MobileFuse: Multimodal Image Fusion at the Edge

H. Perreault[1], B. Debaque[1], R. David[1], M-A. Drouin[2], N. Duclos-Hindié[1], and S. Roy[3]

[1]Thales Group, Thales Digital Solutions, Québec, Canada, hughes.perreault@thalesdigitalsolutions.ca
[2]National Research Council of Canada, Ottawa, Canada, Marc-Antoine.Drouin@nrc-cnrc.gc.ca
[3]DRDC Valcartier Research Centre, Québec, Canada, Simon.Roy@drdc-rddc.gc.ca

***Abstract*—The fusion of multiple images from different modalities is the process of generating a single output image that combines the useful information of all input images. Ideally, the information-rich content of each input image would be preserved, and the cognitive effort required by the user to extract this information should be smaller on the fused image than the one required to examine all images. We propose MobileFuse, an edge computing method targeted at processing large amount of imagery in a bandwidth limited environment using depthwise separable Deep Neural Networks (DNNs). The proposed approach is a hybrid between generative and blending based methods. Our approach can be applied in various fields which require low latency interaction with the user or with an autonomous system. The main challenge in training DNNs for image fusion is the sparsity of data with representative ground truth. Registering images from different sensors is a major challenge in itself, and generating a ground truth from them is another massive one. For this reason, we also propose a multi-focus and multi-lighting framework to generate training dataset using unregistered images. We show that our edge network can perform faster than its state-of-the-art baseline, while improving the fusion quality.**

***Keywords*-image fusion; multimodality; edge-based computing**

I. Introduction

The various advancements in computer vision and artificial intelligence in the last decade has caused a growing interest in their use for multiple applications. However, these algorithms are often greedy in computing power and memory. For applications that require mobility, like augmented reality or self-flying drones, this is a problem since this computing power comes at the prices of weight and volume. The use of edge-computing becomes essential, since sending data away for remote computing would cause much bandwidth transmission and latency. In this work, we explore power-efficient visible and thermal image fusion.

Multi-modality image fusion is the process of generating a single output image that combined the useful information of all input ones. Ideally, the information-rich content of each input image would be preserved, and the cognitive effort required by the user to extract this information should be smaller on the fused image than the one required to examine all images. In the full pipeline, images from different sensors would have to be registered [1], [2], [3], [4], [5], [6] before being fed to a fusion network. The proposed approach focuses on multimodal image fusion, and assumes pre-registered input images. Note that, the training method we propose does not require a registered image dataset.

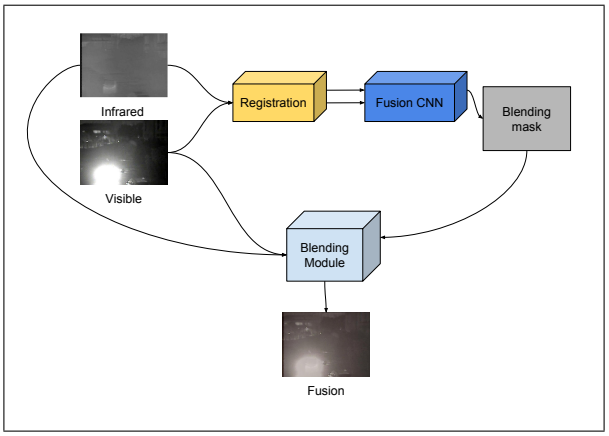


Figure 1: An overview of our method.

Deep neural networks (DNNs) have become a popular tool for multimodal image fusion, because of the quality of the complex representation they learn from data. DNNs can be trained on a dataset of multimodal images and their corresponding ground truth, and then used to fuse new images at test time. The principal difficulty about this is the availability of the ground-truth, as it often does not exist. To alleviate this problem, we develop a procedure to generate a novel multi-focus and multi-lighting dataset from Pascal VOC [7], detailed in section III-B. Our dataset generation procedure can be replicated with any dataset with semantic segmentation ground-truth, which makes it very practical and easy to replicate to other domains.

