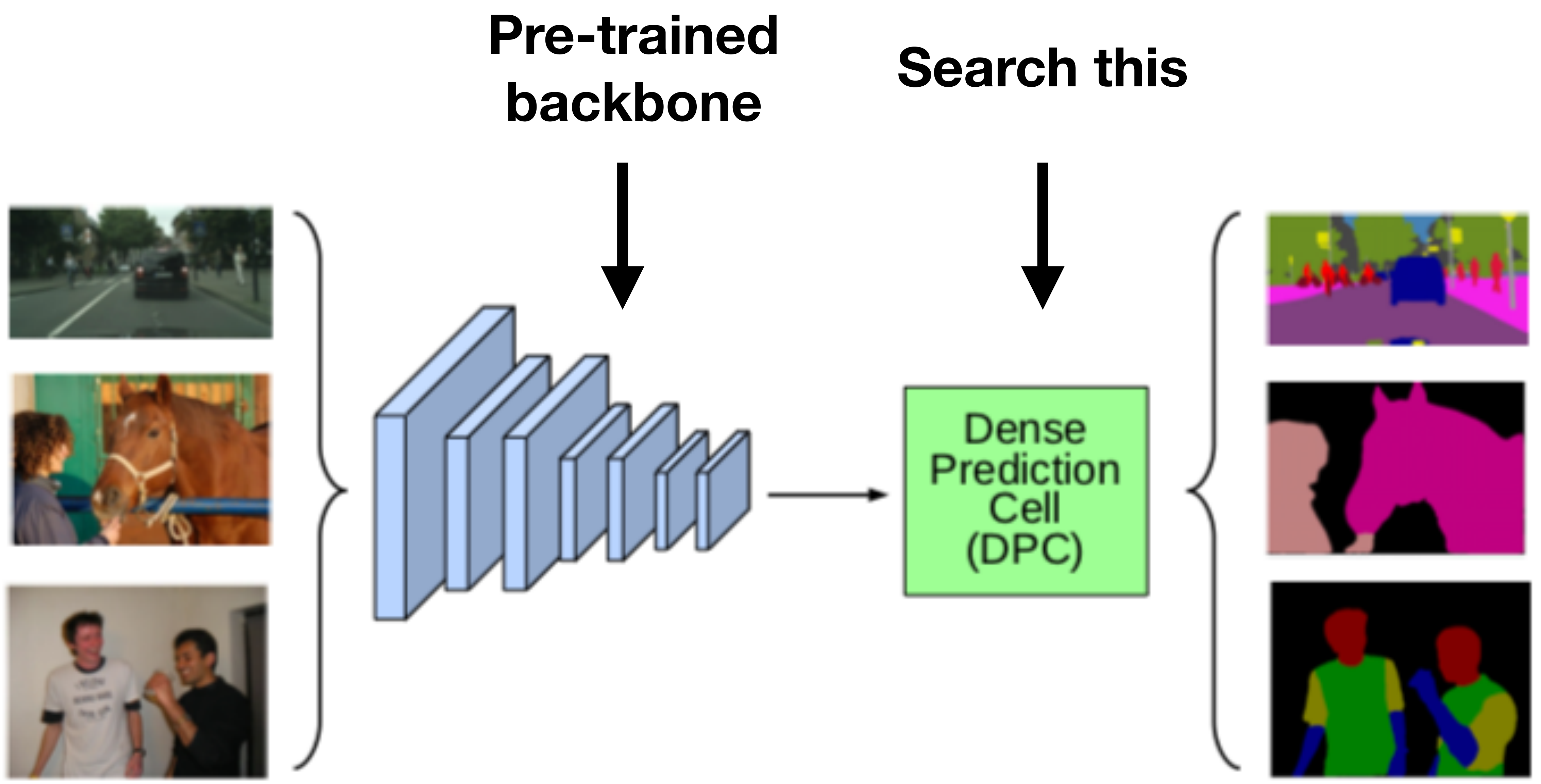
Blueprint for an AutoML paper

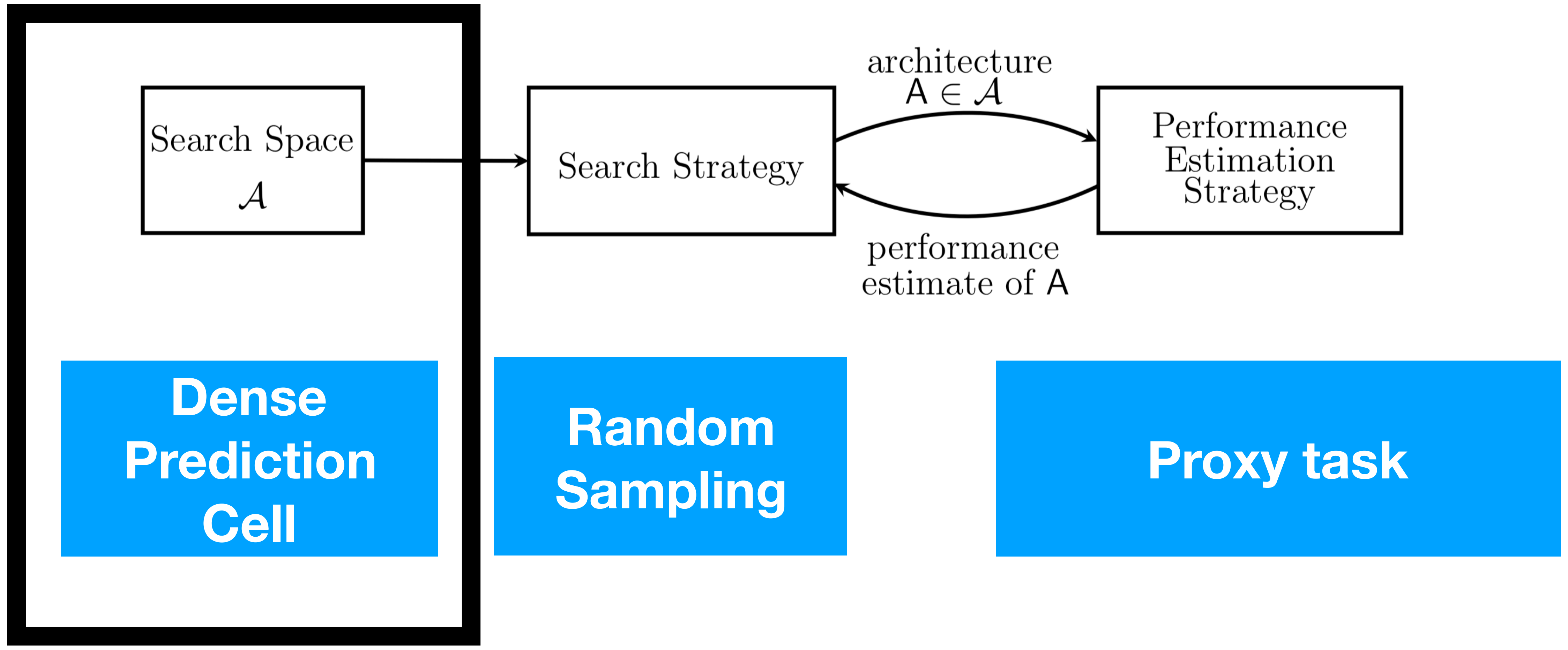


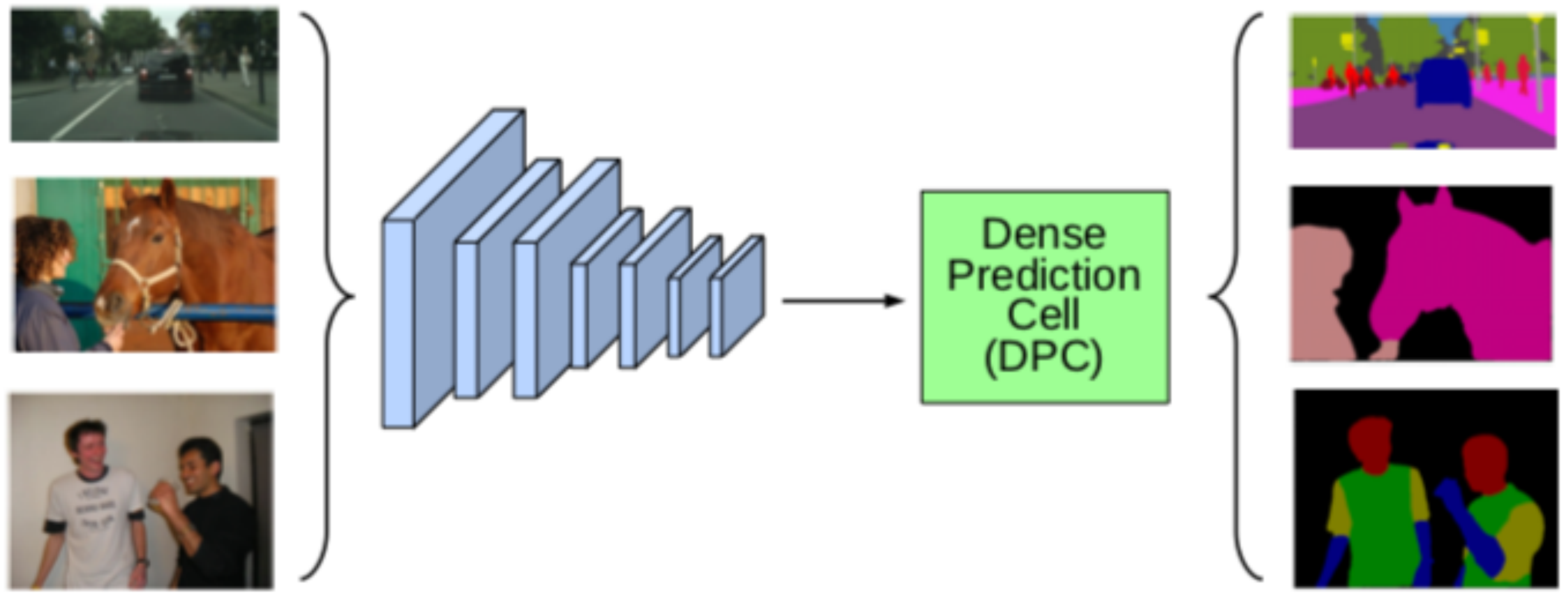
Background
Paper overview
Search space
Sampling strategy
Performance estimation
Results

Search space

- Majority of network arch is fixed
  - MobileNet V2 classification net
  - Xception classification net
- Chop last few layers off classification net and add some new layers (DPC)

