Efficient Multi-Purpose Cross-Attention Based Image Alignment Block for Edge Devices

Bahri Batuhan Bilecen, Alparslan Fişne, Mustafa Ayazoğlu
Aselsan Research
Ankara, Turkey
{batuhanb, afisne, mayazoglu}@aselsan.com.tr

Abstract

Image alignment, also known as image registration, is a critical block used in many computer vision problems. One of the key factors in alignment is efficiency, as inefficient aligners can cause significant overhead to the overall problem. In the literature, there are some blocks that appear to do the alignment operation, although most do not focus on efficiency. Therefore, an image alignment block which can both work in time and/or space and can work on edge devices would be beneficial for almost all networks dealing with multiple images. Given its wide usage and importance, we propose an efficient, cross-attention-based, multi-purpose image alignment block (XABA) suitable to work within edge devices. Using cross-attention, we exploit the relationships between features extracted from images. To make cross-attention feasible for real-time image alignment problems and handle large motions, we provide a pyramidal block based cross-attention scheme. This also captures local relationships besides reducing memory requirements and number of operations. Efficient XABA models achieve real-time requirements of running above 20 FPS performance on NVIDIA Jetson Xavier with 30W power consumption compared to other powerful computers. Used as a sub-block in a larger network, XABA also improves multi-image super-resolution network performance in comparison to other alignment methods.

1. Introduction

Image alignment (image registration) aims to align or match images to a chosen reference image. This task constitutes an important part of many computer vision problems dealing with multiple images either in space and/or time, such as restoration [41, 47], segmentation [21], HDR imaging [45], stereo imaging [40], multi and single-image super-resolution and video super-resolution [24, 43, 44].

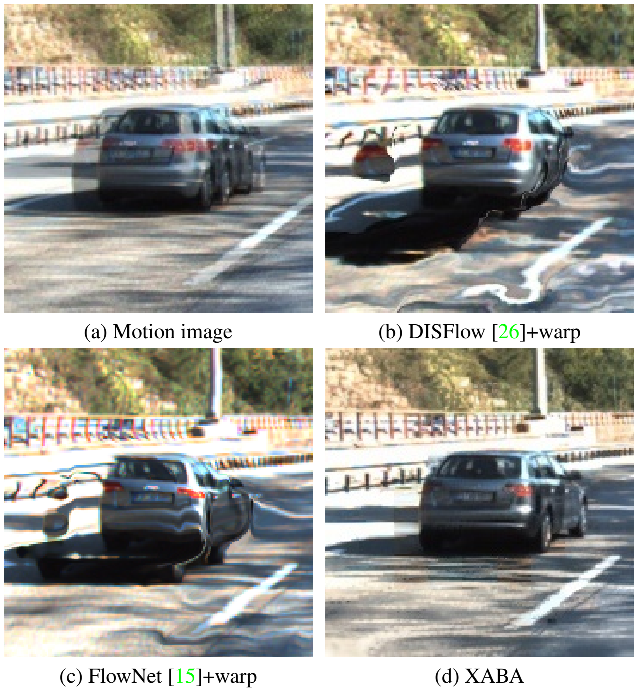

(a) Motion image
(b) DISFlow [26]+warp
(c) FlowNet [15]+warp
(d) XABA

Figure 1. XABA's alignment performance compared to DISFlow and FlowNet based alignment. Motion image represents combination of reference and target images.

To construct a backbone used in many vision-related problems, there have been many image alignment methods/blocks developed to this day with different focuses. These methods can be divided into three main categories: feature-point extraction-based alignment, classical optical flow-based alignment and deep learning-based alignment.

Feature-point extraction-based alignment algorithms utilize extracted feature points to create a global transform matrix between reference and target frames. Later, the resulting matrix is used to warp frames or images onto each other [34]. Classical optical flow-based alignment algorithms compute the flow vectors between reference frame pixels and target frame pixels, then use the flow vectors in the warping process. These methods can vary greatly, from computing affine transforms of image patches using flow vectors [27] to using a contrast constancy assumption and iteratively reducing misalignments between image pairs [10]. Deep learning-based alignment algorithms inherit ideas such as deformable convolutions [13] and deep-learning based optical flow [15], and attention mechanisms [47].

Despite having been implemented in many ways, most of the aforementioned image alignment methods suffer trading between accuracy and speed, especially on low-end devices. Feature-point extraction-based alignment algorithms require the planar scene assumption, which does not hold most of the time. Although having efficient implementations, classical optical-flow algorithms can fail with large motions. They can also cause bottlenecks when used in deep learning pipelines, as it avoids being able to provide an end-to-end deep learning-based solution. This result may result in sub-optimal solution, since optical flow part is not trained and is based on hand-crafted features. This is indeed the problem of integrating classical algorithms into deep learning pipelines.

On the other hand, having the entire solution blocks in deep learning framework helps using hardware at the hand to the full extent easily by using the already supplied deployment tools such as NVIDIA's TensorRT [4] and Intel's OpenVINO [6], and improves the performance of the overall trained pipeline. Besides, deep learning-based alignment methods are fairly more recent and have promising advancements. Deformable convolutions add 2D offsets to sampling locations in traditional convolutions, introducing more adaptation and easing the alignment [18]. Deep learning-based optical flow methods [36, 38] yield accurate alignment; however, they can be quite expensive and hence may not be feasible to run on embedded or edge devices. Recent work on image alignment used attention mechanism because of its native feature matching and transforming properties [7, 47], however direct application of attention operation for image alignment can be memory and computation hungry. The use of attention mechanism started with natural language processing domain with the seminal works [9, 28, 39] and later transferred to image domain [14] and it is known to improve the performances of various networks on image domain [42] as well. Nonetheless, using attention for image alignment is fairly recent and not much study is present, especially focusing on edge device interference efficiency.

