UniNet: Unified Architecture Search with Convolution, Transformer, and MLP

Anonymous authors
Paper under double-blind review

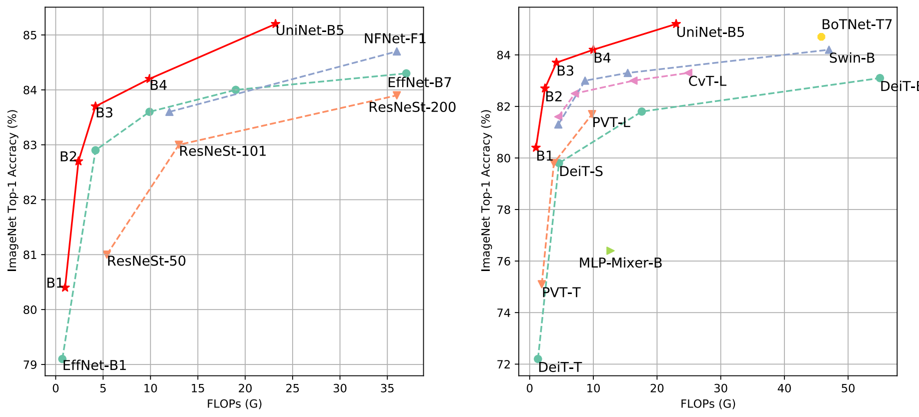
Abstract

Recently, transformer and multi-layer perceptron (MLP) architectures have achieved impressive results on various vision tasks. A few works investigated manually combining those operators to design visual network architectures, and can achieve satisfactory performances to some extent. In this paper, we propose to jointly search the optimal combination of convolution, transformer, and MLP for building a series of all-operator network architectures with high performances on visual tasks. We empirically identify that the widely-used strided convolution or pooling based down-sampling modules become the performance bottlenecks when the operators are combined to form a network. To better tackle the global context captured by the transformer and MLP operators, we propose two novel context-aware down-sampling modules, which can better adapt to the global information encoded by transformer and MLP operators. To this end, we jointly search all operators and down-sampling modules in a unified search space. Notably, Our searched network UniNet (Unified Network) outperforms state-of-the-art pure convolution-based architecture, EfficientNet, and pure transformer-based architecture, Swin-Transformer, on multiple public visual benchmarks, ImageNet classification, COCO object detection, and ADE20K semantic segmentation.

1 Introduction

Convolutional Neural Networks (CNN) dominate the learning of visual representations and show effectiveness on various downstream tasks, including image classification, object detection, semantic segmentation, etc. Recently, convolution-free backbones show impressive performances on image classification (Deng et al., 2009). Vision Transformer (ViT) (Dosovitskiy et al., 2020) firstly shows that pure Transformer architecture can attain state-of-the-art performance when trained on large-scale datasets (e.g. ImageNet-21k, JFT-300M). Data-efficient image Transformers (DeiT) (Touvron et al., 2021b) is competitive with CNNs when trained with ImageNet only. MLP-Mixer (Tolstikhin et al., 2021) introduced a pure multi-layer perceptron (MLP) architecture that can almost match ViT's performance without using the time-consuming attention mechanism.

However, the recent transformer or mixer-based architectures are still manually designed with human experience. The different operators (convolution, attention, MLP-mixer, etc.) have different properties, and how to properly assemble them to form networks is still under investigation. The underlying optimal architectures might be quite different for different dataset sizes and computational budgets. On the one hand, convolutions in CNNs are locally connected and their weights are input-independent, which makes it effective at extracting low-level representations and efficient under the low-data regime. On the other hand, the attention operations in the Transformer capture long-range dependency, and the attention weights are dynamically dependent on the input representations. Hence, it requires a significant amount of data and computational resources. There were recent papers on attempting to manually combine the different types of operators (d'Ascoli et al., 2021; Wu et al., 2021) to form hybrid convolution-transformer visual networks, which, however, lack proper design principles and did not show promising performance. We conducted a pilot study on attempting to directly stack different operators to form networks. As shown in Table 1, however, the straightforward stacking of different operators achieve even worse result than the original ViT with the only self-attention operator.



(a) Comparison with ConvNet (b) Comparison with transformer or hybrid

Figure 1: **ImageNet top-1 accuracy vs FLOPs.** All models are trained with ImageNet-1k dataset. Our UniNet-B5 achieve 85.2% with 23.2G FLOPs, outperforming NFNet-F1 and BoTNet-T7 with 35% and 49% fewer FLOPs respectively.