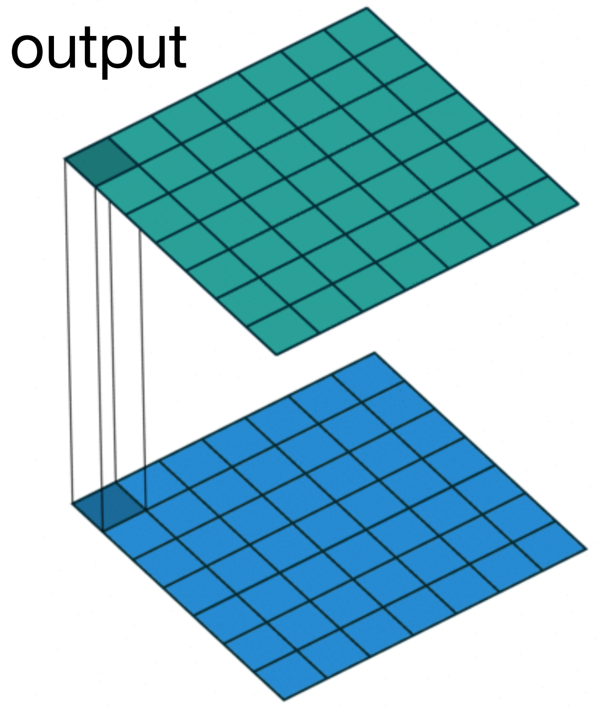




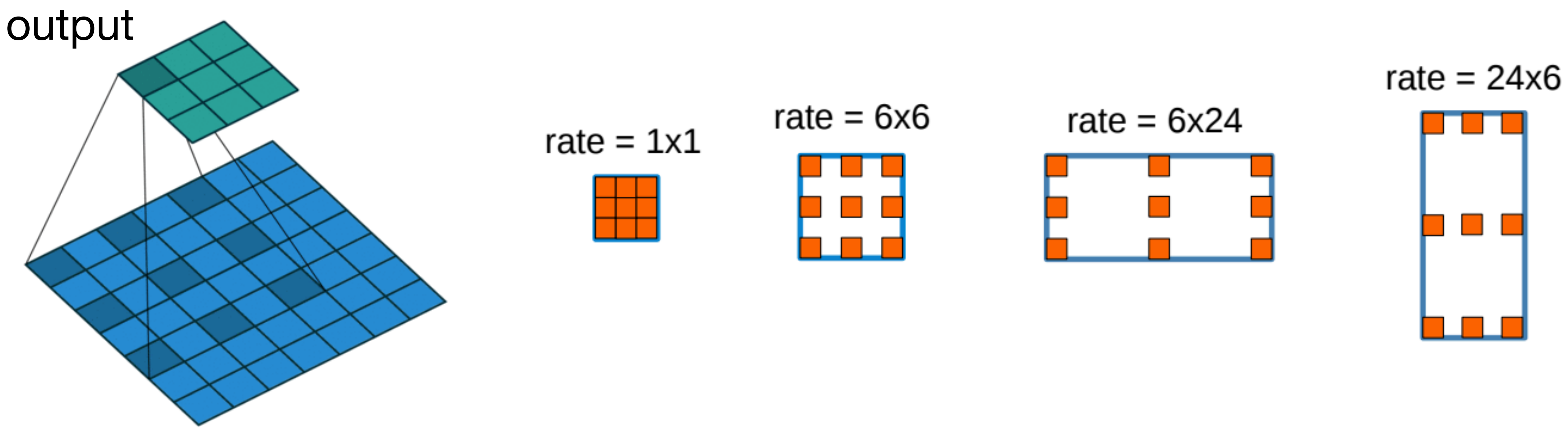


4.2 x 10^{11} search space

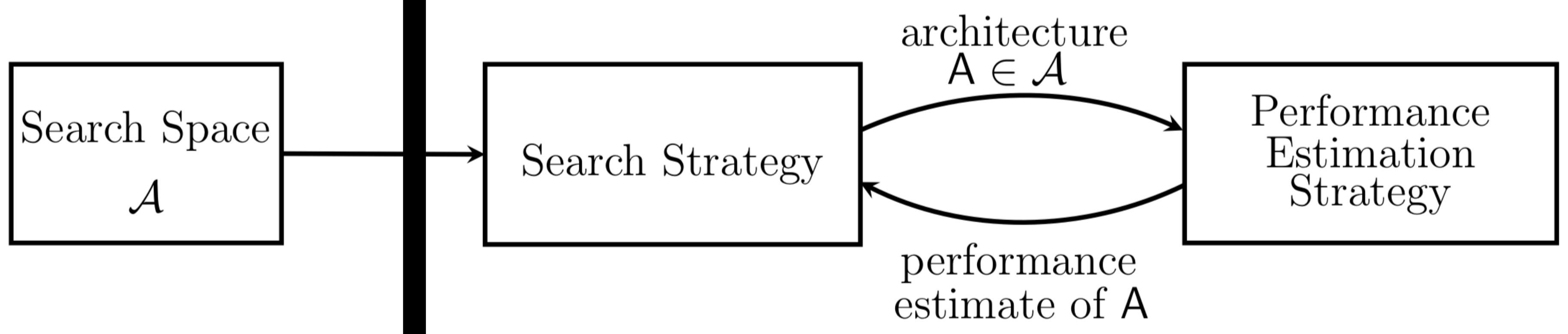
- 1x1 convolution



- 3x3 dilated convolution



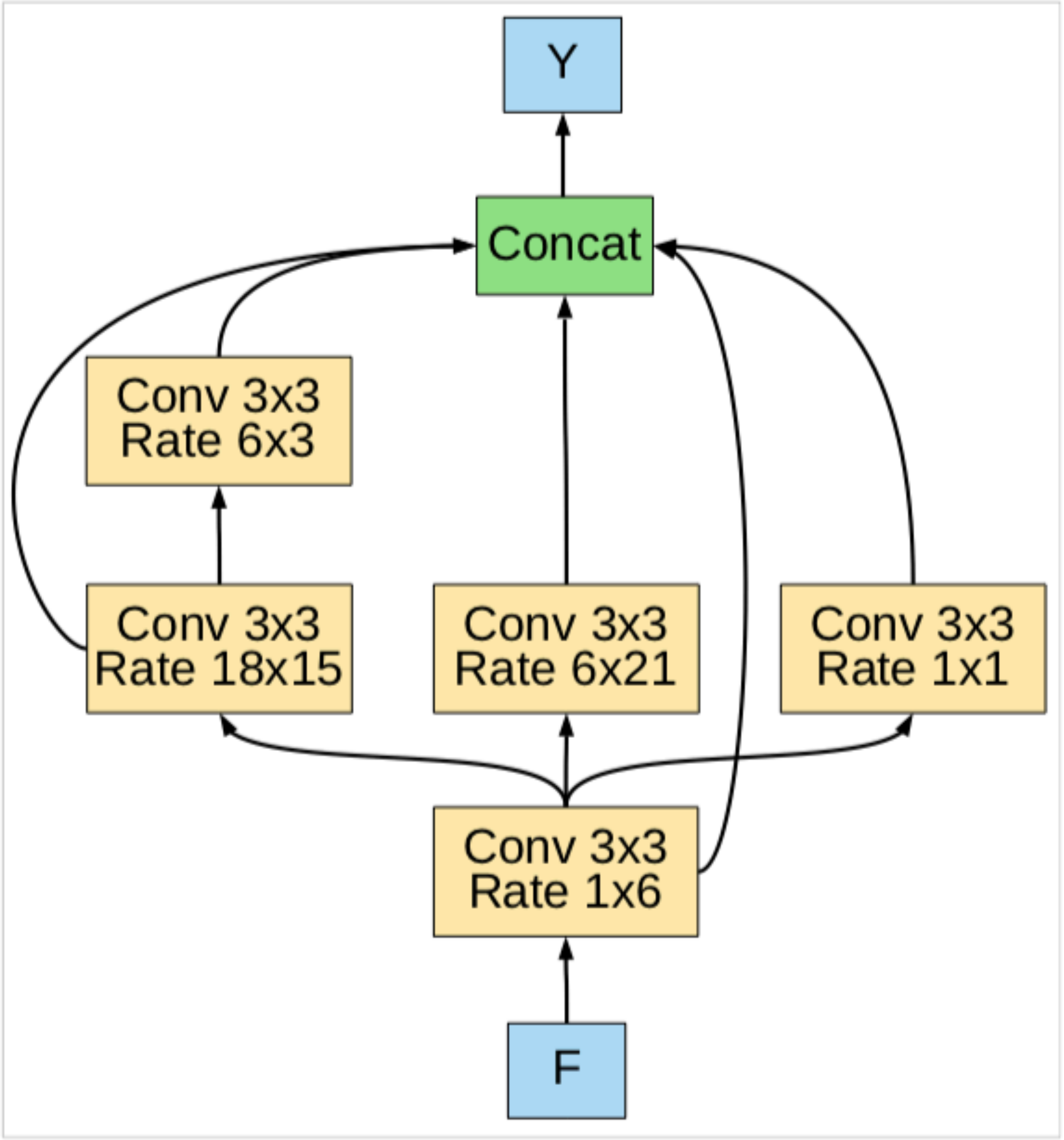
- Average spatial pyramid pooling (downsample, conv1x1, upsample)

