IVIFormer: Illumination-Aware Infrared-Visible Image Fusion via Adaptive Domain-Switching Cross Attention

Supplementary Material

The supplementary material is organized into three sections. Appendix A1 details the used LLM prompts, Appendix A2 outlines the network architecture details, Appendix A3 shows the LCDM's performance, and Appendix A4 presents additional qualitative comparisons.

A1. LLM Prompt Design

We design a specific prompt for the LLM (GPT-4o) to ensure accurate and reliable classification of day and night conditions in images. The prompt is formulated as follows:

> *"Analyze this image and determine whether it is Day or Night. If it is daytime, output 'Day'. If it is nighttime, output 'Night'.*
> *The output must strictly follow this format: - Time : [Classification Result]."*

The Large Language Model (LLM) analyzes the image's characteristics using the specific prompt to determine its temporal condition. By systematically evaluating visual cues such as lighting, color temperature, and atmospheric conditions, the model precisely classifies the image as either 'Day' or 'Night'. This approach ensures a consistent and reliable method of time-of-day identification across diverse visual contexts.

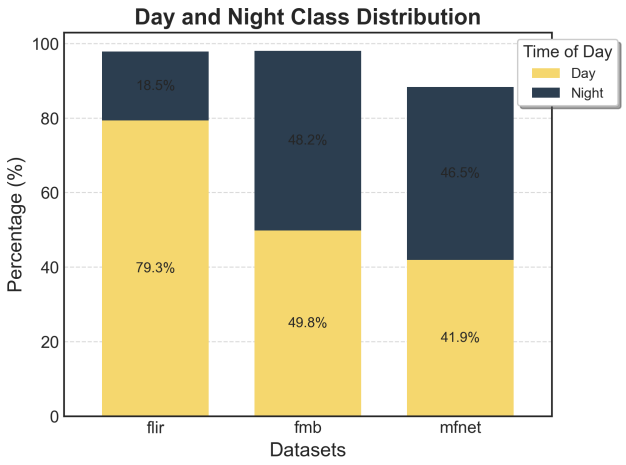

Figure A1. Day and Night Class Distribution Across Datasets

In analyzing the day and night class distribution across three distinct datasets, we observed significant variations in temporal representation. The FLIR dataset exhibits a pronounced imbalance, with daytime images comprising approximately 79.29% of the dataset, while nighttime images account for only 18.53%. In contrast, the FMB dataset demonstrates a nearly balanced distribution, with daytime images at 49.80% and nighttime images at 48.20%. This near-even split indicates a more uniform representation of temporal conditions, which could be beneficial for training models with equal exposure to day and night environments. The MFNET dataset shows a slightly different pattern, with nighttime images marginally outnumbering daytime images (46.46% night versus 41.87% day).

A2. Network Architecture Details

The proposed neural network architecture features a sophisticated encoder-decoder structure with a bottleneck layer, designed to efficiently process and transform visual features through carefully orchestrated computational blocks.

A2.0.1. Encoder Architecture

The encoder pathway consists of 11 sequential blocks organized into three resolution stages. Each stage contains three consecutive processing units followed by a downsampling operation (except for the final stage). The architecture follows this pattern:

- **First Stage (16 channels)**: Three sequential units, each consisting of an IVIF residual block (16→16 channels), an IVIF high-attention block (16 channels with 4 heads), and another IVIF residual block (16→16 channels). A depth downsampling operation follows, maintaining 16 channels.
- **Second Stage (24 channels)**: Three sequential units with IVIF residual blocks (initial transition from 16→24 channels, then 24→24 channels), IVIF high-attention blocks (24 channels with 4 heads), and IVIF residual blocks (24→24 channels). A depth downsampling operation follows, maintaining 24 channels.
- **Third Stage (32 channels)**: Three sequential units with IVIF residual blocks (initial transition from 24→32 channels, then 32→32 channels), IVIF high-attention blocks (32 channels with 4 heads), and IVIF residual blocks (32→32 channels).

A2.0.2. Bottleneck Layer

The bottleneck connecting the encoder and decoder pathways consists of a single IVIF residual block that processes and maintains 32 channels (32→32 channels). This bottleneck preserves essential information while reducing computational complexity.