Model	Configuration	#Params (M)	#FLOPs (G)	Top-1 Acc.
ViT	12 T	22	4.6	**78.0**
MLP-Mixer	18 M	23	4.7	76.8
ViT-MLP	7 T + 7 M	22	4.5	76.5
MLP-ViT	7 M + 7 T	22	4.5	77.8

Table 1: ImageNet top-1 accuracy of different operator combinations. T and M refer to transformer block and mlp-mixer block respectively. Different block numbers are used to keep them comparable.

In this paper, we search the combination of convolution, transformer, and MLP operators, trying to assemble those operators to create novel and high-performance visual network architectures. To the best of our knowledge, we are the first to automatically search the assembly of all types of operators to form novel network architectures. As different operators have distinct characteristics, it is non-trivial to merge them into a super-net that can achieve superior performance.

In addition, we find that the widely used down-sampling modules, such as strided convolution or max pooling, actually become the bottlenecks that hinder the searched architectures to achieve optimal performance. This might be derived from the fact that the different operations searched in our architectures, have distinct characteristics. One of the biggest differences is the receptive field. Convolution has local receptive field and is effective at capturing local features, which welly preserve the spatial structure after transformation. However, attention and MLP-mixer have global receptive field, and each pixel in the output feature map is a weighted sum of all spatial locations, which may destroy the spatial structure. Therefore, the locally-correlated down-sampling module is no longer suitable. To address the above issue, we propose three types of down-sampling modules (DSM), which satisfy different operation combinations. Besides the traditional Local-DSM (L-DSM) that are modeled as strided convolution, we propose Local-Global-DSM (LG-DSM) and Global-DSM (G-DSM) to conduct down-sample based on global context. Both LG-DSM and G-DSM use attention mechanisms to gather global features, but the ways to generate query features for calculating the attention weights are different to capture different characteristics of the context.

We jointly search the operation combination and down-sampling module and network size in a unified search space. We propose a novel scheme to automatically search both the building blocks and the proposed down-sampling modules in a joint manner. To find the optimal architecture, we use Reinforcement Learning approach to optimize accuracy and computation cost simultaneously. Our searched architecture, named UniNet (Unified Network), achieves strong results on various vision tasks. Our UniNet architectures achieve better performance than state-of-the-art pure convolution architecture, EfficientNet, and pure transformer architecture, Swin Transformer. For instance, on ImageNet, our UniNet-B3 is able to achieve 83.7% and 85.2% top-1 accuracy with 4.2G and 23.2G FLOPs respectively. On COCO (Lin et al., 2014), UniNet-B3 achieves 47.9 mAP with 50M parameters, which is also better than the recent local transformer-based architecture Swin-T.

In summary, our contributions are as follows:

1. We are the first to jointly search the optimal combination of convolution, transformer, and MLP to identify high-performance visual neural networks.
2. We reveal the traditional down-sampling module becomes the bottleneck of performance when combining different types of operators. We propose context-aware down-sampling modules, and search them with general processing operations in a joint manner.
3. Our searched hybrid architecture, UniNet, outperforms previous pure convolution architectures EfficientNet and pure transformer architecture Swin Transformer.

2 Related Works

Convolution, Transformer, and MLP. A host of ConvNets have been proposed to push forward the state-of-the-art computer vision approaches such as (He et al., 2016; Szegedy et al., 2015; Tan & Le, 2019). Despite the numerous CNN models, their basic operations, convolution, are the same. Recently, Dosovitskiy et al. (2020) proposed a pure transformer based image classification model ViT, which achieves impressive performance on ImageNet benchmark. DeiT (Touvron et al., 2021b) reveals that well-trained ViT can obtain a better performance-speed trade-off than ConvNets. PVT (Wang et al., 2021b) proposes a pyramid vision transformer, which can be easily transferred to other downstream tasks. On the other hand, recent papers are attempting to use only MLP as the building block. MLP-Mixer (Tolstikhin et al., 2021) and ResMLP (Touvron et al., 2021a) show that a pure MLP architecture can also achieve near state-of-the-art performance.

Combination of different operators. Besides, another line of works tries to combine different operators to form a new network. CvT (Wu et al., 2021) propose to incorporate self-attention and convolution by generating `Q`, `K`, and `V` in self-attention with convolution. CeiT (Yuan et al., 2021) replace the original patchy stem with convolutional stem and add depthwise convolution to FFN layer, which obtains fast convergence and better performance. ConViT (d'Ascoli et al., 2021) tries to unify convolution and self-attention with gated positional self-attention and is more sample-efficient than self-attention. Previous works use a different form to combine those operators and get promising results, but requires tedious manual design, lacking effective guidelines. In this work, we propose a unified search space, in which we can search the combination of different operations automatically.

