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

Train & evaluate models

Iterate to find the most accurate model

20K times

Loop image courtesy Barret Zoph, Quoc Le
Blueprint for an AutoML paper

Search Space $\mathcal{A}$ → Search Strategy → Performance Estimation Strategy

architecture $A \in \mathcal{A}$

performance estimate of $A$

Loop image courtesy Barret Zoph, Quoc Le
Learning straight-line DNNs (simple data)
Learning straight-line DNNs (simple data)

NASNet exceeded human performance on CIFAR and COCO
Learning straight-line DNNs (simple data)

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

Constrained optimization objective for mobile inference latency
**Learning straight-line DNNs (simple data)**

**Learning Transferable Architectures for Scalable Image Recognition**

Barret Zoph  
Google Brain  
barrettzoph@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\textsuperscript{1}, Bo Chen\textsuperscript{2}, Ruoming Pang\textsuperscript{1}, Vijay Vasudevan\textsuperscript{1}, Quoc V. Le\textsuperscript{1}  
\textsuperscript{1}Google Brain, \textsuperscript{2}Google Inc.  
{tanmingxing, bochen, rpang, vrv, qvl}@google.com

**Constrained optimization objective for mobile inference latency**

**DARTS: Differentiable Architecture Search**

Hanxiao Liu  
CMU  
hanxiao1@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)?
What is segmentation? Label each pixel of an image with a 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
Barret Zoph
Maxwell D. Collins
Yuxin Zhu
Hartwig Adam
George Papandreou
Jonathan Shlens

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 representations 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 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 [64, 84, 85, 54, 56, 8], speech recognition [96, 9], and machine translation [89, 82] towards the design of neural network architectures. Recent work has demonstrated successes in automatically designing network architectures, largely focused on single-labeled image classification tasks [100, 101, 52] (but see [100, 85] 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, 85, 86].

Image classification has provided a great starting point because much research effort has identified successful network norms and operate on that may be employed to construct search spaces for architectures [82, 85, 101]. Additionally, image classification is inherently multi-resolution whereby fully convolutional architectures [72, 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 [89, 11, 97], object detection [71, 51], and instance segmentation [28, 33, 9] rely on high resolution image inputs and multi-scale image

- 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
Only learn a single “Dense Prediction Cell”

Sample graphs using random search (Vizier)

A) Train using mobile backbone
   B) Cache feature maps
   C) Early stopping
      (90m per sample = 100x speedup)
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)
Dense Prediction Cell

Random Sampling

Proxy task

Search Space \( \mathcal{A} \) → Search Strategy → Performance Estimation Strategy

- 1x1 convolution
- 3x3 dilated convolution
- Average spatial pyramid pooling (downsample, conv1x1, upsample)

4.2 \times 10^{11} \text{ search space}
4.2 \times 10^{11} \text{ 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 $b$ uniformly at random as well as sampling some points $b$ 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
Proxy task 2: Cache activations

Cache classification network activations and only train new layers (freeze gradient)
Background
Paper overview
Search space
Sampling strategy
Performance estimation
**Results**
Cityscapes Semantic Segmentation

<table>
<thead>
<tr>
<th>Network Backbone</th>
<th>Module</th>
<th>Params</th>
<th>MAdds</th>
<th>mIOU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNet-v2</td>
<td>ASPP [12]</td>
<td>0.25M</td>
<td>2.82B</td>
<td>73.97</td>
</tr>
<tr>
<td>MobileNet-v2</td>
<td>DPC</td>
<td>0.36M</td>
<td>3.00B</td>
<td>75.38</td>
</tr>
<tr>
<td>Modified Xception</td>
<td>ASPP [12]</td>
<td>1.59M</td>
<td>18.12B</td>
<td>80.25</td>
</tr>
<tr>
<td>Modified Xception</td>
<td>DPC</td>
<td>0.81M</td>
<td>6.84B</td>
<td>80.85</td>
</tr>
</tbody>
</table>

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.

