EASNet: Searching Elastic and Accurate Network Architecture for Stereo Matching

Qiang Wang[1,2,3], Shaohuai Shi[4], Kaiyong Zhao[5,3], and Xiaowen Chu[6,4,3*]

[1] Harbin Institute of Technology (Shenzhen), Shenzhen, China
qiang.wang@hit.edu.cn
[2] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen, China
[3] Hong Kong Baptist University, Hong Kong SAR, China
{qiangwang,kyzhao,chxw}@comp.hkbu.edu.hk
[4] The Hong Kong University of Science and Technology, Hong Kong SAR, China
shaohuais@cse.ust.hk
[5] XGRIDS, xgrids.com, Shenzhen, China
kyzhao@xgrids.com
[6] The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
xwchu@ust.hk

Abstract. Recent advanced studies have spent considerable human efforts on optimizing network architectures for stereo matching but hardly achieved both high accuracy and fast inference speed. To ease the workload in network design, neural architecture search (NAS) has been applied with great success to various sparse prediction tasks, such as image classification and object detection. However, existing NAS studies on the dense prediction task, especially stereo matching, still cannot be efficiently and effectively deployed on devices of different computing capability. To this end, we propose to train an elastic and accurate network for stereo matching (EASNet) that supports various 3D architectural settings on devices with different compute capability. Given the deployment latency constraint on the target device, we can quickly extract a sub-network from the full EASNet without additional training while the accuracy of the sub-network can still be maintained. Extensive experiments show that our EASNet outperforms both state-of-the-art human-designed and NAS-based architectures on Scene Flow and MPI Sintel datasets in terms of model accuracy and inference speed. Particularly, deployed on an inference GPU, EASNet achieves a new SOTA 0.73 EPE on the Scene Flow dataset with 100 ms, which is 4.5× faster than LEAStereo with a better quality model.

Keywords: Stereo Matching, Neural Architecture Search

*Corresponding author.

1 Introduction

Stereo matching, also called disparity estimation, is a conventional but important technique widely applied to various computer vision tasks, such as 3D perception, 3D reconstruction and autonomous driving. Stereo matching aims to find dense correspondences between a pair of rectified stereo images. As traditional stereo matching algorithms with manual feature extraction and matching cost aggregation fail on those textureless and repetitive regions in the images due to lack of their prior information, deep neural network (DNN) based methods avoid this failure by efficiently learning the data distribution and have achieved state-of-the-art (SOTA) performance in many public benchmarks [29, 2, 15, 30] in recent years. However, DNN networks for stereo matching should also be well designed to achieve good performance. Existing human-designed stereo networks can be divided into two main classes, the U-shape network with 2D convolution (U-Conv2D) and cost volume aggregation with 3D convolution (CVA-Conv3D).

The U-Conv2D methods leverage the U-shape encoder-decoder structure with 2D convolution layers to directly predict the disparity map. The representative networks are the DispNet/FlowNet series [12, 21, 22, 29] as well as their variants [33, 42]. This category of networks enjoys computing efficiency of 2D convolution. However, recent studies [10] raise some concerns about the generalization ability of the U-Conv2D methods. In contrast, the CVA-Conv3D methods exploit the concept of semi-global matching [19] and construct a 4D feature volume by aggregating features from each disparity-shift to enhance the generalization ability. In CVA-Conv3D, it firstly constructs cost volumes by concatenating left feature maps with their corresponding right counterparts across each disparity candidate [24, 8, 45]. The cost volumes are then automatically aggregated and regressed by 3D convolution layers to produce the disparity map. This branch of methods nowadays achieves SOTA estimation quality and dominates the leader-board of several public benchmarks [15, 30]. However, due to the expensive computing cost of 3D operations, they typically run very slowly and are difficult for real-time deployment even on the modern powerful AI accelerators (e.g., GPUs).

On the other hand, AutoML [18] techniques (e.g., neural architecture search (NAS) [14]) recently become very popular to relieve AI practitioners from manual trial-and-error effort by automating network design. Recent years have witnessed tremendous successes of NAS in various computer vision tasks (e.g., classification [46, 34], object detection [40], and semantic segmentation [31, 26]). However, existing applications of NAS are mainly used on sparse prediction problems like classification and object detection. It would become very challenging to apply NAS to dense prediction problems like stereo matching because of the following two reasons. 1) In general, NAS needs to search through a humongous set of possible architectures to determine the network components, which requires extensive computational costs (e.g., thousands of GPU hours). 2) The memory footprint and the model computation workload of stereo matching networks are much larger than those of sparse prediction networks. Taking an example of two architectures, GANet [45] and ResNet-50 [17], on stereo matching problems and

image classification problems, respectively. To process one sample on an Nvidia Tesla V100 GPU, GANet requires nearly 29 GB of GPU memory and 1.9 seconds inference time (on the Scene Flow [29] dataset), while ResNet-50 only requires 1.5 GB and 0.02 seconds (on the ImageNet [11] dataset). Therefore, directly applying the strategies of sparse prediction in NAS to stereo matching can lead to prohibitive workloads due to the explosion of computational resource demands.