Down-sampling module. In ConvNets, the down-sampling module (DSM) is implemented with strided-Conv or pooling. As DSM breaks the shift invariant of convolution, Zhang (2019) propose anti-aliased DSM to keep it. Besides, a line of works tries to preserve more information when down-sampling with a learnable or dynamic kernel (Gao et al., 2019; Saeedan et al., 2018; Wang et al., 2021a). Most of their approaches are down-sampling based on local context, which we show is not suitable for our unified network. In our work, we propose context-aware DSM and jointly search with operation combinations, which guarantees better performance.

3 Method

3.1 Unified Architecture Search

As discussed in previous works (d'Ascoli et al., 2021), an appropriate combination of convolution and transformer operators can lead to performance improvements. However, the previous approaches (Wu et al., 2021; Yuan et al., 2021) only adopt convolution in self-attention or feed-forward network (FFN) sub-layers and stack them repeatedly. Their approaches did not fully explore their combinations to take advantage of their different characteristics. Besides, effective guidelines of properly assembling those operators are missing in prior works, which is important for designing new architectures.

Prior art (Wang et al., 2021a; Zhang, 2019) show that the down-sampling module plays an important role in visual tasks. Most previous approaches adopt hand-crafted downsampling operations, i.e. strided convolution, max-pooling, or avg-pooling, to down-sample the feature map based on only the local context. However, these operations are specifically designed for ConvNets, and might not be suitable to the transformer or MLP based architectures, which capture representation globally.

In this paper, we jointly search the combination of convolution, transformer, and MLP operators, trying to assemble the operators to create novel and high-performance visual network architectures.

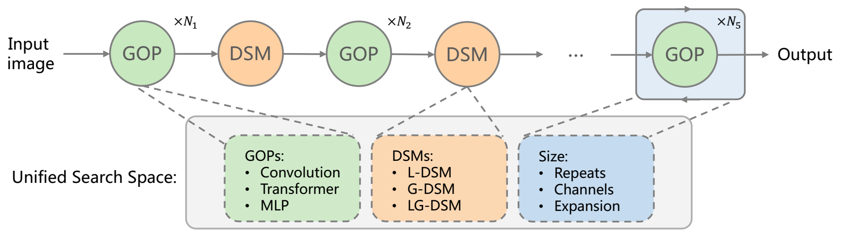


Figure 2: **Unified Architecture Search.** We jointly search the all operators and down-sampling modules and network size in a unified search space.

For better transform features across different operator blocks, we proposed context-aware down-sampling modules. We jointly search the operators, down-sampling modules, and network size in a unified search space. In contrast, previous Neural Architecture Search (NAS) works achieved state-of-the-art performances mainly via searching the network sizes. We show that our proposed unified architecture search can achieve very promising performance with optimal all-operator network architectures.

In the remaining parts of the section, we firstly present how to properly combine different operators into a unified search space and search them jointly. We then present the challenge of incorporating down-sampling modules with different operators, and present our proposed context-aware down-sampling module. Finally, we will introduce our UniNet architecture and NAS pipeline.

3.2 Modeling Convolution, Attention, MLP with a Unified Searchable Form

Recently, transformer and MLP based architectures are able to achieve comparable performance with convolution-based ones on general visual tasks. To achieve better performance, it is intuitive to assemble all the types of operators to build high-performance all-operator networks. Actually, a few works (Wu et al., 2021; Yuan et al., 2021; d'Ascoli et al., 2021) have been studied to empirically combine convolution and self-attention. However, manually search of network architectures is quite time-consuming and cannot ensure optimal performances with different computational budgets.