Motivated by the recent performance & accuracy related studies on attention, the lack of efficient attention-based alignment studies, and the lack of a general-purpose deep learning-based image alignment block applicable to different problems on edge devices; we propose a cross-attention based image alignment block which can be integrated into many deep learning pipelines with ease. We name our proposed method as **XABA (cross-attention based aligner)**. XABA is designed with efficiency and plug-and-play approach in mind that can run in real-time. We divide reference and target images into non-overlapping sub-images, extract features from each sub-image, and efficiently compute a cross-attention matrix to align sub-images in feature-level. Dividing images into blocks allows us to process them in parallel, ease implementations on embedded systems, and force local information extraction. In addition, to boost the proposed methods performance in large motions, the baseline block is applied in a pyramidal fashion to the input image at different scales which effectively increases the attended area while keeping the computational cost requirements at minimum. The final output of the network is used by fusing the pyramidal network outputs with a pixel attention based fusion module. We also do not have hand-crafted hyper-parameters for our cost function unlike [7], which further eases the training process.

For more robustness and adaptation to different applications, we also propose a sparsification scheme to calculate the attention matrix, which can be used to find sparse attention matrices. This can be further exploited for fast matrix multiplication during inference.

Furthermore, as an alternative to the classical softmax non-linearity used in attention matrices, we combined hard-thresholded ReLU (clips above 1) with row-normalization as the activation function. This effectively normalizes the rows of the calculated attention matrix, promotes sparsity, and makes the implementation more suitable to be used in edge devices due to the lightness of the activation function.

The main contributions of this paper are:

- We propose a block-based, pyramidal, multi-purpose, deep-learning based image alignment block using cross-attention.
- We propose an alternative to softmax activation, combining hard-thresholded-ReLU with Normalization.
- The proposed block is efficiency-focused and real-time applicable, proven by multiple tests on edge devices.

2. Related Works

Feature-point based alignment. This method aims to match feature points extracted from images via [20, 34, 46]. Using outlier methods like RANSAC [16] and their derivations [31, 32], some extracted feature points are eliminated. Then, using rest of the matching feature points, a global transformation matrix is generated between images. The major downside of this method is that the motion is restricted to be globally uniform. In other words, every part of the image is being subjected to the same global motion, which is not correct most of the time.

Traditional & deep-learning-based optical flow alignment. Optical flow calculates flow vectors for each pixel between reference image pixels and target image pixels. Since a one-to-one correspondence is obtained between pixels (for occlusion-free regions), images can be aligned accordingly. Optical flow is an under-constrained problem, hence additional constraints like brightness consistency and confining to small movements are required. Lucas-Kanade [27] and Horn-Schunck [19] are two of the most well-known classical flow algorithms, with other variations also present [37]. Classical flow algorithms are still being developed, one of the recent ones being DISFlow [26] which focuses on time complexity. Deep-learning based optical-flow, on the contrary, are much more recent which gained momentum with FlowNet [15]. FlowNet proposed using a convolutional network in flow estimation for the first time. FlowNet2 [23] offered using correlation layers as an improvement. It also included a new stacked architecture including image warping, and sub-networks for small displacements. PWC-Net [36] refined flow in a coarse-to-fine manner and utilized feature warping to reduce the network size. In addition to performing better than FlowNet2, PWC-Net also has a smaller network size. Recent algorithms like RAFT [38] started to utilize correlation volumes and recurrent structures to further increase the flow quality. RIFE [22] directly estimates intermediate flow estimations from a low-framerate video for frame interpolation purposes. Even though most deep-learning based flow methods have high accuracies, they are not applicable on embedded environments due to their high computational demand at practically meaningful image sizes.

Attention and attention-based alignment. Bahdanau [9] and Luong [28] models, also known as additive and multiplicative/dot-product attentions respectively, are the first remarkable studies about attention. Later on, Google proposed their attention-based network architecture, the Transformer [39]. All these studies were done on natural language processing; however, attention-based solutions in computer vision problems were also starting to emerge. Inspired by non-local means denoising algorithm [11], Wang et al. [42] proposed a non-local building block which captures long-range dependencies within feature maps. This study prepared the ground for many vision applications, such as classification [33, 48], object detection [12, 17], image segmentation [21, 30], image super-resolution [40], and video super-resolution [24, 43, 44]. Regardless, the idea of attention is fairly new on image alignment algorithms. Only a handful of studies [7, 43, 47] are present, most of which do not focus on efficiency and deployment on embedded environments.

3. Proposed Method

3.1. Background information

Attention mechanism. The attention mechanism proposed in [39] and [28] has three main concepts: key, query and value.

Query (Q) is the input vector for which the attention is desired to be calculated. Query is compared with all keys (K) by taking the dot products between Q and K, creating a key-query matrix (QK^T). Higher values in the resulting matrix indicate higher correlation between the relevant elements of Q and K. The matrix is then normalized with some constant $(\sqrt{d_k})$ and its softmax is calculated to make the sum of each row 1. At the end, multiplying with values (V) gives the attention (1).

$$Att(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V = AV \tag{1}$$

In self-attention, Q, K and V are equal to each other. Naturally, the equality does not hold for cross-attention. In our study, Q represents the feature matrix of the reference image, the image to be aligned. K is the feature matrix of the target image. V is the reference image itself.