To avoid such a problem, Saikia et al. [36] propose AutoDispNet that searches the architecture based on the U-Conv2D methods, and it limits its search space on three different cell structures rather than the full architecture. Although AutoDispNet saves search time, it still cannot surpass the existing SOTA CVA-Conv3D methods (e.g. AANet [43]) in both model accuracy and inference speed. Later, Cheng [10] leveraged task-specific human knowledge in the search space design to reduce the demands of computational resources in searching architectures, and proposed an end-to-end hierarchical NAS network named LEAStereo, which achieved the SOTA accuracy among the existing CVA-3D methods. However, LEAStereo takes 0.3 seconds of model inference even on a high-end Nvidia Tesla V100 GPU, which is far away from the requirement of real-time applications. Moreover, both AutoDispNet and LEAStereo attempt to find a specialized network architecture, and train it from scratch, thus cannot be scaled to different devices. Notice that the deployment of stereo matching applications can have diverse computing resource constraints, from high-end cloud servers to low-end edge devices or robotics embedded devices. To meet the latency requirement of a new given device, the above two methods need to re-search and re-train the model, which requires large labor. The recent proposed once-for-all (OFA) network [5] tries to support diverse architectural settings, but it only explores the sparse prediction problem like image classification. In summary, existing stereo matching methods including human-designed architectures and searched architectures cannot well fulfill real-world deployment requirements which need to consider model accuracy, inference speed, and training cost.

To this end, we propose to train an elastic and accurate stereo matching network, EAS-Net, which enables model deployment on devices of different computing capability to guarantee the inference speed without additional training while the model accuracy can be maintained. Furthermore, our EASNet does not need to re-train or re-search the architecture for deployment on any new devices. In this paper, we make three-fold contributions:

- Based on the pipeline of the CVA-3D methods, we propose an end-to-end stereo matching network named EASNet that contains four function modules. We allow the network to search for both the layer level and the network level structures in a huge network candidate space.
- To efficiently train EASNet, we develop a multiple-stage training scheme for EASNet to reduce the model size across diverse dimensions of network architecture parameters including depth, width, kernel size and scale. Our training strategy can significantly improve the prediction accuracy of sub-networks sampled from the largest full EASNet structure, which enables

flexible deployment according to the target device computing capability and latency requirement without any additional model training.
– We conduct extensive experiments to evaluate the effectiveness of EASNet on several popular stereo datasets among three GPUs of different computing power levels. Under all deployment scenarios, EASNet outperforms both the human-designed and NAS-based networks in terms of model accuracy and inference speed.

2 Related Work

2.1 Manual DNN Design for Stereo Matching

In recent years, many deep learning methods have been proposed for stereo matching by extracting effective features from a pair of stereo images and estimating their correspondence cost, which can be classified into 2D with the U-shape network (U-Conv2D) and 3D with cost volume aggregation (CVA-Conv3D) methods.

On the one hand, in the U-Conv2D networks, the U-shape network architecture mainly utilizes 2D convolution layers [29][33] to estimate disparity, which takes a pair of rectified stereo images as input and generates the disparity by direct regression. However, the pure 2D CNN architectures are difficult to capture the matching features such that the estimation results are not good. On the other hand, the 3D methods with cost volume aggregation, named CVA-Conv3D, are further proposed to improve the estimation performance [44][24][8][45][32], which apply 3D convolutions to cost volume aggregation. The cost volume is mainly constructed by concatenating left feature maps with their corresponding right counterparts across each disparity level [24][8], and the features of the generated cost volumes can be learned by 3D convolution layers. Nowadays the top-tier CVA-Conv3D methods [45, 43, 10] have achieved very good accuracy on various public benchmarks. However, the key limitation of CVA-Conv3D is its high computation resource requirements, which makes them be difficult for real-world deployment. For example, GANet [45] and LEAStereo [10] take 1.9 seconds and 0.3 seconds respectively on predicting the disparity map of a stereo pair of 960×540 even using a very powerful Nvidia Tesla V100 GPU. Though they achieve good accuracy, it is difficult to deploy them for real-time inference.

2.2 NAS-based Stereo Matching

To lessen the effort dedicated to designing network architectures, AutoML [18], especially NAS [14, 46, 34, 28, 4], has become an increasingly active research area over the past few years. While most of the early studies [1, 3, 37, 27, 35, 28] have proven the effectiveness of NAS in many sparse prediction tasks, the extension to dense prediction tasks, such as semantic segmentation [31, 9] and stereo matching [36, 10], is still at an early stage. AutoDispNet is the first work that adopts the DARTS NAS method to search the efficient basic cell structure for

the U-Conv2D method. However, to reduce the prohibitive search space, it only searches partially three different cell structures rather than the full architecture methods, and thus achieves limited model accuracy and generalization. In contrast, LEAStereo [10] leveraged the domain knowledge of stereo matching and designed a hierarchy end-to-end pipeline, which allows the network to automatically select the optimal structures. However, LEAStereo is still difficult to be deployed on modern AI processors due to the high computational cost. Furthermore, to meet the latency requirement on the new target device, LEAStereo needs to tune the network search space accordingly, followed by re-searching and re-training the model, where the expensive network specialization is unavoidable.