Additionally, we present MobileFuse, a fast and light multimodal image fusion network. MobileFuse is based on MobileNet [8], and thanks to depthwise separable convolutions, can reach the very low number of 33K parameters. It can run at 150 fps on GPU, and 30 fps on CPU. As it is similar in architecture, IFCNN [9] is used as a baseline for comparing the fusion quality and speed improvement.

Furthermore, most deep fusion methods are purely generative, meaning that the network directly outputs the resulting image. Although they usually perform well, they are not exempt from making mistakes, as the resulting image is

based on training data. This could potentially lead to a lack of trust in the fusion results from the user. To improve this weakness, and prevent the network from hallucinating altogether, we introduce a novel blending module. The module uses a different output strategy than purely generative methods. Instead of training the network to output an image directly, we train the network to generate a blending map M such that:

$$Output = (image_1 \times M) + (image_2 \times (1 - M)), \quad (1)$$

where the blending operation is done inside the network, and the loss is based on the resulting image, and not the map itself. The results from this module are more trustworthy than generative methods, as well as producing images with a higher fusion quality, as shown in Table II and Table III.

The resulting pipeline (seen in Figure 1) is a light fusion CNN that takes two images as inputs, extracts a latent representation from them with a multi-stream architecture, outputs a blending mask, generates an output image from the blending mask, and is trained end-to-end on the generated fusion.

The contributions in this paper are:

1) We propose MobileFuse, a novel deep light network architecture to perform image fusion, and demonstrate its quality against our baseline.
2) We present a novel output strategy for image fusion, which is a hybrid between generative and blending based methods.
3) We introduce a new procedure to generate fusion datasets with corresponding ground-truths, as long as the semantic segmentation is available.

The rest of the paper is organized as follows: related work are first presented, then our method and implementation details are detailed, followed by the experiments descriptions, results, and conclusion.

II. Related work

A. Image fusion

The following is an overview of techniques used to perform multimodal image fusion:

The **encoder-decoder** architectures [10], [11], [12], [13] are commonly used for image fusion. They consist of using a CNN or other method to extract features from the input images, and then combining the features using a decoder network. To ensure that the decoder learns something useful, the latent, or intermediary representation, has to be smaller than the input. This is the encoding part. The decoder network is then trained to minimize the difference between the fused decoded output and the ground truth.

The **multi-stream** architectures [14], [15], [9], [16] process each input image separately and combine the outputs in a final stage. The streams can either share parameters, or be trained completely separately, depending on the situation. This approach allows the processing of each modality individually, which could be better parallelized than other frameworks. It also gives the advantage of letting each stream of the network learn the correct feature extraction mechanisms for each modality, which might be very different. In the case of infrared and visible image fusion for instance, the texture, contrast, and saliency recognition is extremely different in each stream.

In the **multi-task learning** approach, the network is trained to perform multiple tasks simultaneously, such as image fusion and image segmentation [17]. This approach can improve the overall performance, as the loss on other tasks can often help the network avoid overfitting and learn more general latent representations. It can also be useful for real-time processing, if other tasks are needed anyway. Other notable works include [18], [19], [20].

Different **loss functions** can be used to train the network, and often have a great impact on the end result. Using a loss which is not representative enough of the task can be devastating for the results. For example, a combination of mean squared error and perceptual loss to train the network has been used [9]. Other work bringing originality in using loss functions include [21], [22], [23], [24]

Many variations and combinations of these frameworks have been proposed to improve fusion quality, such as incorporating attention mechanisms [25], [26], [27], [28] and adversarial training [29], [30], [31], [32], [33], [34].

B. Speeding-up deep neural networks

The following is an overview of techniques used to speed up neural networks:

Model pruning is a technique to reduce the size and complexity of neural networks by removing neurons and connections that have minimal impact on the network's accuracy. The idea is to simplify the network in order to get a good trade-off between speed and quality. The network is first trained, then the importance of each neuron is evaluated based on criteria such as the weight's magnitude or activation. Based on a threshold, the least impactful neurons are then removed, reducing the network's size and computational complexity. Pruning can result in significant reduction in model size and computation, but can also cause accuracy loss, so validation is necessary in order to reach the desired trade-off. The most notable example in this field include [35], [36], [37], [38], [39].