3.2. Baseline Block - Interframe Aligner

Our *baseline block* structure which we refer as *interframe aligner block* is given in Fig. 3. Using cascaded convolutional layers with skip connections and ReLU activations, features of the input images (I_1, I_2) are extracted. Note that the parameters of this residual feature extraction network is shared between images. These features are then convolved with 1x1 kernels to reduce the number of operations. Note that depending on the application, the layer with 1x1 kernel may or may not share the same parameters with its parallel branch. For instance, 1x1 convolutions can be shared in aligning two RGB images as done in our experiments, however; to match images in different domains (such as thermal and RGB images) further adaptations may be needed.

After the feature extraction, resulting features are sent to tensor-to-block (T2B) operator. T2B operator divides the feature images into non-overlapping patches and stacks these blocks in batch dimension. In this sense, it is somewhat similar to pixel unshuffling [35] where the unshuffled pixels are stacked on the channel dimension. This enables us to only match spatially closer features with each other while allowing parallel processing.

To find the correlation between two image features, T2B outputs are matrix multiplied to create a dot-product attention matrix as described in Eq. (1). After normalization along the rows of the result with a non-linearity (softmax, or hard-thresholded ReLU with Normalization), A, the *attention matrix* between two input image features is generated.

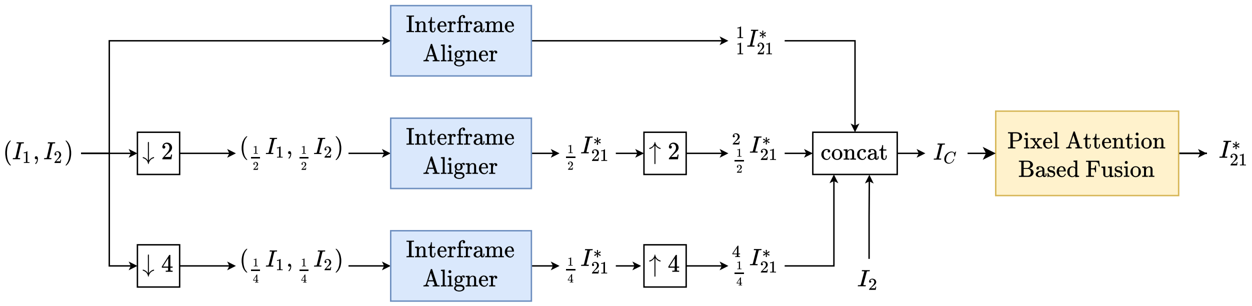

(I_1, I_2)
Interframe Aligner
${}^{1}_{1}I^*_{21}$
↓ 2
$({}_{\frac{1}{2}}I_1, {}_{\frac{1}{2}}I_2)$
Interframe Aligner
${}_{\frac{1}{2}}I^*_{21}$
↑ 2
${}^{2}_{\frac{1}{2}}I^*_{21}$
↓ 4
$({}_{\frac{1}{4}}I_1, {}_{\frac{1}{4}}I_2)$
Interframe Aligner
${}_{\frac{1}{4}}I^*_{21}$
↑ 4
${}^{4}_{\frac{1}{4}}I^*_{21}$
concat
I_2
I_C
Pixel Attention Based Fusion
I^*_{21}

Figure 2. Pyramidal global alignment block. concat denotes tensor concatenation. ↑ and ↓ denote upscaling and downscaling operations, respectively.

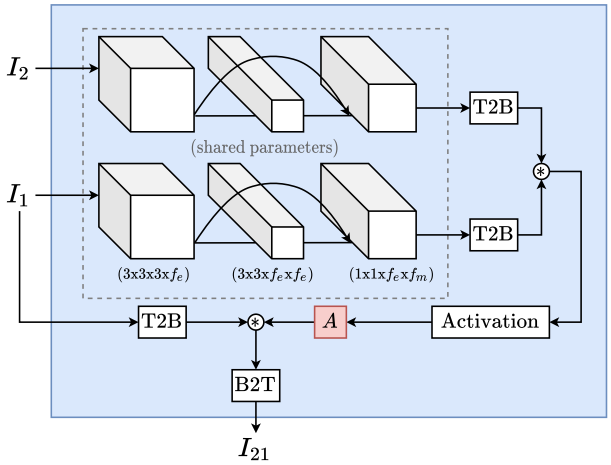


Figure 3. Baseline block - interframe aligner. Reference (I_1) and target frames (I_2) are sent as inputs and aligned frame I_{21} is obtained at the output. f_e and f_m are modifiable dimension parameters for feature extraction. T2B is the operator transforming tensors into non-overlapping blocks and reorganize the tensor in the batch dimension, while B2T is the inverse operator. $*$ is matrix multiplication operation. *Activation* is the function to normalize the attention matrix, A.