Recent studies [4, 6, 39, 23, 16, 20, 25] take the hardware capability into account to search the network. As one of the existing SOTA studies, the once-for-all (OFA) network allows direct deployment under various computing devices and constraints by selecting only a part of the original full one without additional model training. However, they only discuss the case for image classification, while the domain knowledge and processing pipeline of stereo matching are much different from the sparse prediction task of OFA. In this paper, we propose a novel network named EASNet to search elastic and accurate stereo matching networks, and design a specialized network search space according to the prior geometric knowledge of stereo matching. Our EASNet covers a wide range of search dimensions (kernel size, width, depth, and scale). With negligible accuracy loss and without any extra model fine-tuning, our EASNet can be directly deployed on different scenarios of computing power and resource constraint, which refers to its "elastic" and "accurate" characteristics.

3 Our Method: EASNet

In this section, we present our proposed elastic network structure for stereo matching, EASNet, that covers four function modules in the search and training pipeline. We support up to four search dimensions for different modules in EASNet. We firstly describe the architecture search space of each function module, including their basic unit and supported search dimensions. Then we introduce the training approach across four search dimensions to maximizing the average model accuracy of all the derived sub-networks in EASNet.

3.1 The Architecture Space of EASNet

In this subsection, we introduce the overview structure of EASNet. As illustrated in Fig. 1, EASNet is composed of four modules: feature pyramid construction, cost volume, cost aggregation, and disparity regression and refinement. The functions of these modules are benefited from prior human knowledge in stereo matching and success of previous hand-crafted network architecture design. EASNet enables its flexibility and effectiveness by providing different levels of support of neural architecture search for these four modules.

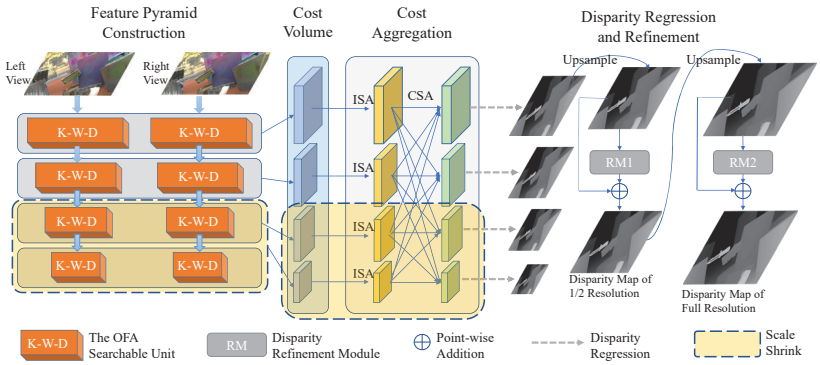

Fig. 1: The model structure of our proposed EASNet. It contains four modules with different functions derived from the domain knowledge of stereo matching. The OFA searchable unit applies the similar methodology in [5]. The parts covered by the shallow yellow dotted blocks can be alternatively skipped when applying scale shrinking.

Feature Extraction. In feature extraction, we need to extract multi-scale features from the input left and right images and construct a feature pyramid for the latter cost volume stage. Thus, we design a sequence of searchable units (similar to [5]) that cover three important dimensions of CNNs, i.e., depth, width, and kernel size. The ith unit produces features maps of $1/(3 \times 2^{i-1})$ resolution by setting stride=2 for the first convolution layer and stride=1 for the rest in it. For example, in our experimental setting, there are totally four units providing different resolutions of feature maps from 1/3 to 1/24. We also enable each unit to use arbitrary numbers of layers (denoted as elastic depth) as that of OFA [5]. Then we allow each layer to use arbitrary numbers of channels (denoted as elastic width) and arbitrary kernel sizes (denoted as elastic kernel size). In our experimental setting, the depth of each unit is chosen from $\{2, 3, 4\}$; the width expansion ratio in each layer is chosen from $\{2, 4, 6, 8\}$; the kernel size is chosen from $\{3, 5, 7\}$. Therefore, with 4 units, we have roughly $((3 \times 4)^2 + (3 \times 4)^3 + (3 \times 4)^4)^4 \approx 10^{13}$ different architectures. Since all of these sub-networks share the same weights, we only require 5M parameters to store all of them. Without sharing, the total model size will be extremely large, which is impractical.

We further introduce one more dimension of the network search space, the total scale of the feature pyramid (denoted as elastic scale) to EASNet. As proved by existing studies [10], the number of scales in a feature pyramid can significantly affect the model accuracy of disparity prediction. Deeper feature pyramids typically provide better prediction accuracy but require more computational efforts. Thus, we allow our EASNet to skip some high levels of feature maps and fine-tune the low levels. Take an example shown in Fig. 1, the part covered by the shallow golden dotted blocks can be alternatively skipped during model fine-tune and inference. The scale of feature pyramid is chosen from $\{2, 3, 4\}$ in our experiments.

Cost Volume. After the feature pyramids of the left and right images are constructed, we then establish the multi-scale 3D cost volume by correlating left and right image features at corresponding scales with the point-wise multiplication operation, which is similar to AANet [43].

$$\mathbf{C}(d,h,w) = \frac{1}{N}\langle \mathbf{F}_l^s(h,w), \mathbf{F}_r^s(h,w-d)\rangle, \tag{1}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product of two feature vectors and N is the channel number of extracted features. $\mathbf{F}_l^s$ denotes the feature maps of the scale s extracted from the left view, and $\mathbf{F}_r^s$ refers to the ones from the right view. $\mathbf{C}(d,h,w)$ is the matching cost at location (h,w) for disparity candidate d. Thus, S scales of feature pyramid produce S 3D cost volume. The raw cost volume in this module will be then fed into the cost aggregation module. In the cost volume module, we also support elastic scale which can be chosen from [2,3,4]. The chosen scale is naturally consistent with the scale number of feature pyramid.