Quantization is a technique used to reduce the memory and computational requirements of DNNs by converting the network's parameters from high-precision floating point values to integer values. This reduces the number of bits required to represent each parameter, enabling the deployment of larger and more complex networks on edge devices. Quantization can also result in faster computation, as low-precision arithmetic operations are typically faster than high-precision operations. However, quantization can also lead to

accuracy loss, as the reduced precision can result in rounding errors and loss of information. To mitigate this, various quantization techniques have been proposed [40], [41], [42], [43].

Network architecture design for speeding up neural networks consists of selecting and arranging the building blocks of a DNN to reduce its computational and memory requirements while keeping its accuracy as high as possible. This often involves choosing lightweight building blocks, such as depthwise separable convolutions, or reducing the number of layers. The choice of activation functions, loss functions, and optimizers can also impact the speed of the network. The design process often requires trade-offs between speed and accuracy, and is often an experimental process, with models being modified and refined based on results. By designing efficient network architectures [8], [44], [45], [46], [47], [48], [1], it is possible to significantly speed up neural network training and inference.

III. Methodology

A. Mobile fusion network

When developing our network, we had three objectives in mind: making it faster, lighter, and improving the fusion quality. To tackle the first and second objectives, we adapted one of the most popular edge network available, MobileNet. MobileNet introduced the depthwise separable convolution, which divides the standard 2D convolution with a 3D filter into two operations, one 2D convolution with a filter of depth one (in other words, a 2D filter), followed by a 1×1 convolution of depth N, N being the depth of feature map to be processed. Using these as building blocks, we developed a classic multi-stream architecture, seen in Figure 2. Our network uses MAX as the fusion operation, as it proved to be best by test.

In some fusion deep networks, we noticed a color fidelity issue that was making the fused image unnatural to the eye. In order to alleviate this issue and to further improve the fusion quality, we developed a blending module. The blending module takes as input a blending mask produced by the network, normalized between 0 and 1, and both input images, and combines them using Equation 1. The output of this module is the generated fusion image. This operation is fully differentiable, and the network is trained on the resulting output image, not the mask itself. The loss is a simple MSE. Some variations of this blending module are evaluated in an ablation study (shown in table II and III).

B. Fusion dataset generation

To train the network, a large quantity of fusion images with ground-truth are needed. These images are not easily available, thus the need to create a new dataset of our own. The difficulty arises from the fact that not only visible to thermal registration is challenging, but even with perfectly aligned images, the best fusion is subjective, and actually depends on the task. We developed a procedure that creates input images in such a way that the network can learn to recognize high quality parts of each input.

From Pascal VOC [7], we generated both a multi-lighting and multi-focus fusion dataset (see Figure 3).

For the multi-lighting part, we created images to be fused by increasing or decreasing the pixel values. This generated an over-saturated image, and a darkened one that were used as inputs to the CNN. For the multi-focus part, we use instance segmentation annotations to create a foreground / background separation. From this separation, we could blur the rest using a Gaussian filter, thus creating two images to be fused, the foreground and the background. These two images can be fed to our multi-stream CNN. In both cases, the original image is used as ground-truth.

In the feature extraction step, one could expect that the neural network learns to capture the most sharp and textured parts of the image. Thus, the network is expected to prioritize the input with the most content-rich information, whatever the modalities used for training. In our experiments related to visible and thermal image fusion, this approach performs well.

C. Implementation details

The network was implemented in PyTorch [49] and trained with PyTorch Lightning [50]. From our dataset, a training, validation and testing set were fixed in advance. To optimize data diversity, multi-focus and multi-lightning images were balanced in each training batches. The hardware used in this research is an Intel® Core™ i7-8750H CPU, an NVIDIA GeForce GTX 1060 Mobile for the GPU, and a Raspberry Pi 3 Model B+ for the edge device.