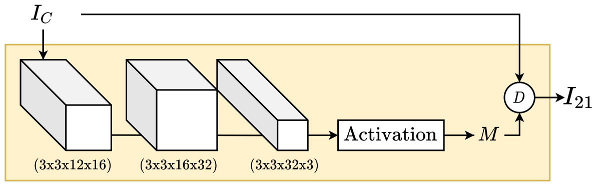


Figure 4. Pixel attention based fusion block. Using the outputs of interframe aligners, it generates a mask M to be used in fusing the aligned images. D denotes the following operation: $D(A, B) = \sum_i A[i] \cdot B[i]$, where $A[i]$ and $B[i]$ have the same dimensions for all i, and $\cdot$ denotes element-wise matrix multiplication operation. For the resulting tensor after *concat* operation (I_C), each i^{th} dimension represents an aligned image (${}^{K^{-1}}_{K}I^*_{21}$).

Note that this matrix A transfers images from K (key) domain to Q (query) domain via an adaptive linear combination and blending. This linear combination map is then used for aligning I_1 and I_2. Block-to-tensor (B2T) operation is applied at the end to reverse the effects of T2B operator, which is similar to pixel shuffling [35] where it operates along the channel dimension.

As discussed above, the baseline block is good enough to capture and align small displacements between features. However, to be able to capture large motions, non-overlapping block size should be increased which is a parameter of T2B operator. Unfortunately, increasing the block size increases the size of A, and hence it increases the computational load and memory requirements which is not suitable for edge devices. As an alternative to handle large motions, we propose a pyramidal processing scheme which is more efficient and suitable for edge devices.

3.3. Pyramidal Global Alignment Block

As described in Sec. 3.2, our baseline interframe aligner is suitable for capturing local feature matches. In other words, it is good at capturing small displacement of features between images. To effectively handle large motions and effectively increase the block size while being computationally light, we propose another block which we refer as *Pyramidal Global Alignment Block*. This block encapsulates different number of baseline blocks dedicated to work with different scales of the input images (Fig. 2).

An input image pair (I_1, I_2) is sent to the alignment block. Each baseline interframe aligner takes down scaled input image pairs $({}_KI_1, {}_KI_2)$, where the downscaling factors are denoted by $K = \{1, \frac{1}{2}, \frac{1}{4}, \ldots \frac{1}{n}\}$. Interframe aligners generate the aligned frames $({}_KI^*_{21})$, all of which have different resolutions due to different downscalings. For all outputs of interframe aligners, upsampling is applied with the same scaling factor and therefore are scaled back to their original resolution (${}^{K^{-1}}_{K}I^*_{21}$). The individual outputs of the baseline blocks for different scales are fused into a single image using the Pixel Attention Based Fusion Block. Final aligned frame result, I^*_{21} is obtained at the end.

3.4. Pixel Attention Based Fusion Block

Individual outputs of the baseline blocks for different scales constitute candidate images and hence candidate pixels. Inherently, at unit scale level, the image resolution is high but only small displacements are handled. At $\frac{1}{2}$ scale, level medium displacements are handled but the image resolution is lower. At $\frac{1}{4}$ scale level, very large displacements are handled; however, the resolution is at its lowest. Given these, a selection mechanism is needed. *Pixel Attention Based Fusion Block* is used for this purpose in such a way that it tries to combine different outputs to form a single image.

The fusion block inherits a cascaded convolutional network which generates a mask M using all interframe aligner outputs, shown in Fig. 4. Concatenated aligned images (I_C) are passed through the CNN and the activation function, which performs normalization in the channel dimension. This ensures that the contribution of all interframe aligner output energies are unchanged. Each channel of the resulting mask M are then used as a multiplicative mask for the corresponding images in I_C and all masked images are summed up to obtain the final image. Note that the mask in this case determines the combination ratio of all interframe aligner outputs from different scales.

4. Experiments

For the experiments, we used Kitti [29] dataset which is commonly used in image alignment, stereo and optical flow benchmarking. Kitti includes 200 training and 200 test stereo scene pairs, captured in rural and city traffic. Images are in RGB and lossless png format, with resolutions not the same among all images but all around 1250x375.

The experiments can be divided into three different sections. In the first experiment, we used XABA by itself for image alignment. In the second experiment, we used the image pairs of the Kitti dataset and posed a Multi Image Super-Resolution problem (MISR) and showed the performance of XABA in combination with a Single Image Super-Resolution (SISR) network to solve MISR problem. In the third experiment, for the different parameter settings and block sizes of XABA, we have taken measurements from NVIDIA Jetson Xavier and showed real-time capabilities of the proposed method.

4.1. Training Details

For the first and second experiments, we used Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and used maximum learning rate = 2e-3 with Knee learning rate scheduling [25] and warm up for all of our experiments. Mini batch size of 8 is used and the models are trained for 170 epochs, where each epoch consumes the training images 10 times. Each mini-batch is composed of image patches cropped from random images from the training set and standard geometric transformations such as rotate & flip were used as a form of data augmentation. We used 320x320 as crop size and for the second experiment (Sec. 4.3) the low resolution cropped images were created by 4 times downscaling the original images. In both experiments, Charbonnier loss was used with $\epsilon = 0.1$ as defined in (2). Charbonnier loss is the smoother version of L1-loss, which is known to have better convergence characteristics than L2-loss.

$$Charbonnier(x) = \sqrt{x^2 + \epsilon^2} \tag{2}$$

4.2. Alignment Performance

For this experiment, we used images from Kitti dataset paired in time as inputs to XABA and tried to align these images and warp the reference image to the target image. An example pair and alignment results of the different methods can be seen in Fig. 5