We introduce a unified search space that contains General Operators (GOPs, including convolution, transformer, and MLP), and then search for the optimal combination of those operators jointly. Compared with the prior art, we use a unified form to characterize each operator. Specifically, we use inverted residual (Sandler et al., 2018) to model a general operator block, which first expands the input channel $\mathtt{c}$ to a larger size $\mathtt{ec}$, and later projects the $\mathtt{ec}$ channels back to $\mathtt{c}$ for residual connection. The $\mathtt{e}$ is denoted as the expansion ratio, which is usually a small integer number, e.g., 4. The general operator block is modeled as

$$\mathrm{y} = \mathrm{x} + \mathtt{Operation}(\mathrm{x}), \tag{1}$$

where $\mathtt{Operation}$ can be convolution, MLP, or self-attention operators, and x, y represent input and output features, respectively. For convolution, we place the convolution operation inside the bottleneck (Sandler et al., 2018), which can be expressed as

$$\mathtt{Operation}(\mathrm{x}) = \mathtt{Proj}_{\mathtt{ec}\to\mathtt{c}}(\mathtt{Conv}(\mathtt{Proj}_{\mathtt{c}\to\mathtt{ec}}(\mathrm{x}))). \tag{2}$$

The $\mathtt{Conv}$ operation can be either regular convolution or depth-wise convolution ($\mathtt{DWConv}$) (Chollet, 2017), and the $\mathtt{Proj}$ represents a linear projection. For self-attention and MLP, operating on the large bottleneck feature map can be quilt slow. Following previous works (Dosovitskiy et al., 2020; Tolstikhin et al., 2021), we separate them from the bottleneck for computation efficiency, and the $\mathtt{Proj}$ is implemented inside the FFN (Vaswani et al., 2017) sub-layer. Each transformer block has a query-key-value attention sub-layer and an FFN sub-layer. For MLP block, we implement with transpose-FFN-transpose like Tolstikhin et al. (2021).

$$\mathrm{y} = \mathrm{y}' + \mathtt{FFN}(\mathrm{y}'), \tag{3}$$

$$\text{where} \quad \mathrm{y}' = \mathrm{x} + \mathtt{SA}(\mathrm{x}) \text{ or } \mathrm{x} + \mathtt{MLP}(\mathrm{x}),\ \mathtt{FFN}(\mathrm{y}') = \mathtt{Proj}_{\mathtt{ec}\to\mathtt{c}}(\mathtt{Proj}_{\mathtt{c}\to\mathtt{ec}}(\mathrm{y}')), \tag{4}$$

where $\mathtt{SA}$ can be either vanilla self-attention or local self-attention $\mathtt{LSA}$, and $\mathtt{MLP}$ refers to token-mixing operation.

There are two main advantages of representing the different operators in a unified format and search space: (1) We can characterize each operator with the same set of configuration hyper-parameters except for the operation type, e.g. expansion rate and channel size. As a result, the search space for the operator is much simplified, which can speed up the search process. (2) Under the same network size configuration, each operator block has a similar computation cost. The comparison between different operator combinations is fairer, which is extremely important for NAS (Tan et al., 2019).

3.3 CONTEXT-AWARE DOWN-SAMPLING MODULES

As discussed in Section 3.1, the down-sampling module (DSM) plays an important role in visual tasks. In addition to hand-crafted DSM (i.e. max-pooling or avg-pooling), a few works (Saeedan et al., 2018; Gao et al., 2019; Wang et al., 2021a) tried to preserve more information when down-sampling with learnable or dynamic kernel. Most of their approaches are down-sampling based on local context, which suits for ConvNets well. However, in our unified search space, operators with different receptive filed can be assembled unrestrictedly to form a novel architecture, where the local context may be destroyed and therefore the previous approaches will fail.

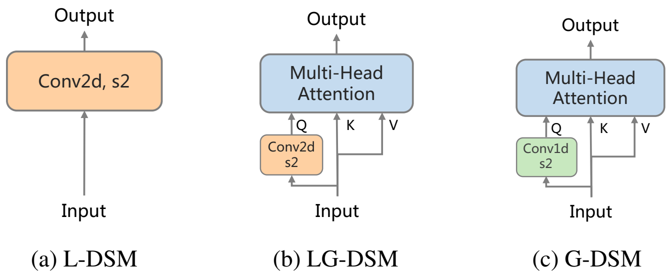


Figure 3: Structures of the context-aware down-sampling modules.

In this paper, we propose context-aware DSM, which is instanced with L-DSM, LG-DSM, and G-DSM. The main difference between those DSMs is the context used when down-sampling. For L-DSM, only local context is involved, which fits ConvNets well as shown in previous works. For G-DSM, only global context is used to down-sample, which may fit other operations, e.g., transformer. The LG-DSM combines the characteristics of L-DSM and G-DSM. It uses both local and global context to downsample. Our intuition is that one of the largest dissimilarities of different operators is the receptive field. Transformer and MLP naturally have global receptive filed, while convolution has local receptive field, e.g., 3×3. When combining those operators, there is no single optimal DSM that satisfies all scenarios.

