Searching Efficient Neural Architecture with Multi-resolution Fusion Transformer for Appearance-based Gaze Estimation

Vikrant Nagpure
Honda Motor Co., Ltd.
Tokyo, Japan
vikrant_nagpure@jp.honda

Kenji Okuma
Honda Motor Co., Ltd.
Tokyo, Japan
kenji_okuma@jp.honda

Abstract

For aiming at a more accurate appearance-based gaze estimation, a series of recent works propose to use transformers or high-resolution networks in several ways which achieve state-of-the-art, but such works lack efficiency for real-time applications on edge computing devices. In this paper, we propose a compact model to precisely and efficiently solve gaze estimation. The proposed model includes 1) a Neural Architecture Search(NAS)-based multi-resolution feature extractor for extracting feature maps with global and local information which are essential for this task and 2) a novel multi-resolution fusion transformer as the gaze estimation head for efficiently estimating gaze values by fusing the extracted feature maps. We search our proposed model, called GazeNAS-ETH, on the ETH-XGaze dataset. We confirmed through experiments that GazeNAS-ETH achieved state-of-the-art on Gaze360, MPIIFaceGaze, RTGENE, and EYEDIAP datasets, while having only about 1M parameters and using only 0.28 GFLOPs, which is significantly less compared to previous state-of-the-art models, making it easier to deploy for real-time applications.

1. Introduction

Human gaze is an important indicator of human attention. A wide range of applications use eye-gaze estimation from monocular images, which attracts a significant interest in computer vision for understanding human cognition [33] and human behavior [15]. It is also commonly used in driver fatigue estimation [42, 21], human-computer interactions [46, 32], and virtual reality [31, 40]. The conventional model-based methods estimate the human gaze by building a geometric eye model [19].

In recent years, appearance-based gaze estimation methods, that directly learn a mapping function from human face expressions to the human gaze, have made significant progress. Since the face appearances vary a lot due to personal or environmental factors such as head poses and illuminations, the problem of an appearance-based gaze estimation has inevitable complications [7]. Thus, learned mapping functions should be highly non-linear to attend the whole appearance and catpure appearance-based complications. The recent development of convolutional neural network (CNN) based methods show convincing results [49, 8]. Meanwhile, several large scale datasets are prepared and made publicly available to facilitate the gaze estimation research [45, 48, 23, 18, 27].

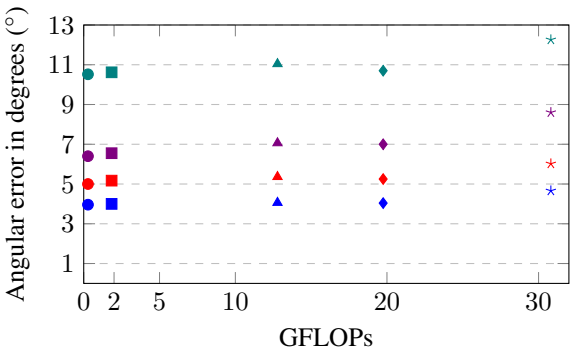

Figure 1. Comparisons of angular error and FLOPs of state-of-the-art methods on different gaze estimation datasets. Here blue is for MPIIFazeGaze, red is for EYEDIAP, violet is for RTGENE, green is for Gaze360, ○ is for **Our GazeNAS**, □ is for GazeTR, △ is for Gaze360, ◇ is for CADSE and ⋆ is for RTGENE. The optimal model should be near origin having both a low error and a small number of FLOPs

Recently, transformers, originally proposed by [37] for the natural language processing tasks, are used for the gaze estimation task. As transformers are capable of capturing the global context, their applications in computer vision tasks demonstrate an excellent performance. GazeTR [9] uses a hybrid ViT [17] for appearance-based gaze estimation tasks and achieves state-of-the-art results in several datasets. In [3], authors show the effectiveness of several HR-Net [38] based methods for the gaze estimation task.

The HR-Net and transformers based works are computationally expensive and hence are not feasible for the real time applications in practice. On the other hand, NAS based methods are quite popular due to their efficiency in other tasks such as object detection [1], segmentation [5] and human pose estimation [12]. Recently, in HR-NAS [16], they use HR-Net[38] based search space with light-weight transformers and achieve state-of-the-art results in several tasks with a reasonable computational cost.

For the real time applications of gaze estimation, we need a efficient and accurate neural architecture. With a recent development of NAS in other major tasks, extending its applicability to the problem of gaze estimation is the main topic of our work. In this paper, we propose to solve this problem by using a NAS based efficient feature extractor with powerful gaze estimation head. The proposed model includes 1) a NAS-based multi-resolution feature extractor for extracting feature maps with global and local information which are essential for this task and 2) a novel multi-resolution fusion transformer as the gaze estimation head for efficiently estimating gaze values by fusing the extracted feature maps.

Instead of searching models on every dataset of gaze estimation task, we propose to search a model on only one dataset then validate it on other datasets. This significantly reduces the training time. Our searched neural architecture, called GazeNAS-ETH, is searched on ETH-XGaze [45] which has a wide range of gaze values and head poses and a large size of dataset. Experiments show that the GazeNAS-ETH outperforms state-of-the-art methods on several gaze estimation benchmarks with the least computational budget. In fact, our GazeNAS-ETH requires only 1.027M parameters and 0.28GFLOPs enabling it to be used for real time applications.