IV. Experiments

A. Datasets

We perform our experiments on two thermal and visible image datasets, VIFB [51] and M3FD [52], which are public and open-source. Both of them have a completely different domain from our training data, which do not even contain any thermal imagery.

Our model already was trained, validated, and tested on our novel Pascal VOC Fusion dataset, but to avoid over fitting, the fusion experiments were performed on real visible-thermal datasets without ground-truth. For this reason, we used an aggregation of metrics commonly used in the literature.

B. Metrics

Defining meaningful metrics for image fusion quality is a challenging issue. Often times, the metrics used do not seem to be particularly useful, or methods can be aggressively optimized for a given metric. For this reason, numerous metrics are used in this work and aggregated to determine the best methods. Even the aggregation strategies could be

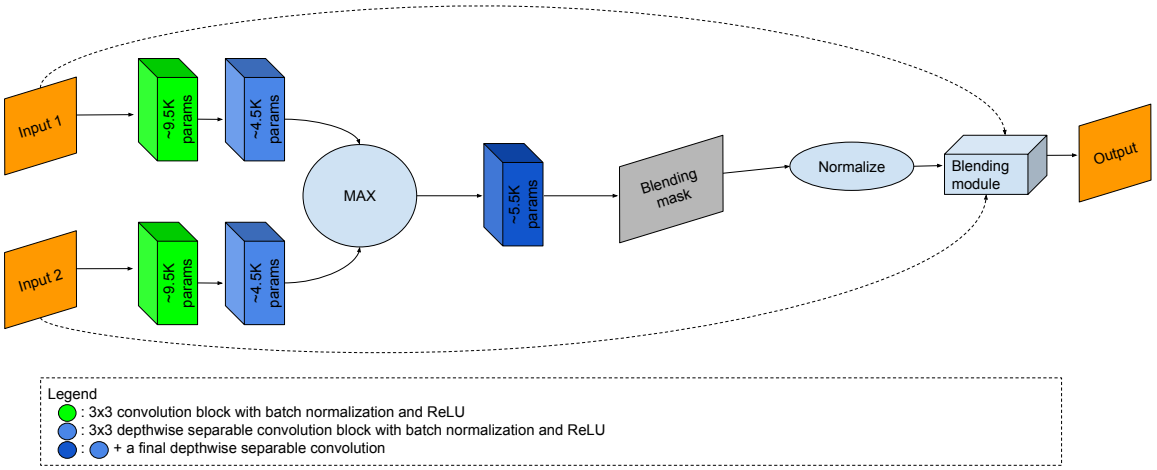
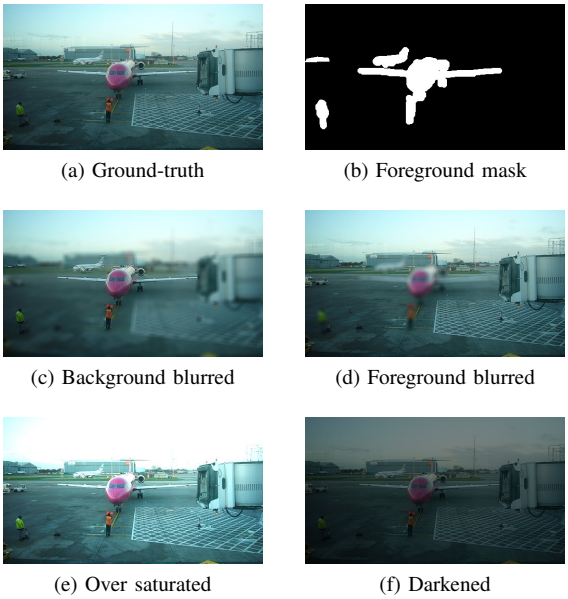


Figure 2: An overview of our network. The blending module consists of Equation 1. Better seen in color. The number of parameters for each component of the proposed network are given.