Cost Aggregation. The cost aggregation module is used to compute and aggregate matching costs from the concatenated cost volumes. We apply the stacked Adaptive Aggregation Modules (AAModules) for flexible and efficient cost aggregation, as it can simultaneously estimate the matching cost in the views of intra-scale and cross-scale. An AAModule consists of S adaptive Intra-Scale Aggregation (ISA) modules and an adaptive Cross-Scale Aggregation (CSA) module for S pyramid levels.

For each scale of the cost volume, ISA can address the popular edge-fattening problem in object boundaries and thin structures by enabling sparse adaptive location sampling. In [43], ISA is implemented by dilated convolution. In particular, we use the same implementation of ISA in [43], which is a stack of three layers (i.e., 1×1 convolution, 3×3 deformable convolution and 1×1 convolution) and a residual connection.

Assume that the resulting cost volume after ISA is $\tilde{C}^s$, we apply the CSA module to explore the correspondence among different scales of $\tilde{C}^s$. To estimate the cross-scale cost aggregation of the scale s, we adopt

$$\hat{C}^s = \sum_{k=1}^{S} f_k(\tilde{C}^s), s = 1,2,...,S \tag{2}$$

where f_k is a function to adaptively combine the cost volumes from different scales. We adopt the same definition of f_k as HRNet [38], which is defined as

$$f_k = \begin{cases} \mathcal{I}, k = s, \\ (s-k)\ 3\times 3 \text{ convs with stride} = 2, k < s, \\ \text{unsampling } \oplus 1\times 1 \text{ conv}, k > s. \end{cases} \tag{3}$$

where $\mathcal{I}$ denotes the identity function and $\oplus$ indicates bilinear up-sampling to the same resolution followed by a 1×1 convolution to align the number of channels. In f_k, when $k < s$, $(s-k)$ 3×3 convolutions with stride=2 are used for $2^{(s-k)}$ times down-sampling to make the resolution consistent.

In the cost aggregation module, we also support elastic scale which can be chosen from [2,3,4]. The chosen scale is naturally consistent with the scale number of feature pyramid. Notice that for S scales of cost volumes, the total number of combinations is $S^2/2$. Removing some scales can considerably reduce the computational efforts of cost aggregation.

Disparity Regression and Refinement. For each scale of the aggregated cost volumes, we use disparity regression as proposed in [24] to produce the estimated disparity map. The probability of each disparity d is calculated from the predicted cost C_s^d via the softmax operation $\sigma(\cdot)$. The estimated disparity $\hat{d}$ is calculated as the sum of each disparity candidate d weighted by its probability.

$$\hat{d} = \sum_{d=0}^{D_{\max}-1} d \times \sigma(c_d) \tag{4}$$

where $D_{\max}$ is the maximum disparity range, $\sigma(\cdot)$ is the softmax function, and c_d is the aggregated matching cost for disparity candidate d. As discussed in [24], this regression has been proved to be more robust than using a convolution layer to directly produce the one-channel disparity map. In our EASNet, it will predict S scales of disparity maps of different resolutions, from 1/3 to $1/(3\times2^{S-1})$.

Notice that the largest regressed disparity map is only 1/3 of the original resolution. To produce the full resolution of disparity, we apply the same two refinement modules in StereoDRNet [7] to hierarchically upsample and refine the predicted 1/3 disparity. The two refinement modules upsample the predicted disparity map from 1/3 to 1/2 and then 1/2 to full resolution, respectively.

3.2 Training EASNet

As discussed in [5], directly finetuning the network from scratch leads to prohibitive training cost and interference of model quality among different sub-networks. To efficiently train EASNet, we extend the progressive shrinking (PS) strategy of OFA to support our specialized search space of stereo matching. We first start with training the largest neural network (denoted as the full EASNet) with the maximum kernel size (K=7), depth (D=4), width (W=8) and scale (S=4). Then we perform four stages to finetune EASNet to support different dimensions of elastic factors.

Elastic Kernel Size, Depth and Width. To search networks of different kernel sizes (K), depths (D) and widths (W), we apply the progressive shrinking strategy in [5], which is an effective and efficient training method to prevent interference among different sub-networks. First, we support elastic kernel size which can choose from $\{3, 5, 7\}$ at each layer, while the depth and width remain the maximum values. This is achieved by introducing kernel transformation matrices which share the kernel weights. For each layer, we have one 25×25 matrix and one 9×9 matrix that are shared among different channels, to transform the largest 7×7 kernels. Second, we support elastic depth. For a specific D, we keep the first D layers and skip the last $(N - D)$ layers (N is the original number of

layers), which results that one depth setting only corresponds to one combination of layers. Third, we support elastic width. We introduce a channel sorting operation to support partial widths, which reorganizes the channels according to their importance (i.e., L1 norm of weights). Until now, we have finished three stages of model finetune.