Our **main contributions** are: (1) We are the first to employ and analyse NAS for the gaze estimation task. (2) We propose a novel multi-resolution fusion transformer based gaze regression head which is efficient as well as accurate to predict gaze values from multi-resolution features. (3) We propose to use ETH-XGaze [45] as the dataset for searching neural architectures for the gaze estimation tasks. (4) Extensive experiments show that our GazeNAS-ETH achieves state-of-the-art results while having a computationally efficient architecture for real time applications.

2. Related Work

2.1. Gaze Estimation

Recently, several CNN-based approaches are proposed with a significant performance improvement [47]. In [10] the authors explore the asymmetry between two eyes and propose asymmetric regression on a four-stream CNN to estimate gaze from eye images. The work in [29] proposes to estimate gaze from the pictorial representation of eye images. A dilated convolutional network to capture subtle changes in eye images is proposed in [6]. In [39], a CNN-based approach is used to align the feature extracted with adversarial learning and also incorporate bayesian inference for improving prediction accuracy.

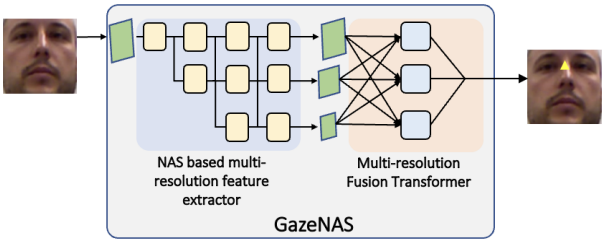

Figure 2. The proposed method called GazeNAS. We extract multi-resolution features from face image using multi-resolution feature extractor searched through NAS. These multi-resolution features are then fused to estimate gaze value using our multi-resolution fusion transformer.

A coarse-to-fine network to integrate face and eye images is proposed in [8], where a basic gaze is estimated from face images and refined with eye images. A recent use of HR-Net [38] achieves a competitive accuracy in [3]. Recently [28] introduced self-attention with convolution and de-convolution to solve low generalization problem of gaze estimation. However, all of these methods are not efficient enough for real time applications. Therefore, more efficient gaze estimation models are still required for the real time applications.

2.2. Transformers

Transformer is originally introduced by [37] for natural language processing (NLP) tasks. The transformer architecture contains only self-attention layers, layer normalization and multi-layer perceptron layers. Compared with recurrent networks, the self-attention layers have global computations and perfect memory to make transformers more suitable for long sequence tasks. The transformer-based methods are the current state-of-the-art methods for NLP tasks [14].

Transformers are recently quite popular in computer vision tasks as well. Recent works integrate CNNs with transformers to achieve a better performance in object detection and instance segmentation tasks [4] [11] [50]. The Vision Transformer (ViT) is proposed by [17], where they divide an image into non-overlapping patches and apply a conventional transformer architecture into these patches for image classification.

For the gaze estimation task, transformers are applied in GazeTR [9], where they apply ViT on the feature maps from a CNN and achieve state-of-the-art results effectively in various gaze estimation benchmarks.

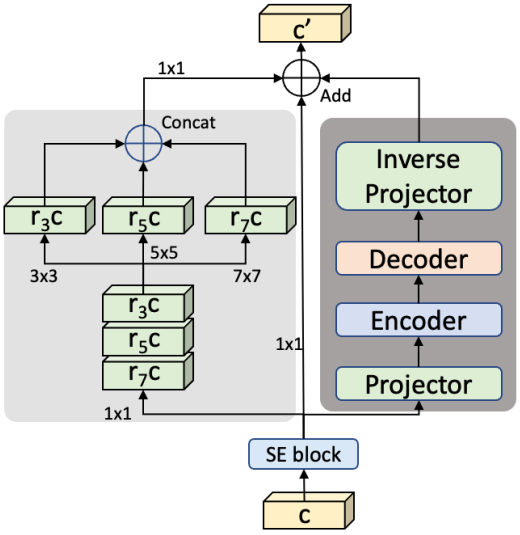

Figure 3. Architecture of search block containing a lightweight transformer path denoted by dark gray region, a MixConv path with 3×3, 5×5 and 7×7 kernels denoted by light grey region, and a residual connection

2.3. Neural Architecture Search for efficient models

For efficient neural architecture search, early approaches mainly used reinforcement learning [51] and evolution algorithms [34, 25]. Usually, these methods are computationally expensive. To improve the efficiency of the search process, differentiable search methods were proposed by works such as Darts [24, 22, 41] and ProxylessNAS [2]. Here, they formulate the search space as a super-graph where the adoption of an operator depends on the probability represented by a continuous importance weight, allowing to use gradient descent for an efficient search of the architecture. Due to the multi-scale feature modeling capability of mixed convolution [36, 26], it is also adopted in NAS search spaces. Recently, model expansion based methods are proposed to expand the search space from operators to other hyper-parameters such as input resolutions, channel numbers, and layer numbers [1, 43]. In order to search for efficient models, the existing methods usually borrow efficient operators from manually designed networks, such as depth-wise convolution and Inverted Residual Block[35]. Recently, HR-NAS [16] incorporated transformer into the search space to have more powerful operators and achieve state-of-the-art performance in various tasks.