The proposed DSMs are visualized in Figure 3. To down-sample based on global cues, we utilize the self-attention mechanism to capture global context, which is missed by the prior art. However, vanilla self-attention directly applies a linear transformation on input `x`, which is not applicable to down-sample a feature map. To address this issue, we replace the original linear transformation of a query with down-sampling operations. Specifically, for G-DSM, we use `Conv1D` with stride 2 to down-sample the query. To note that, there is no direct local context preserved after the transformation of G-DSM. For LG-DSM, we first reshape the flattened token sequences back to the spatial grid and apply `Conv2D` with stride 2 to down-sample the query, and then flatten the query back to calculate the attention weights.

Compared with previous works, which mainly try to improve ConvNets, our proposed DSMs are not designed for a specific architecture. Our motivation is that the optimal DSM is not fixed for different operators. For example, the optimal DSM may be L-DSM for ConvNets, but G-DSM for transformer. As thousands of operator combinations will be trained in our NAS process, it is unfeasible to decide which DSM to use by hand. To obtain the optimal architecture, we jointly search DSMs with the General Operators. An interesting result in our experiments is that our searched DSM matches our assumption. In our searched optimal architecture, L-DSM is used between operators with local receptive field while LG-DSM is favored by operators with global receptive field.

3.4 UNINET ARCHITECTURE

As shown in recent studies, combining different operators (Wu et al., 2021; Yuan et al., 2021) can bring performance improvements. Most previous approaches only repeatedly stack the same operation in the whole architecture and use different channels in different stages. These approaches don't permit stage diversity, which we show is crucial for achieving high accuracy.

On the contrary, in our UniNet, the operators are not fixed but searched from the unified search space. We construct our UniNet architecture in a multi-stage fashion, which can be easily transferred to downstream tasks. Between two successive stages, one of our proposed DSMs will be inserted to reduce the spatial dimension. We jointly search the GOP and DSM per stage separately. The GOP could be different for different stages but repeated in one stage, which can greatly reduce the search space size as pointed out before (Tan et al., 2019). The overall architecture and search space are shown in Figure 2.

Thanks to our unified form of GOPs, the network size of each stage can be configured with repeat number `r`, channel size `c`, and expansion ratio `e`. To obtain better computation-accuracy trade-off, we jointly search the network size with the GOP and DSM. For GOP, we search for convolution, transformer, MLP and their promising variants, i.e., {`SA`, `LSA`, `Conv`, `DWConv`, `MLP`}, as mentioned in Section 3.2; for `e`, we search for {2, 3, 4, 5, 6}. Following previous work, we search the network size based on a given architecture, e.g. EfficientNet (Tan & Le, 2019), and the channels and repeats will be changed according to it. For `c` and `r`, we search for {0.5, 0.75, 1.0, 1.25, 1.5} and {-2, -1, 0, 1, 2} respectively. Suppose we partition the network into K stages, and each stage has a sub search space of size S. Then the total search space is S^N. In our implementation, K is set to 5 and S equals 125. As a result, our search space size is about 10^{16}.

3.5 Search Algorithm

We utilize Reinforcement Learning to search for the optimal architecture automatically. Concretely, we follow previous work (Liu et al., 2021a) and map an architecture in the search space to a list of tokens, which are determined by a sequence of actions generated by an RNN. The RNN is optimized using PPO algorithm (Schulman et al., 2017) by maximizing the expected reward. In our implementation, we simultaneously optimize accuracy and the theoretical computation cost (FLOPs). To handle the multi-objective optimization problem, we use a weighted product customized as Tan et al. (2019) to approximate Pareto optimal. For one sampled architecture m, the reward is formulated as $r(m) = a(m) \times \frac{t}{f(m)}^{\alpha}$, where function $a(m)$ and $f(m)$ return the accuracy and the FLOPs of m, t is the target FLOPs, and α is a weight factor that balances the accuracy and computation cost.

During the search process, thousands of combinations of GOPs and DSMs are trained on a proxy task with the same setting, which gives us a fair comparison between those combinations. When the RNN converges, the top-k architectures with the highest reward will be trained with full setting, and the top-performing one will be kept for model scaling and transferring to other downstream tasks.