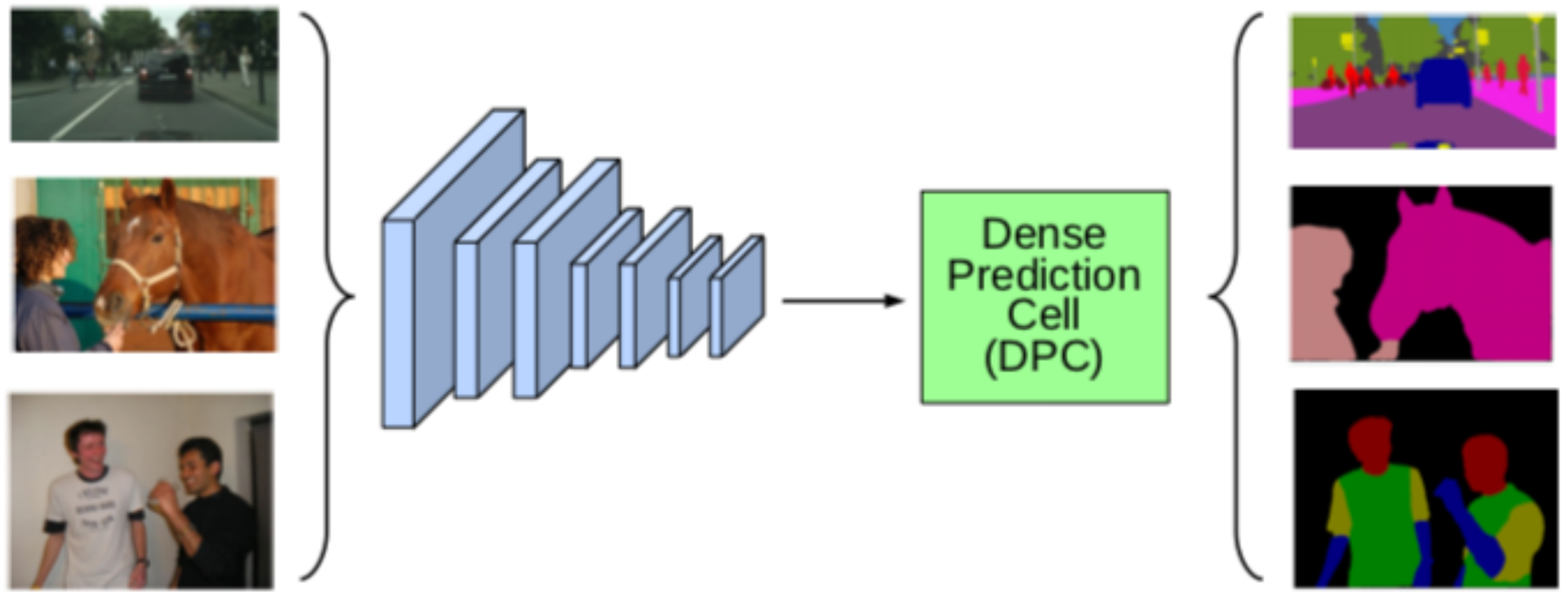


Dense Prediction Cell

Random Sampling

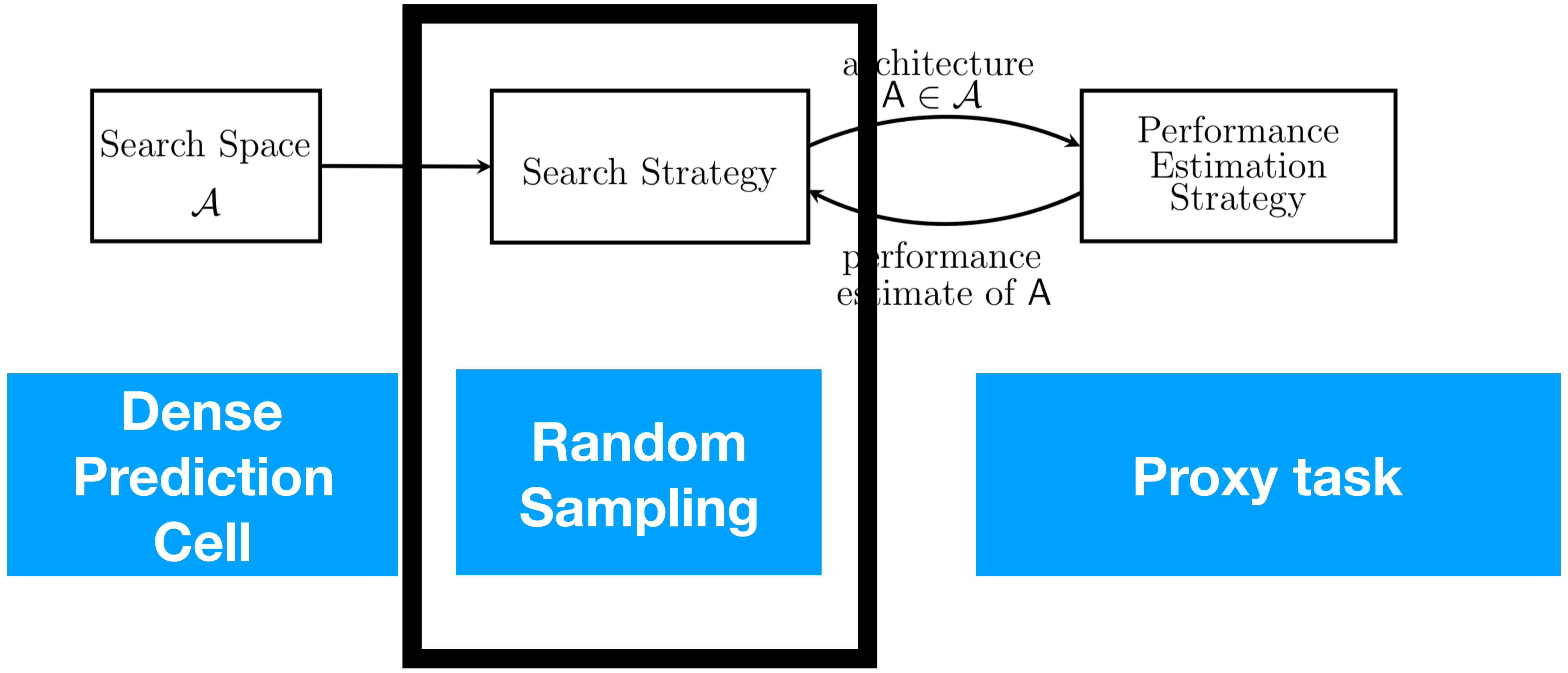
Proxy task

