Low light image denoising

I implement the **MIRNet** model for low-light image enhancement, a fully-convolutional architecture that learns an enriched set of features that combines contextual information from multiple scales, while simultaneously preserving the high-resolution spatial details.

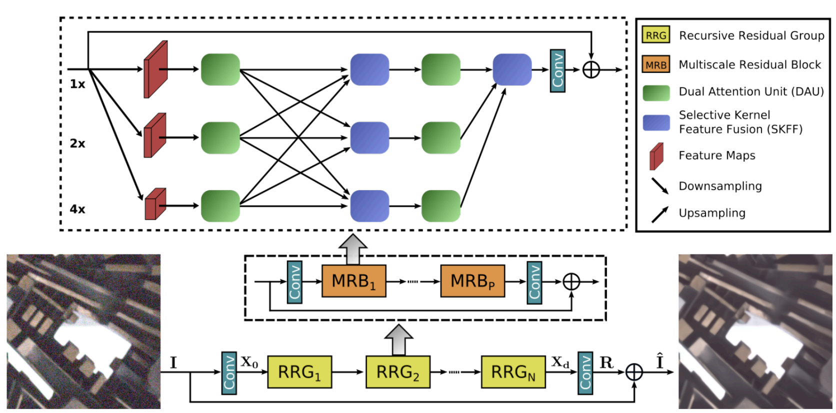
The pipeline follows as

1) Loading and preparing images
2) Ensuring dimension uniformity
3) Normalizing pixel values
4) Defining model
5) Training model
6) Plotting result graphs

Link of the paper: https://arxiv.org/abs/2003.06792

Here are the main features of the MIRNet model:

- A feature extraction model that computes a complementary set of features across multiple spatial scales, while maintaining the original high-resolution features to preserve precise spatial details.
- A regularly repeated mechanism for information exchange, where the features across multi-resolution branches are progressively fused together for improved representation learning.
- A new approach to fuse multi-scale features using a selective kernel network that dynamically combines variable receptive fields and faithfully preserves the original feature information at each spatial resolution.
- A recursive residual design that progressively breaks down the input signal in order to simplify the overall learning process, and allows the construction of very deep networks.

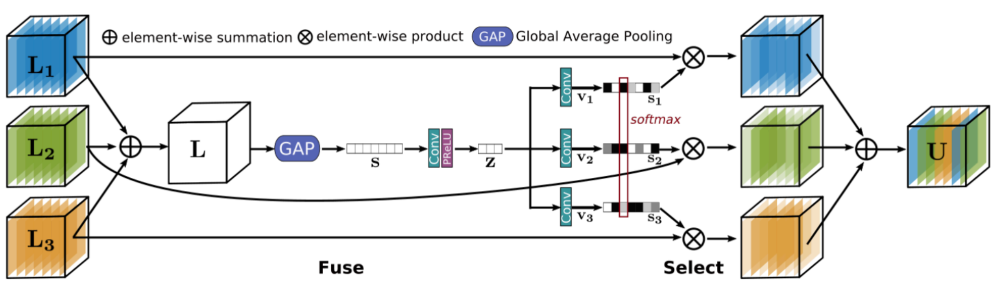


Selective Kernel Feature Fusion

The Selective Kernel Feature Fusion or SKFF module performs dynamic adjustment of receptive fields via two operations: **Fuse** and **Select**. The Fuse operator generates global feature descriptors by combining the information from multi-resolution streams. The Select operator uses these descriptors to recalibrate the feature maps (of different streams) followed by their aggregation.

Fuse: The SKFF receives inputs from three parallel convolution streams carrying different scales of information. We first combine these multi-scale features using an element-wise sum, on which we apply Global Average Pooling (GAP) across the spatial dimension. Next, we apply a channel- downscaling convolution layer to generate a compact feature representation which passes through three parallel channel-upscaling convolution layers (one for each resolution stream) and provides us with three feature descriptors.