(a) Ground-truth (b) Foreground mask

(c) Background blurred (d) Foreground blurred

(e) Over saturated (f) Darkened

Figure 3: An example of the image transformations applied on Pascal VOC [7] to create our dataset.

debatable, in our case, we ended using an average ranking (the rank on each metric, averaged). Note that for some metrics, large values indicate higher qualities, while for others smaller values are preferable. The metrics used in this work are presented in Table I:

C. Fusion experiment

The methods were first trained on our Pascal VOC fusion dataset, with a full cycle of validation and testing. In a later stage, they were fed the VIFB and M3FD images without having seen them in the first phase. The resulting fused images are then evaluated with our metrics. Since we use so many metrics, the aggregation strategy is to use the average ranking on each metric, for each method. For example, if a method obtains first rank on one metric, then third on another one, and fourth on the final one, the average ranking would be $(1 + 3 + 4)/3 = 2.\overline{6}$. The results are shown in Table II and Table III.

Acronym	name	↑ / ↓
AG	Average gradient	↑
CE	Cross entropy	↓
EI	Edge intensity	↑
MI	Mutual information	↑
PSNR	Peak signal-to-noise ration	↑
RMSE	Root mean squared error	↓
SF	Spatial frequency	↑
SSIM	Structural similarity index measure	↑
SD	Standard deviation	↑

Table I: A description of the metrics used for evaluation. ↑ and ↓ mean greater or smaller is better, respectively.

D. Benchmarking

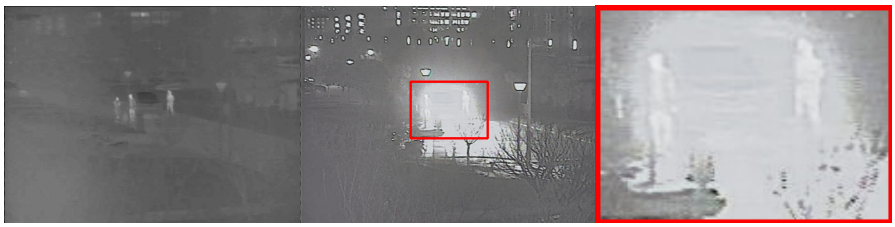
For the speed benchmarking, results are shown in Table IV. The models were compared in the same conditions and hardware. For the CPU and the Raspberry Pi 3 B+, the methods were wrapped into an ONNX [53] file, and ran on the device using ONNX Runtime.

For the bandwidth benchmarking, results are shown in Table V. Four scenarios are explored, and their bandwidth cost in MB/s are shown, both for the upload and the download. Experiments were conducted using the average input and output image size from the entire M3FD dataset, and assuming a fixed 30 fps frame rate. The four explored scenarios are remote and edge computing, paired with the constraint of needing the result remotely, and needing the result locally.

Visible image Ours Ours (zoomed)

Infrared image IFCNN IFCNN (zoomed)

Visible image Ours Ours (zoomed)



Infrared image IFCNN IFCNN (zoomed)

Visible image Ours Ours (zoomed)

Infrared image IFCNN IFCNN (zoomed)

Figure 4: Qualitative results on the VIFB (row 1-4) and M3FD (row 5-6) dataset.

Table II: Results on the VIFB dataset. Best result in **bold**.

Methods\Metrics	AG ↑	CE ↓	EI ↑	EN ↑	MI ↑	psnr ↑	rmse ↓	SF ↑	ssim ↑	SD ↑	avg. rank ↓
wavelet-max	3.903	2.210	8.411	6.781	1.128	45.315	0.138	10.686	1.199	43.479	3.000
IFCN	4.874	2.213	8.234	6.668	1.103	45.371	0.135	12.414	1.164	37.385	3.800
Blending only	3.350	1.666	8.428	6.712	1.114	45.934	0.124	10.154	1.228	34.608	2.700
Generative only	4.545	2.214	8.243	6.703	1.110	45.478	0.134	11.976	1.171	39.492	3.300
MobileFuse (ours)	3.724	1.909	8.451	6.815	1.114	45.808	0.127	10.806	1.210	37.589	**2.200**

Table III: Results on the M3FD dataset. Best result in **bold**.