3. Methodology

In order to solve gaze estimation both precisely and efficiently, we propose to use a NAS based efficient feature extractor with powerful gaze estimation head as shown in Fig. 2. Inspired by successful applications of high-resolution network[38] in [3] for gaze estimation, our feature extractor is a modified version of HR-NAS[16]. To efficiently predict gaze values from multi-resolution feature maps, we propose a multi-resolution fusion transformer architecture as the gaze estimation head. In this section, firstly, we briefly describe the NAS-based feature extractor. We then introduce our multi-resolution fusion transformer which acts as the regression head. Finally, we summarize the entire pipeline along with the resource-aware search strategy.

3.1. NAS based feature extractor

In this section we briefly describe our NAS based feature extractor. We modify and adapt the feature extractor proposed in HR-NAS[16] for the gaze estimation task. Here we describe the search block and super-net architecture used in NAS.

3.1.1 Search block

As shown in Fig. 3, the search block contains three paths: a MixConv[36], a residual path, and a light-weight transformer[16] for extracting more global context. The number of convolution channels in the MixConv and the number of tokens in the lightweight transformer are searchable parameters.

For simplicity here we define a search block with 3×3, 5×5, 7×7 kernels. In the rest of this paper, we call a channel of the depthwise convolutions or a token in lightweight transformers a search unit. Let the input of search block be of c feature channels. A Squeeze-and-Excitation (SE)[20] block is applied to input to enhance its feature representation. In the MixConv path, the input channels are expanded by a point-wise 1×1 convolution to $(r_3 + r_5 + r_7)c$ dimension, where r_i is the expansion ratio for $i \times i$ convolution. The output is split accordingly, which are then fed into depth-wise convolutions with kernel sizes 3×3, 5×5, 7×7 respectively. Then the outputs from all the convolutions are concatenated, which is followed by another 1×1 convolution layer to reduce the channels to desired output channels c'.

In the lightweight transformer path, a Projector, which is used to reduce the computational cost, is applied on input features by projecting input features of size $c \times h \times w$ to a reduced size of $n \times s \times s$. Here, n denotes the number of queries and $s \times s$ is the reduced spatial size. Now, the transformer is applied over the projected input. Then an inverse Projector is applied on the output of transformer to inversely project it to desired output size. More importantly, there is a residual connection in the search block to handle the case when all search units of a search block become zero during the search. The residual connection has a point-wise 1×1 convolution to get desired output size. The outputs from the MixConv path and lightweight transformer are added along with residual connection to get the output

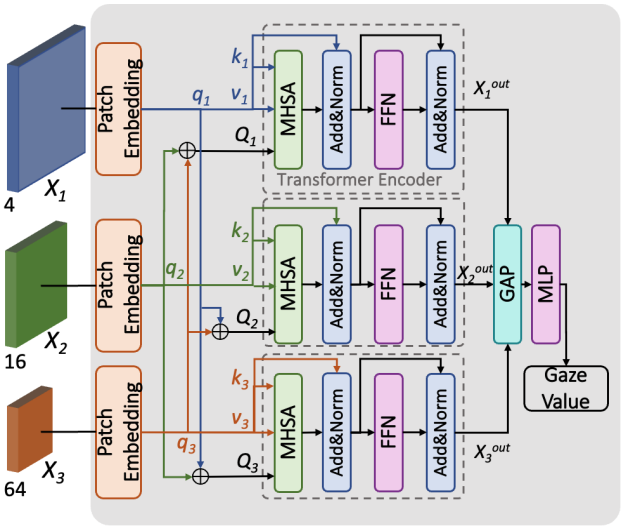

Figure 4. Architecture of our Multi-resolution Fusion Transformer. Here, MHSA is the Multi-head Self Attention layer, FFN is Feed Forward Network, GAP is Global Average Pooling layer, and MLP is Multi-layer Perceptron

of the search block.

Inspired by MixConv[36], where the authors use a different set of kernels at different stage of the network. Similarly we also adapt our search block by having different combinations of kernels at different stages of the network. We detail these modifications in 3.4.

3.1.2 Super-Net Architecture

Inspired by [3, 38, 16], we design a multi-branch search space that contains both multi-scale features and global contexts while maintaining high-resolution representations throughout the network.

The super-net architecture is shown in Fig. 5. The network consists of two modules: the parallel module and the fusion module. Both of the two modules are constructed with our search block. The parallel module obtains larger receptive fields and multi-scale features by stacking search blocks in each branch. A fusion module is used after a parallel module to exchange information across multiple branches. An extra lower-resolution branch is also generated from the previously lowest resolution branch. For each output branch, all its neighboring input branches are fused by using the search block to unify their feature maps. For example, a $1/8$ output branch integrates information of $1/4, 1/8$, and $1/16$ input branches.

As shown in Fig. 5, after two convolutions which decrease the feature resolution to 1/4 of the input image size, we start with this high-resolution branch and gradually add high-to-low resolution branches through fusion modules, and connect the multi-resolution branches in parallel through parallel modules. Finally, we reduce the channel dimension of multi-branch features by applying a point-wise 1×1 convolution layer to decrease computation of the estimation head and then connect the output to our multi-resolution fusion transformer.

3.2. Multi-resolution fusion transformer

After obtaining the multi-branch features, an intuitive solution is to resize and aggregate features and connect it to transformer encoder directly. The transformer architecture utilizes its self-attention mechanism to capture the correlations across patches.