Elastic Scale. The scale search covers all four function modules in EASNet, and can be chosen from $\{2, 3, 4\}$. Take an example of choosing the scale S for the largest scale N. Starting from feature pyramid construction, we keep the first S scales of feature maps and skip the rest, which forms a S-level of feature pyramid. Then for the cost volume, $N - S$ cost volumes are naturally removed. Next for the cost aggregation, we only need to process $S^2/2$ combinations instead of $N^2/2$. Finally, the last $N - S$ scales of disparity maps are also skipped. In our experiments, this scale shrinking operation not only preserves the accuracy of larger sub-networks but also significantly reduces the network inference time.

Loss Function. Given a pair of rectified stereo RGB images, our EASNet takes them as inputs and produces $S + 2$ disparity maps of different scales, where the first S scales (denoted by $\hat{d_s^i}$) are generated by the AAModules and the rest two are generated by the refinement modules. We denote $\hat{d_h^i}$ as the first refinement result and $\hat{d_f^i}$ as the second one. Assume that the input image size is $H \times W$. For each predicted $\hat{d_i}$, it is first bilinearly upsampled to the full resolution. Then we adopt the pixel-wise smooth L1 loss to calculate the error between the predicted disparity map $\hat{d_i}$ and the ground truth d_i,

$$L_s(d_s, \hat{d_s}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d_s^i - \hat{d_s^i}), s \in [1, ..., S] \tag{5}$$

where N is the number of pixels of the disparity map, d_s^i is the i^{th} element of $d_s \in \mathcal{R}^N$ and

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{6}$$

The predicted refinement results $\hat{d_h}$ and $\hat{d_f}$ also follow the same smooth L1 loss calculation, denoted by L_h and L_f respectively. The final loss function is a weighted summation of losses over all disparity predictions as

$$L = \sum_{s=1}^{S} w_s L_s(d_s, \hat{d_s}) + w_h L_h(d_h, \hat{d_h}) + w_f L_f(d_f, \hat{d_f}). \tag{7}$$

In our experimental setting, the loss weights for the two lowest scale in (7) are set to 1/3 and 2/3, while the rest are all set to 1.0.

4 Evaluation

4.1 Experimental Settings

We conduct extensive experiments on four popular stereo datasets: Scene Flow [29], MPI Sintel [2], KITTI 2012 [15] and KITTI 2015 [30]. We use the training

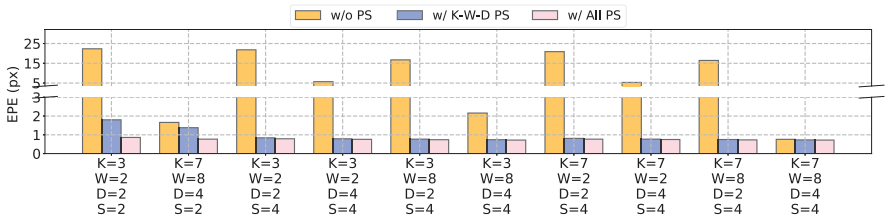

Fig. 2: Scene Flow EPE (px) performance of sub-networks extracted from the full EASNet. K: kernel size, W: width, D: depth, S: Scale.

set of 35,454 samples of Scene Flow to train our EASNet, and then evaluate it on the test set of Scene Flow. For Scene Flow and MPI Sintel, we use end-point error (EPE) to measure the accuracy of the methods, where EPE is the mean disparity error in pixels. For KITTI 2012 and KITTI 2015, we report EPE and official metrics (e.g., D1-all) in the online leader board.

We implement our EASNet using PyTorch 1.8. First, we train the full network for 64 epochs with an initial learning rate of 1×10^{-3}. Then we follow the schedule described in Section 3.2 to further fine-tune the full network, which contains five 25-epoch stages. The initial learning rate of each stage is set to 5×10^{-4}, which is decayed by half every 10 epochs.

To compete for the methods in the online official leader board, we also fine-tune our EASNet on two KITTI datasets. We use a crop size of 336×960, and first fine-tune the pre-trained Scene Flow model on mixed KITTI 2012 and 2015 training sets for 1000 epochs. The initial learning rate is 1×10^{-3} and it is decreased by half every 300 epochs. Then another 1000 epochs are trained on the separate KITTI 2012/2015 training set for benchmarking, with an initial learning rate of 1×10^{-4} and the same learning rate schedule as before.

As for data pre-processing, we follow the steps in [43], including color normalization and random cropping. For all the stages, we use the Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) optimizer to train EASNet. The network is trained with a batch size of 16 on 8 V100 GPUs. The entire architecture search optimization takes about 48 GPU days. Although AutoDispNet and LEAStereo only take 42 and 10 GPU days, the training cost of model re-searching and re-training can be prohibitive when they are applied to new computing devices.

To validate the deployment flexibility and efficiency, we benchmark our EASNet on three Nvidia GPUs with different computing levels, including the server-level Tesla V100, the desktop-level GTX 2070, and the inference-level Tesla P40.