Methods\Metrics	AG ↑	CE ↓	EI ↑	EN ↑	MI ↑	psnr ↑	rmse ↓	SF ↑	ssim ↑	SD ↑	avg. rank ↓
wavelet-max	3.432	1.602	9.488	6.508	1.176	45.683	0.056	10.254	1.436	31.475	3.0
IFCNN	4.378	2.173	9.536	6.532	1.146	45.751	0.054	11.712	1.420	27.505	3.2
Blending only	2.606	1.467	9.440	6.460	1.140	46.491	0.051	8.939	1.466	26.626	3.2
Generative only	5.799	1.882	9.682	6.583	1.137	45.803	0.178	14.691	1.321	27.517	2.9
MobileFuse (ours)	3.148	1.803	9.638	6.603	1.134	46.177	0.052	9.976	1.452	28.132	**2.7**

Table IV: Results of our speed and memory experiments. Best result in **bold**. All results are produced with 256×256 images for memory constraints on the edge device.

Metric \Method	IFCNN	MobileFuse (ours)
FPS (GPU) ↑	134	**152**
FPS (CPU) ↑	16	**30**
FPS (Rasp. Pi 3B+) ↑	0.7	**1.5**
# params. ↓	∼130K	∼**33K**

Table V: Results of our bandwidth transmission experiments. All results assume a frame rate of 30 fps. Best result in **bold**.

	Remote computing		Edge computing	
Bandwidth cost (MB/s)	upload	download	upload	download
Result needed on device	51.17	31.01	**0**	**0**
Result needed remotely		0	**31.01**	**0**

V. Results

A. Discussion

As we can see thanks to the quantitative evaluation in Table II and Table III, our method seems to generate more useful information than the baseline network IFCNN, as well as the wavelet fusion method.

Additionally, we demonstrate the utility of the blending module by training two other versions of our network, the blending only and generative only. The generative only approach is trained to directly output the ground-truth image, without using the blending module. The blending only approach is trained to output a blending mask, by using the foreground mask as ground-truth. As we can see in Figure 3, if $image_1$ is (c) and $image_2$ is (d), then the perfect blending mask would be (b). To clarify further, the main difference between MobileFuse and the blending only version is that MobileFuse is trained with the loss on the generated image, and the blending only is trained with the loss on the blending mask. Overall, the quantitative results of both of these approaches demonstrate that using the hybrid approach of the differentiable blending module works better.

Furthermore, our method runs slightly faster than IFCNN on GPU, and close to 200% faster on CPU and on an edge device (seen in Table IV), gaining vital time which could be used to perform other useful computation, or just performs smoother rendering of the scene.

It also goes without saying that Table V demonstrates the bandwidth gains one can achieve using edge computing. Even in the worst scenario where the result of the fusion would have to be transferred back, not having to transfer the inputs for remote computing achieves a gain of 165% of total bandwidth. Of course, the device does not have to transfer back the result in the case of remote computing, since the server already computed it.

Beyond the quantitative evaluation, Figure 4 shows some relevant qualitative examples on the VIFB and M3FD dataset. We can observe that our method performs just as well as or better than IFCNN for vital parts of the fusion, but also renders the background of the scenes much better. On the tested imagery, IFCNN seems to have a bias for averaging the inputs and tends to incorporate much more infrared than visible everywhere, causing an impression of over saturation in the fusion image. For example, the trees are more natural and better contrasted with our method, and fusion images seem to be less grainy. The example on rows 5-6 demonstrates the better conservation of color fidelity of MobileFuse.

VI. Conclusion

We introduce MobileFuse, a lightweight image fusion method based on a depthwise separable deep network. The proposed method is suitable for edge computing in bandwidth limited environments. It is a hybrid between generative and blending based methods. On a low-power platform, the proposed method is about 200% faster than IFCNN and on average outperforms IFCNN. Different strategies for further speeding up MobileFuse are discussed. We also propose a strategy for creating image fusion datasets using multi-focus and multi-lighting. This strategy for creating training datasets does not require registered thermal and visible

imagery pairs. Future work might include further speed and memory improvements. We also expect to explore human-centric quality metrics.