A2.0.3. Decoder Architecture

The decoder pathway mirrors the encoder but in reverse order, progressively recovering spatial resolution through 11 blocks organized in three stages:

- **First Stage (32 channels)**: Three sequential units with IVIF residual blocks (32→32 channels), IVIF high-attention blocks (32 channels with 4 heads), and IVIF residual blocks (the final one transitioning from 32→24 channels). An upsampling operation follows, maintaining 24 channels.
- **Second Stage (24 channels)**: Three sequential units with IVIF residual blocks (24→24 channels), IVIF high-attention blocks (24 channels with 4 heads), and IVIF residual blocks (the final one transitioning from 24→16 channels). An upsampling operation follows, maintaining 16 channels.
- **Third Stage (16 channels)**: Three sequential units with IVIF residual blocks (16→16 channels), IVIF high-attention blocks (16 channels with 4 heads), and IVIF residual blocks (16→16 channels).

A2.1. Efficiency Analysis

We conducted a comprehensive performance analysis comparing the proposed ADS-CA module with the baseline spatial attention mechanism across various configurations. Our experiments covered different input resolutions (32×32, 64×64, 128×128), feature dimensions (16, 24, 32), and attention head counts (2, 4, 8), with a batch size of 1 for all test scenarios.

A2.1.1. Memory Efficiency

The ADS-CA module demonstrates remarkable memory efficiency compared to the baseline spatial attention mechanism, as illustrated in Figure A2. The memory consumption differences become increasingly pronounced at higher resolutions:

- **At 32×32 resolution**: ADS-CA consumes approximately 29.1-30.0 MB across all configurations, whereas the baseline requires 44.8-93.4 MB, representing a 1.5-3.1× increase in memory usage.
- **At 64×64 resolution**: The memory efficiency gap widens significantly, with ADS-CA requiring only 31.7-161.8 MB compared to the baseline's 286.5-1184.8 MB, yielding a 4.2-37.3× memory advantage.
- **At 128×128 resolution**: The difference becomes dramatic, with ADS-CA utilizing just 167.2-178.3 MB versus the baseline's 4261.2-16558.3 MB, reflecting a 23.9-98.4× improvement in memory efficiency.

Particularly notable is the scaling behavior with respect to resolution. While the baseline model's memory consumption increases quadratically with spatial dimensions due to the attention computation across all pixel positions, the ADS-CA module's memory requirement grows linearly, as it applies attention along the channel dimension rather than the spatial dimension.

A2.1.2. Computational Efficiency

The inference time measurements reveal equally substantial performance benefits:

- **At 32×32 resolution**: Both approaches exhibit comparable inference times (approximately 0.44 ms for ADS-CA versus 0.26 ms for baseline).
- **At 64×64 resolution**: ADS-CA maintains consistent inference times (0.44-0.46 ms) while the baseline shows increased latency (1.09-3.57 ms), representing a 2.4-8.0× speed advantage for ADS-CA.
- **At 128×128 resolution**: The computational efficiency gap becomes most pronounced, with ADS-CA maintaining inference times of around 0.45-0.46 ms compared to the baseline's 14.48-57.72 ms, yielding a 31.5-126.8× speed advantage.

Notably, the inference time of ADS-CA remains nearly constant across all resolutions, demonstrating the scalability advantage of channel-wise attention compared to the spatial attention mechanism.

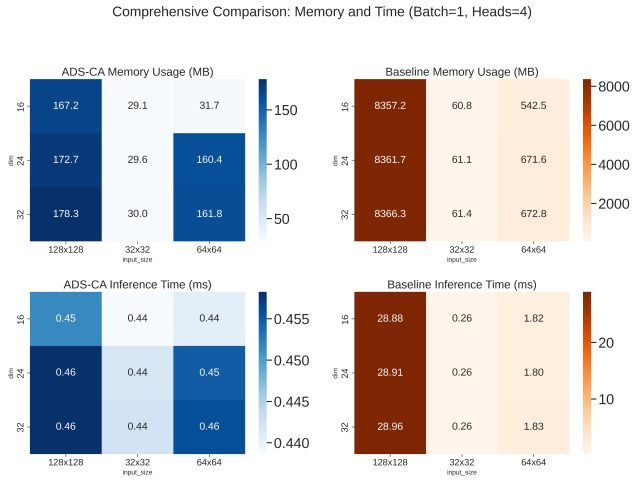

Figure A2. Comprehensive comparison of memory usage (top) and inference time (bottom) between ADS-CA (left) and baseline (right) attention mechanisms with batch size=1 and 4 attention heads. The heat maps illustrate performance metrics across different input resolutions (32×32, 64×64, 128×128) and feature dimensions (16, 24, 32). Note the significant scale differences between ADS-CA and baseline, particularly for the 128×128 resolution where ADS-CA demonstrates up to 50× memory efficiency and 65× computational efficiency.