4 Experimental Setup

To find the optimal architecture in our search space, we directly search on a large dataset, ImageNet. We reserve 50K images from the training set as a validation set. For each sampled architecture, we train it for 5 epochs. After that, we calculate the reward of the architecture with its FLOPs and the accuracy on the validation set. We set the target FLOPs t and weight factor α in reward function to 550M and -0.07 respectively. During the search process, totally 2K models are trained on the proxy task. After that, we fully train the top 5 architectures on ImageNet and preserve the top-performing one for model scaling and transferring to other downstream tasks.

For regular ImageNet training, we mostly follow the training strategy in DeiT (Touvron et al., 2021b), except that we use small augmentation for small models and heave augmentation for large models as shown by Steiner et al. (2021). When the input resolution exceeds 256×256, we train our models with 256×256 and finetune on the large resolution for training efficiency. Besides, we also transfer UniNet to other downstream tasks, e.g., object detection on COCO and semantic segmentation on ADE20K. For COCO training, we use the typical framework Mask R-CNN and train on wildly-used 1x (12 epochs) and 3x (36 epochs) schedules. For ADE20K training, we use the UperNet framework and train with the same setting as Liu et al. (2021b).

5 Main Results

In this section, we firstly present our searched UniNet architecture. We then show the performance of the scaled UniNets on classification, object detection, and semantic segmentation.

Model	Family	Input Size	#FLOPs (G)	#Params (M)	Top-1 Acc.
EffNet-B0 (Tan & Le, 2019)	C	224	0.39	5.3	77.1
EffNetV2-B0‡ (Tan & Le, 2021)	C	240	0.7	7.4	78.7
DeiT-Tiny (Touvron et al., 2021b)	T	224	1.3	5.7	72.2
PVT-Tiny (Wang et al., 2021b)	T	224	1.9	13.2	75.1
ConViT-Ti+ (d'Ascoli et al., 2021)	H	224	2	10	76.7
UniNet-B0	H	160	0.56	11.9	79.1
EffNet-B2 (Tan & Le, 2019)	C	260	1	9.2	80.1
EffNetV2-B1‡ (Tan & Le, 2021)	C	260	1.2	8.1	79.8
RegNetY-4G (Radosavovic et al., 2020)	C	224	4	20.6	80
DeiT-Small (Touvron et al., 2021b)	T	224	4.3	22	79.8
PVT-Small (Wang et al., 2021b)	T	224	3.8	24.5	79.8
UniNet-B1	H	192	0.99	14	80.4
EffNet-B3 (Tan & Le, 2019)	C	300	1.8	12	81.6
EffNetV2-B3‡ (Tan & Le, 2021)	C	300	3	14	82.1
Swin-T (Liu et al., 2021b)	T	224	4.5	29	81.3
CvT-13-NAS (Wu et al., 2021)	H	224	4.1	18	82.2
ConViT-B+ (d'Ascoli et al., 2021)	H	224	30	152	82.5
UniNet-B2	H	224	2.4	22.5	82.7
EffNet-B4 (Tan & Le, 2019)	C	380	4.2	19	82.9
NFNet-F0 (Brock et al., 2021)	C	256	12.4	71.5	83.6
Swin-B (Liu et al., 2021b)	T	224	15.4	88	83.5
CvT-21 (Wu et al., 2021)	H	384	24.9	32	83.3
UniNet-B3	H	256	4.2	31.8	83.7
UniNet-B4	H	256	9.9	73.5	84.2
EffNet-B7 (Tan & Le, 2019)	C	600	37	66	84.3
EffNetV2-M‡ (Tan & Le, 2021)	C	480	24	54	85.1
NFNet-F2 (Brock et al., 2021)	C	352	62.6	193.8	85.1
BoTNet-T7 (Srinivas et al., 2021)	T	384	45.8	75.1	84.7
UniNet-B5	H	384	23.2	73.5	85.2

Table 2: UniNet performance on ImageNet. All UniNet models are trained with ImageNet dataset with 1.28m images. C, T, and H denotes convolution, transformer, and hybrid architecture respectively. ‡: all EfficientNetV2 models are trained with progressive learning.

5.1 The Searched UniNet

Table 3 shows our searched UniNet-B0 architecture. Our searched architecture has the following characteristics: (1) UniNet uses DWConv at early stages, while uses transformer at later stages. (2) UniNet chooses L-DSM to down-sample in the DWConv stage, while uses LG-DSM for transformer. Surprisingly, the searched operator combination somewhat matches previous empirical finds (Xiao et al., 2021) or manual designed architecture (Gao et al., 2021), which show the effectiveness of our proposed unified architecture search.