REFERENCES

[1] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "An improved one millisecond mobile backbone," *arXiv preprint arXiv:2206.04040*, 2022.

[2] G.-A. Bilodeau, A. Torabi, P.-L. St-Charles, and D. Riahi, "Thermal–visible registration of human silhouettes: A similarity measure performance evaluation," *Infrared Physics & Technology*, vol. 64, pp. 79–86, 2014.

[3] V. Sanchez, G. Prince, J. P. Clarkson, N. M. Rajpoot, *et al.*, "Registration of thermal and visible light images of diseased plants using silhouette extraction in the wavelet domain," *Pattern Recognition*, vol. 48, no. 7, pp. 2119–2128, 2015.

[4] A. Torabi, G. Massé, and G.-A. Bilodeau, "An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 210–221, 2012.

[5] B. Debaque, H. Perreault, J.-P. Mercier, M.-A. Drouin, R. David, B. Chatelais, N. Duclos-Hindie, and S. Roy, "Thermal and visible image registration using deep homography," in *2022 25th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2022.

[6] T. Pouplin, H. Perreault, B. Debaque, M. Drouin, N. Duclos-Hindie, and S. Roy, "Multimodal deep homography estimation using a domain adaptation generative adversarial network," in *2022 IEEE International Conference on Big Data (Big Data)*, pp. 3635–3641, IEEE, 2022.

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[9] Y. Zhang, Y. Liu, P. Sun, H. Yan, X. Zhao, and L. Zhang, "Ifcnn: A general image fusion framework based on convolutional neural network," *Information Fusion*, vol. 54, pp. 99–118, 2020.

[10] L. Jian, X. Yang, Z. Liu, G. Jeon, M. Gao, and D. Chisholm, "Sedrfuse: A symmetric encoder–decoder with residual block network for infrared and visible image fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2020.

[11] H. Xu, M. Gong, X. Tian, J. Huang, and J. Ma, "Cufd: An encoder–decoder network for visible and infrared image fusion based on common and unique feature decomposition," *Computer Vision and Image Understanding*, vol. 218, p. 103407, 2022.

[12] L. Jian, X. Yang, Z. Liu, G. Jeon, M. Gao, and D. Chisholm, "A symmetric encoder-decoder with residual block for infrared and visible image fusion," *arXiv preprint arXiv:1905.11447*, 2019.

[13] J.-L. Yin, B.-H. Chen, Y.-T. Peng, and C.-C. Tsai, "Deep prior guided network for high-quality image fusion," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2020.

[14] X. Zhu, S. Li, Y. Gan, Y. Zhang, and B. Sun, "Multi-stream fusion network with generalized smooth l 1 loss for single image dehazing," *IEEE Transactions on Image Processing*, vol. 30, pp. 7620–7635, 2021.

[15] Y. Hu, M. Lu, and X. Lu, "Driving behaviour recognition from still images by using multi-stream fusion cnn," *Machine Vision and Applications*, vol. 30, pp. 851–865, 2019.

[16] D. LEI, J. DU, L. ZHANG, and W. LI, "Multi-stream architecture and multi-scale convolutional neural network for remote sensing image fusion," *Journal of Electronics & Information Technology*, vol. 44, no. 200792, p. 237, 2022.

[17] Y. Liu, F. Mu, Y. Shi, and X. Chen, "Sf-net: A multi-task model for brain tumor segmentation in multimodal mri via image fusion," *IEEE Signal Processing Letters*, vol. 29, pp. 1799–1803, 2022.

[18] L. Qu, S. Liu, M. Wang, and Z. Song, "Transmef: A transformer-based multi-exposure image fusion framework using self-supervised multi-task learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 2126–2134, 2022.

[19] A. Fang, X. Zhao, and Y. Zhang, "Cross-modal image fusion guided by subjective visual attention," *Neurocomputing*, vol. 414, pp. 333–345, 2020.

[20] Q. Zhang and M. D. Levine, "Robust multi-focus image fusion using multi-task sparse representation and spatial context," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2045–2058, 2016.