Although the transformer encoders can inherently model the multi-resolution features jointly to some extent with a simple concatenation, the strong correlations between different resolution features are not fully exploited by the vanilla transformer since features are concatenated together. To address this we introduce our multi-resolution fusion transformer, referred as MRFT.

The proposed MRFT structure is shown in Fig.4. For 3 branches in the network, MRFT has X_i as inputs, where $i \in [1, 3]$. Here $X_i \in \mathbb{R}^{h_i \times w_i \times c_i}$, where (h_i, w_i) is the resolution of the i^{th} input feature map and c_i is the number of channels. As in ViT[17], we reshape every input feature map X_i into a sequence of flattened 2D patches $x_i \in \mathbb{R}^{n_i \times ({p_i}^2 \cdot c_i)}$, where (p_i, p_i) is the resolution of each feature patch, and $n_i = h_i w_i / {p_i}^2$ is the resulting number of patches, which also serves as the effective input sequence length for the transformer encoder.

Each sequence of flattened 2D patches is mapped to three matrices: feature query matrix q_i, key matrix k_i and value matrix v_i by linear transformations. The transformer query matrices are defined as:

$$Q_1 = T_1(q_2 +\!\!+ q_3), Q_2 = T_2(q_1 +\!\!+ q_3), Q_3 = T_3(q_1 +\!\!+ q_2)$$

where $+\!\!+$ is the channel wise concatenation operation and T_i is the transformation function to transform input to same size as k_i. By doing this, the high resolution features are empowered by the other low resolution features mostly comprising of local features. On the other hand, the low resolution features are provided with global information from other high resolution features.

The outputs X_i^{out} are represented as follows:

$$\begin{aligned} x_i' &= \mathrm{LN}(\mathrm{MHSA}(Q_i, k_i, v_i) + x_i) \\ X_i^{out} &= \mathrm{LN}(\mathrm{FFN}(x_i') + x_i') \end{aligned} \tag{1}$$

where $\mathrm{MHSA}(.)$ represents the Multi Head Self Attention block, $\mathrm{FFN}(.)$ denotes the Feed Forward Network, $\mathrm{LN}(.)$ is the layer normalization operator. Here we apply only one-layer of Transformer encoder to have less computation cost. The final gaze values are predicted by applying global average pooling (GAP) layer and MLP layer on outputs X_i^{out}. The main difference between our MRFT and

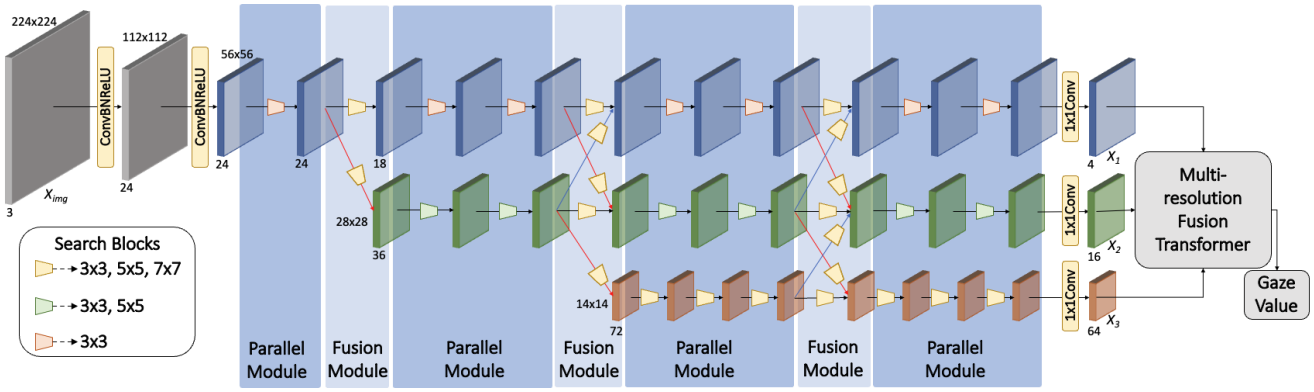

Figure 5. Architecture of our Super-Net. Here ConvBNReLU is a 3×3 convolution with Batch-Normalization and ReLU activation. Different search blocks are used in the parallel module. We show Search block architecture in Fig. 3 and MRFT in Fig. 4

ViT[17] lies in the usage of GAP layer rather than an extra learnable classification token and better fusion strategy than simple concatenation.

3.3. Baseline

To analyze NAS methods for gaze estimation tasks, we propose two baseline models having diffrenet number of branches in the super-net architecture. As our single-branch baseline model we use feature extractor from AtomNAS with a vanilla transformer encoder as estimation head. As our multi-branch baseline architecture we use an HR-NAS[16] feature extractor coupled with a vanilla transformer encoder[17] as estimation head. The vanilla transformer is inspired from the GazeTR[9] work. The multi-resolution features from HR-NAS are resized and concatenated before connecting to the vanilla transformer.

3.4. GazeNAS

First, we describe our GazeNAS and its difference with HR-NAS. Then, we briefly describe the search strategy. Finally, we describe the loss functions used to search the model.

As shown in 5, our GazeNAS has a 3 branch structure, since we have not observed significant performance gains from using a 4 branch structure like HR-NAS[16]. In MixConv[36] the convolution options in search block differ by the size of the feature maps. Basically, the lower resolution features are provided with large kernels options for better accuracy and high resolution features are provided with only small kernels to save computation cost. Inspired by this, in our parallel module the search block of 1^{st} branch has only 3×3, 2^{nd} branch has 3×3, 5×5 and 3^{rd} branch has 3×3, 5×5, 7×7 kernels. In the fusion module, the search block contains 3×3, 5×5, 7×7 kernels for a better fusion of features across branches. All search blocks contains lightweight transformers.