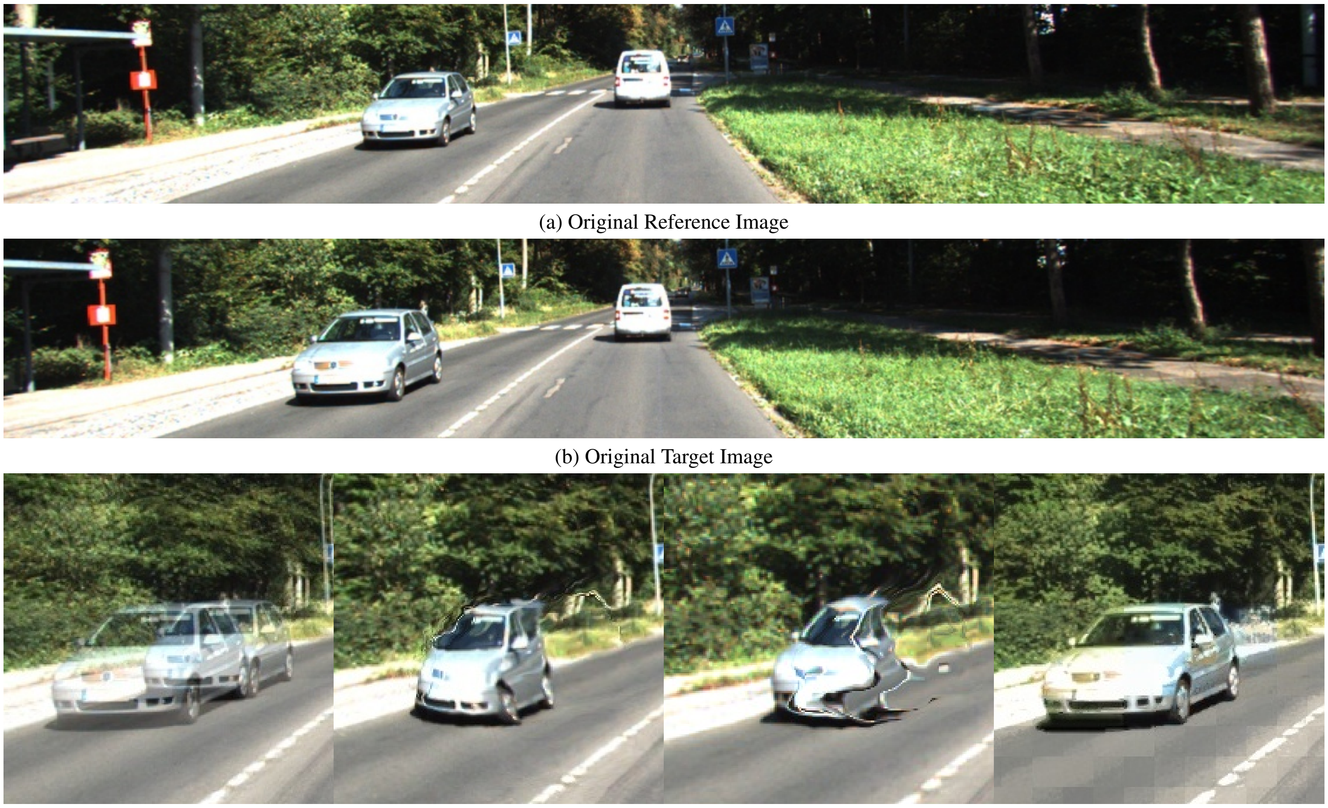
As shown in Fig. 5, optical flow based methods' performance drastically drops whenever there is a large motion. This is basically due to the fact that these methods constraint the change in between frames to planar geometric motion with 1-to-1 pixel correspondence. These constraints from the point of the view of attention mechanism are indeed equivalent to limiting the Attention matrix, A, to a permutation matrix where there is one and only one entry being 1 for all of its rows. However, in our case, we can ”relax” the permutation matrix constraint by letting the sum of each row to 1 (by using softmax or hard-thresholded ReLU with Normalization), rather than forcing only a single element to be 1 in each row. This relaxation allows contribution of multiple pixels and blending, which warps the reference image to the target image with better performance, which can be seen in Tab. 1. Effects of different parameters of XABA on image alignment performance can be seen in Tab. 2.

Methods	PSNR
FlowNet-c [15] + warp	18.404
DISFlow [26] + warp	19.891
XABA (Softmax)	**27.920**

Table 1. PSNR results for different alignment algorithms. Ours has the highest PSNR.

4.3. Super-Resolution Performance

To show the effectiveness of XABA as a sub-network of a greater network, we conducted a multi-image super-resolution experiment. Two different images of the same scene is given as an input to a network to find x4 higher resolution image of the same scene. For this experiment, we selected XLSR [8] as the SISR baseline network. Then



(a) Original Reference Image

(b) Original Target Image

(c) Motion Image

(d) DISFlow

(e) FlowNet

(f) XABA(HT+N)

Figure 5. Example Images from Kitti Dataset for Image Alignment. Note that rather than distorting the reference image to match with the target image under planar 1-to-1 constraint, our proposed method prefers transferring a combination of features from the reference to the target image for alignment which better fits the image.