Stage	Operator		Network Size		
	GOP	DSM	e	c	r
0	DWConv	L-DSM	4	48	2
1	DWConv	L-DSM	6	80	4
2	DWConv	L-DSM	3	128	4
3	Transformer	LG-DSM	2	128	4
4	Transformer	LG-DSM	5	256	8

Table 3: UniNet-B0 architecture. GOP and DSM represent General Operators and down-sampling module respectively. DWConv and Transformer are are described in Section 3.2.

Most previous transformer based architectures usually outperform convolution based architecture in high FLOPs region, but underperform in low FLOPs region. To prove the effectiveness of our searched UniNet, we scale up and down our search UniNet-B0. The FLOPs number varies from 0.56G to 23.2G, which covers both the mobile and large settings. We utilize the compound scaling (Tan & Le, 2019) to scale depth, width, and resolution simultaneously. An exception is that we increase image resolution more slowly, which is more efficient as Bello et al. (2021) shows.

5.2 ImageNet Classification Performance

Table 2 presents the performance comparison of our searched UniNet and other architectures, including convolution based, transformer based, and hybrid architectures. Our search UniNet has better accuracy and computation efficiency than previous ConvNets, transformer, or hybrid architectures.

As shown in Table 2, under mobile setting, our UniNet-B0 achieves 79.1% top-1 accuracy with 0.56G FLOPs, outperforming EfficientNetV2-B0 (Tan & Le, 2021) with less FLOPs. In the middle FLOPs region, our UniNet-B3 achieves 83.7% top-1 accuracy with 4.2G FLOPs, which outperforms pure convolution based EfficientNet-B4, pure transformer based Swin-B, and hybrid architecture CvT-21. For larger models, our UniNet-B5 achieve 85.2% with 23.2G FLOPs, outperforming NFNet-F2 and BoTNet-T7 with 63% and 49% fewer FLOPs respectively. Figure 1 further visualizes the comparison with on accuracy and FLOPs.

5.3 Object Detection and Semantic Segmentation Performance

Backbone	#Params (M)	Mask R-CNN 1x		Mask R-CNN 3x		UperNet
	Det/Seg	AP@box	AP@mask	AP@box	AP@mask	mIoU (%)
ResNet18	31/ -	34.0	31.2	36.9	33.6	-
ResNet50	44/ -	38.0	34.4	41.0	37.1	-
PVT-Tiny	33/ -	36.7	35.1	39.8	37.4	-
UniNet-B1	28/38	40.5	37.5	44.4	40.1	42.7
ResNet101	63/86	40.4	36.4	42.8	38.5	44.9
PVT-Small	44/ -	40.4	37.8	43.0	39.9	-
Swin-T	48/60	42.2	39.1	46.0	41.6	44.5
UniNet-B3	50/59	45.6	41.6	47.9	43.3	49.0

Table 4: Object detection, instance segmentation, and semantic segmentation performance on the COCO val2017 and ADE20K val set. All UniNet models are pretrained on the ImageNet-1K dataset.

For object detection and semantic segmentation, we pick UniNet-B1 and UniNet-B3 and use them as the feature extractors of detection and segmentation frameworks. We compare our UniNet with other convolution or transformer based architectures. For COCO object detection, we use Mask-RCNN framework and compare the performance under 1x and 3x schedules. For ADE20K semantic segmentation we use UperNet framework and report mIoU (%) for different architectures under same the training setting.

As shown in Table 4, our searched UniNet consistent outperforming convolution based ResNet (He et al., 2016) and transformer based PVT (Wang et al., 2021b) or Swin-Transformer (Liu et al., 2021b). UniNet-B1 achieves 40.5 AP@box, which is 3.8 points better than PVT-Tiny but with 15% fewer parameters. UniNet-B4 achieves 45.6 AP@box with 1x schedule and 47.9 AP@box with 3x schedule, which is 3.4 points and 1.9 points better than Swin-T respectively. For ADE20K semantic segmentation, we achieve 49% mIoU with 59M parameters. Compared with transformer based Swin-T, our UniNet outperforms 4.5% mIoU with similar parameters. Besides, compared with convolution based ResNet101, we get 4.1% higher mIoU with 31% fewer parameters. All the results show the generalization of our searched UniNet.

6 Ablative Studies and Analysis

In this section, we study the impact of joint search of General Operations, and discuss the importance of context-aware down-sampling modules (DSMs).