4.2 Experimental Results

Results of Different Sub-networks. Fig. 2 reports the Scene Flow EPEs of sub-networks derived from the full EASNet of different training schemes. Due to space limits, we take 10 sub-networks for comparison, and each of them is

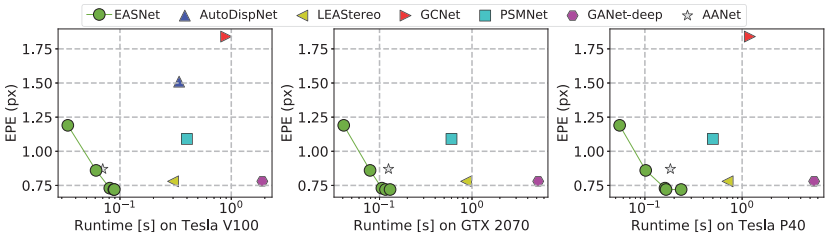

Fig. 3: Our proposed method, EASNet, sets a new state-of-the-art on the Scene Flow *test* dataset with much fewer parameters and much lower inference time. The data points on the EASNet line indicate different sub-networks sampled from its largest full network structure.

denoted as "(K=k, W=w, D=d, S=s)". It represents a sub-network that has d layers for all units in the feature extraction module with the expansion ratio and kernel size set to w and k for all layers, and s scales throughout all the function modules in EASNet. "w/o PS" indicates that we only train the largest full EASNet without model finetune, while "w/ K-W-D PS" and "w/ All PS" indicate that the full EASNet is fine-tuned using progressive shrinking (PS) of the first three stages (kernel size, width, and depth) and the complete five stages, respectively. Without PS, the model accuracy is significantly degraded while shrinking width and depth (seen from the last four sub-networks.). After performing PS for K-W-D, the accuracy of all the sub-networks can be improved by a significant margin. Moreover, our proposed shrinking scheme on scale (S) and refinement (R) can further reduce nearly 50% of the estimation error (seen from the first two sub-networks). Specifically, without architecture optimization, our complete PS scheme can still achieve 0.86 of average EPE using only 0.78 M parameters under the architecture setting (K=3, W=2, D=2, S=2), which is on par with AANet (EPE: 0.87, 3.9 M parameters). In contrast, without the additional PS for scale and refinement, it only achieves 1.8, which is 0.94 worse.

EASNet under Different Hardware Computing Capability. Fig. 3 summarizes the results of different sub-networks extracted from EASNet under three GPUs. We also plot the results of other existing SOTA methods for comparison. First, EASNet outperforms all the other methods with Pareto optimality of both model accuracy and inference time. Take the desktop GPU GTX 2070 as an example. EASNet achieves a new SOTA 0.73 EPE with the runtime of 0.12 s, being 0.14 lower EPE than AANet that has similar inference performance. To achieve similar accuracy of AANet, EASNet performs 0.08 s on GTX 2070, which is 33.3% lower than AANet. Second, since our EASNet only needs one time of training and does not need any further fine-tuning efforts when being deployed on devices of different computing capability, we can directly choose the sub-network from the full EASNet according to the latency requirement, while other methods cannot. For example, if we set the inference latency goal to

Table 1: Quantitative results on Scene Flow dataset. The runtime is measured on Nvidia Tesla P40. Bold indicates the best. Underline indicates the second best. Parentheses indicate that the results are reported by the original paper on Nvidia Tesla V100.

Method	**Params** [M]	**EPE** [px]	**Runtime** [s]
PSMNet [8]	5.22	1.09	0.5
GANet-deep [45]	6.58	0.78	5.5
AANet [43]	3.9	0.87	0.18
AutoDispNet-CSS [36]	37	1.51	(0.34)
LEAStereo [10]	<u>1.81</u>	0.78	0.71
DeepPruner (best) [13]	7.1	0.86	0.18
DeepPruner (fast) [13]	7.1	0.97	**0.06**
EASNet-L	5.07	**0.72**	0.24
EASNet-M	3.03	<u>0.73</u>	0.16
EASNet-S	**0.78**	0.86	<u>0.10</u>

100 ms, for both the existing human-designed and NAS methods, only AANet on Tesla V100 can satisfy the requirement. However, our EASNet can provide a sub-network of competitive accuracy on all the three devices, i.e., 0.72 EPE with 0.09 s on Tesla V100, 0.73 EPE with 0.1 s on GTX 2070, and 0.86 EPE with 0.1 s on Tesla P40. This proves the flexibility and efficiency of our EASNet.

Benchmark Results on Scene Flow. For the rest of experiments, we pick three sub-networks from the full EASNet, EASNet-L (K=7, W=8, D=4, S=4), EASNet-M (K=7, W=8, D=2, S=4), and EASNet-S (K=3, W=2, D=2, S=2).

We compare our EASNet networks with five SOTA methods, including three hand-crafted and two NAS-based networks on Scene Flow [29] test set with 192 disparity level. In Table 1, we can observe that EASNet-M achieves the best performance using only near half of parameters in comparison to the SOTA hand-crafted methods (e.g., GANet [45]). Furthermore, the previous SOTA NAS-based method AutoDispNet [36] has 10× more parameters than our EASNet-M. Our smallest sub-network EASNet-S can still achieve much better accuracy than AutoDispNet and comparable accuracy to AANet with much fewer parameters and faster inference speed. As for the model runtime, EASNet-L outperforms the accuracy SOTA, GANet [45] and LEAStereo [10] by 3× and 22× respectively. EASNet-M achieves Pareto optimality in both accuracy and speed among all the methods. We show some of the qualitative results in Fig. 4. Our EASNet outperforms AutoDispNet in terms of estimation quality and achieves competitive accuracy to LEAStereo with only one third of inference time on P40.