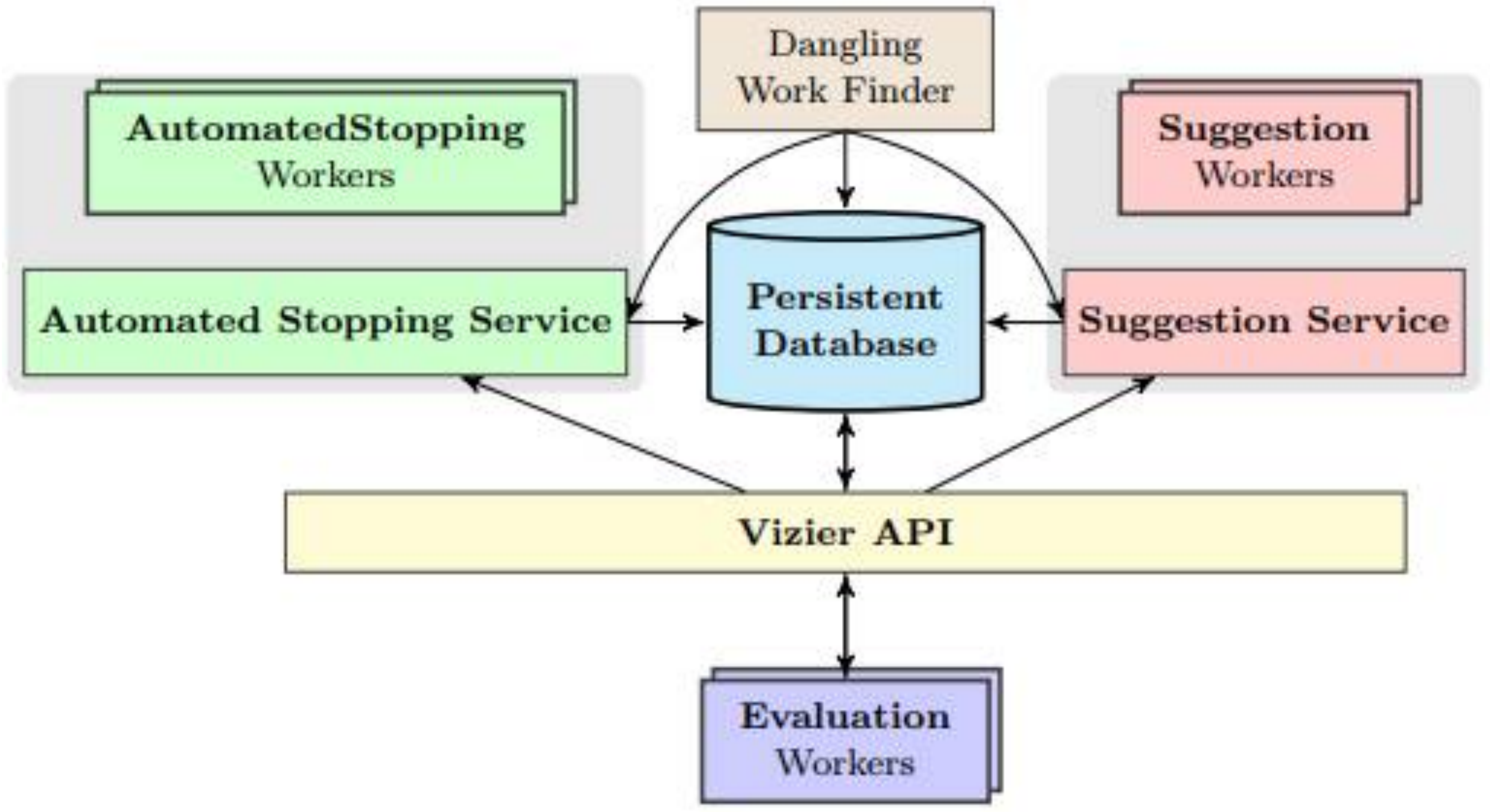




4.2 x 10^{11} search space

Background
Paper overview
Search space
Sampling strategy
Performance estimation
Results



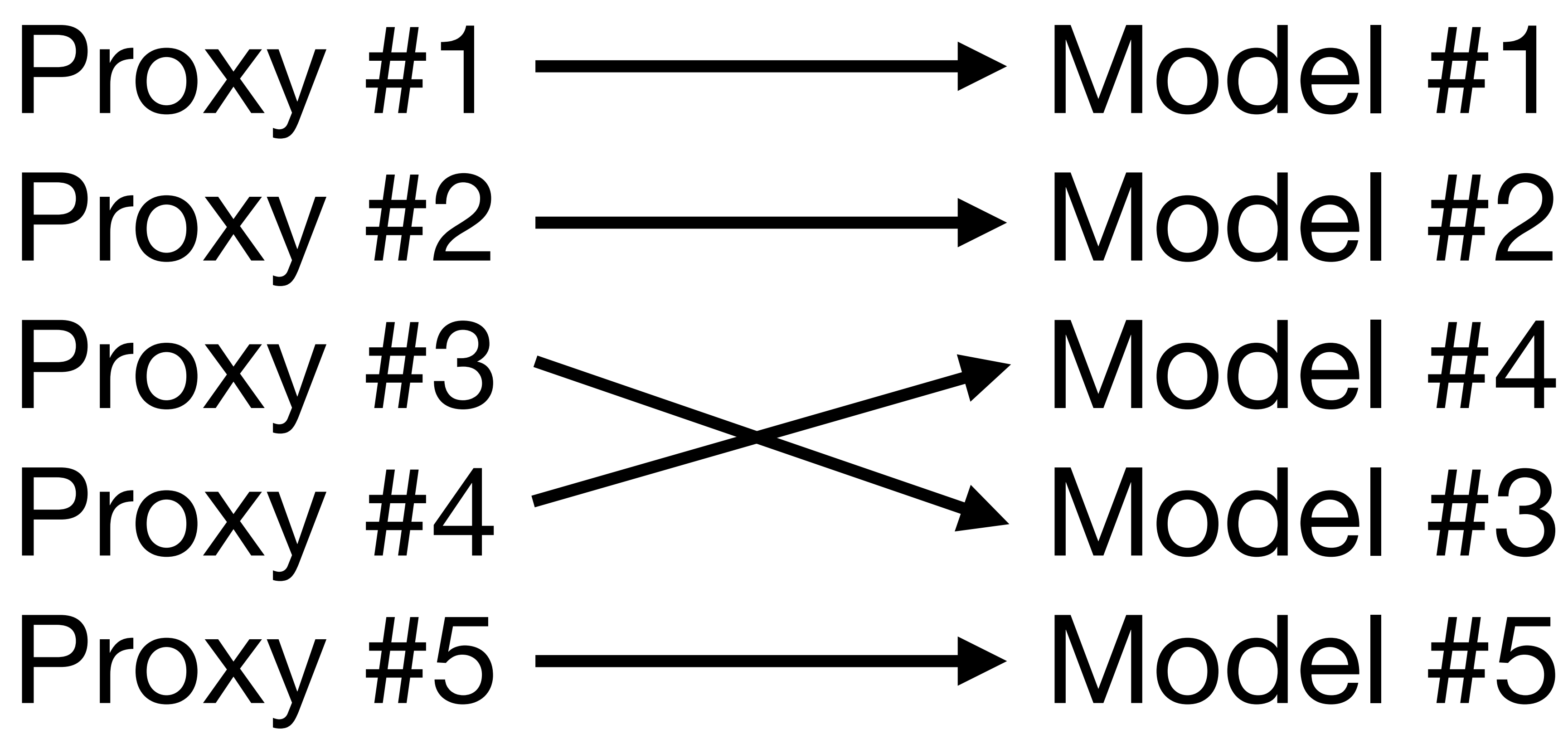


Our search space size is on the order of 10^{11} and we adopt the *random search* algorithm implemented by Vizier [30], which basically employs the strategy of sampling points $\boldsymbol{b}$ uniformly at random as well as sampling some points $\boldsymbol{b}$ near the currently best observed architectures. We refer the interested readers to [30] for more details. Note that the *random search* algorithm is a simple yet powerful method. As highlighted in [101], random search is competitive with reinforcement learning and other learning techniques [52].

Background
Paper overview
Search space
Sampling strategy
Performance estimation
Results

