6 Appendix

6.1 Architecture of RefineNet

Following the previous work [41], we design a RefineNet with an encoder-decoder architecture similar to U-Net and a context extractor. The context extractor and encoder part have similar architectures, consisting of four convolutional blocks, and each of them is composed of two 3×3 convolutional layers, respectively. The decoder part in the FusionNet has four transpose convolution layers.

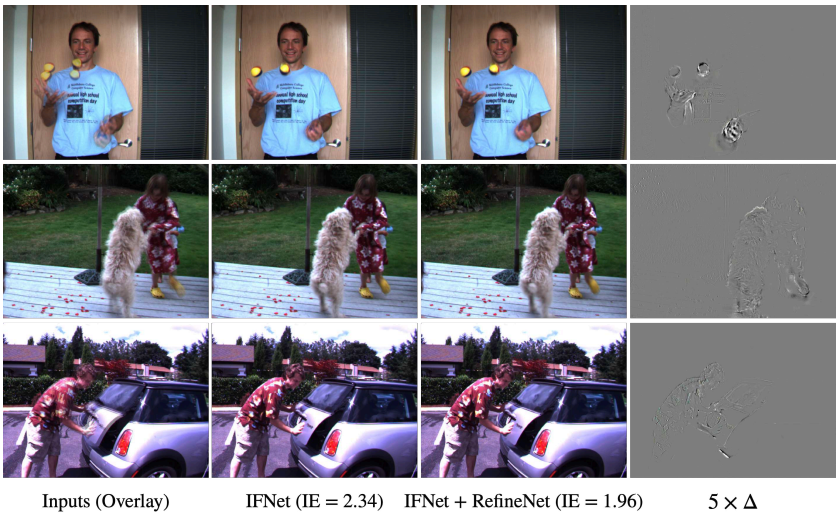

Fig. 10: **Visualization of the effect of RefineNet on M.B. [2] benchmark.**

Specifically, the context extractor first extracts the pyramid contextual features from input frames separately. We denote the pyramid contextual feature as C_0: $\{C_0^0, C_0^1, C_0^2, C_0^3\}$ and C_1: $\{C_1^0, C_1^1, C_1^2, C_1^3\}$. We then perform backward warping on these features using estimated intermediate flows to produce aligned pyramid features, $C_{t\leftarrow 0}$ and $C_{t\leftarrow 1}$. The origin frames I_0, I_1, warped frames $\widehat{I}_{t\leftarrow 0}, \widehat{I}_{t\leftarrow 1}$, intermediate flows $F_{t\rightarrow 0}, F_{t\rightarrow 1}$ and fusion mask M are fed into the encoder. The output of $i-th$ encoder block is concatenated with the $C_{t\leftarrow 0}^i$ and $C_{t\leftarrow 1}^i$ before being fed into the next block. The decoder parts finally produce a reconstruction residual Δ. And we will get a refined reconstructed image $clamp(\hat{\mathbf{I}}_t + \Delta, 0, 1)$, where $\hat{\mathbf{I}}_t$ is the reconstruct image before the RefineNet. We show some visualization results in Figure 10. RefineNet seems to make some uncertain areas more blurred to improve quantitative results.

6.2 Selection of Building Operators

We focus on introducing a simplified VFI pipeline without bells and whistles. So we choose building operators with intentional restraint. Exploring model com-