In summary, the **main difference** between feature extractor of HR-NAS[16] and our GazeNAS lies in 1) different kernel options in the search block depending on the feature map size 2) the number of branches in the network 3) Use of Squeeze and Excitation (SE)[20] block in search block

Search strategy: For the search strategy, we adopt a progressive shrinking NAS paradigm which generates lightweight models by discarding some of the convolution channels and Transformer queries during training. Following Darts[24], we introduce an importance factor $\alpha > 0$ that can be learned jointly with the network weights for each search unit of the searching block. We then progressively discard those with low importance while maintaining overall performance. Inspired by [44, 26, 16], we add a resource-aware L1 penalty on α, which effectively pushes importance factors of high computational costs to zero. Specifically, the L1 penalty of a search unit is weighted by the amount of the reduction in computational cost $\Delta > 0$ (i.e. FLOPs in this case):

$$\Delta_i = \begin{cases} 3 \times 3 \times h \times w & \text{if } i \text{ is a } 3 \times 3 \text{ conv} \\ 5 \times 5 \times h \times w & \text{if } i \text{ is a } 5 \times 5 \text{ conv} \\ 7 \times 7 \times h \times w & \text{if } i \text{ is a } 7 \times 7 \text{ conv} \\ O_T(n') - O_T(n'-1) & \text{if } i \text{ is a transformer token} \end{cases}$$

where O_T is the FLOPs of the transformer defined in HR-NAS[16], n' is the number of remaining tokens. Note that Δ's for search units of convolutions are fixed, while in the Transformer, Δ's is a function of the number of remaining tokens. It is worth mentioning that, although FLOPs is not always the best measure of latency, we use it anyway as it is the most widely and easily used metric. This can be easily adapted to use other metrics, e.g., latency and energy cost.

With the added resource-aware penalty term, the overall training loss is:

$$L = L_1(g_t, g_p) + \lambda \sum_{i \in A} \Delta_i |\alpha_i|$$

where L_1 denotes the standard L1 loss, g_t denotes the ground truth gaze value, g_p denotes the predicted gaze value, λ denotes the coefficient of the L1 penalty term, and A denotes the set of all available search units in the network.

During training, after every few epochs, we progressively remove the search units whose importance factors are below a predefined threshold ϵ and re-calibrate the running statistics of Batch Normalization (BN) layers. Note that if all tokens of a search block are removed, the search block will degenerate into a residual path, as shown in Fig. 3.

When the search ends, the remaining structure not only represents the best accuracy-efficiency trade-offs, but also has the optimal low-level/high-level and local/global feature combination for the gaze estimation task.

4. Experiments

4.1. Implementation details

4.1.1 Dataset for NAS & pre-training

In order to search an efficient neural architecture for the gaze estimation task, we use ETH-XGaze[45] dataset. It contains a total of 1.1M images of 110 subjects. We use the training set in ETH-XGaze for pre-training, which contains 765K images of 80 subjects. The evaluation set is divided into within-dataset and person specific evaluations, each including 15 people. We used the within-dataset as the test set for pre-training validation. The dataset provides the normalized data, which we directly fed into the model.

4.1.2 Datasets for evaluation

For the comprehensive evaluation of the searched neural architecture using GazeNAS, we select the following datasets for evaluation: MPIIFaceGaze[48], Gaze360[23], EyeDIAP[27] and RT-GENE[18]. For the direct comparison with state-of-the-art methods on these datasets, we keep the datasets similar to the previous works. More specifically, we follow [9] to process all the datasets as well as the evaluation protocol. After data-preprocessing, MPIIFaceGaze contains 45K images of 15 subjects. We perform leave-one-person-out evaluation on it. EYEDIAP contains 16K images of 14 subjects. We perform four-folder cross validation on it. Gaze360 contains 84K images of 54 subjects for training and 16K images of 15 subjects for test. RT-GENE contains 92K images of 13 subjects. A three-folder cross validation is performed in RT-GENE. The EYEDIAP and MPIIFaceGaze datasets have relatively limited head pose and gaze range, therefore assumed as benchmarks in a controlled environment. Gaze360 and RT-GENE has relatively wide head pose and gaze range and hence represent performance in unrestrained environments.

4.1.3 Training

We search for an efficient neural architecture by using GazeNAS on ETH-XGaze (GazeNAS-ETH). The whole code structure is implemented using PyTorch[30][1] and trained on NVIDIA Tesla A100 GPUs. The input size is set to 224×224. The initial learning rate is set to 0.001 with the batch size 369 on 3 Tesla A100 GPUs for 50 epochs, and decays by 0.97 every 5 epochs. We adopt an Adam optimizer with momentum 0.9 and weight decay 1e-5. We also employ the exponential moving average (EMA) with decay 0.9999. By setting the coefficient of the L1 penalty term λ to 1.0e-5, we get our GazeNAS-ETH model.

As for the evaluation of GazeNAS-ETH on evaluation datasets, we freeze the model architecture and train on evaluation datasets. We use pre-trained weights on ETH-XGaze as the initial values of parameters. The λ is set to 0, as no further pruning is required. For all four evaluation datasets, the learning rate is set to 0.0005 and RMSprop optimizer is used to train the model. All other hyper-parameters remain same as before.

4.1.4 Evaluation

For the gaze estimation task, the most common evaluation metric is an angular gaze error. We use it to compare with other gaze estimation methods, where a smaller error represents a better model.

4.2. Comparison with state-of-the-art

We compare the performance of our proposed model GazeNAS-ETH and the state-of-the-art methods, which showed competitive performance in gaze estimation, with the MPIIFaceGaze, Gaze360, EYEDIAP and RT-GENE datasets. The results are shown in Table 1. In the table, methods corresponding to category A are CNN or RNN-based gaze estimation models namely FullFace[48], RT-GENE[18], Dilated-Net[6], CA-Net[8] and Gaze360[23]. The methods in category B are those that use a transformer. There is one more difference between category A and B models, the models in category A are ImageNet[13] pretrained whereas the models in category B are ETH-Xgaze[45] pretrained.

Table 2 shows the number of parameters for each method and the FLOPs required to derive the results. The results show that GazeNAS-ETH has better estimation of gaze values compared to state-of-the-art methods on all the evaluation datasets while having only about 1.027M parameters and using only 0.28GFLOPs. Therefore, GazeNAS-ETH achieved state-of-the-art performance with the least computational budget. More specifically, when compared to the

[1]To ensure reproducibility, we will release the code.

Table 1. Comparison with State-of-the-art methods. Our proposed GazeNAS-ETH achieves state-of-the-art results in all four datasets. $*$ indicates the model backbone is pre-trained on the ImageNet dataset, † indicates the model is pre-trained on the ETH-XGaze dataset

Category	Method	MPIIFaceGaze[48]	Gaze360[23]	RT-GENE[18]	EYEDIAP[27]
A	FullFace [48]$*$	4.93°	14.99°	10.00°	6.53°
	RT-GENE [18]$*$	4.66°	12.26°	8.60°	6.02°
	Dilated-Net [6]$*$	4.42°	13.73°	8.38°	6.19°
	CA-Net [8]$*$	4.27°	11.20°	8.27°	5.27°
	Gaze360 [23]$*$	4.06°	11.04°	7.06°	5.36°
B	CADSE [28]†	4.04°	10.70°	7.00°	5.25°
	GazeTR-Hybrid [9]$*$†	4.00°	10.62°	6.55°	5.17°
	Our GazeNAS-ETH†	$\mathbf{3.96^\circ}$	$\mathbf{10.52^\circ}$	$\mathbf{6.40^\circ}$	$\mathbf{5.00^\circ}$

Table 2. Specification of the state-of-the-art models. Our proposed GazeNAS-ETH requires very less computational budget compared to other state-of-the-art models, making it easier to deploy for real-time applications

Method	# of Params.	# of FLOPs	Running Time(ms)
RT-GENE[18]	82.0 M	30.81 G	467
Gaze360[23]	14.6 M	12.78 G	276
CADSE[28]	74.8 M	19.75 G	379
GazeTR-Hybrid[9]	11.4 M	1.84 G	N/A
Our GazeNAS-ETH	**1.027** M	**0.28** G	**22**

state-of-the-art model GazeTR[9], the performance in MPIIFaceGaze dataset increased by 0.04°, in Gaze360 dataset by 0.1°, in RT-GENE dataset by 0.15° and in EYEDIAP dataset by 0.17°.

The proposed model's performance can be seen in Figure 6. It visualizes some qualitative results of gaze estimation on various face images from different dataset.

4.3. Comparison with Baseline models

As mentioned in section 3.3, we experiment with two baseline models to observe the performance of both single-branch and multi-branch NAS methods on gaze estimation tasks. We prefer to use AtomNAS[26] over a single-branch HR-NAS[16] due to its better performance in other tasks. Both models are searched on ETH-XGaze. We conducted evaluation experiments on four datasets. The results are shown in Table 3. The single-branch baseline (AtomNAS+ViT) uses a smaller number of parameters and FLOPs but the performance of the multi-branch baseline (HR-NAS+ViT) is better by a significant margin proving the impact of multi-branch network for gaze estimation task as proved by previous works. Since both of these baselines have ViT as the estimation head, their performance is comparable to other state-of-the-art methods due to better representation power of transformers.

The HR-NAS+ViT uses a more number of parameters and FLOPs than our GazeNAS. The possible reason for this is the difference in number of branches as well as the convolution options in the search block. Even after using less computation, the performance of our GazeNAS is better compared to HR-NAS+ViT on all evaluation datasets, showing the impact of our MRFT based gaze estimation head. Due to its better fusion and representation capability of both global and local features, the number of parameters required in feature extractor becomes less compared to HR-NAS+ViT.