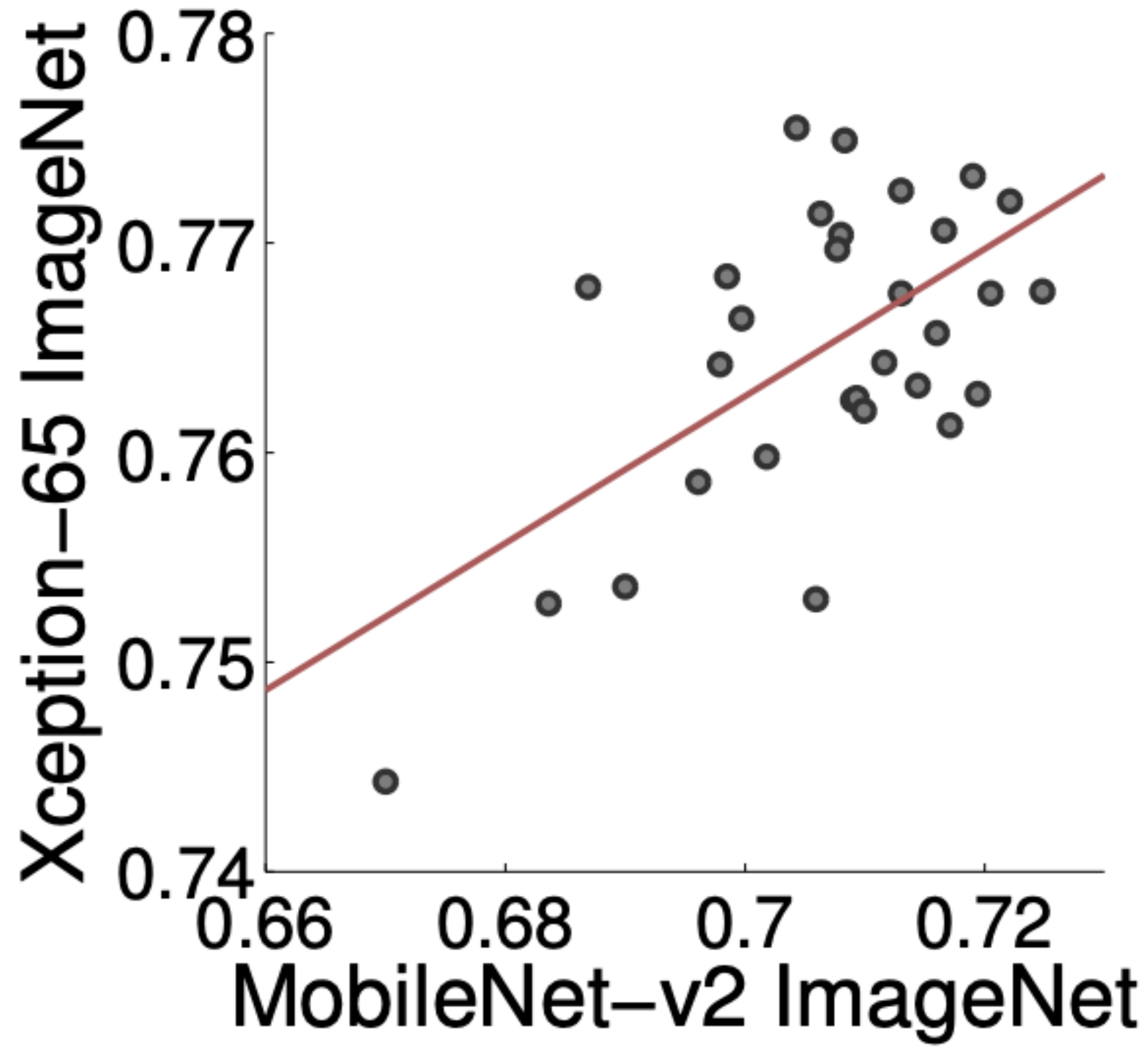
Faster NAS using proxy tasks

- **IDEA:** Estimate architecture performance using a proxy task
- The better the proxy task is, the more efficient search is
- Key contribution of this paper is task-specific proxy tasks



Proxy task 1: Train using MobileNet

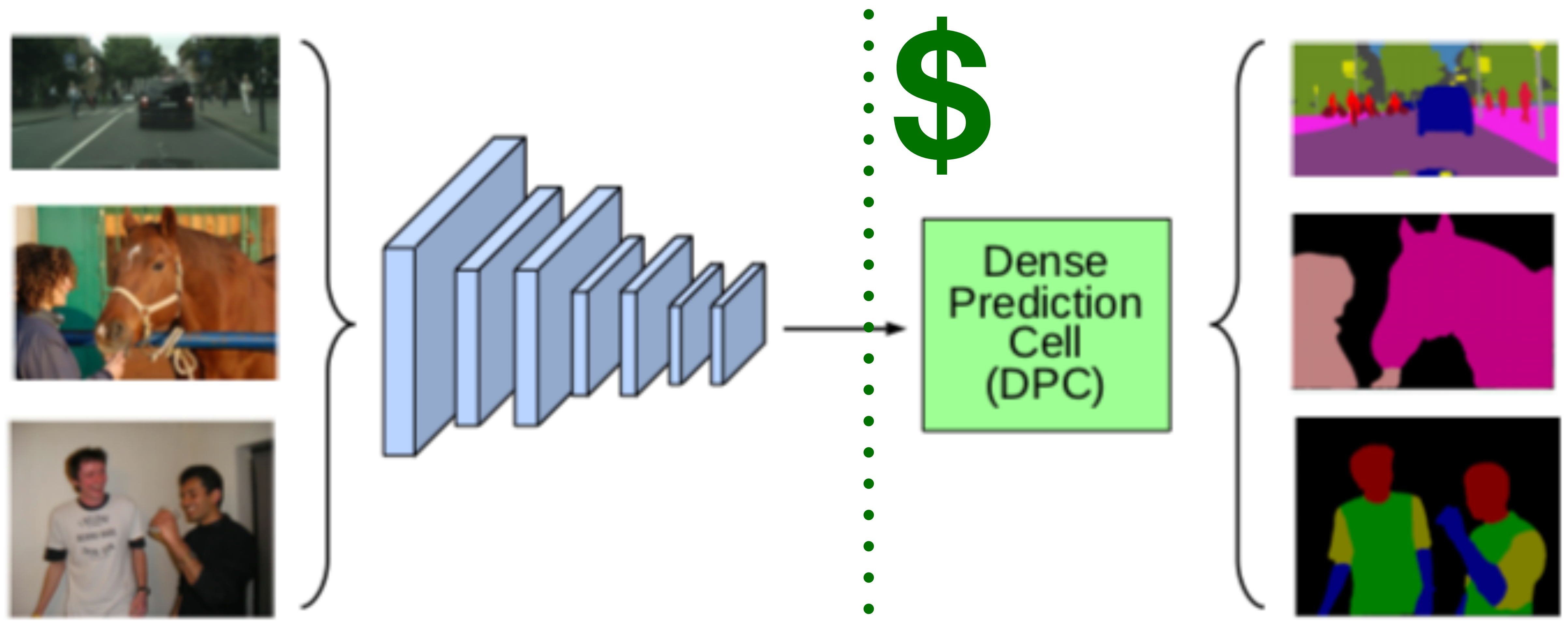
- **Predict final accuracy by using a smaller classification network**
  - *Xception:* 21% top-1 error, 22M params
  - *MobileNet v2:* 28% top 1 error, 3.4M params