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

Table 2: Cityscapes test set performance across leading competitive models.
Person-part identification

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

Table 3: PASCAL-Person-Part validation set performance.
### PASCAL VOC scene understanding

<table>
<thead>
<tr>
<th>Method</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mIOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncNet [95]</td>
<td>95.3</td>
<td>76.9</td>
<td>94.2</td>
<td>80.2</td>
<td>85.3</td>
<td>96.5</td>
<td>90.8</td>
<td>96.3</td>
<td>47.9</td>
<td>93.9</td>
<td>80.0</td>
<td>92.4</td>
<td>96.6</td>
<td>90.5</td>
<td>91.5</td>
<td>70.9</td>
<td>93.6</td>
<td>66.5</td>
<td>87.7</td>
<td>80.8</td>
<td>85.9</td>
</tr>
<tr>
<td>DFN [93]</td>
<td>96.4</td>
<td>78.6</td>
<td>95.5</td>
<td>79.1</td>
<td>86.4</td>
<td>97.1</td>
<td>91.4</td>
<td>95.0</td>
<td>47.7</td>
<td>92.9</td>
<td>77.2</td>
<td>91.0</td>
<td>96.7</td>
<td>92.2</td>
<td>91.7</td>
<td>76.5</td>
<td>93.1</td>
<td>64.4</td>
<td>88.3</td>
<td>81.2</td>
<td>86.2</td>
</tr>
<tr>
<td>DeepLabv3+ [14]</td>
<td>97.0</td>
<td>77.1</td>
<td>97.1</td>
<td>79.3</td>
<td>89.3</td>
<td>97.4</td>
<td>93.2</td>
<td>96.6</td>
<td>56.9</td>
<td>95.0</td>
<td>79.2</td>
<td>93.1</td>
<td>97.0</td>
<td>94.0</td>
<td>92.8</td>
<td>71.3</td>
<td>92.9</td>
<td>72.4</td>
<td>91.0</td>
<td>84.9</td>
<td>87.8</td>
</tr>
<tr>
<td>ExFuse [96]</td>
<td>96.8</td>
<td>80.3</td>
<td>97.0</td>
<td>82.5</td>
<td>97.8</td>
<td>96.3</td>
<td>92.6</td>
<td>96.4</td>
<td>53.3</td>
<td>94.3</td>
<td>78.4</td>
<td>94.1</td>
<td>94.9</td>
<td>91.6</td>
<td>92.3</td>
<td>81.7</td>
<td>94.8</td>
<td>70.3</td>
<td>90.1</td>
<td>83.8</td>
<td>87.9</td>
</tr>
<tr>
<td>MSCI [48]</td>
<td>96.8</td>
<td>76.8</td>
<td>97.0</td>
<td>80.6</td>
<td>89.3</td>
<td>97.4</td>
<td>93.8</td>
<td>97.1</td>
<td>56.7</td>
<td>94.3</td>
<td>78.3</td>
<td>93.5</td>
<td>97.1</td>
<td>94.0</td>
<td>92.8</td>
<td>72.3</td>
<td>92.6</td>
<td>73.6</td>
<td>90.8</td>
<td>85.4</td>
<td>88.0</td>
</tr>
<tr>
<td>DPC</td>
<td>97.4</td>
<td>77.5</td>
<td>96.6</td>
<td>79.4</td>
<td>87.2</td>
<td>97.6</td>
<td>90.1</td>
<td>96.6</td>
<td>56.8</td>
<td>97.0</td>
<td>77.0</td>
<td>94.3</td>
<td>97.5</td>
<td>93.2</td>
<td>92.5</td>
<td>78.9</td>
<td>94.3</td>
<td>70.1</td>
<td>91.4</td>
<td>84.0</td>
<td>87.9</td>
</tr>
</tbody>
</table>

Table 4: PASCAL VOC 2012 test set performance.
Dense Prediction Cells learned

#1
- Concat
- Conv 3x3 Rate 6x3
- Conv 3x3 Rate 18x15
- Conv 3x3 Rate 6x21
- Conv 3x3 Rate 1x1
- Conv 3x3 Rate 1x6
- Y
- F

#2
- Concat
- Conv 3x3 Rate 12x21
- Conv 3x3 Rate 21x15
- Conv 3x3 Rate 6x1
- Conv 1x1
- Conv 3x3 Rate 3x6
- Y
- F

#3
- Concat
- Conv 3x3 Rate 21x21
- Conv 3x3 Rate 12x1
- Conv 3x3 Rate 1x6
- Conv 1x1
- Conv 1x1
- Y
- F
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?