Benchmark Results on Sintel and KITTI. We evaluate the model generalization of EASNet on the other two datasets, MPI Sintel and KITTI. Table 2 shows the results. Sintel is tested without any model finetune. EASNet-L achieves a much lower EPE than DispNet-CSS and AutoDispNet-CSS. Besides, after being finetuned on the KITTI training data, EASNet still shows satisfying accuracy with the state of the art on the common public benchmarks. EASNet-L

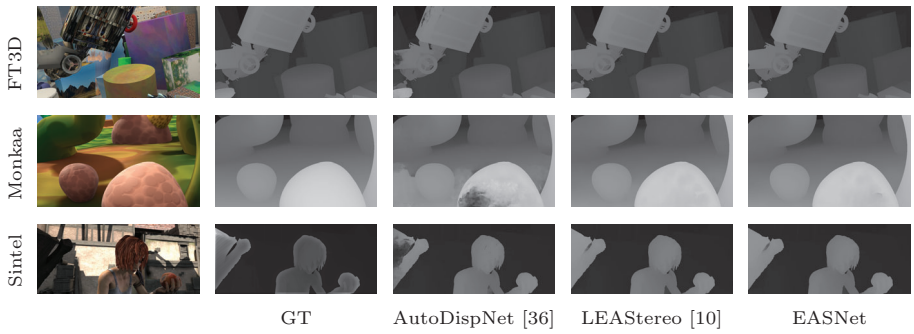

Fig. 4: Disparity predictions for the testing data of FlyingThings3D (FT3D), Monkaa and MPI Sintel. The leftest column shows the left images of the stereo pairs. The rest four columns respectively show the disparity maps estimated by (a) ground truth, (b) AutoDispNet [36], (c) LEAStereo [10], and (d) our EASNet.

Table 2: Quantitative results on other stereo datasets. Entries enclosed by parentheses indicate if they were tested on the target dataset without model finetuning. "DN-CSS" is short for DispNet-CSS. "ADN-CSS" is short for AutoDispNet-CSS. The time is measured on Nvidia Tesla V100 for KITTI resolution.

Method	**Sintel** (clean)	**KITTI** (2012)		**KITTI** (2015)		**Time** [s]
	EPE *train*	EPE *train*	Out-noc *test*	EPE *train*	D1-all *test*	
ADN-CSS [36]	(2.14)	(**0.93**)	1.70%	(**1.14**)	2.18%	0.34
GCNet [24]	-	-	1.77%	-	2.87%	0.9
GANet [45]	-	-	1.19%	-	1.81%	1.9
AANet [43]	-	-	1.91%	-	2.55%	0.07
LEAStereo [10]	-	-	**1.13%**	-	**1.65%**	0.3
DeepPruner (best) [13]	-	-	-	-	2.15%	0.18
$\leq$ 100 ms						
DN-CSS [22]	(2.33)	(1.40)	1.82%	(1.37)	2.19%	0.08
DeepPruner (fast) [13]	-	-	-	-	2.59%	0.06
MADNet [41]	-	-	-	-	4.66%	**0.02**
EASNet-L	(**1.58**)	(1.91)	1.89%	(1.90)	2.70%	0.09
EASNet-M	(1.59)	(1.91)	1.96%	(2.18)	2.89%	0.08
EASNet-S	(1.95)	(2.44)	2.57%	(2.32)	3.43%	0.06

achieves the best or second best accuracy among the methods of less than 100 ms. EASNet-S also achieves competitive accuracy with much lower latency. We show some of the qualitative results in Fig. 5. Our EASNet is able to capture the disparity information of those thin objects, such as street light and road fence.

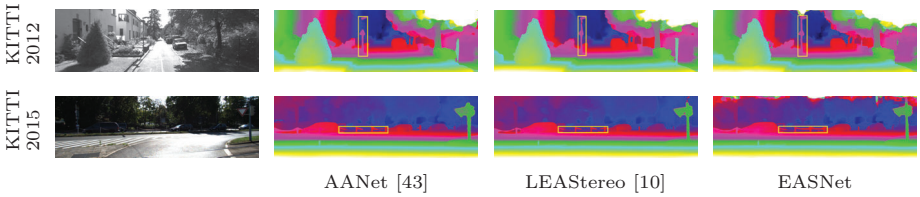

Fig. 5: Disparity predictions for KITTI 2012 and 2015 testing data. The leftest column shows the left images of the stereo pairs. The rest three columns show the disparity maps estimated by existing methods and our EASNet.

4.3 Discussion

There are several hints from the experiments. 1. The scale of the feature extraction module does not need to be large to achieve a good performance, due to the fact that the sub-network of (S=2) has similar accuracy to that of (S=4) with our training strategy; 2. The inference time drop of EASNet-S mainly comes from shrinking the feature extraction units and the whole scale (nearly 58% of overhead); 3. It is observed that skipping the first refinement module can further improve the inference speed of EASNet-S by 40% but increase the EPE from 0.86 to 1.19. Notice that we yet do not explore network search techniques for the refinement module. This indicates that our EASNet still has great potentials for deriving smaller sub-networks with the consistent accuracy.