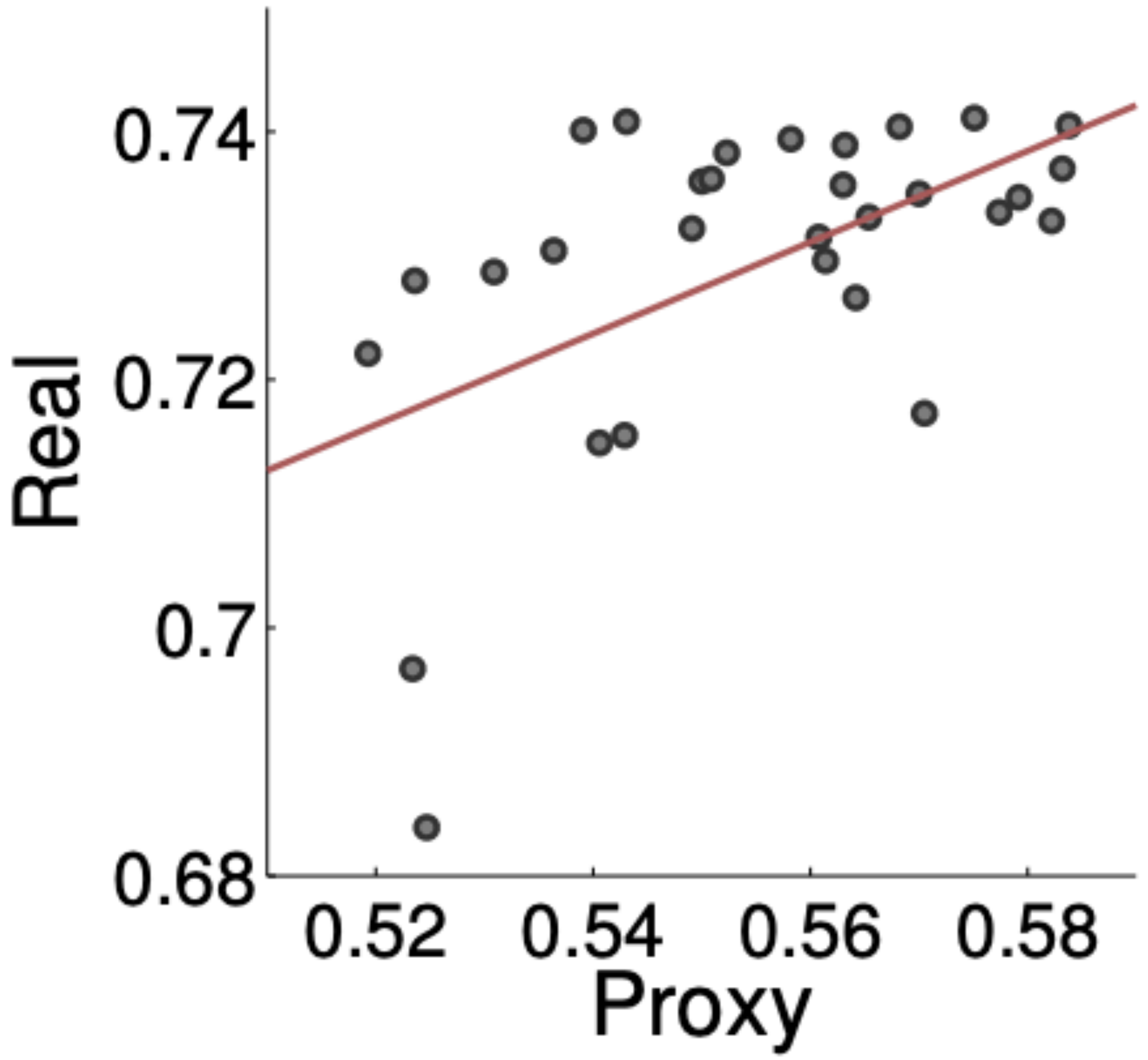


(a) $\rho = 0.36$

Proxy task 2: Cache activations

Cache classification network activations and only train new layers (freeze gradient)





(b) $\rho = 0.47$

Background
Paper overview
Search space
Sampling strategy
Performance estimation
Results

Cityscapes Semantic Segmentation

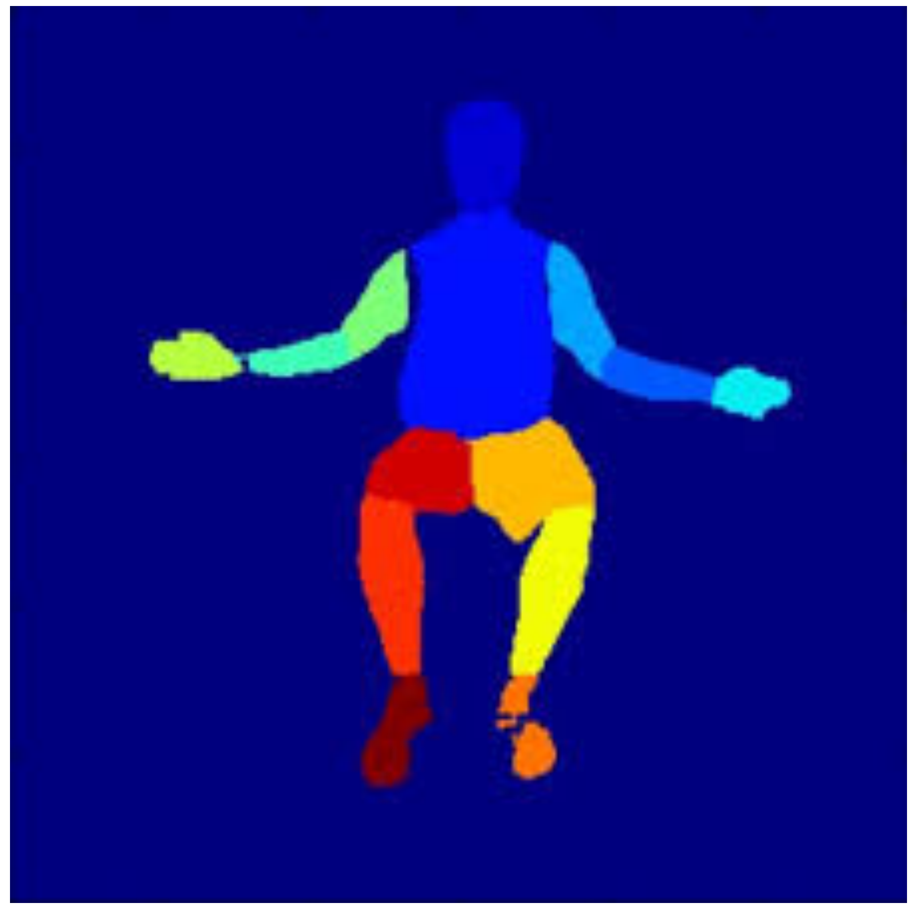
Network Backbone	Module	Params	MAdds	mIOU (%)
MobileNet-v2	ASPP [12]	0.25M	2.82B	73.97
MobileNet-v2	DPC	0.36M	3.00B	**75.38**
Modified Xception	ASPP [12]	1.59M	18.12B	80.25
Modified Xception	DPC	0.81M	6.84B	**80.85**

Table 1: Cityscapes *validation* set performance (labeling IOU) across different network backbones (output stride = 16). ASPP is the previous state-of-the-art system [12] and DPC indicates this work. Params and MAdds indicate the number of parameters and number of multiply-add operations in each multi-scale context module.

Method	road	sidewalk	building	wall	fence	pole	light	sign	vege.	terrain	sky	person	rider	car	truck	bus	train	mbike	bicycle	mIOU
PSPNet [97]	**98.7**	86.9	93.5	58.4	63.7	67.7	76.1	80.5	93.6	72.2	95.3	86.8	71.9	96.2	77.7	91.5	83.6	70.8	77.5	81.2
Mapillary Research [6]	98.4	85.0	93.7	**61.8**	**63.9**	67.7	77.4	80.8	93.7	71.9	95.6	86.7	72.8	95.7	79.9	93.1	**89.7**	72.6	78.2	82.0
DeepLabv3+ [14]	**98.7**	87.0	**93.9**	59.5	63.7	**71.4**	**78.2**	**82.2**	**94.0**	73.0	**95.9**	88.0	73.3	96.4	78.0	90.9	83.9	73.8	78.9	82.1
DPC	**98.7**	**87.1**	93.8	57.7	63.5	71.0	78.0	82.1	**94.0**	**73.3**	95.4	**88.2**	**74.5**	**96.5**	**81.2**	**93.3**	89.0	**74.1**	**79.0**	**82.7**

Table 2: Cityscapes *test* set performance across leading competitive models.