Model parameters		**PSNR**
# of interframe aligners	Block size	
1	10x10	20.004
1	20x20	22.873
2 (1,2)	10x10	23.151
2 (1,2)	20x20	26.090
3 (1,2,4)	10x10	25.149
3 (1,2,4)	20x20	**27.920**

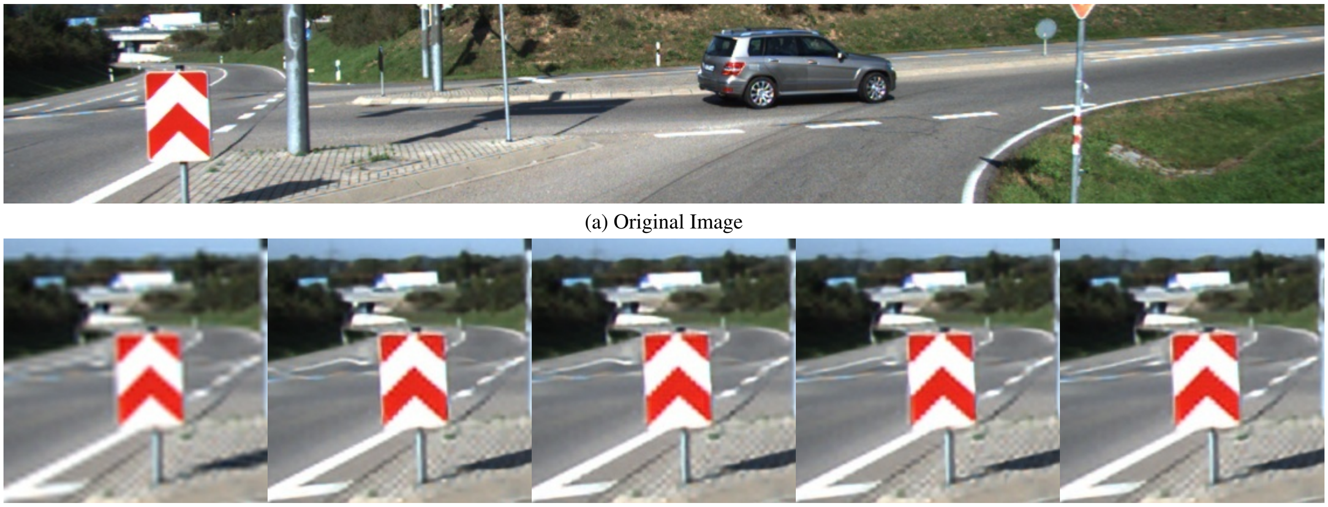
Table 2. PSNR values on Kitti test set for different model parameters. Values in the parenthesis in number of interframe aligners denote the downsample and upsample factors for each interframe aligner. (f_e, f_m) are chosen as (32,16) in interframe aligner block. Non-overlapping block size affects T2B and B2T operations. 3 interframe aligners with block size of 20 perform the best.

only by changing the input filters to accept two input RGB images, we constructed so called XLSR_MISR. By combining XLSR_MISR with DISFlow+Warp and XABA, we constructed XLSR_MISR + DIS + Warp and XLSR_MISR + XABA, respectively. Here for XABA, we used two different activation functions. PSNR values are calculated using the original high resolution images with RGB outputs.

As seen in Tab. 3, when an input tensor adjusted SISR method (XLSR_MISR) is combined with our XABA (HT+N), the performance is improved by 0.1dB. Furthermore, the SISR method is chosen to be with low number of parameters to limit its receptive field to a limited region. This causes degradation on the performance on MISR problem, as shown in XLSR_MISR without any alignment. The usage of correct alignment with XABA shows the effectiveness of our algorithm. Although DISFlow corrects and aligns the relevant data in the receptive field region, Kitti image pairs usually have large motion and large motions cannot be effectively compensated. An example output of the different alignment methods combined with the super-resolution network can be seen in Fig. 6.

4.4. Embedded Benchmarks Performance

In this experiment, we have investigated the computational performance of our proposed method XABA on an embedded computation device. For this purpose, NVIDIA Jetson AGX Xavier GPU [2] was used in the benchmark


(a) Original Image

(b) Bicubic

(c) No Alignment

(d) DISFlow

(e) XABA(Softmax)

(f) XABA(HT+N)

Figure 6. Example Images from Kitti Dataset. Note that the lanes are distorted on the other methods while XABA block helps the super-resolution network to better distinguish the lanes, which match with the original image better.

SR Type	Alignment	PSNR	Δ PSNR
Bicubic	×	23.965	-1.26
XLSR	×	25.223	0.00
XLSR_MISR	×	25.216	-0.01
XLSR_MISR	DISFlow + Warp	25.231	0.01
XLSR_MISR	XABA (Softmax)	25.289	0.07
XLSR_MISR	XABA (HT+N)	**25.323**	**0.10**

Table 3. PSNR values of Kitti test set used in super resolution (SR). In XABA, 3 interframe aligners are used with block size 20x20 and $(f_e, f_m) = (32, 16)$. Our method with hard threshold + normalize activation function (HT+N) outperforms the others. HT+N also outperforms Softmax in terms of timing performance for same block sizes (Tab. 4).

tests. To have a better understanding of the computational performance of the block, the high-performance computing benchmarks are also provided for reference. For the high-performance tests NVIDIA GeForce RTX 3080 GPU [1] was used. For the inference measurements PyTorch models were exported to ONNX file format, which were then converted to TensorRT engine using NVIDIA TensorRT-Command Line Wrapper, trtexec [5]. As for TensorRT, we used v7.1.3 and v8.2 for Jetson AGX Xavier GPU and GeForce RTX 3080, respectively. Jetson AGX Xavier GPU has CUDA cores. In this study, we preferred two power consumption modes, which are 15 Watts and 30 Watts for benchmarking on Jetson AGX Xavier GPU. The RTX 3080 GPU has 8704 CUDA cores and consumes 320 Watts as maximum power.