5 Conclusion and Future Work

In this paper, we proposed EASNet, an elastic and accurate stereo matching network that leverages the domain knowledge of the CVM-Conv3D methods to design a specialized search space covering enormous architecture settings. To efficiently train EASNet with the target of maximizing the accuracy of all the sub-networks, we use the progressive shrinking strategy to support the specialized network search space of four dimensions, including depth, width, kernel size, scale and refinement. Superior to the previous studies that design and train a neural network for each deployment scenario, our EASNet can quickly generate the sub-networks that satisfy the deployment requirement of accuracy and latency. Validated on public benchmarks among three devices of different computing capability, our EASNet achieves Pareto optimality in terms of model accuracy and inference speed among all state-of-the-art CVA-3D deep stereo matching architectures (human designed and NAS searched).

In the future, we plan to apply NAS to search elastic and efficient units for cost aggregation and disparity refinement, which owns great potential for deriving smaller sub-networks with the consistent accuracy. Furthermore, how to combine the network search strategies of DARTS (mainly for operator and cell link) and OFA (mainly for cell/layer/block hyper-parameter) is also an in-

teresting and potential direction of searching an efficient and effective network structure for stereo matching.

Acknowledgements

This work was supported in part by the Hong Kong RGC GRF grant under the contract HKBU 12200418, grant RMGS2019_1_23 and grant RMGS21EG01 from Hong Kong Research Matching Grant Scheme, the NVIDIA Academic Hardware Grant, and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

References

1. Bello, I., Zoph, B., Vasudevan, V., Le, Q.V.: Neural optimizer search with reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 459–468 (06–11 Aug 2017)
2. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European conference on computer vision. pp. 611–625. Springer (2012)
3. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J.: Reinforcement learning for architecture search by network transformation. Proceedings of the AAAI Conference on Artificial Intelligence **4** (2017)
4. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J.: Efficient architecture search by network transformation. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (Apr 2018)
5. Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once for all: Train one network and specialize it for efficient deployment. In: International Conference on Learning Representations (2020)
6. Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware (2018)
7. Chabra, R., Straub, J., Sweeney, C., Newcombe, R., Fuchs, H.: Stereodrnet: Dilated residual stereonet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
8. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
9. Chen, W., Gong, X., Liu, X., Zhang, Q., Li, Y., Wang, Z.: Fasterseg: Searching for faster real-time semantic segmentation (2019)
10. Cheng, X., Zhong, Y., Harandi, M., Dai, Y., Chang, X., Li, H., Drummond, T., Ge, Z.: Hierarchical neural architecture search for deep stereo matching. Advances in Neural Information Processing Systems **33** (2020)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
12. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)

13. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4383–4392 (2019)
14. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey **20**(1), 1997–2017 (Jan 2019)
15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
16. Hao, C., Zhang, X., Li, Y., Huang, S., Xiong, J., Rupnow, K., Hwu, W., Chen, D.: Fpga/dnn co-design: An efficient design methodology for 1ot intelligence on the edge. In: 2019 56th ACM/IEEE Design Automation Conference (DAC). pp. 1–6 (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
18. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems **212**, 106622 (2021)
19. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence **30**(2), 328–341 (2007)
20. Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification (2017)
21. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
22. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In: The European Conference on Computer Vision (ECCV) (September 2018)
23. Jiang, W., Zhang, X., Sha, E.H.M., Yang, L., Zhuge, Q., Shi, Y., Hu, J.: Accuracy vs. efficiency: Achieving both through fpga-implementation aware neural architecture search. In: Proceedings of the 56th Annual Design Automation Conference 2019. DAC '19 (2019)
24. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 66–75 (2017)
25. Lin, J., Rao, Y., Lu, J., Zhou, J.: Runtime neural pruning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 2178–2188 (2017)
26. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
27. Liu, H., Simonyan, K., Vinyals, O., Fernando, C., Kavukcuoglu, K.: Hierarchical representations for efficient architecture search (2017)
28. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search (2018)
29. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4040–4048 (2016)

30. Menze, M., Heipke, C., Geiger, A.: Joint 3d estimation of vehicles and scene flow. In: ISPRS Workshop on Image Sequence Analysis (ISA) (2015)
31. Nekrasov, V., Chen, H., Shen, C., Reid, I.: Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
32. Nie, G.Y., Cheng, M.M., Liu, Y., Liang, Z., Fan, D.P., Liu, Y., Wang, Y.: Multi-level context ultra-aggregation for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3283–3291 (2019)
33. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2017)
34. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 4780–4789 (Jul 2019)
35. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 2902–2911 (06–11 Aug 2017)
36. Saikia, T., Marrakchi, Y., Zela, A., Hutter, F., Brox, T.: Autodispnet: Improving disparity estimation with automl. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
37. Stanley, K.O., Miikkulainen, R.: Evolving Neural Networks through Augmenting Topologies. Evolutionary Computation **10**(2), 99–127 (06 2002)
38. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
39. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
40. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
41. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 195–204 (2019)
42. Wang, Q., Shi, S., Zheng, S., Zhao, K., Chu, X.: FADNet: A fast and accurate network for disparity estimation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA 2020). pp. 101–107 (2020)
43. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1959–1968 (2020)
44. Zbontar, J., LeCun, Y., et al.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17**(1-32), 2 (2016)
45. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

46. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)