4.4. Ablation Study

To confirm the validity of our search block design, we conduct the following ablation study by removing some components from the entire pipeline: 1) without MixConv and 2) without lightweight transformer. (See Table 3).

a) w/o MixConv To investigate the effect of the MixConv layer in our search block, we replaced the MixConv layer with 3×3 convolution layer in all search blocks. We conducted experiments on four datasets to ensure consistency, and the results are shown in Table 3. When the MixConv layer was applied in the search block, the performance improved from 0.20° to 0.70°, which shows the significance of MixConv in proposed method. It seems to be because MixConv layer extracts both global and local information more efficiently which is significant for gaze estimation task.

b) w/o lightweight transformation To check the effect of lightweight transformers in our search block, we replace the lightweight transformer layer with skip connection in all search blocks. As earlier, we conduct experiments on four datasets and the results are shown in Table 3. The results clearly show the impact of lightweight transformer on the performance of our GazeNAS-ETH. More specifically, the performance improved by 0.25° to 0.75°. As lightweight transformer enhances the global context within the search block, it is significant in our search block for gaze estimation task.

Table 3. Ablation study and baseline models

Method	Params	FLOPs	MPII [48]	Gaze360[23]	EYEDIAP[27]	RT-GENE[18]
AtomNAS + ViT (single branch)	0.9M	250M	4.35°	10.90°	5.29°	7.50°
HR-NAS + ViT (multi-branch)	1.1M	320M	4.25°	10.65°	5.20°	6.83°
GazeNAS (MRFT)	1.027M	280M	3.96°	10.53°	5.00°	6.40°
w/o MixConv	1.024M	275M	4.31°	10.71°	5.21°	7.10°
w/o lightweight transformer	0.4M	233M	4.28°	10.88°	5.25°	7.15°

5. Discussions

5.1. Dataset for searching models

Previous NAS works for other tasks have always used same dataset for both searching model and testing performance. For the case of gaze estimation, we think that searching model on every dataset would be computationally very expensive. Since most of the datasets in gaze estimation are small and have a limited range of gaze values and head pose. We propose to use ETH-XGaze for searching neural architecture due to its high range of gaze values and large dataset size. We validate the searched network architecture GazeNAS-ETH on other datasets. GazeNAS-ETH is able to achieve state-of-the-art on other datasets. This indicates that for gaze estimation, searched model on ETH-XGaze is easily able to generalize to other datasets, saving time and resources of searching models on every datasets of the task.

5.2. NAS in gaze estimation

We follow popular AtomNAS and HR-NAS to design feature extractors of baseline models in this paper. We use popular ViT as the gaze estimation head. Both of these models are able to achieve competitive results against previous state-of-the-art models. This indicates that NAS based methods are suited for gaze estimation. In order to further improve the performance, we propose our MRFT gaze estimation head in GazeNAS. Through our experiments we validate that GazeNAS-ETH outperforms previous state-of-the-art methods. This indicates that NAS based methods not only requires very less computational budget but can also achieve state-of-the-art performance. This enables real-time applications of gaze estimation task.

5.3. Limitations

The searched model on ETH-XGaze using our GazeNAS method is able to perform well on many datasets after fine tuning on that dataset. One of the major limitations of NAS based methods is a low cross-dataset performance. We perform experiments to see the cross-dataset performance which is not competitive enough to the state-of-the-art method [28]. This suggests that we may need to increase the computational budget for better cross-dataset generalizability in this task.

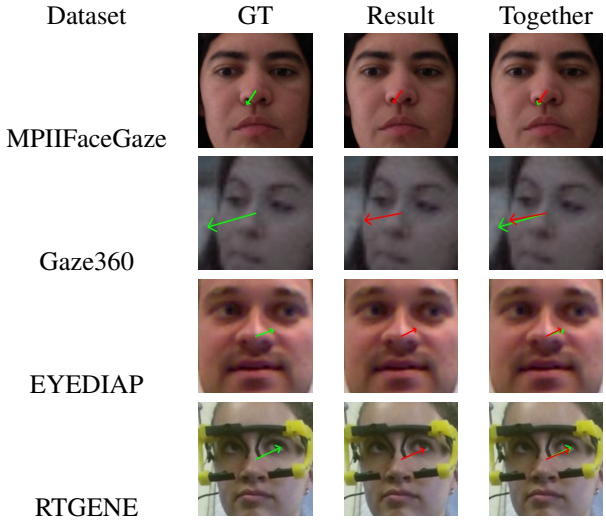

Figure 6. Proposed method GazeNAS-ETH's Gaze estimation results on various dataset face images. The first row images are ground truth for gaze, and the second row are the estimation result by the proposed network, and third row are both shown together

6. Conclusions

For gaze estimation, we are the first to explore the effectiveness of using the NAS-based methods. We introduce a novel multi-resolution fusion transformer based gaze estimation head, which effectively fuses global context with local features for accurate gaze estimation. We propose a modified HR-NAS based feature extractor for the task of gaze estimation. We propose to use only one dataset for searching neural architecture rather than individually searching model on all datasets. We choose ETH-XGaze [45] as the dataset for searching neural architectures in the gaze estimation tasks and validate it on other datasets. Through rigorous experiments on four public datasets [49, 23, 18, 27], we validate that our proposed GazeNAS-ETH outperforms other state-of-the-art methods that are either CNN-based or transformer-based in terms of both accuracy and computational cost. More specifically, our GazeNAS-ETH uses only 1M parameters and 0.28 GFLOPs, which is much less than previous state-of-the-art models and hence can be easily deployed for real time applications on embedded devices at edge.

References

[1] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *ArXiv*, abs/1908.09791, 2020.

[2] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *ArXiv*, abs/1812.00332, 2019.