6.1 Single Operator vs. General Operators

Previous work (Tan et al., 2019) focus on the network size search, which uses single operator, convolution, as the main feature extractor. In comparison, we jointly search the combination of different General Operators (GOPs), i.e., convolution, transformer, MLP, and their promising variants. To verify the importance of GOPs, we remove MLP and transformer and their variants from our search

Model	#FLOPs (G)	#Params (M)	Top-1 Acc.
UniNet-B05	0.56	11.5	**79.1**
UniNet-B05 w/ Conv-Only	0.59	11.0	77.7

Table 5: Performance on ImageNet of UniNet with different search settings. Conv-Only represents search UniNet with convolution operator only.

space, and re-run the search experiment under the same setting. After search, we fully train the top-5 architectures with highest reward on ImageNet, and report the best performance.

As shown in Table 5, our joint search of GOPs outperforms Conv-Only search by a large margin. The result verifies the effectiveness of our joint search of GOPs, which can combine the characteristics of different operators. To note that, the DSMs are also searched in the Conv-Only search experiment. However, the top-performing architecture chooses L-DSM in all its stages.

6.2 Fixed vs. Context-Aware Down-Sampling Module

Model	#FLOPs (G)	#Params (M)	Top-1 Acc.
UniNet	0.56	11.5	**79.1**
w/ L-DSM	0.54	11.3	78.5
w/ G-DSM	0.77	12.7	76.8
w/ LG-DSM	0.72	14.1	78.9

Table 6: Performance on ImageNet of UniNet with different DSMs. To note that, the result of traditional strided-conv based down-sampling module is shown in row 2.

Model	#FLOPs (G)	#Params (M)	Top-1 Acc.
PVT-Tiny	1.9	13.2	75.1
w/ DSM	2.0	14.3	**77.5**
Swin-T	4.5	29.0	81.2
w/ DSM	4.7	30.0	**81.6**

Table 7: Performance comparison on ImageNet of different backbones when equipped with our proposed DSMs.

When combining different operators into a unified network, the traditional down-sampling module, which is mainly implemented with strided-conv or pooling, could be sub-optimal. To verify the effectiveness of our proposed context-aware DSMs, we replace the DSMs of our search UniNet with one fixed DSM, and compare their performance under the same training setting.

As shown in Table 6, our searched UniNet consistently outperforms its variants that use single fixed DSM in all stages. Although we see that using G-DSM or LG-DSM in all stages brings more computation and parameters, the performance does not become better. The result emphasizes the importance of our joint search of GOPs and DSMs.

Besides, we transfer our proposed DSMs to other popular transformer based architectures, e.g. Swin-Transformer (Liu et al., 2021b) and PVT (Wang et al., 2021b). Both Swin and PVT have 4 stages. Borrowing the result from our searched UniNet, we use L-DSM for the first two stages while LG-DSM for the latter two stages. As shown in Table 7, our proposed DSMs improve PVT-Tiny and Swin-T for 2.4% and 0.4% respectively. To note that, PVT uses a strided-conv to down-sample. As discussed in Section 3.3, it is harmful to the main operator in PVT, transformer, which has global receptive field. On the contrary, our proposed DSMs are able to down-sample based on both local and global context, and can greatly improve the performance.

7 Conclusion

In this paper, we propose to jointly search the combination of convolution, transformer, and MLP. We empirically identify that the widely-used strided convolution or pooling based down-sampling modules become the performance bottlenecks when the operators are combined to form a network. To further improve the performance, we propose context-aware down-sampling modules and jointly search them with all operators. Our searched UniNet outperforms state-of-the-art pure convolution-based architecture, EfficientNet, and pure transformer-based architecture, Swin-Transformer, on ImageNet classification, COCO object detection, and ADE20K semantic segmentation.

References

Irwan Bello, William Fedus, Xianzhi Du, Ekin D Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *arXiv preprint arXiv:2103.07579*, 2021.

Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*, 2021.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021.

Ziteng Gao, Limin Wang, and Gangshan Wu. Lip: Local importance-based pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3355–3364, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Jihao Liu, Ming Zhang, Yangting Sun, Boxiao Liu, Guanglu Song, Yu Liu, and Hongsheng Li. Fnas: Uncertainty-aware fast neural architecture search. *arXiv preprint arXiv:2105.11694*, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021b.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10428–10436, 2020.

Faraz Saeedan, Nicolas Weber, Michael Goesele, and Stefan Roth. Detail-preserving pooling in deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9108–9116, 2018.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16519–16529, 2021.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe++: Unified content-aware reassembly of features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021a.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021b.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.