[21] Y. Qi, S. Zhou, Z. Zhang, S. Luo, X. Lin, L. Wang, and B. Qiang, "Deep unsupervised learning based on color un-referenced loss functions for multi-exposure image fusion," *Information Fusion*, vol. 66, pp. 18–39, 2021.

[22] S. Eghbalian and H. Ghassemian, "Multi spectral image fusion by deep convolutional neural network and new spectral loss function," *International Journal of Remote Sensing*, vol. 39, no. 12, pp. 3983–4002, 2018.

[23] C. Cheng, C. Sun, Y. Sun, and J. Zhu, "Stylefuse: An unsupervised network based on style loss function for infrared and visible image fusion," *Signal Processing: Image Communication*, vol. 106, p. 116722, 2022.

[24] Z. Zhu, X. Yang, R. Lu, T. Shen, X. Xie, and T. Zhang, "Clf-net: Contrastive learning for infrared and visible image fusion network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[25] B. Yang and S. Li, "Visual attention guided image fusion with sparse representation," *Optik*, vol. 125, no. 17, pp. 4881–4888, 2014.

[26] J. Li, H. Huo, C. Li, R. Wang, and Q. Feng, "Attentionf-

gan: Infrared and visible image fusion using attention-based generative adversarial networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 1383–1396, 2020.

[27] H. Li, X.-J. Wu, and T. Durrani, "Nestfuse: An infrared and visible image fusion architecture based on nest connection and spatial/channel attention models," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9645–9656, 2020.

[28] J. Liu, X. Fan, J. Jiang, R. Liu, and Z. Luo, "Learning a deep multi-scale feature ensemble and an edge-attention guidance for image fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 105–119, 2021.

[29] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-gan: An unsupervised pan-sharpening method for remote sensing image fusion," *Information Fusion*, vol. 62, pp. 110–120, 2020.

[30] H. Xu, J. Ma, and X.-P. Zhang, "Mef-gan: Multi-exposure image fusion via generative adversarial networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 7203–7216, 2020.

[31] H. Zhang, Z. Le, Z. Shao, H. Xu, and J. Ma, "Mff-gan: An unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion," *Information Fusion*, vol. 66, pp. 40–53, 2021.

[32] H. Zhang, J. Yuan, X. Tian, and J. Ma, "Gan-fm: Infrared and visible image fusion using gan with full-scale skip connection and dual markovian discriminators," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1134–1147, 2021.

[33] Q. Li, L. Lu, Z. Li, W. Wu, Z. Liu, G. Jeon, and X. Yang, "Coupled gan with relativistic discriminators for infrared and visible images fusion," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7458–7467, 2019.

[34] Y. Wang, S. Xu, J. Liu, Z. Zhao, C. Zhang, and J. Zhang, "Mfif-gan: A new generative adversarial network for multi-focus image fusion," *Signal Processing: Image Communication*, vol. 96, p. 116295, 2021.

[35] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *arXiv preprint arXiv:1810.05270*, 2018.

[36] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *arXiv preprint arXiv:1710.01878*, 2017.

[37] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, and M. Jaggi, "Dynamic model pruning with feedback," *arXiv preprint arXiv:2006.07253*, 2020.

[38] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

[39] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[40] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5918–5926, 2017.

[41] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[42] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-s. Hua, "Quantization networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7308–7316, 2019.

[43] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *International conference on machine learning*, pp. 2849–2858, PMLR, 2016.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[45] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

[46] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

[47] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[48] D. Li, X. Wang, and D. Kong, "Deeprebirth: Accelerating deep neural network execution on mobile devices," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[49] A. e. a. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[50] W. Falcon et al., "Pytorch lightning," *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, vol. 3, 2019.

[51] X. Zhang, P. Ye, and G. Xiao, "Vifb: A visible and infrared image fusion benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 104–105, 2020.

[52] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo, "Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5802–5811, June 2022.

[53] J. Bai, F. Lu, K. Zhang, *et al.*, "Onnx: Open neural network exchange." https://github.com/onnx/onnx, 2019.