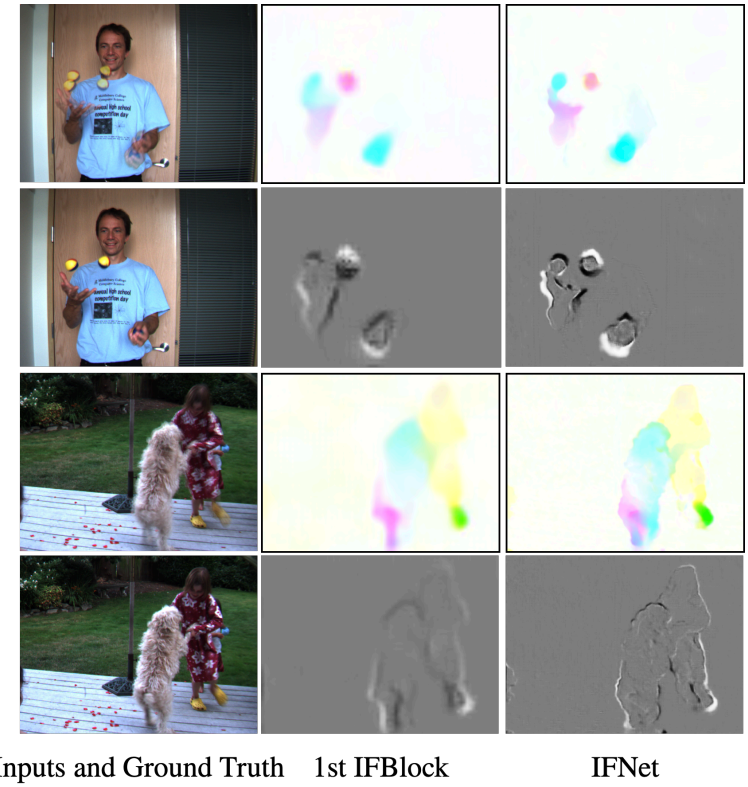

Fig. 11: **Visualization of intermediate flow $F_{t\to 0}$ and blend mask M.** We show that stack 3 IFblocks can get finer intermediate flow and blend mask.

pression is orthogonal to our approach. Our pipeline can be further sped up by manual model design, or Neural Architecture Search (NAS) approaches (left to future work). Furthermore, plain Conv is highly supported by NPU embedded in display devices and provides convenience for customized requirements.

6.3 Intermediate Flow Visualization

In Figure 12, we provide visual results of our IFNet and compare them with the linearly combined bi-directional optical flows [22]. IFNet produces clear motion boundaries.

6.4 Model Efficiency Comparison

Recall that we aim to explore real-time models instead of refreshing SOTA with larger models. Our models are suitable for real-time processing scenarios (display devices, live streaming, games) and media post-processing. However, to the best of our knowledge, currently published papers do not test the

Fig. 12: **Visual comparison between linearly combined bi-directional flows [22] and the result of IFNet.**

speed of each state-of-the-art VFI model on same hardware, and rarely report the complexity of the model. Some previous works [3,45] report runtime data from the MiddleBury public leaderboard without indicating running devices. These data are reported by the submitters of various methods. A more verifiable survey comes from EDSC (Table 10) [9]. We collect the models of each paper and test them on a NVIDIA TITAN X(Pascal) GPU with same hardware. The code can be found on https://github.com/megvii-research/ECCV2022-RIFE/blob/main/benchmark/testtime.py. RIFE use $16ms$ for interpolating a 640×480 frame, $31ms$ for 720p frame, $68ms$ for 1080p frame. The relationship between runtime and resolution is roughly linear.

Take into account the imprecise comparison issues that TTA may introduce. We can use other techniques to get large models. We replace TTA with "multiply the number of hidden filter's channel by a factor of 1.5". And we train this new large model from scratch. The difference between its performance and RIFE-Large is almost negligible. $2\times$ TTA do not change the performance curve of our model. Using TTA, we can get a larger model without training.

6.5 Other Details

Training dynamic. We study the dynamic during the RIFE training. As shown in Figure 13, the privileged distillation scheme helps RIFE converge to better performance. Furthermore, we try to adjust the weights of losses. We found that larger scale ($10\times$) of weights will cause the model to not converge and smaller weights ($0.1\times$) will slightly reduce model performance. We found that the effect

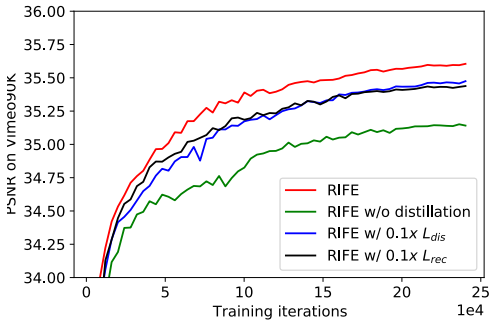

Fig. 13: **PSNR on Vimeo90K benchmark during the whole training process.** The distillation scheme helps RIFE converge to better performance

of our distillation method is similar to regularization techniques, making the models easier to train.

Supervision of mask. Further experiment show that adding supervision of fusion mask has no effect. More detailed distillation design may be a future research direction. When fixing $\widehat{I}_{t\leftarrow 0}$ and $\widehat{I}_{t\leftarrow 1}$, M can be directly learned from the ground truth using the reconstruction loss.