A3. LCDM Performance Evaluation

This section provides a detailed quantitative evaluation of the final trained LCDM model. The model was evaluated on a test set, which was not used during the training. Ta-

ble 6 summarizes the key classification performance metrics, demonstrating the model's high efficacy in distinguishing between day and night images.

Metric	Score
Accuracy	0.9796
Precision	0.9785
Recall	0.9785
F1-Score	0.9785

Table 6. Performance metrics of the LCDM model on the test dataset.

A4. Additional Qualitative Comparisons

Additional qualitative comparisons are provided to further illustrate the performance and effectiveness of our method.

Figure A3. Qualitative comparisons with different models.

Figure A4. Qualitative comparisons with different models.

Figure A5. Qualitative comparisons with different models.

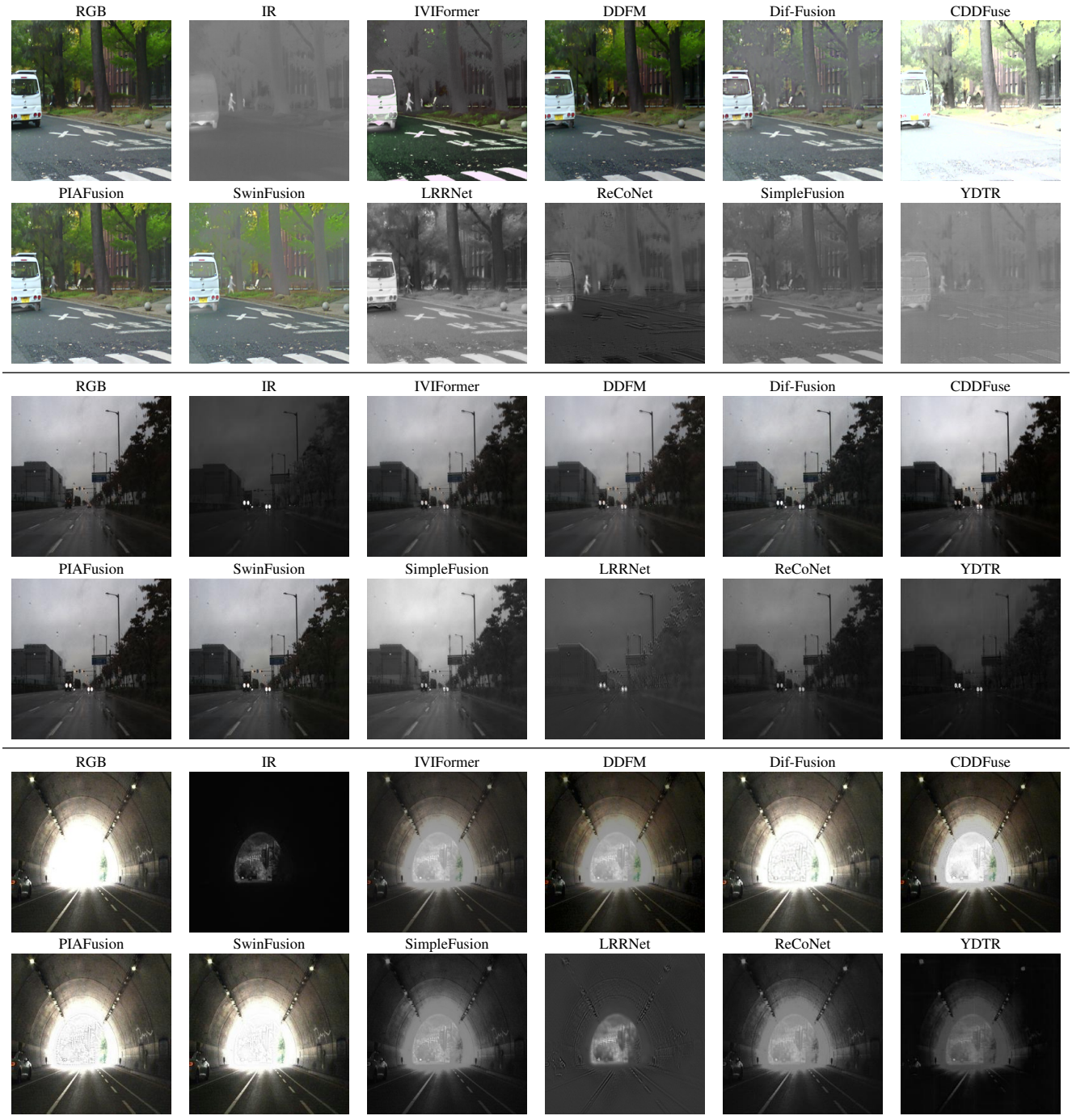

Figure A6. Qualitative comparisons with different models.