The inference benchmark results of image alignment models were separately produced with floating-point16 (FP16) and floating-point32 (FP32) operations. All the training has been conducted with FP32, as it is known that FP16 inference most of the time does not hurt the performance while being two times computationally light [3].

4.5. Timing Results and Power-Efficiency Analysis

The inference performances of different configurations of the proposed model are comparatively demonstrated in this section. Note that Tab. 4 indicates timing performances of global and local alignment models. It can be seen that using FP16 for inference is faster than using FP32. On the other hand, the global alignment inference performance can reach about 170 frame per seconds (FPS) on high performance computing device, which is included in the table as a reference to compare it with Jetson Xavier's performance.

As it can be seen from the Tab. 4, it is possible to run the proposed block in real-time on Jetson for some specific configuration such as XABA10fp16Soft which indicates the non-overlapping block size for T2B operator is 10 and FP16 is used for inference and activation in Attention module is softmax. Note that by using Hard-Thresholding we could increase the performance of the similar block using softmax in all of the cases and this simple change added almost +3FPS to most of the configurations. Note that local alignment section is also given in the table to get a better understanding of the effects of pyramidal processing and baseline block. Also note that local alignment block can be seen as a pyramid with 0 depth and can still have meaningful usages for aligning smaller displacements such as aligning frames of a video stream.

Fig. 7 presents FPS performance results of global align-

Timing Performance*,ms	Global Alignment Inference						Local Alignment Inference					
Units	**AGX (15 W)**		**AGX (30 W)**		**RTX 3080**		**AGX (15 W)**		**AGX (30 W)**		**RTX 3080**	
DataType&Models	*FP16*	*FP32*	*FP16*	*FP32*	*FP16*	*FP32*	*FP16*	*FP32*	*FP16*	*FP32*	*FP16*	*FP32*
XABA20fp32Soft	594.2	904.4	304.8	473.2	39.9	64.5	381.4	596.3	202.4	322.7	26.4	42.9
XABA10fp32Soft	368.5	556.9	186.6	282.2	22.1	41.7	210.5	339.5	110.9	176.9	14.2	25.4
XABA10fp16Soft	86.7	121.9	**43.5**	61.4	6.5	10.2	46.7	69.7	24.2	36.5	4.4	6.2
XABA20fp16Soft	146.7	215.1	74.5	111.2	9.9	15.9	91.6	140.7	48.1	76.4	7.4	10.5
XABA10fp16HTN	74.1	110.5	**38.6**	56.1	5.9	9.8	40.9	64.9	20.5	32.7	4.3	5.9
XABA20fp16HTN	120.2	193.6	61.1	102.7	8.9	15.3	72.6	126.9	36.5	65.4	6.2	12.4

*With the contributions of Alperen Kalay (alperenkalay@aselsan.com.tr), Aselsan Research.

Table 4. The Global & Local Alignment Inference Benchmark Results. The name of the model is encoded as XABA[block_size][fp32—fp16][Soft—HTN]. Since the Global Alignment Inference includes Local Alignment Inference computation, we investigate only Global Alignment Inference for real-time performance. The bold results indicate real-time performance.

ment inference models, while Fig. 8 demonstrates the power efficiency performances for global alignment models.

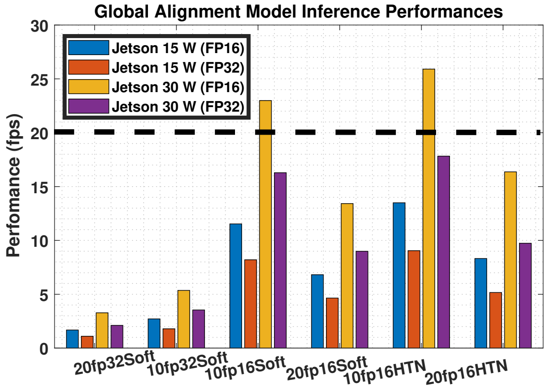


Figure 7. The global alignment inference performance benchmarks in terms of FPS for different block configurations.

The computing performance of XABA10fp16Soft inference model meets real-time requirement in terms of FPS for image alignment processing according to the results in Fig. 7. Indeed, the bottleneck analysis was performed with NVIDIA Profiler for this model. According to bottleneck analysis, the computing time of softmax layer showed that this layer has dominant computation compared to other layers. This inspired our Hard-Thresholding activation proposal which replaces softmax function in activation function. The global alignment inference performance has been increased to 26 FPS from 23 FPS by using hard thresholding and FP16 precision in XABA10fp16HTN. This performance result provides 10% speedup compared to previous inference model (XABA10fp16Soft).

The power efficiency experiments proved that Jetson AGX GPU provides the most power efficient computation with respect to comparative results in Fig. 8. According to Fig. 8, the Jetson AGX GPU performance provides about 1.6x power efficiency compared to RTX3080 GPU under FP16 precision. From this point of view, this edge device meets real-time image alignment processing requirement in FP16 inference case with 30 Watts power consumption. Although RTX 3080 GPU has high computation performance, its power efficiency performance gave drastically lower result compared to Jetson AGX GPU.