[3] Xin Cai, Boyu Chen, Jiabei Zeng, Jiajun Zhang, Yunjia Sun, Xiao Wang, Zhilong Ji, Xiao Liu, Xilin Chen, and Shiguang Shan. Gaze estimation with an ensemble of four architectures. *arXiv*, abs/2107.01980, 2021.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

[5] Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, G. Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, 2018.

[6] Zhaokang Chen and Bertram E. Shi. Appearance-based gaze estimation using dilated-convolutions. In *ACCV*, 2018.

[7] Yihua Cheng, Yiwei Bao, and Feng Lu. Puregaze: Purifying gaze feature for generalizable gaze estimation. 03 2021.

[8] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:10623–10630, 04 2020.

[9] Yihua Cheng and Feng Lu. Gaze estimation using transformer. arXiv preprint arXiv:2105.14424, 05 2021.

[10] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[11] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. In *NeurIPS*, 2020.

[12] Xiyang Dai, Dongdong Chen, Mengchen Liu, Yinpeng Chen, and Lu Yuan. Da-nas: Data adapted pruning for efficient neural architecture search. In *ECCV*, 2020.

[13] 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*, pages 248–255. Ieee, 2009.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[15] Philipe Dias, Damiano Malafronte, Henry Medeiros, and Francesca Odone. Gaze estimation for assisted living environments. pages 279–288, 03 2020.

[16] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, Zhiwu Lu, and Ping Luo. Hr-nas: Searching efficient high-resolution neural architectures with lightweight transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[18] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[19] E.D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, 53(6):1124–1133, 2006.

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[21] Qiang Ji and Xiaojie Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8:357–377, 10 2002.

[22] X. Jin, Jiang Wang, Joshua Slocum, Ming-Hsuan Yang, Shengyang Dai, Shuicheng Yan, and Jiashi Feng. Rc-darts: Resource constrained differentiable architecture search. *ArXiv*, abs/1912.12814, 2019.

[23] Petr Kellnhofer, Adrià Recasens, Simon Stent, W. Matusik, and A. Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6911–6920, 2019.

[24] Hanxiao Liu, K. Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *ArXiv*, abs/1806.09055, 2019.

[25] Zhichao Lu, Ian Whalen, Vishnu Naresh Boddeti, Yashesh D. Dhebar, K. Deb, E. Goodman, and W. Banzhaf. Nsga-net: A multi-objective genetic algorithm for neural architecture search. *ArXiv*, abs/1810.03522, 2018.

[26] Jieru Mei, Yingwei Li, Xiaochen Lian, X. Jin, Linjie Yang, A. Yuille, and Jianchao Yang. Atomnas: Fine-grained end-to-end neural architecture search. *ArXiv*, abs/1912.09640, 2020.

[27] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. EYEDIAP. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, mar 2014.

[28] Jun O Oh, Hyung Jin Chang, and Sang-Il Choi. Self-attention with convolution and deconvolution for efficient eye gaze estimation from a full face image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4992–5000, June 2022.

[29] Seonwook Park, Adrian Spurr, and Otmar Hilliges. *Deep Pictorial Gaze Estimation*, pages 741–757. 09 2018.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[31] Anjul Patney, Joohwan Kim, Marco Salvi, Anton Kaplanyan, Chris Wyman, Nir Benty, Aaron Lefohn, and David Luebke. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery.

[32] Thammathip Piumsomboon, Gun Lee, Robert W. Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39, 2017.

[33] Rima-Maria Rahal and Susann Fiedler. Understanding cognitive and affective mechanisms in social psychology through eye-tracking. *Journal of Experimental Social Psychology*, 85:103842, 2019.

[34] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Y. Suematsu, Jie Tan, Quoc V. Le, and A. Kurakin. Large-scale evolution of image classifiers. *ArXiv*, abs/1703.01041, 2017.

[35] M. Sandler, Andrew G. Howard, Menglong Zhu, A. Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[36] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels. *BMVC 2019*, 07 2019.

[37] 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*, pages 5998–6008, 2017.

[38] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[39] Kang Wang, Rui Zhao, Hui Su, and Qiang Ji. Generalizing eye tracking with bayesian adversarial learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11899–11908, 2019.

[40] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao. Gaze prediction in dynamic 360â ° immersive videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5333–5342, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[41] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, X. Chen, Guo-Jun Qi, Qi Tian, and H. Xiong. Pc-darts: Partial channel connections for memory-efficient differentiable architecture search. *ArXiv*, abs/1907.05737, 2019.

[42] Hyo Sik Yoon, Na Rae Baek, Noi Quang Truong, and Kang Ryoung Park. Driver gaze detection based on deep residual networks using the combined single image of dual near-infrared cameras. *IEEE Access*, 7:93448–93461, 2019.

[43] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas S. Huang, Xiaodan Song, and Quoc V. Le. Bignas: Scaling up neural architecture search with big single-stage models. In *ECCV*, 2020.

[44] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas S. Huang. Slimmable neural networks. *CoRR*, abs/1812.08928, 2018.

[45] Xucong Zhang, Seonwook Park, T. Beeler, D. Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *ECCV*, 2020.

[46] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Everyday eye contact detection using unsupervised gaze target discovery. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 193–203, New York, NY, USA, 2017. Association for Computing Machinery.

[47] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, 2015.

[48] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. 11 2016.

[49] Xucong Zhang, Yusuke Sugano, Mario Fritz, and A. Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:162–175, 2019.

[50] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

[51] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. *ArXiv*, abs/1611.01578, 2017.