Person-part identification



Method	head	torso	u-arms	l-arms	u-legs	l-legs	bkg	mIOU
Liang *et al.* [47]	82.89	67.15	51.42	48.72	51.72	45.91	97.18	63.57
Xia *et al.* [89]	85.50	67.87	54.72	54.30	48.25	44.76	95.32	64.39
Fang *et al.* [25]	87.15	72.28	57.07	56.21	52.43	50.36	**97.72**	67.60
DPC	**88.81**	**74.54**	**63.85**	**63.73**	**57.24**	**54.55**	96.66	**71.34**

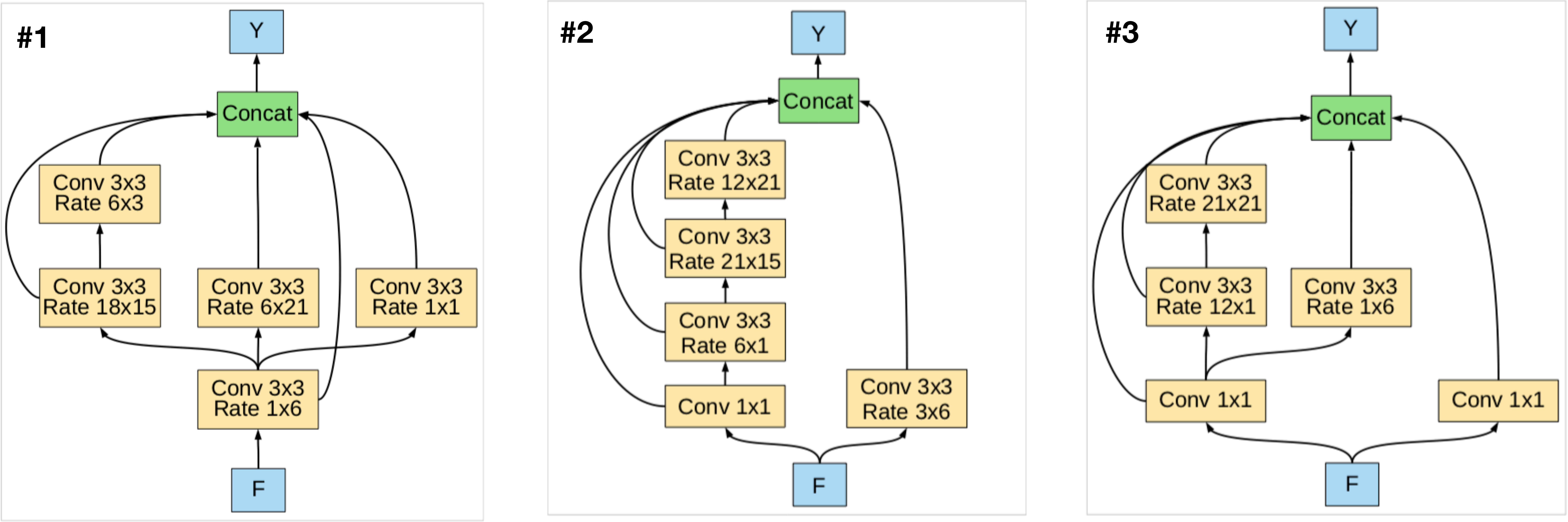
Table 3: PASCAL-Person-Part *validation* set performance.

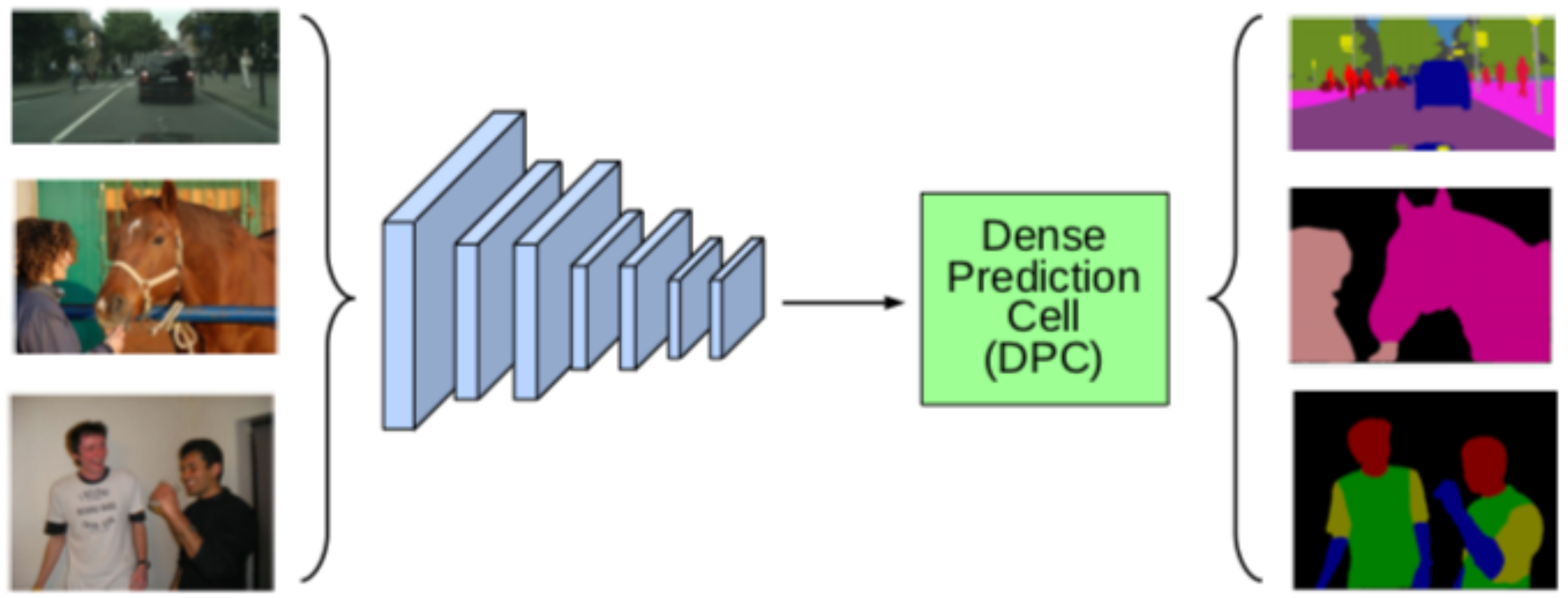
PASCAL VOC scene understanding

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIOU
EncNet [95]	95.3	76.9	94.2	80.2	85.3	96.5	90.8	96.3	47.9	93.9	**80.0**	92.4	96.6	90.5	91.5	70.9	93.6	66.5	87.7	80.8	85.9
DFN [93]	96.4	78.6	95.5	79.1	86.4	97.1	91.4	95.0	47.7	92.9	77.2	91.0	96.7	92.2	91.7	76.5	93.1	64.4	88.3	81.2	86.2
DeepLabv3+ [14]	97.0	77.1	**97.1**	79.3	**89.3**	97.4	93.2	96.6	**56.9**	95.0	79.2	93.1	97.0	**94.0**	**92.8**	71.3	92.9	72.4	91.0	84.9	87.8
ExFuse [96]	96.8	**80.3**	97.0	**82.5**	87.8	96.3	92.6	96.4	53.3	94.3	78.4	94.1	94.9	91.6	92.3	**81.7**	**94.8**	70.3	90.1	83.8	87.9
MSCI [48]	96.8	76.8	97.0	80.6	**89.3**	97.4	**93.8**	**97.1**	56.7	94.3	78.3	93.5	97.1	**94.0**	**92.8**	72.3	92.6	**73.6**	90.8	**85.4**	**88.0**
DPC	**97.4**	77.5	96.6	79.4	87.2	**97.6**	90.1	96.6	56.8	**97.0**	77.0	**94.3**	**97.5**	93.2	92.5	78.9	94.3	70.1	**91.4**	84.0	87.9

Table 4: PASCAL VOC 2012 *test* set performance.

Dense Prediction Cells learned





Some discussion points

- What are **new application** *areas* for NAS?
  - Ideas? object detection, speech generation, GANs?
- Does NAS **un-democratize ML**?
  - Google leads the training compute arms race
- Will the NAS workload influence how **hardware should look**?
- Seems like significant domain knowledge is necessary to develop SoTA NAS methods — is NAS most useful as a research productivity tool?