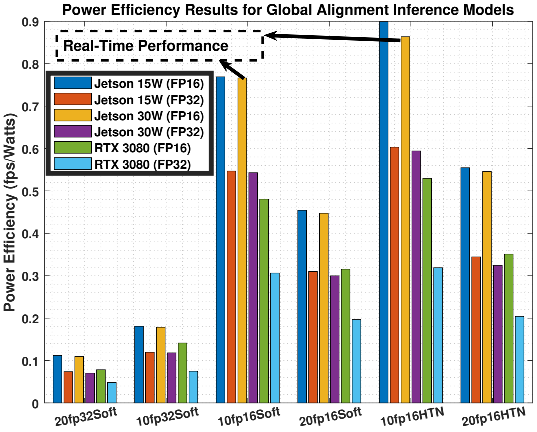


Figure 8. The global alignment inference power efficiency for different computation designs.

5. Conclusion

In this study, we proposed our multi-purpose, cross-attention based image alignment block, XABA. By processing the images in blocks inside a pyramidal block based alignment structure, we capture local relationships with minimal computational need. Focusing on efficiency, we further prove with tests that XABA can run in real-time on edge devices such as NVIDIA Jetson Xavier.

Our experiments reveal that XABA can outperform common optical-flow based alignment methods. We have also shown that XABA can be used as a sub-network aligner in larger deep-learning based scenarios like single and multi-image super-resolution with good performance. Embedded benchmarks and power analyses further prove that pyramidal structure of XABA allows us to realize a power-efficient image aligner.

References

[1] NVIDIA GeForce RTX 3080 GPU. https://www.nvidia.com/tr-tr/geforce/graphics-cards/30-series/rtx-3080_msm_moved. Accessed: 2022-03. 7

[2] NVIDIA Jetson AGX Xavier GPU. https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit. Accessed: 2022-03. 6

[3] NVIDIA Mixed Precision. https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html. Accessed: 2022-03. 7

[4] NVIDIA TensorRT. https://developer.nvidia.com/tensorrt. Accessed: 2022-03-07. 2

[5] NVIDIA TensorRT Command-Line Wrapper. https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html#trtexec. Accessed: 2022-03. 7

[6] OpenVINO. https://docs.openvino.ai/latest/index.html. Accessed: 2022-03. 2

[7] Davide Abati, Amir Ghodrati, and Amirhossein Habibian. Efficient video super resolution by gated local self attention. *2021 British Machine Vision Conference (BMVC)*, 2021. 2, 3

[8] Mustafa Ayazoglu. Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. *CoRR*, abs/2105.10288, 2021. 5

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 2, 3

[10] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, pages 25–36, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 2

[11] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, 2005. 3

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 3

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[14] Alana de Santana Correia and Esther Luna Colombini. Attention, please! a survey of neural attention models in deep learning, 2021. 2

[15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 1, 2, 3, 5

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 2

[17] Chaoxu Guo, Bin Fan, Jie Gu, Qian Zhang, Shiming Xiang, Veronique Prinet, and Chunhong Pan. Progressive sparse local attention for video object detection, 2019. 3

[18] Zhou Haiyun, Xiang Xuezhi, Zhang Rongfang, Zhai Mingliang, and Syed Masroor Ali. Learning optical flow via deformable convolution and feature pyramid networks. In *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pages 26–30, 2019. 2

[19] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. 3

[20] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Robust image alignment with multiple feature descriptors and matching-guided neighborhoods. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2015. 2

[21] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3

[22] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation, 2020. 3

[23] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017. 3

[24] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3

[25] Nikhil Iyer, V. Thejas, Nipun Kwatra, Ramachandran Ramjee, and Muthian Sivathanu. Wide-minima density hypothesis and the explore-exploit learning rate schedule. *CoRR*, abs/2003.03977, 2020. 5

[26] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. volume 9908, pages 471–488, 10 2016. 1, 3, 5

[27] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, volume 81, 04 1981. 2, 3

[28] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015. 2, 3

[29] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[30] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Space-time memory networks for video object segmentation with user guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):442–455, 2022. 3

[31] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038, 2013. 2

[32] Miftahur Rahman, Xueyuan Li, and Xufeng Yin. Dl-ransac: An improved ransac with modified sampling strategy based on the likelihood. In *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pages 463–468, 2019. 2

[33] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models, 2019. 3

[34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 1, 2

[35] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 3, 4

[36] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 2, 3

[37] Javier Sánchez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, 3:137–150, 07 2013. 3

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. 2, 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 3

[40] Longguang Wang, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning parallax attention for stereo image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12242–12251, 2019. 1, 3

[41] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1954–1963, 2019. 1

[42] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2, 3

[43] Weilei Wen, Wenqi Ren, Yinghuan Shi, Yunfeng Nie, Jingang Zhang, and Xiaochun Cao. Video super-resolution via a spatio-temporal alignment network. *IEEE Transactions on Image Processing*, 31:1761–1773, 2022. 1, 3

[44] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 1, 3

[45] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton Van Den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019. 1

[46] Zhuoqian Yang, Tingting Dan, and Yang Yang. Multi-temporal remote sensing image registration using deep convolutional features. *IEEE Access*, 6:38544–38555, 2018. 2

[47] Yongyi Yu, Mingzhe Liu, Huajun Feng, Zhihai Xu, and Qi Li. Split-attention multi-frame alignment network for image restoration. *IEEE Access*, 8, 2020. 1, 2, 3

[48] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10073–10082, 2020. 3