Select: This operator applies the softmax function to the feature descriptors to obtain the corresponding activations that are used to adaptively recalibrate multi-scale feature maps. The aggregated features are defined as the sum of product of the corresponding multi-scale feature and the feature descriptor.

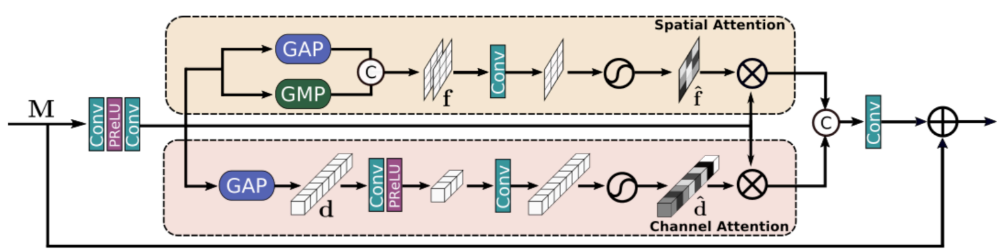


Dual Attention Unit

The Dual Attention Unit or DAU is used to extract features in the convolutional streams. While the SKFF block fuses information across multi-resolution branches, we also need a mechanism to share information within a feature tensor, both along the spatial and the channel dimensions which is done by the DAU block. The DAU suppresses less useful features and only allows more informative ones to pass further. This feature recalibration is achieved by using **Channel Attention** and **Spatial Attention** mechanisms.

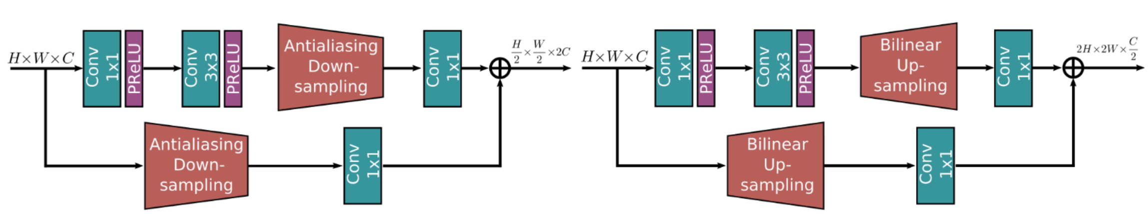
The **Channel Attention** branch exploits the inter-channel relationships of the convolutional feature maps by applying squeeze and excitation operations. Given a feature map, the squeeze operation applies Global Average Pooling across spatial dimensions to encode global context, thus yielding a feature descriptor. The excitation operator passes this feature descriptor through two convolutional layers followed by the sigmoid gating and generates activations. Finally, the output of Channel Attention branch is obtained by rescaling the input feature map with the output activations.

The **Spatial Attention** branch is designed to exploit the inter-spatial dependencies of convolutional features. The goal of Spatial Attention is to generate a spatial attention map and use it to recalibrate the incoming features. To generate the spatial attention map, the Spatial Attention branch first independently applies Global Average Pooling and Max Pooling operations on input features along the channel dimensions and concatenates the outputs to form a resultant feature map which is then passed through a convolution and sigmoid activation to obtain the spatial attention map. This spatial attention map is then used to rescale the input feature map.



Multi-Scale Residual Block

The Multi-Scale Residual Block is capable of generating a spatially-precise output by maintaining high-resolution representations, while receiving rich contextual information from low-resolutions. The MRB consists of multiple (three in this paper) fully-convolutional streams connected in parallel. It allows information exchange across parallel streams in order to consolidate the high-resolution features with the help of low-resolution features, and vice versa. The MIRNet employs a recursive residual design (with skip connections) to ease the flow of information during the learning process. In order to maintain the residual nature of our architecture,



(a) Downsampling module (b) Upsampling module residual resizing modules are used to perform downsampling and upsampling operations that are used in the Multi-scale Residual Block.

Training

- We train MIRNet using **Charbonnier Loss** as the loss function and **Adam Optimizer** with a learning rate of `1e-4`.
- We use **Peak Signal Noise Ratio** or PSNR as a metric which is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation.