Convergence Analysis of MAP based Blur Kernel Estimation Supplementary

Sunghyun Cho
DGIST
scho@dgist.ac.kr

Seungyong Lee
POSTECH
leesy@postech.ac.kr

Table of Contents

MATLAB Source Code for Figure 6

```
% To make Fig. 6 in the paper
function psf = deblur_single_level(blurred, k_size)
    lambda_l = 0.0006;
    alpha = 0.1;
    lambda_k = 0.001;
    num_iters = 20;

    % initial PSF
    psf = zeros(k_size);
    psf((k_size(1)+1)/2,(k_size(2)+1)/2) = 1;

    % prepare gradient maps of blurred
    blurred = edgetaper(blurred,fspecial('gaussian',60,10));
    bx = imfilter(blurred, [0 -1 1], 'corr', 'circular');
    by = imfilter(blurred, [0;-1;1], 'corr', 'circular');
    blurred_g = cat(3, bx, by);

    % alternating estimation
    for iter=1:num_iters
        latent_g = deconv_sps(blurred_g,psf,lambda_l,alpha);

        [energy data prior_l prior_k] = energy_func(...
                        latent_g, blurred_g, psf, ...
                        lambda_l, alpha, lambda_k);
        fprintf('%d\t%f\t%f\t%f\t%f\n', ...
            iter, energy, data, prior_l, prior_k);

        psf = estimate_psf(blurred_g, latent_g, psf_size, lambda_k);
    end
end
```

```
function [energy data prior_l prior_k] ...
        = energy_func(latent, blurred, psf, lambda_l, alpha, lambda_k)
    tau = 0.01;

    K = psf2otf(psf, size(blurred));
    b = real(ifft2(fft2(latent).*K));
    diff = b - blurred;
    data = sum(diff(:).^2);

    w = max(abs(latent),tau).^(alpha-2);
    prior_l = sum(w(:).*latent(:).^2);
    prior_k = sum(psf(:).^2);

    energy = data + lambda_l*prior_l + lambda_k*prior_k;
end
```

MATLAB Source Code for Figure 6

```
function latent = deconv_sps(blurred, psf, lambda_l, alpha, num_iters)
    tau = 0.01;
    if ~exist('num_iters', 'var')
        num_iters = 15;
    end
     % initial latent image
    B = fft2(blurred);
    K = psf2otf(psf, size(blurred));
    L = conj(K).*B./(conj(K).*K+lambda_l);
    latent = real(ifft2(L));
    % iterative update using IRLS
    for iter=1:num_iters
        w = max(abs(latent),tau).^(alpha-2);
        latent = deconv_L2(blurred,latent,psf,lambda_l,w,5);
    end
end

function latent = deconv_L2(blurred, latent, psf, ...
                        lambda_l, weight, n_iters)
    img_size = size(blurred);
    psf_f = psf2otf(psf, img_size);
    b = real(ifft2(fft2(blurred).*conj(psf_f)));
    % run conjugate gradient
    p.lambda_l = lambda_l;
    p.psf_f = psf_f;
    p.weight = weight;
    latent = conjgrad(latent, b, n_iters, 1e-4, @Ax, p);
end

function y = Ax(x, p)
    y = real(ifft2(conj(p.psf_f).* p.psf_f.*fft2(x)));
    y = y + p.lambda_l*p.weight.*x;
end
```

```
function x=conjgrad(x, b, n_iters, tol, Ax_func, func_param)
    r = b - Ax_func(x,func_param);
    p = r;
    rsold = sum(r(:).^2);

    for iter=1:n_iters
        Ap = Ax_func(p,func_param);
        alpha = rsold/sum(p(:).*Ap(:));
        x=x+alpha*p;
        r=r-alpha*Ap;
        rsnew=sum(r(:).^2);
        if sqrt(rsnew)<tol
            break;
        end
        p=r+rsnew/rsold*p;
        rsold=rsnew;
    end
end
```

MATLAB Source Code for Figure 6

```
function psf = estimate_psf(blurred, latent, psf_size, lambda_k)
    B = fft2(blurred);
    L = fft2(latent);
    Bx = B(:,:,1);
    By = B(:,:,2);
    Lx = L(:,:,1);
    Ly = L(:,:,2);
    Lap = psf2otf([0 -1 0;-1 4 -1; 0 -1 0], ...
                      [size(blurred,1), size(blurred,2)]);
    K = (conj(Lx).*Bx + conj(Ly).*By) ...
         ./ (conj(Lx).*Lx + conj(Ly).*Ly + lambda_k.*Lap);
    psf = real(otf2psf(K, psf_size));
    psf(psf < max(psf(:))*0.05) = 0;
    psf = psf / sum(psf(:));
end
```

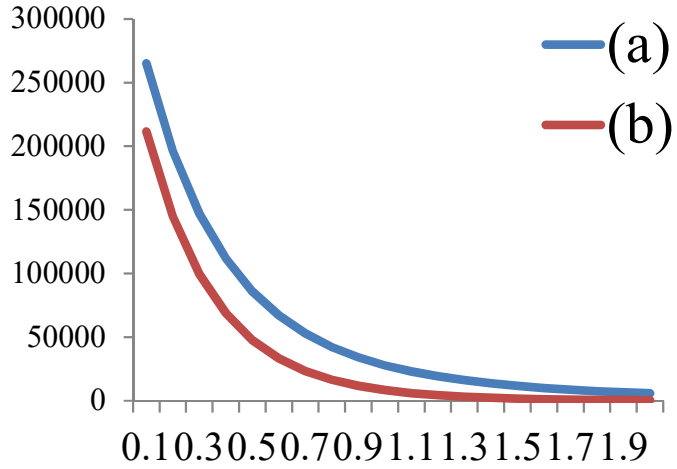
```
% To find the globally optimal latent image
%  for the no-blur case
function l = find_optimal_no_blur(blurred, lambda_l, alpha)
    LUT = make_LUT(lambda_l, alpha);
    bx = blurred(:,:,1);
    by = blurred(:,:,2);
    lx = interp1(LUT(1,:), LUT(2,:), bx(:), 'linear', 'extrap');
    lx = reshape(lx, size(bx));
    ly = interp1(LUT(1,:), LUT(2,:), by(:), 'linear', 'extrap');
    ly = reshape(ly, size(by));
    l = cat(3, lx, ly);
end

function LUT = make_LUT(lambda_l, alpha)
    % for every b find the optimal l
    % which minimizes |b-l|^2 + lambda_l |l|^alpha
    v = -1:0.1/256:1;
    [b l] = meshgrid(v,v);
    tau = 0.01;
    w = max(abs(l),tau).^(alpha-2);
    prior_l = w.*l.^2;
    energies = (b-l).^2 + lambda_l * prior_l;
    [min_energies min_indices] = min(energies);
    LUT = [v; l(min_indices)];
end
```

Figure 1

(a) Sharp image and
a blur kernel

(b) Blurred image and
a delta kernel



(c) Sparsity prior values
of (a) and (b)

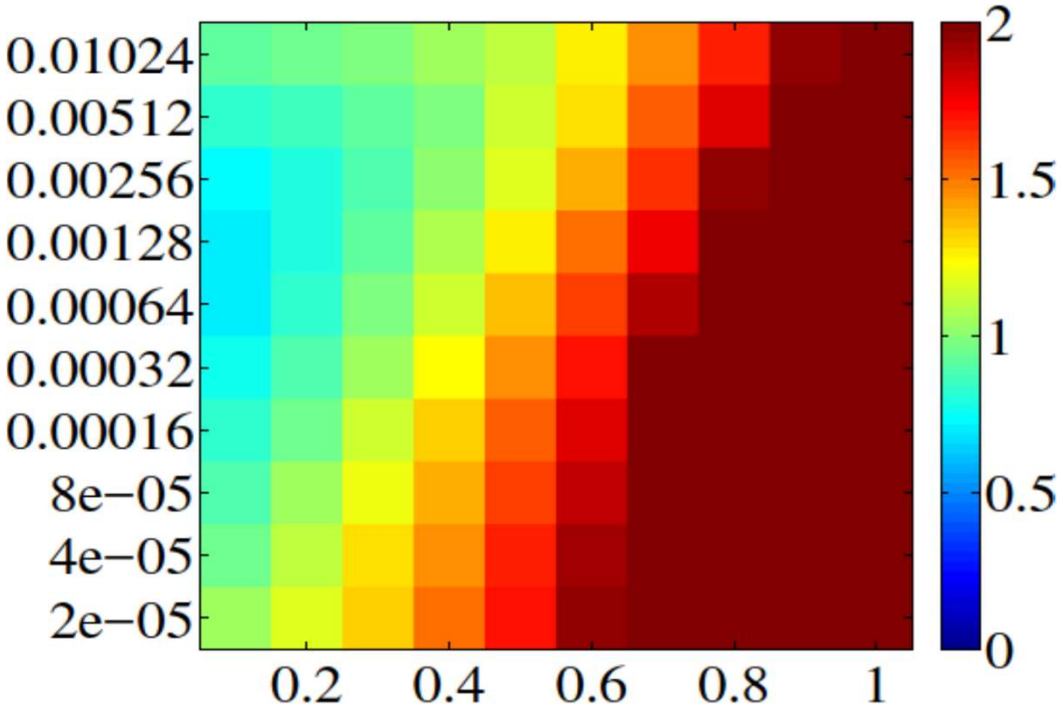
Figure 2

(a) $\hat{l}_{gt}^{\mathrm{IRLS}}$

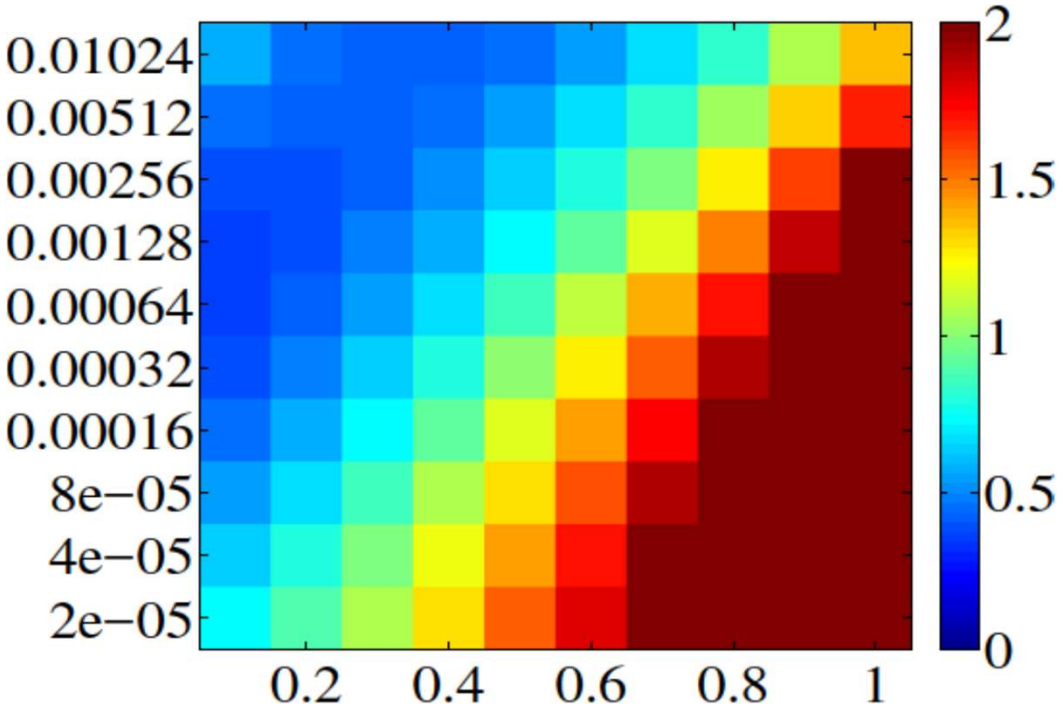
(b) $\hat{l}_{\delta}^{\mathrm{IRLS}}$

(c) $\hat{l}_{\delta}^{\mathrm{opt}}$

Figure 3



(a) $f^{\mathrm{IRLS}}(k_{gt})/f^{\mathrm{opt}}(k_{\delta})$



(b) $\rho_l(\hat{l}_{gt}^{\mathrm{IRLS}})/\rho_l(\hat{l}_{\delta}^{\mathrm{opt}})$

Figure 4

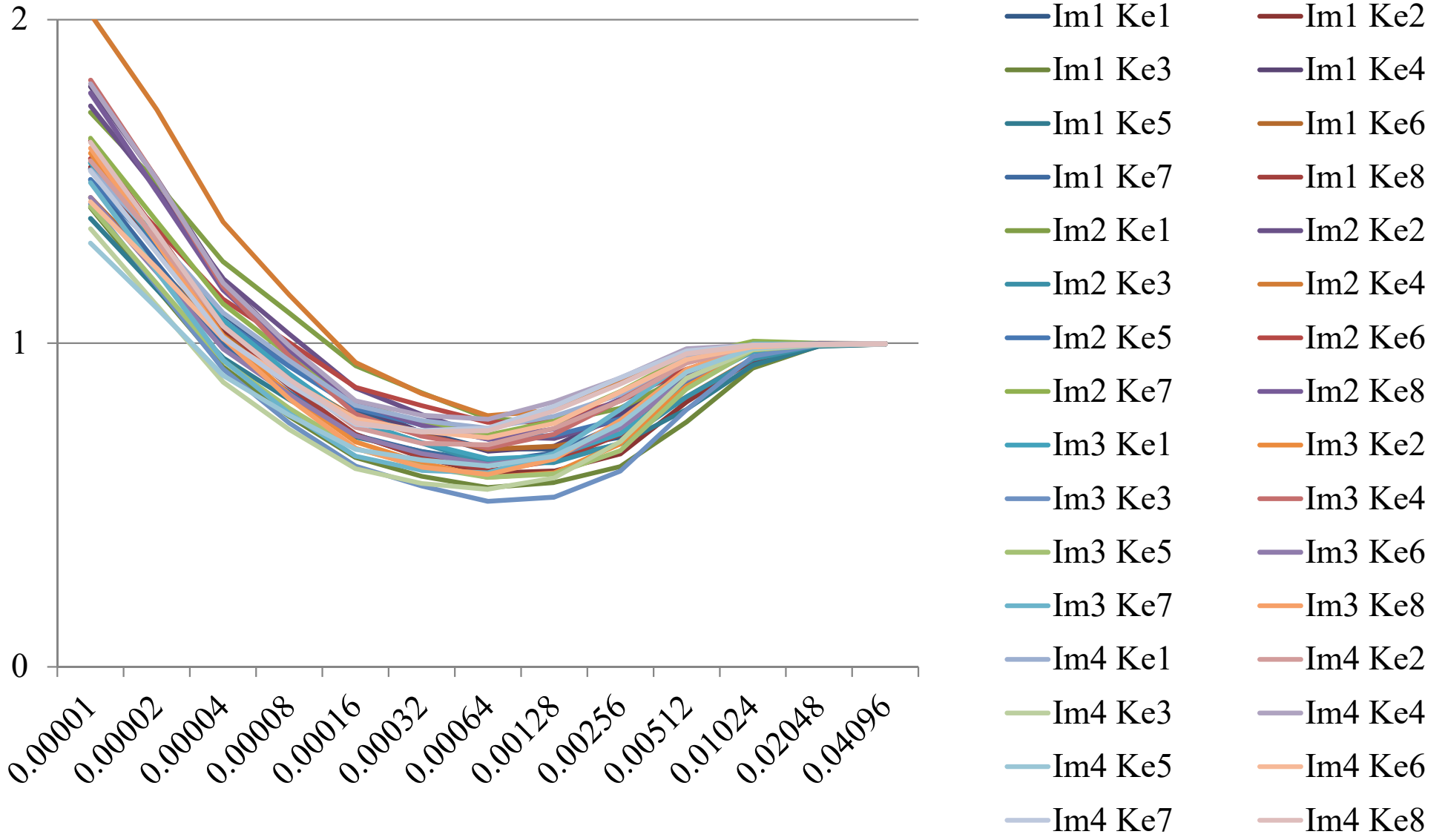

2
1
0
0.00001
0.00002
0.00004
0.00008
0.00016
0.00032
0.00064
0.00128
0.00256
0.00512
0.01024
0.02048
0.04096
Im1 Ke1
Im1 Ke2
Im1 Ke3
Im1 Ke4
Im1 Ke5
Im1 Ke6
Im1 Ke7
Im1 Ke8
Im2 Ke1
Im2 Ke2
Im2 Ke3
Im2 Ke4
Im2 Ke5
Im2 Ke6
Im2 Ke7
Im2 Ke8
Im3 Ke1
Im3 Ke2
Im3 Ke3
Im3 Ke4
Im3 Ke5
Im3 Ke6
Im3 Ke7
Im3 Ke8
Im4 Ke1
Im4 Ke2
Im4 Ke3
Im4 Ke4
Im4 Ke5
Im4 Ke6
Im4 Ke7
Im4 Ke8

Figure 5

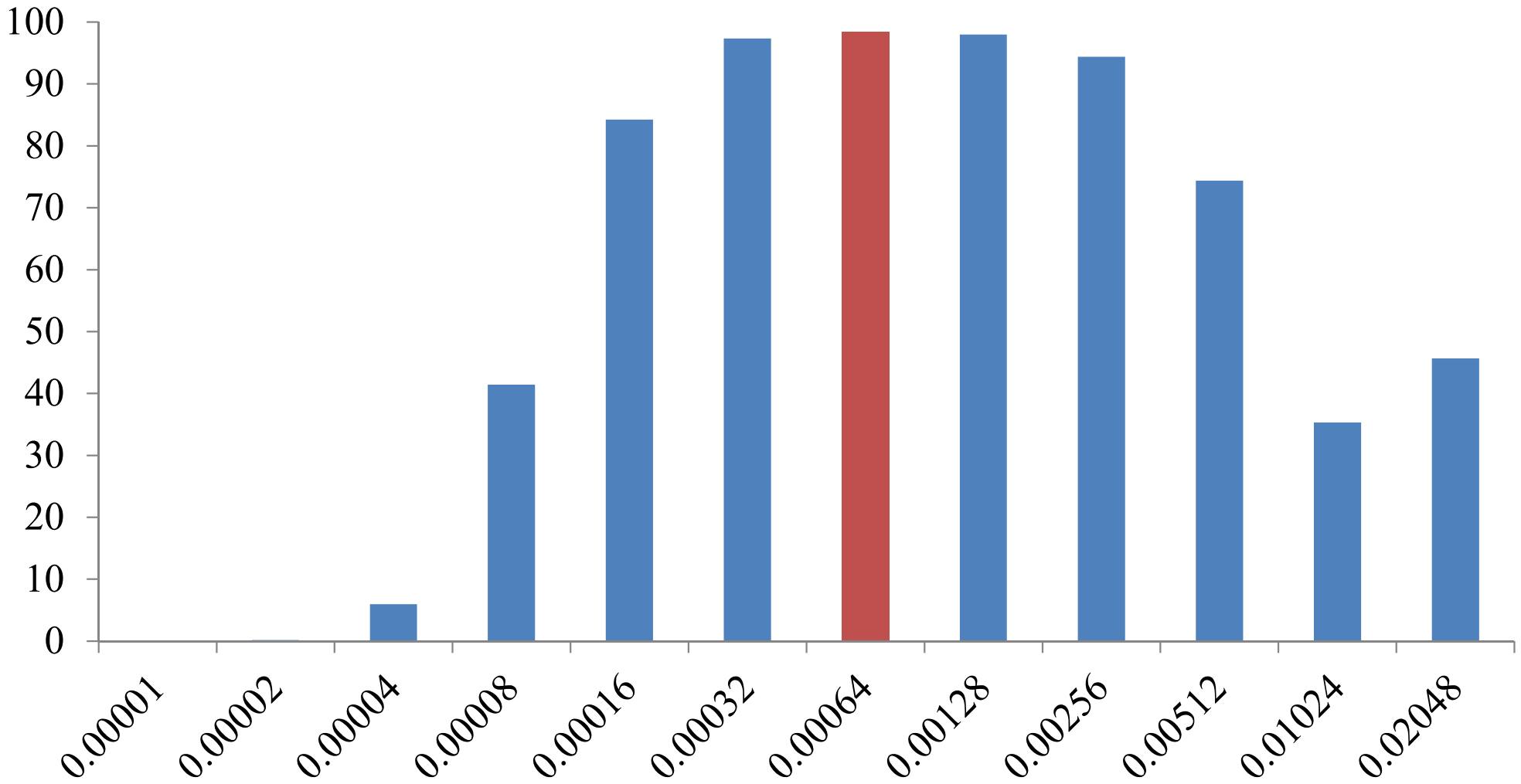
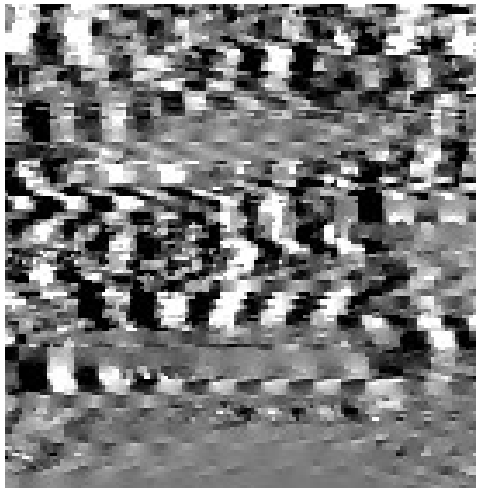

100
90
80
70
60
50
40
30
20
10
0
0.00001
0.00002
0.00004
0.00008
0.00016
0.00032
0.00064
0.00128
0.00256
0.00512
0.01024
0.02048

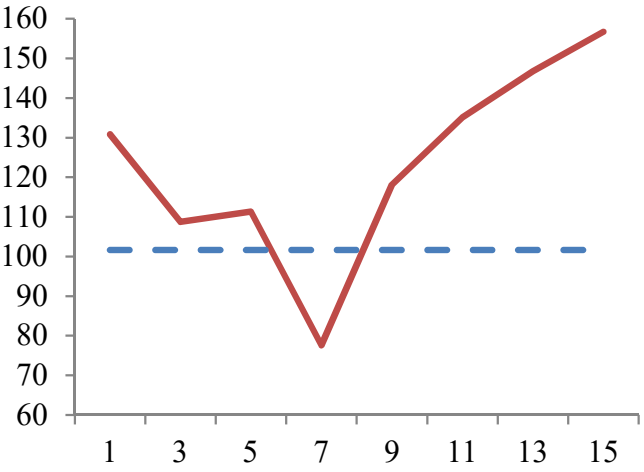
Figure 6

(a) Latent image from the blur kernel of length 1

(b) Latent image from the blur kernel of length 7

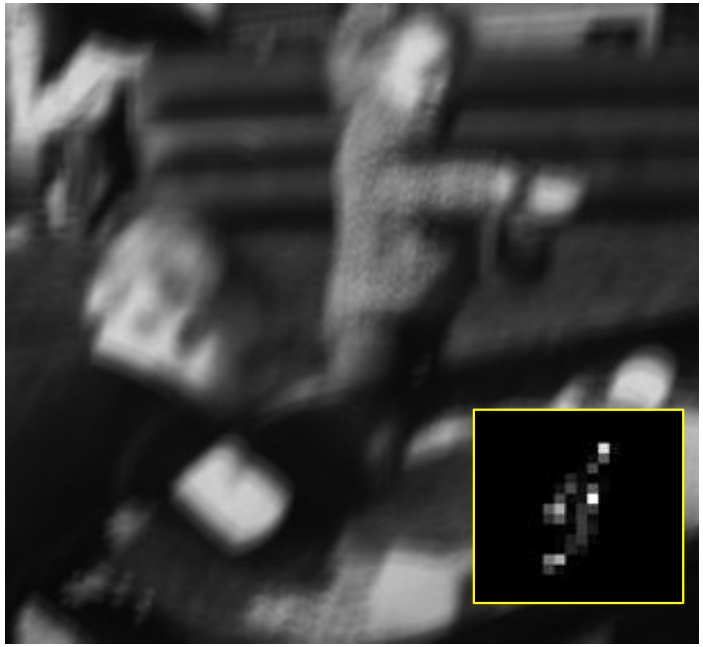


(c) Latent image from the blur kernel of length 15

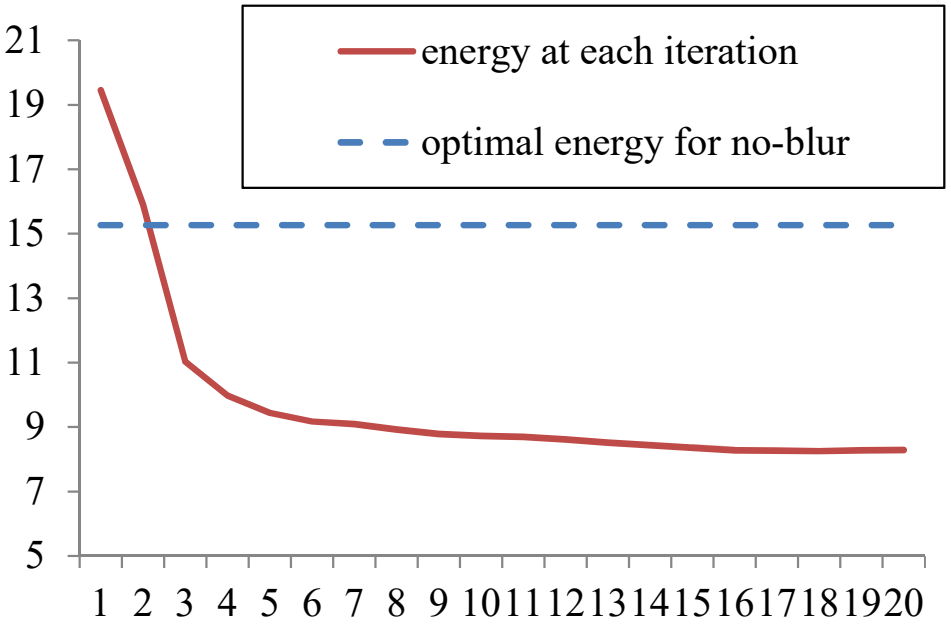


(d)

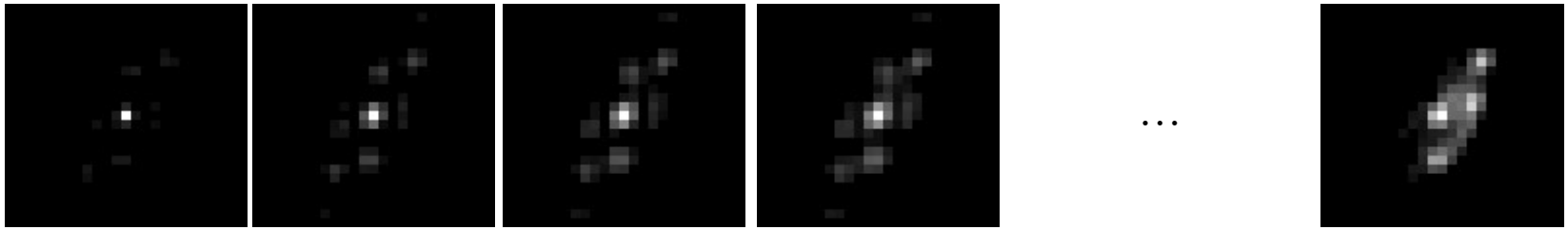
Figure 7



(a) Blurred image &
its ground truth blur kernel



(b) Energy values along iterations



(c) Blur kernels at different iterations

Figure 8

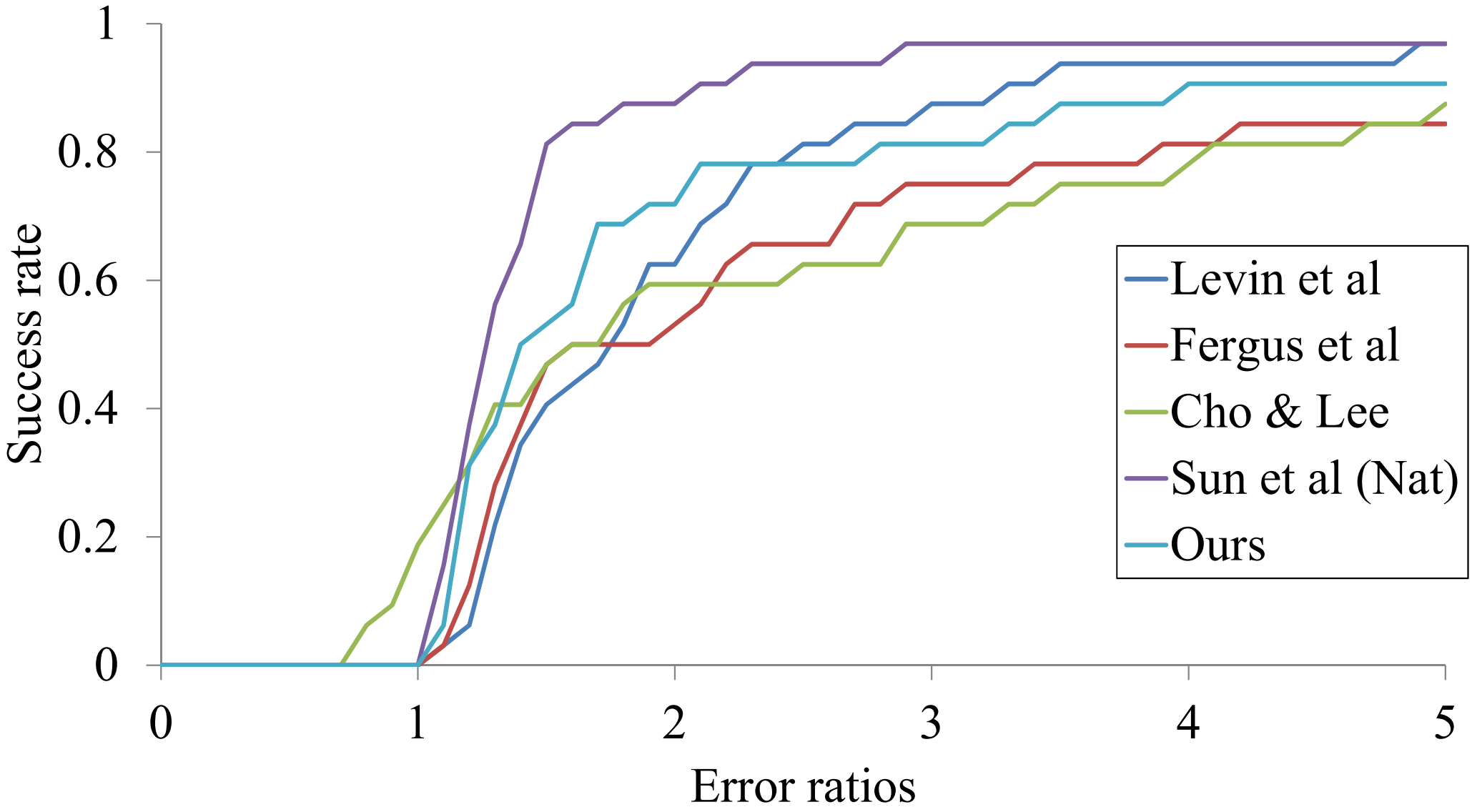
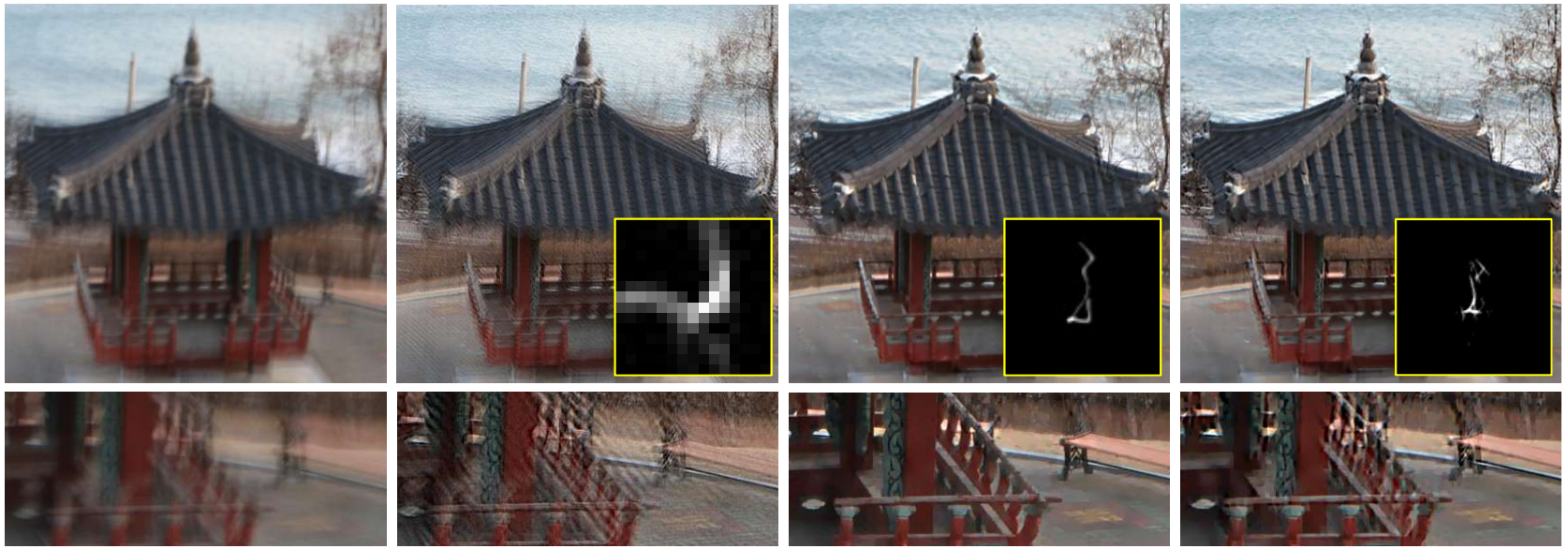

Success rate
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
Error ratios
Levin et al
Fergus et al
Cho & Lee
Sun et al (Nat)
Ours

Figure 9



(a) Blurred image

(b) 15x15 kernel
Energy: 124.2

(c) 75x75 kernel
Energy: 121.0

(d) 115x115 kernel
Energy: 130.2

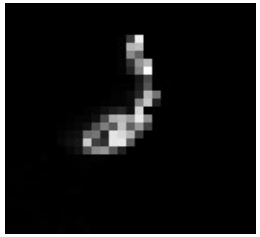
(e) Blurred image

(f) 35x35 kernel
Energy: 19.4

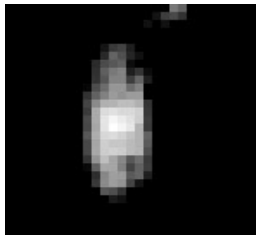
(g) 55x55 kernel
Energy: 20.2

(h) 75x75 kernel
Energy: 18.9

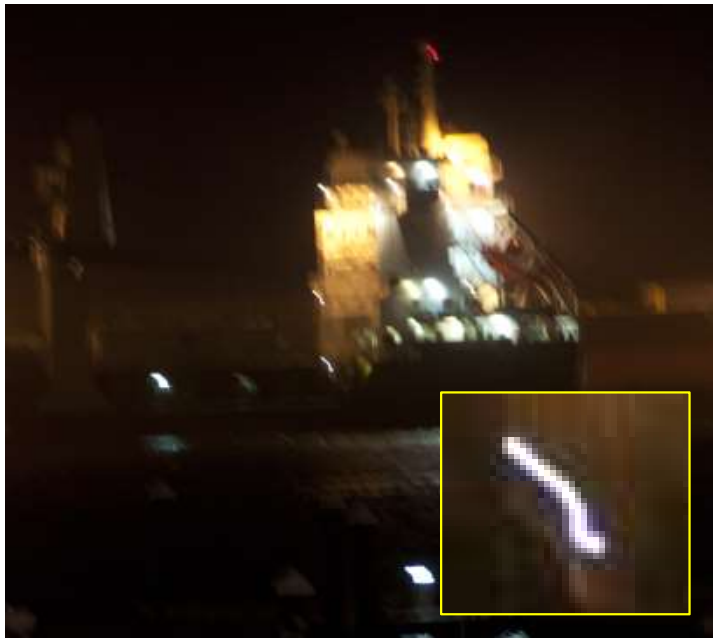
Figure 10

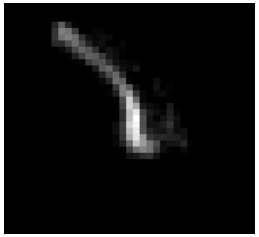


43.8

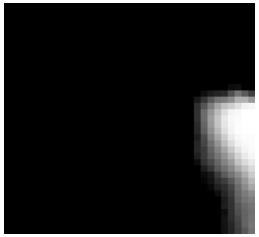


66.7

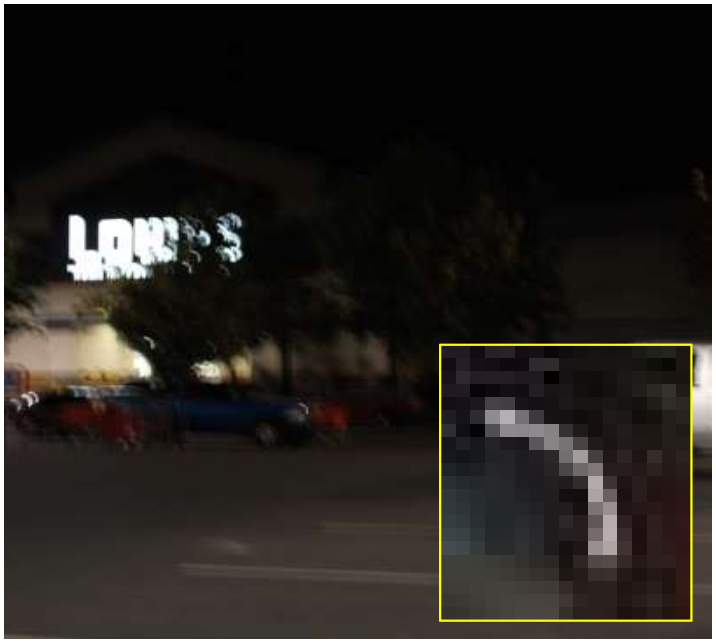


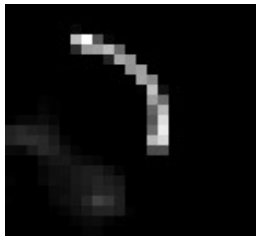


219.2

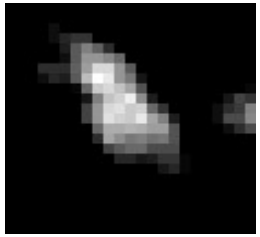


258.1

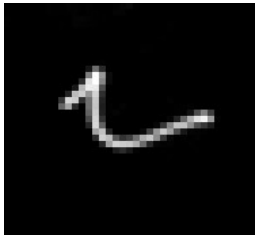




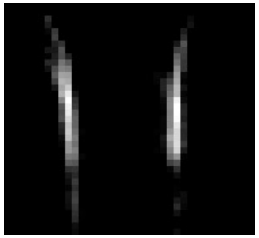
113.2



181.5

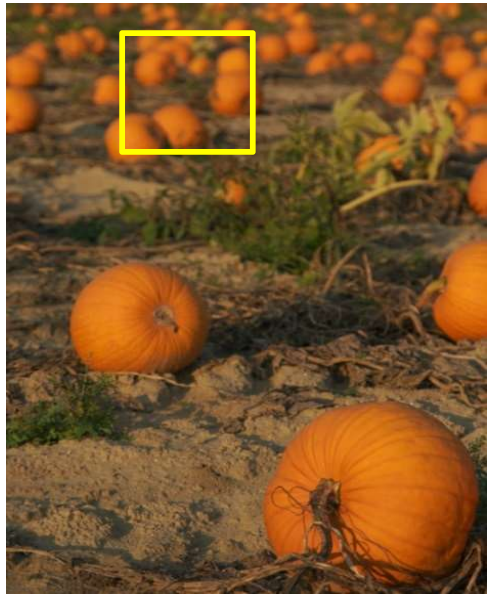


123.1

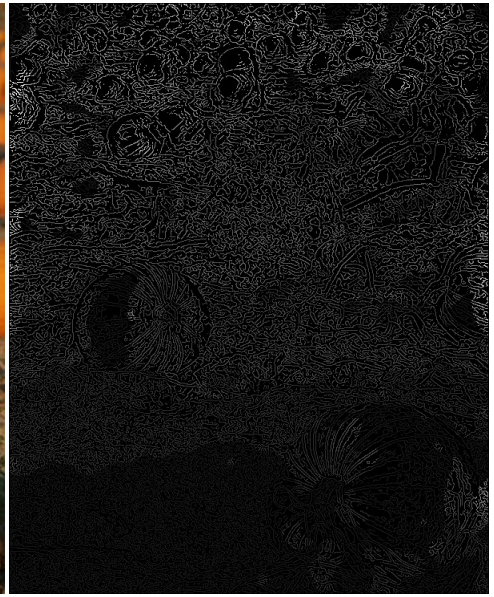


165.7

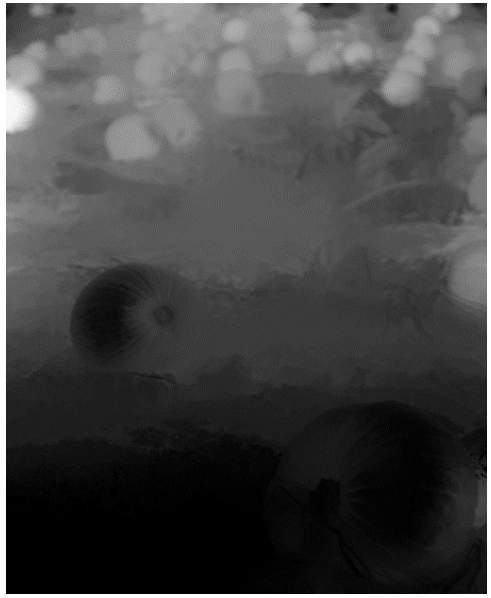
Figure 11



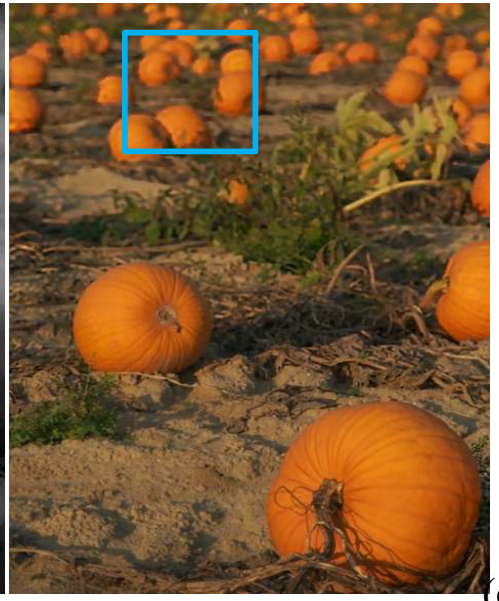
(a) Input image



(b) Sparse defocus map



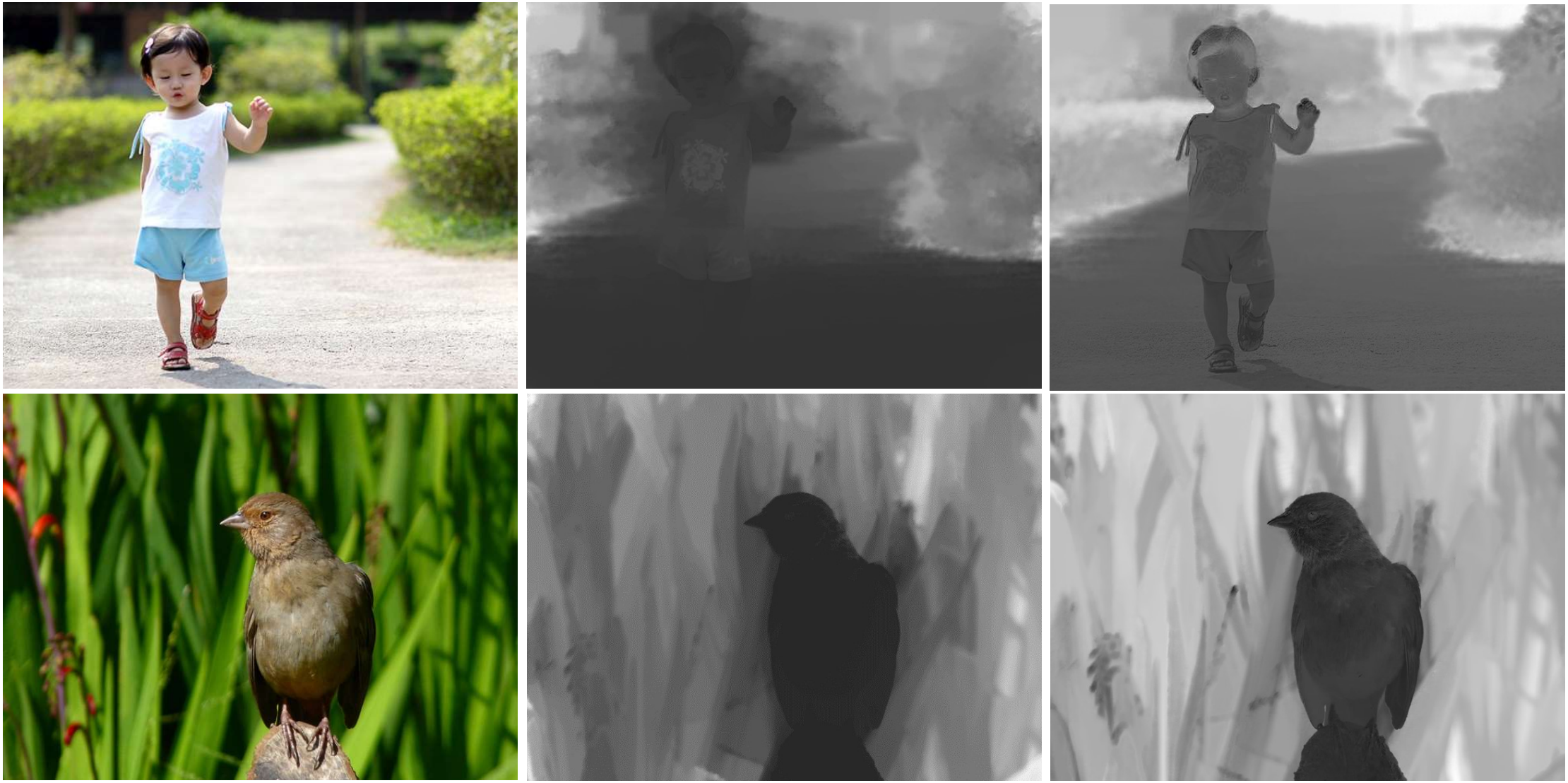
(c) Dense defocus map



(d) All-focus image

(e) Magnified views of (a) & (d)

Figure 12



(a) Input images (b) Our defocus maps (c) Zhuo and Sim [25]

Figure 8 Our Results

Error ratio: 1.18

Error ratio: 1.11

Error ratio: 1.26

Error ratio: 1.31

Error ratio: 1.22

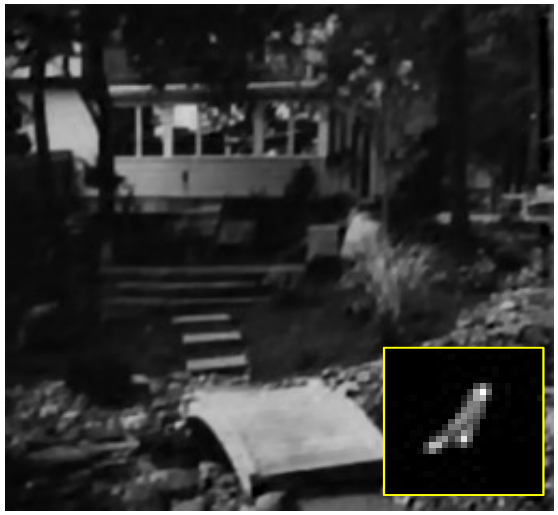
Error ratio: 1.66

Error ratio: 1.53

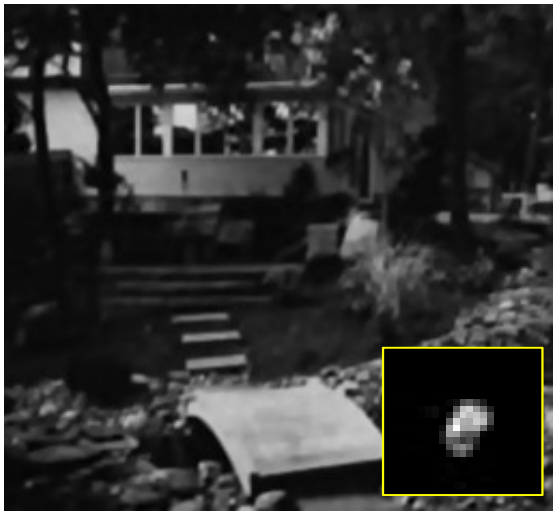
Error ratio: 1.35

Figure 8 Our Results

Error ratio: 1.16



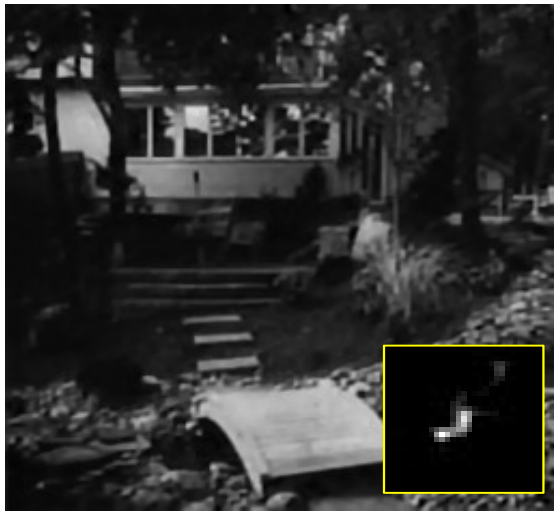
Error ratio: 1.03



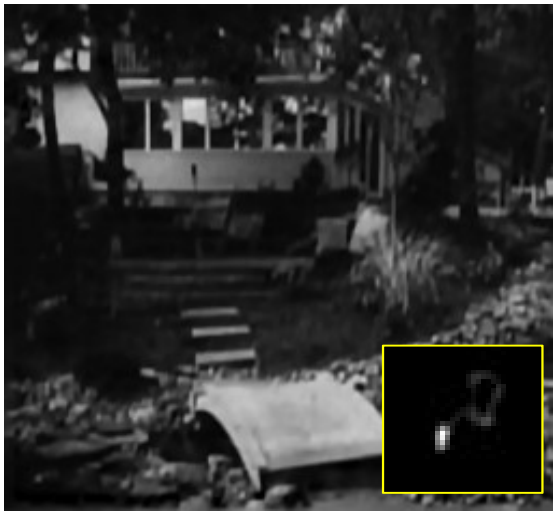
Error ratio: 1.14

Error ratio: 1.11

Error ratio: 1.14



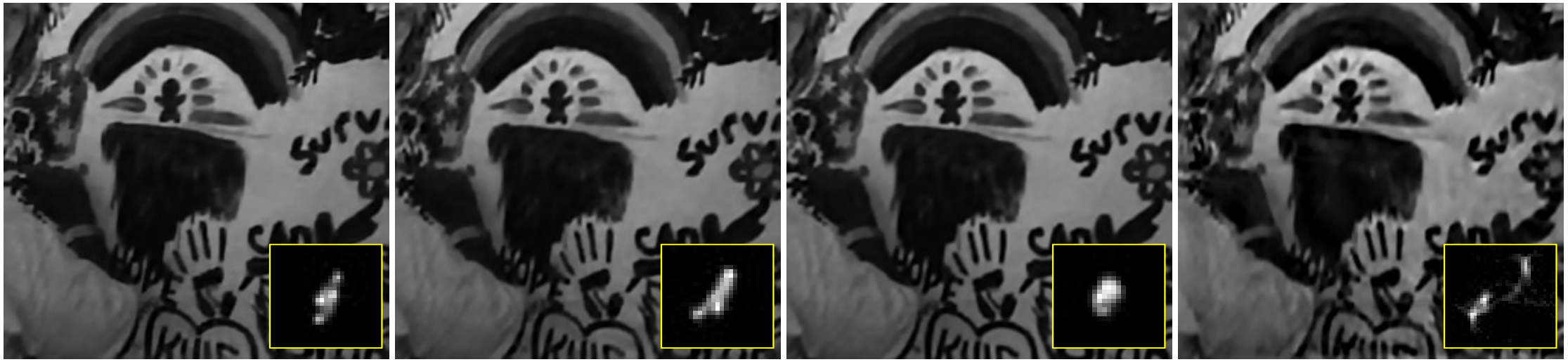
Error ratio: 2.06



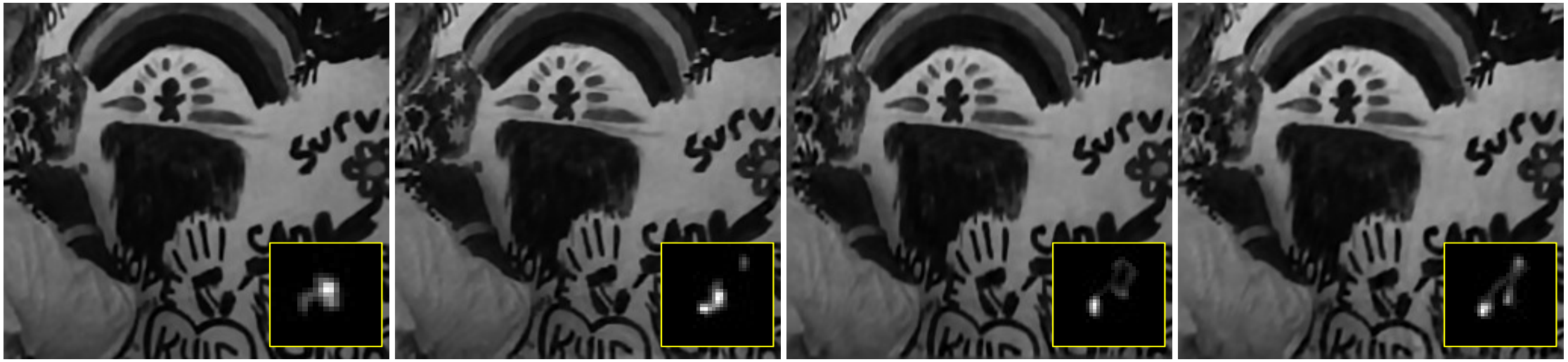
Error ratio: 1.48

Error ratio: 29.19

Figure 8 Our Results



Error ratio: 1.08 Error ratio: 1.15 Error ratio: 1.19 Error ratio: 3.47



Error ratio: 1.34 Error ratio: 2.79 Error ratio: 1.60 Error ratio: 1.34

Figure 8 Our Results

Error ratio: 1.61 Error ratio: 2.02 Error ratio: 1.88 Error ratio: 3.99

Error ratio: 1.62 Error ratio: 3.26 Error ratio: 9.84 Error ratio: 5.55

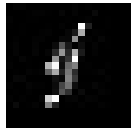
Experimental Results using Sun et al.'s dataset [19]

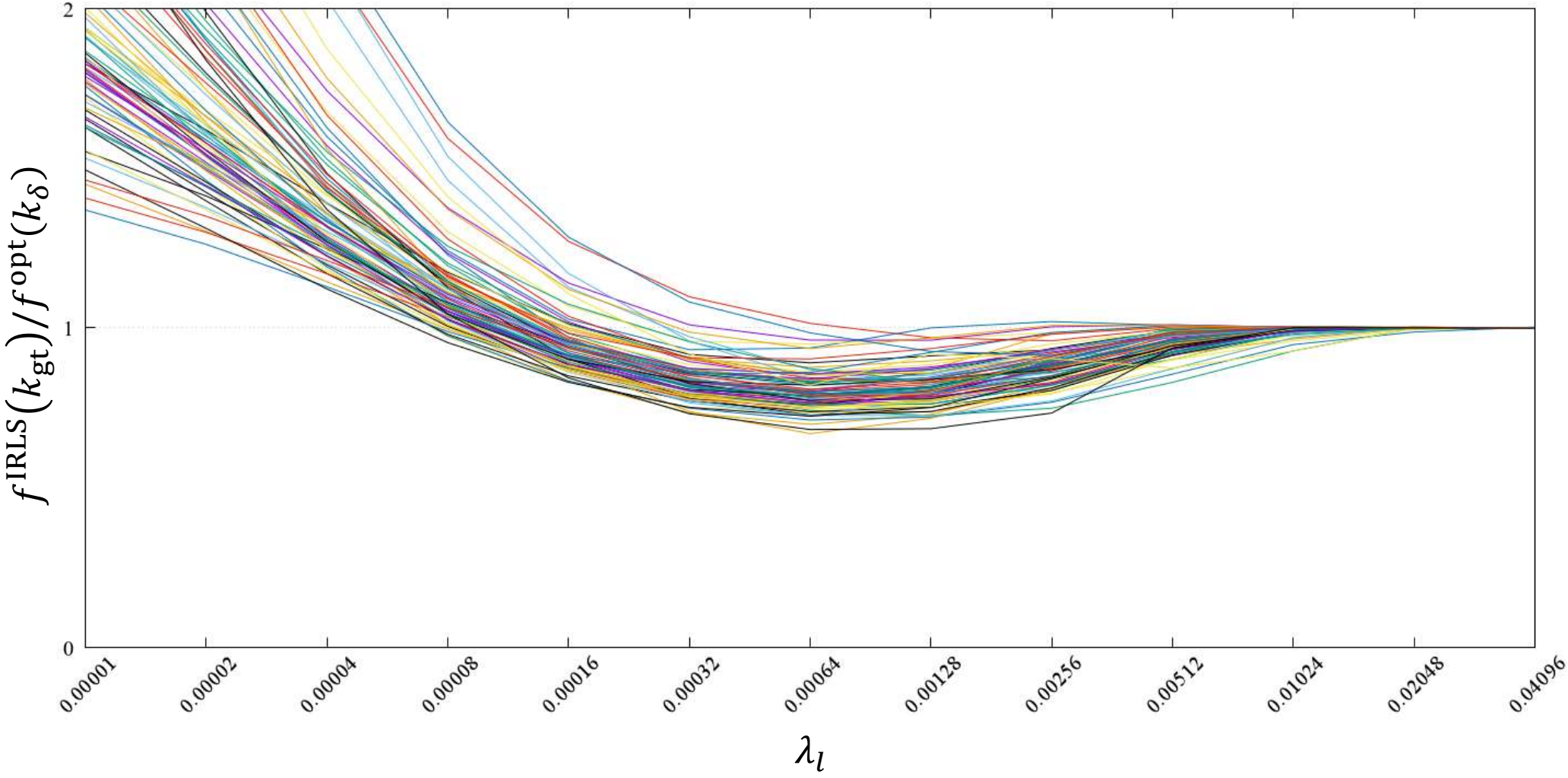
- Plots in the following pages show $f^{\mathrm{IRLS}}(k_{gt})/f^{\mathrm{opt}}(k_{\delta})$ with respect to different λ_l's
  - For each kernel, and then for each image.
  - A couple of them are shown in Fig. 4 in the paper.

- $f^{\mathrm{IRLS}}(k_{gt})/f^{\mathrm{opt}}(k_{\delta}) < 1$ means that the ground truth blur kernel is preferred to the delta kernel by the energy function.

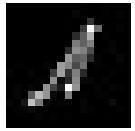
- Sun et al.'s dataset consists of 640 synthetically blurred images generated from
  - 80 sharp images ranging from natural to man-made scenes, and
  - 8 blur kernels.

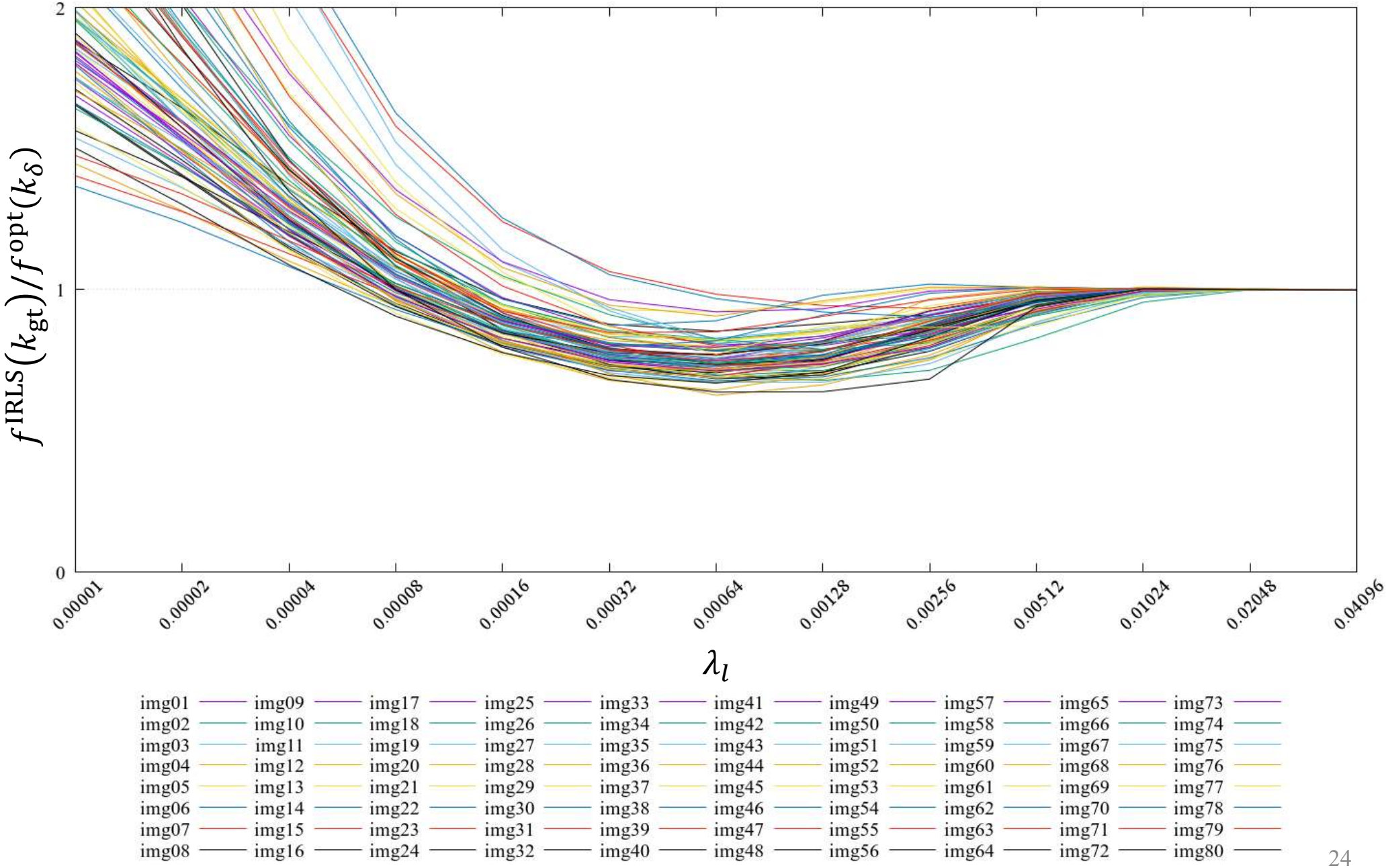
Sun et al.'s dataset - Kernel 1



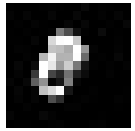


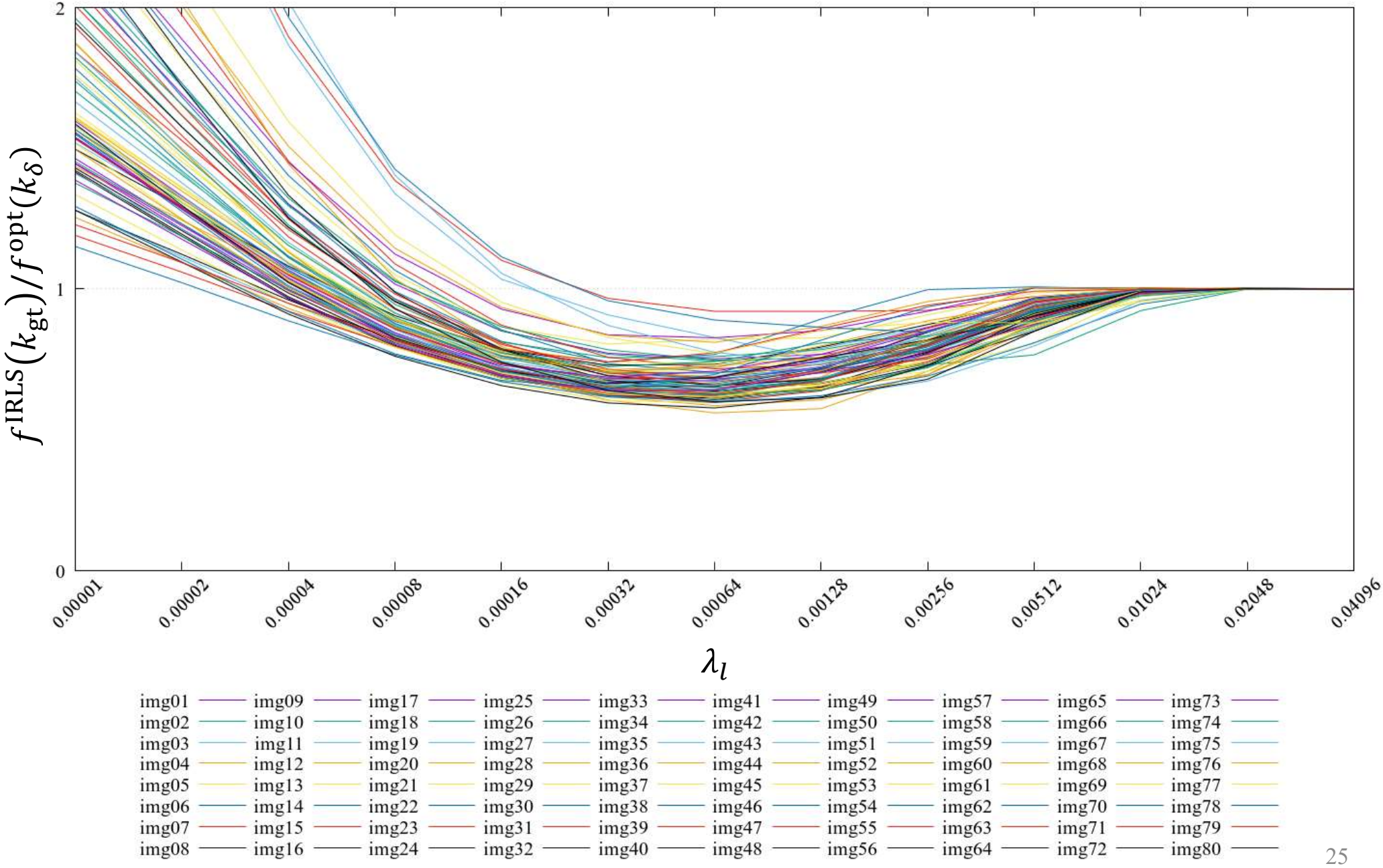
Sun et al.'s dataset - Kernel 2



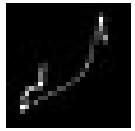


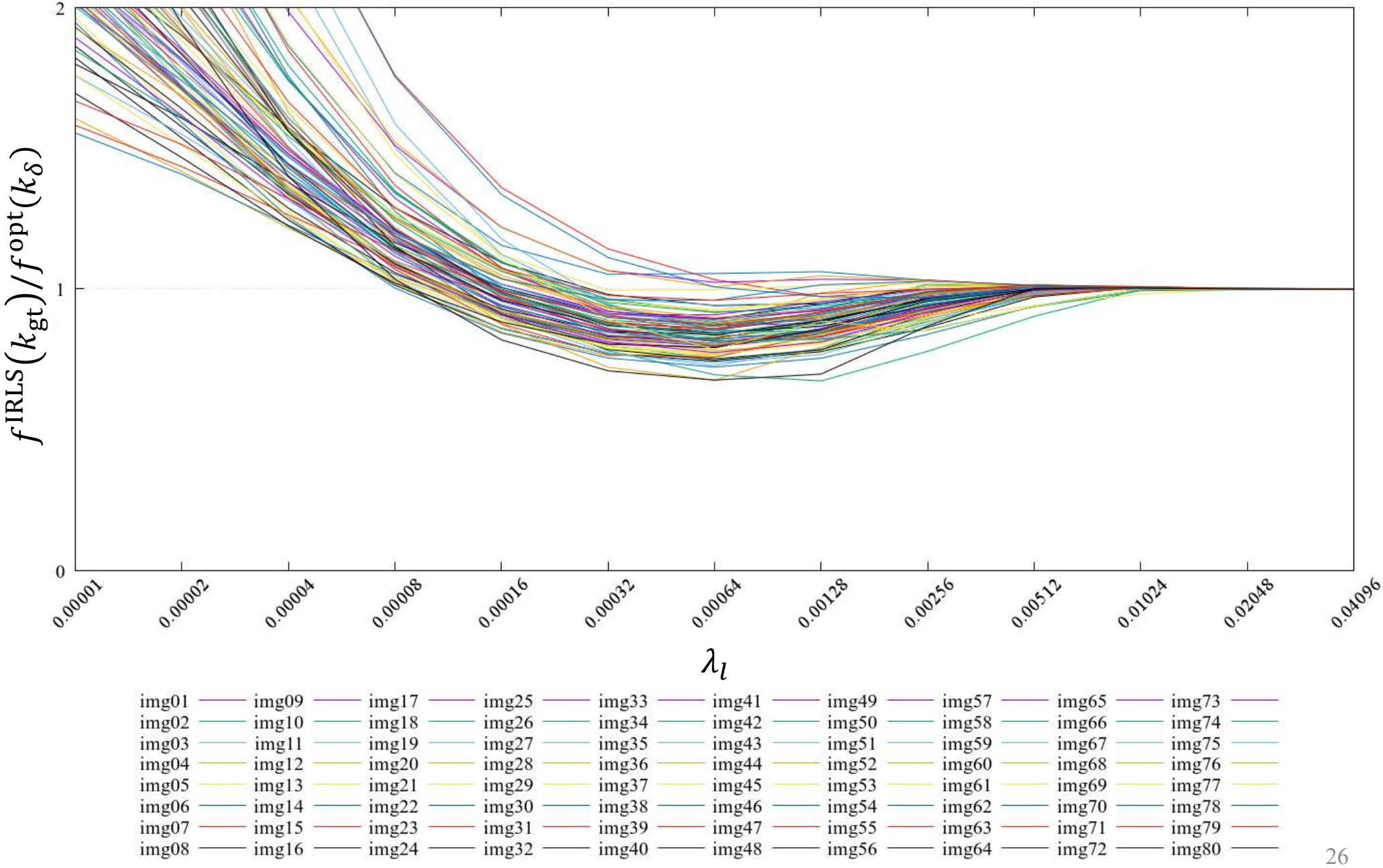
Sun et al.'s dataset - Kernel 3



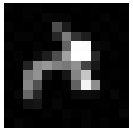


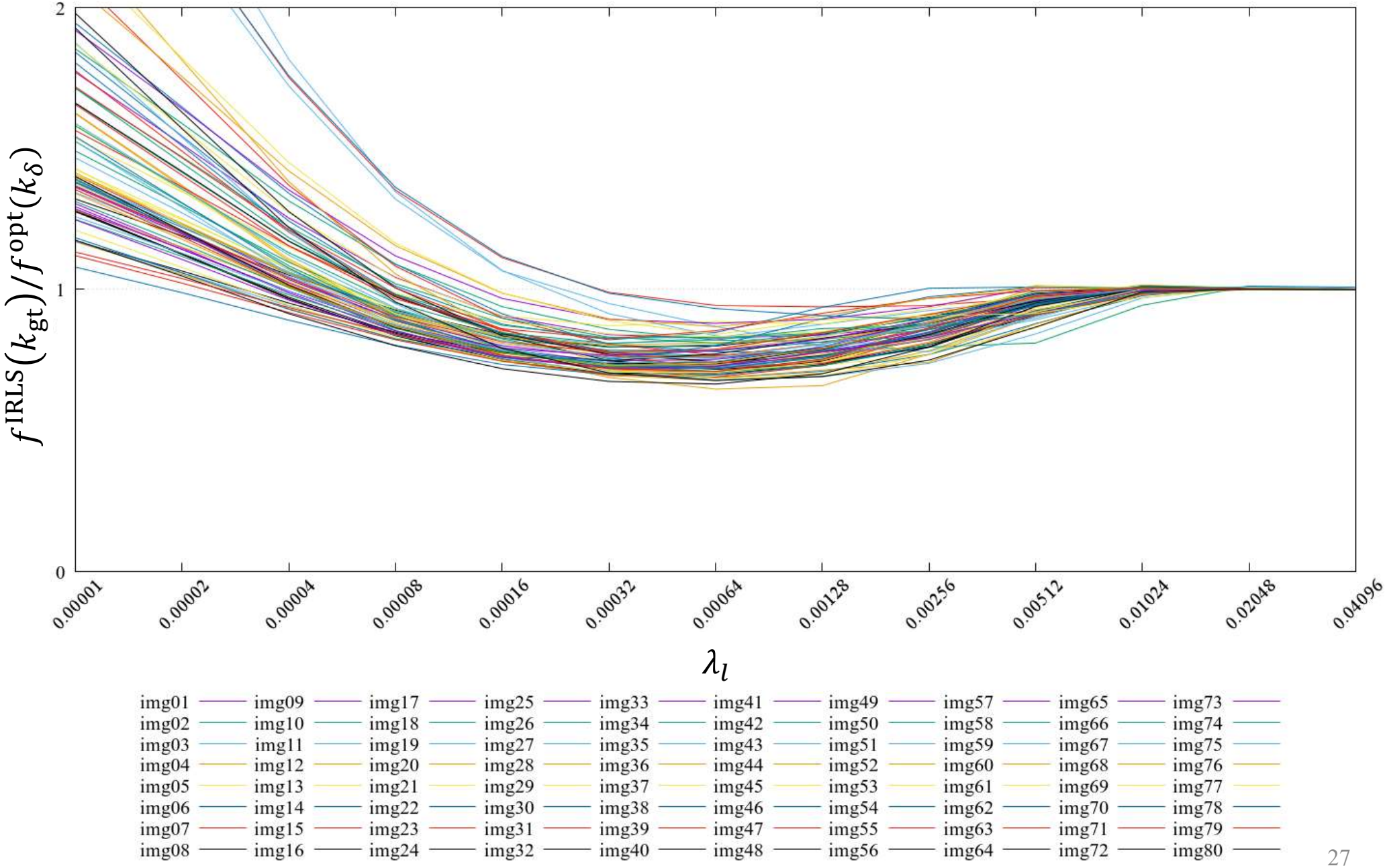
Sun et al.'s dataset - Kernel 4



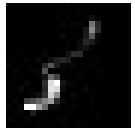


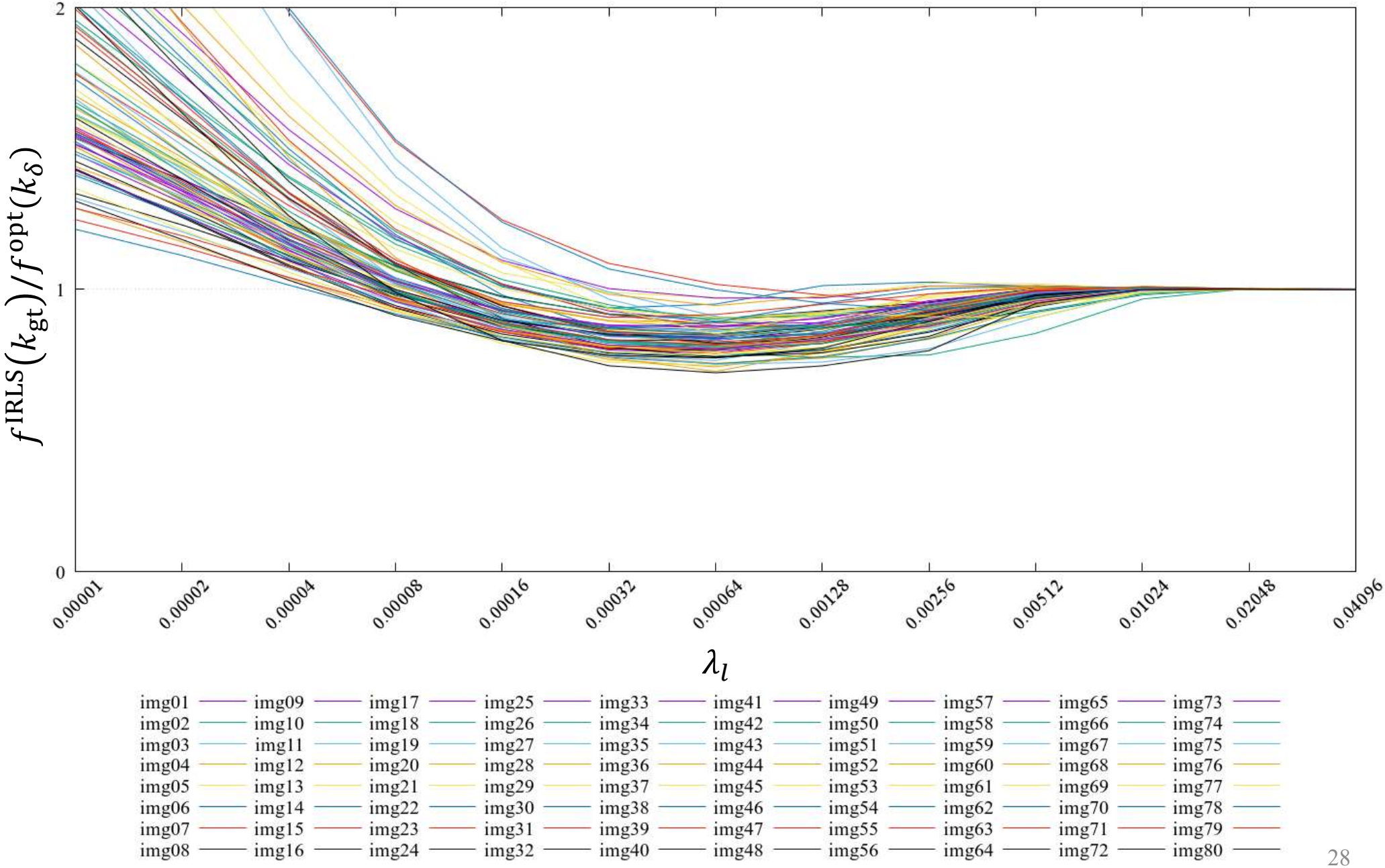
Sun et al.'s dataset - Kernel 5



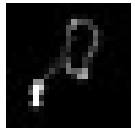


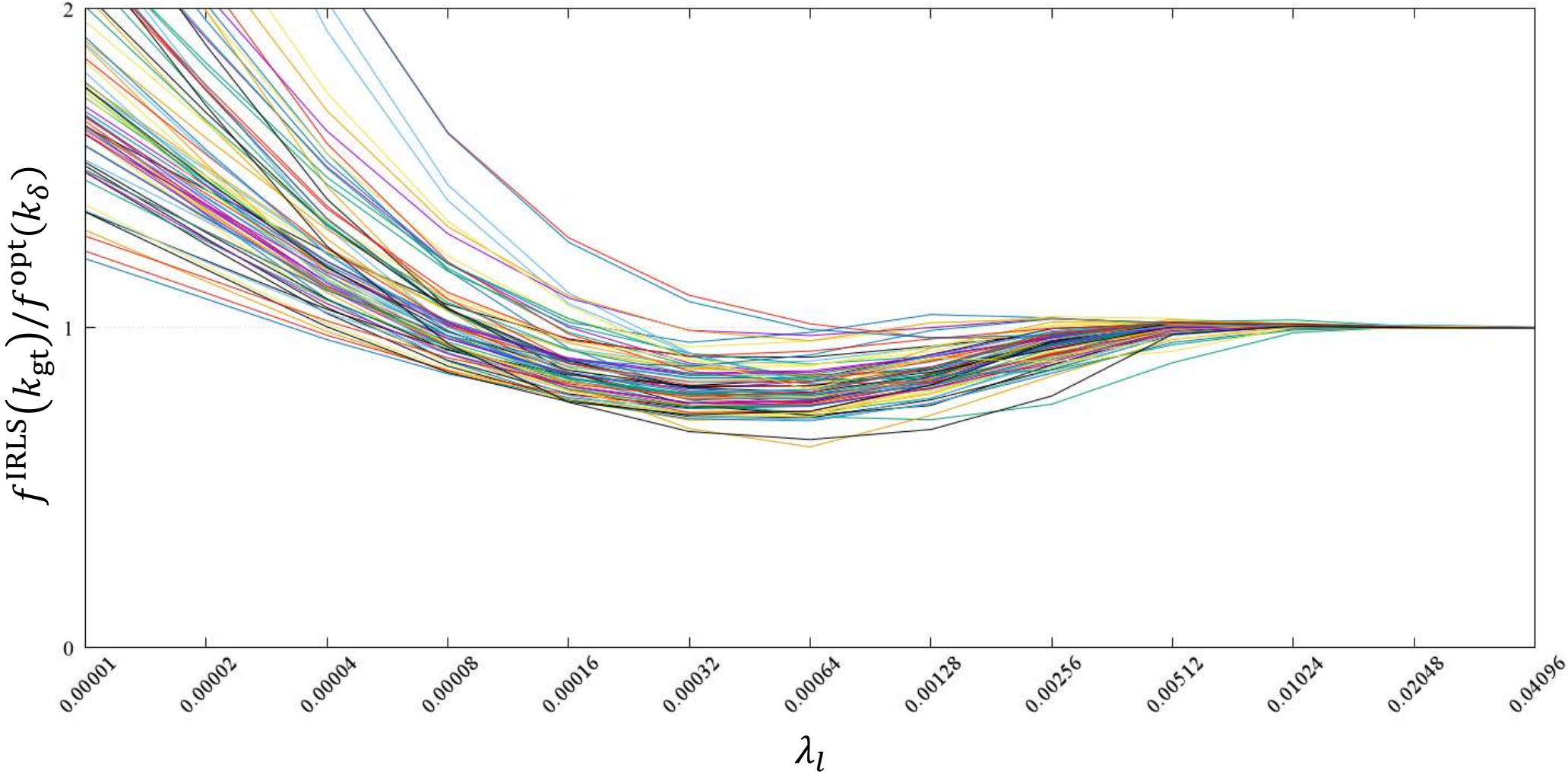
Sun et al.'s dataset - Kernel 6



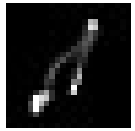


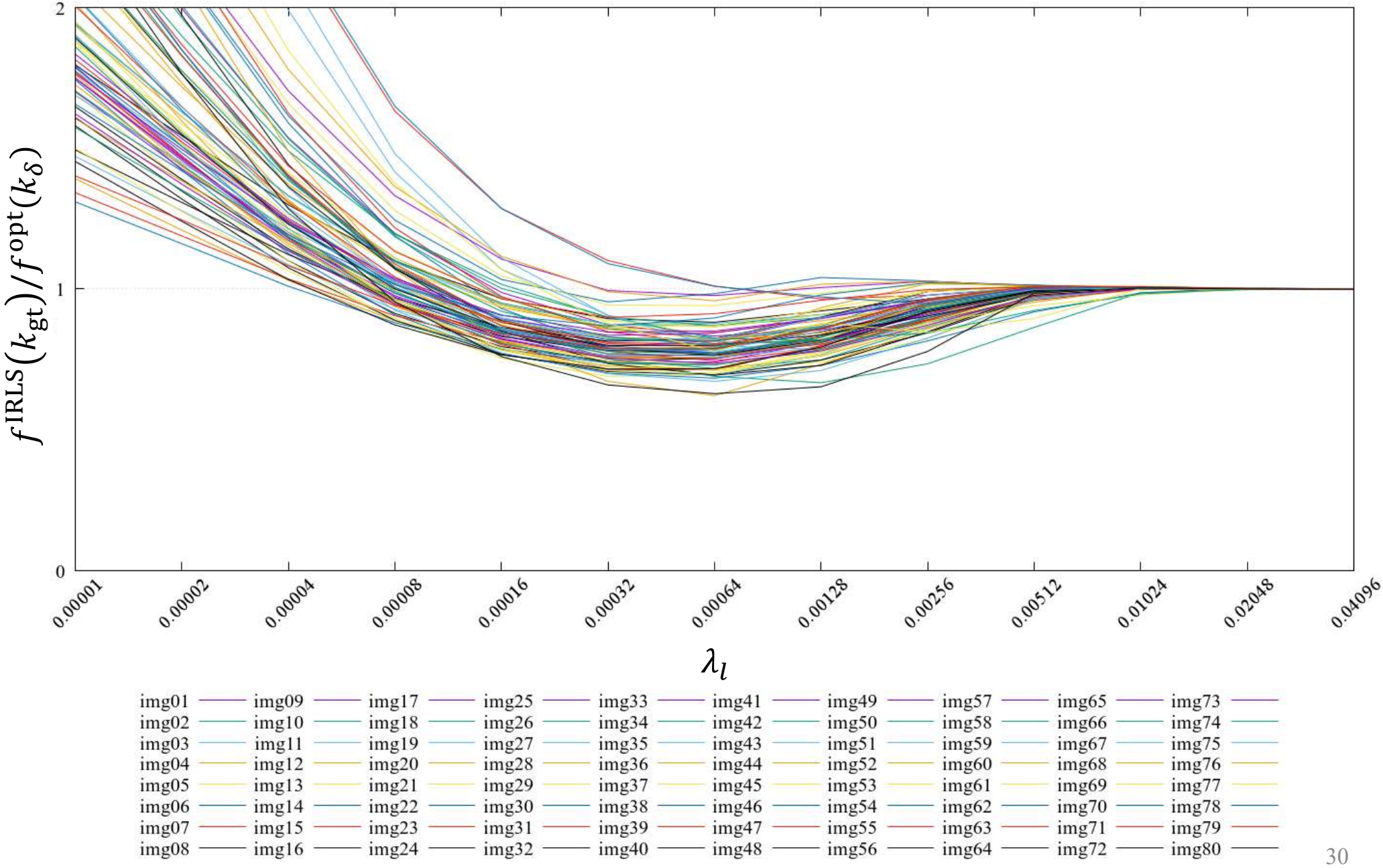
Sun et al.'s dataset - Kernel 7



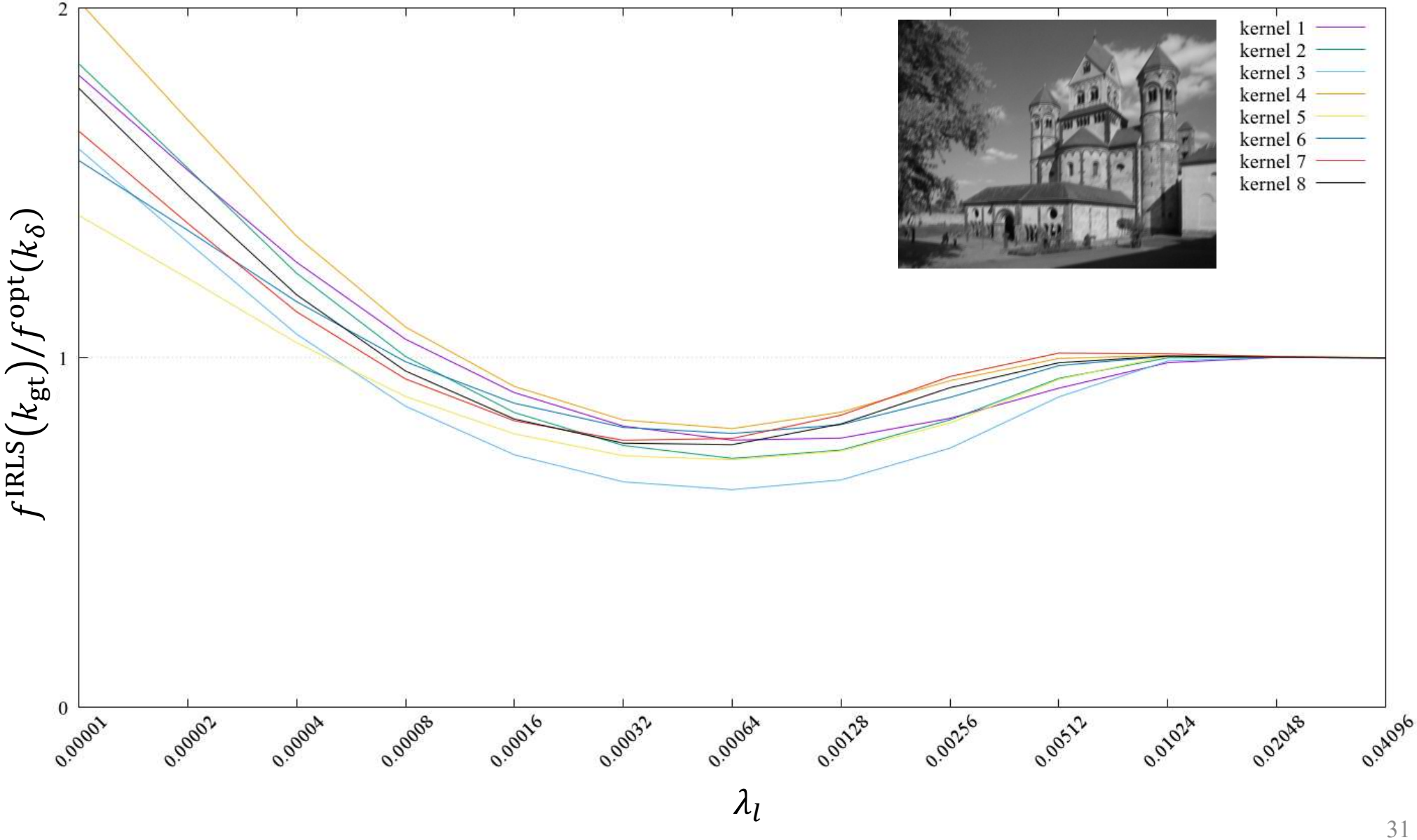


Sun et al.'s dataset - Kernel 8

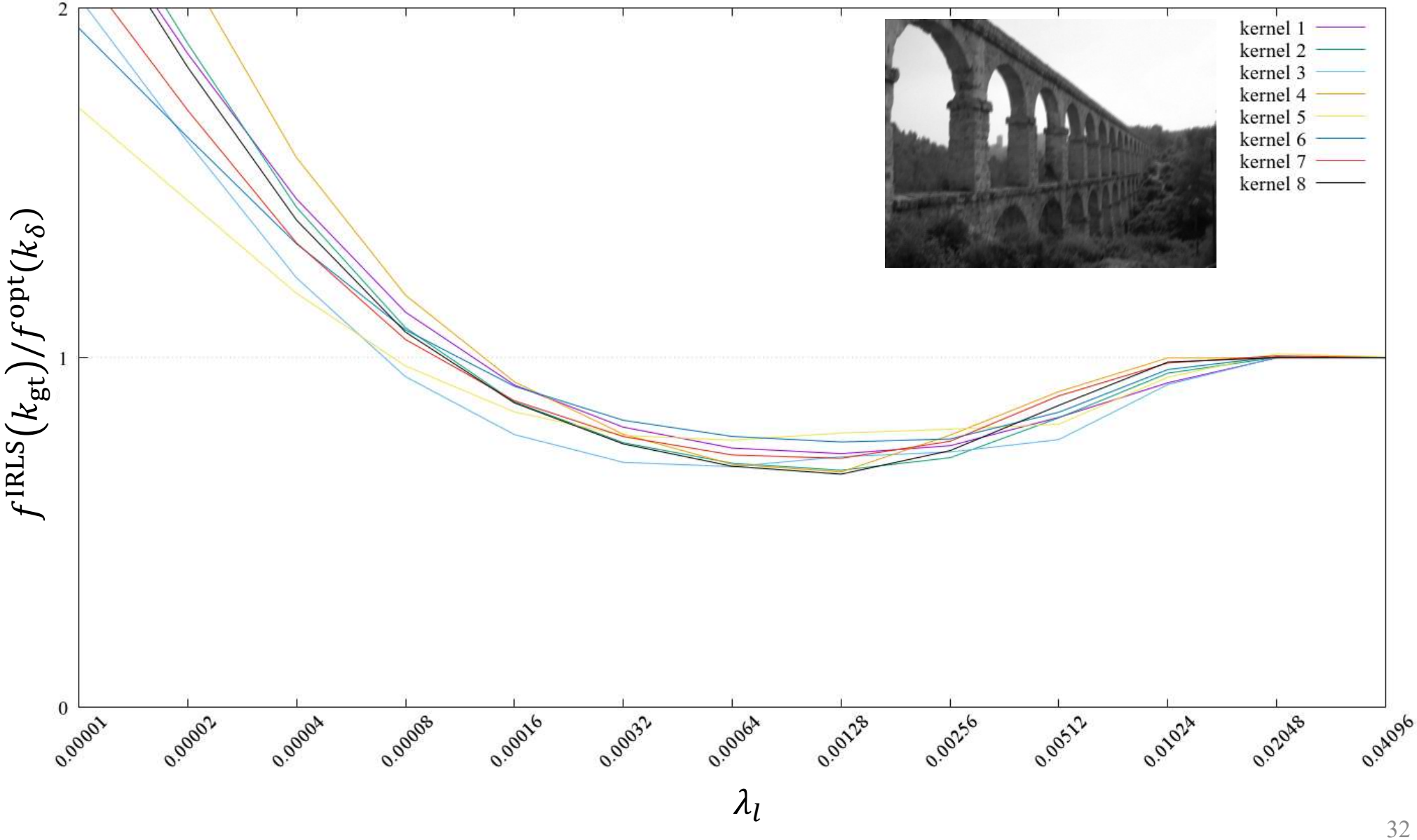




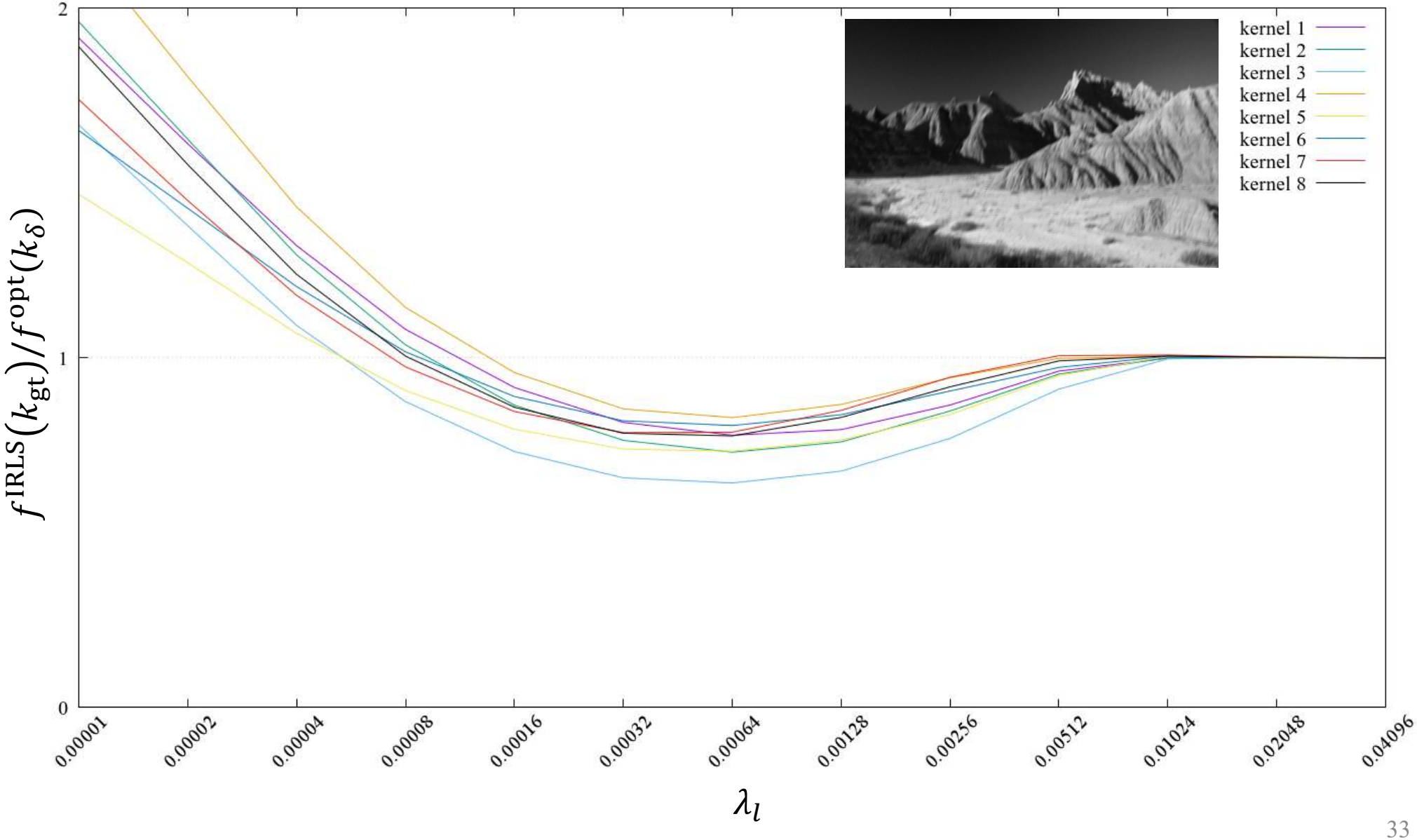
Sun et al.'s dataset - Image 01



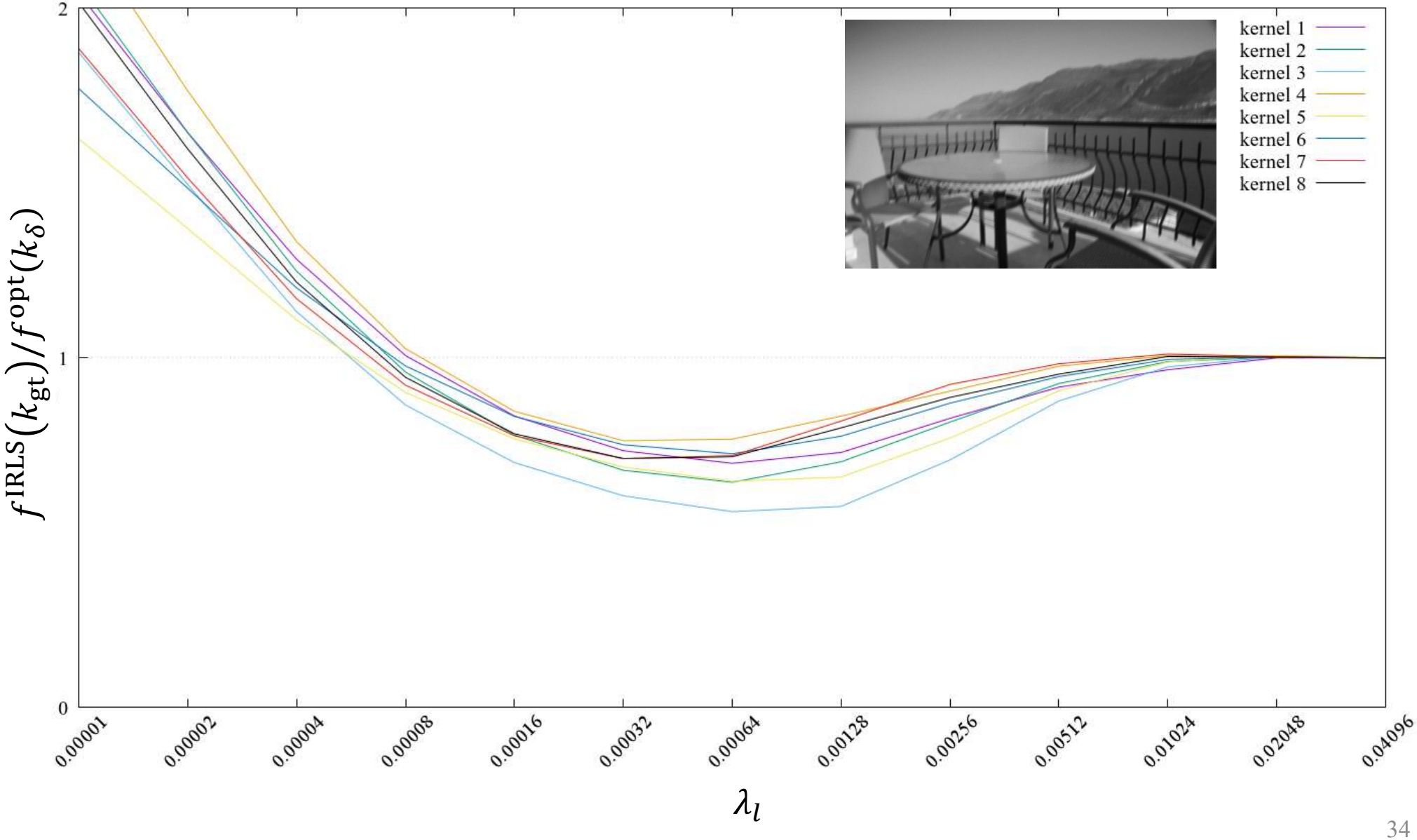
Sun et al.'s dataset - Image 02



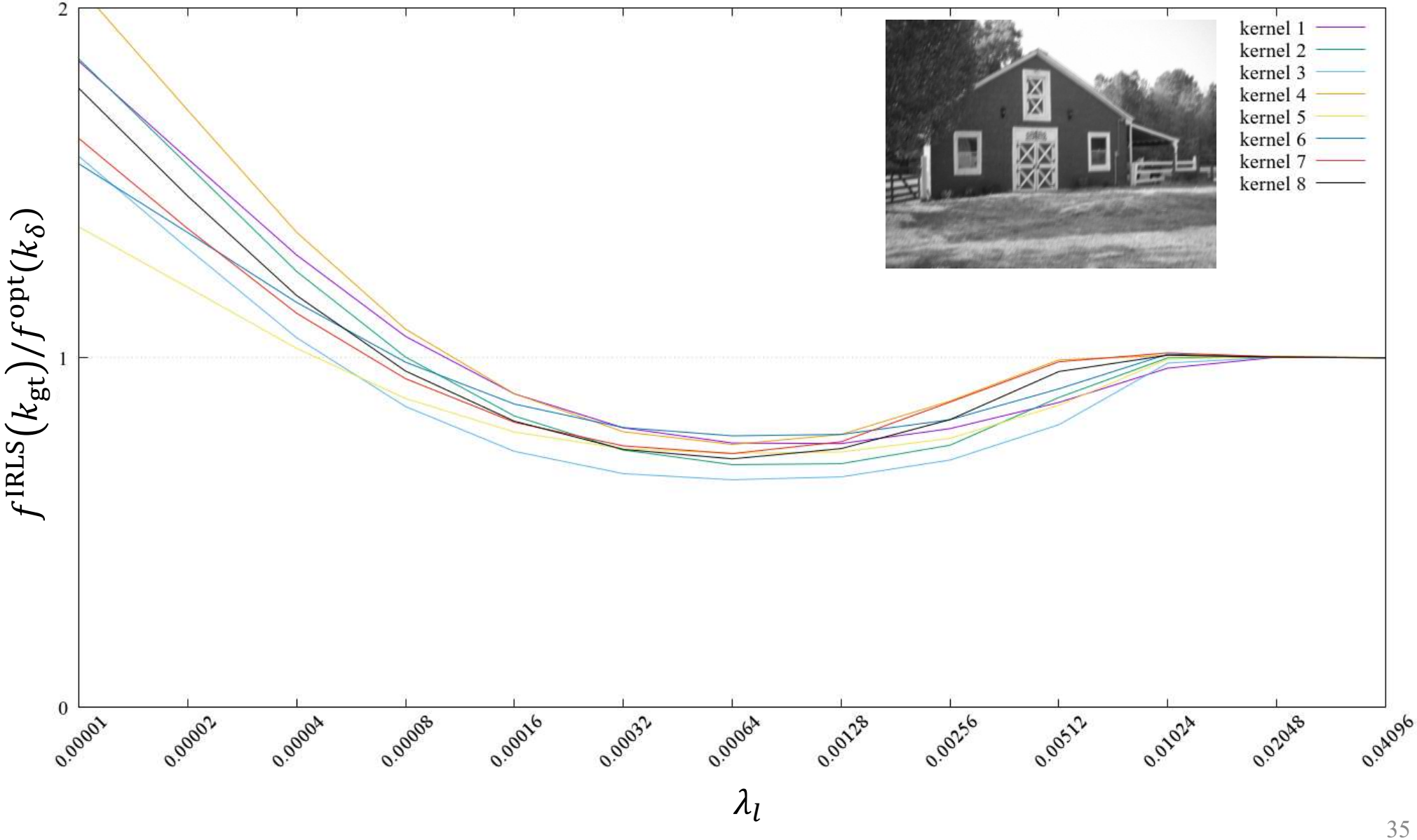
Sun et al.'s dataset - Image 03



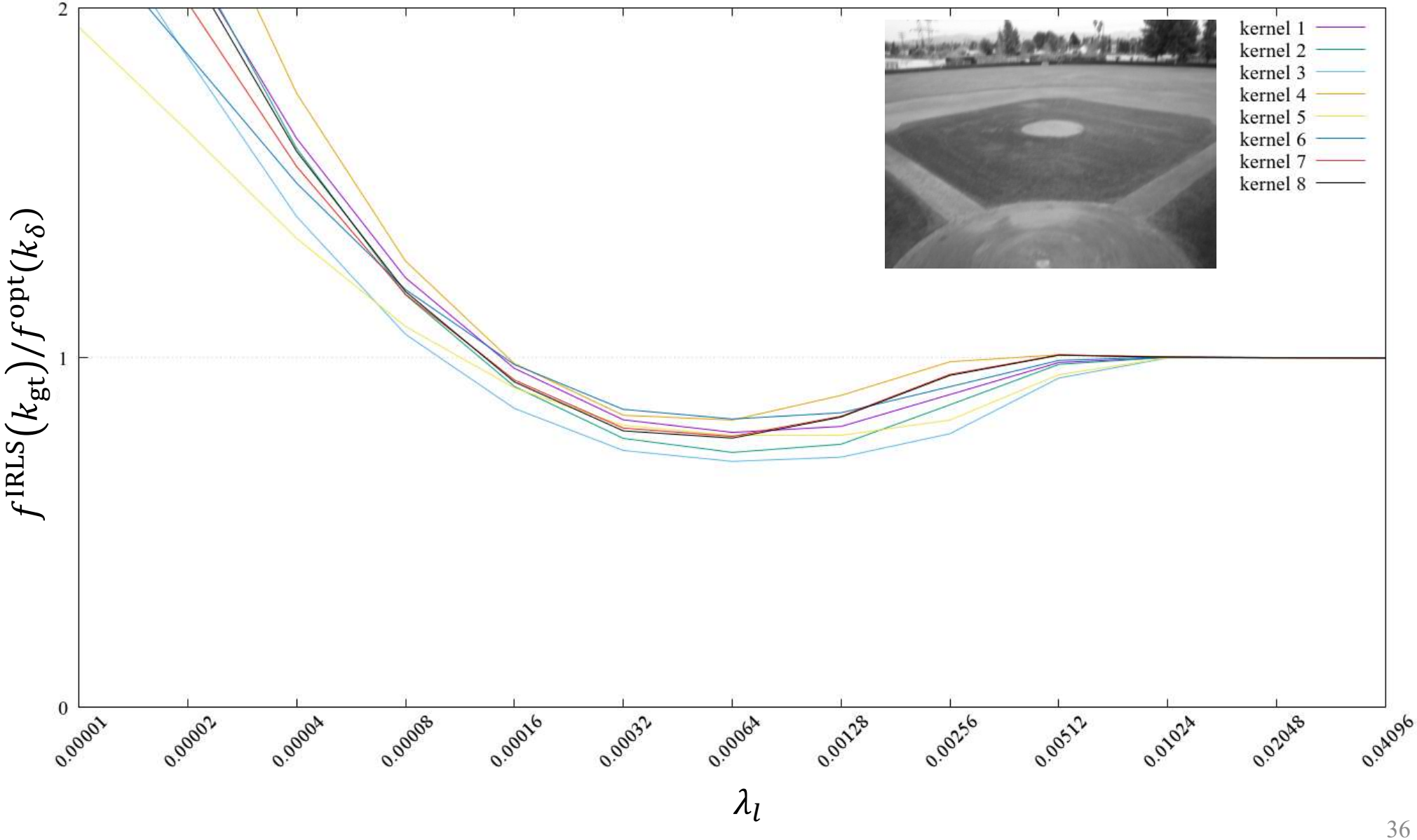
Sun et al.'s dataset - Image 04



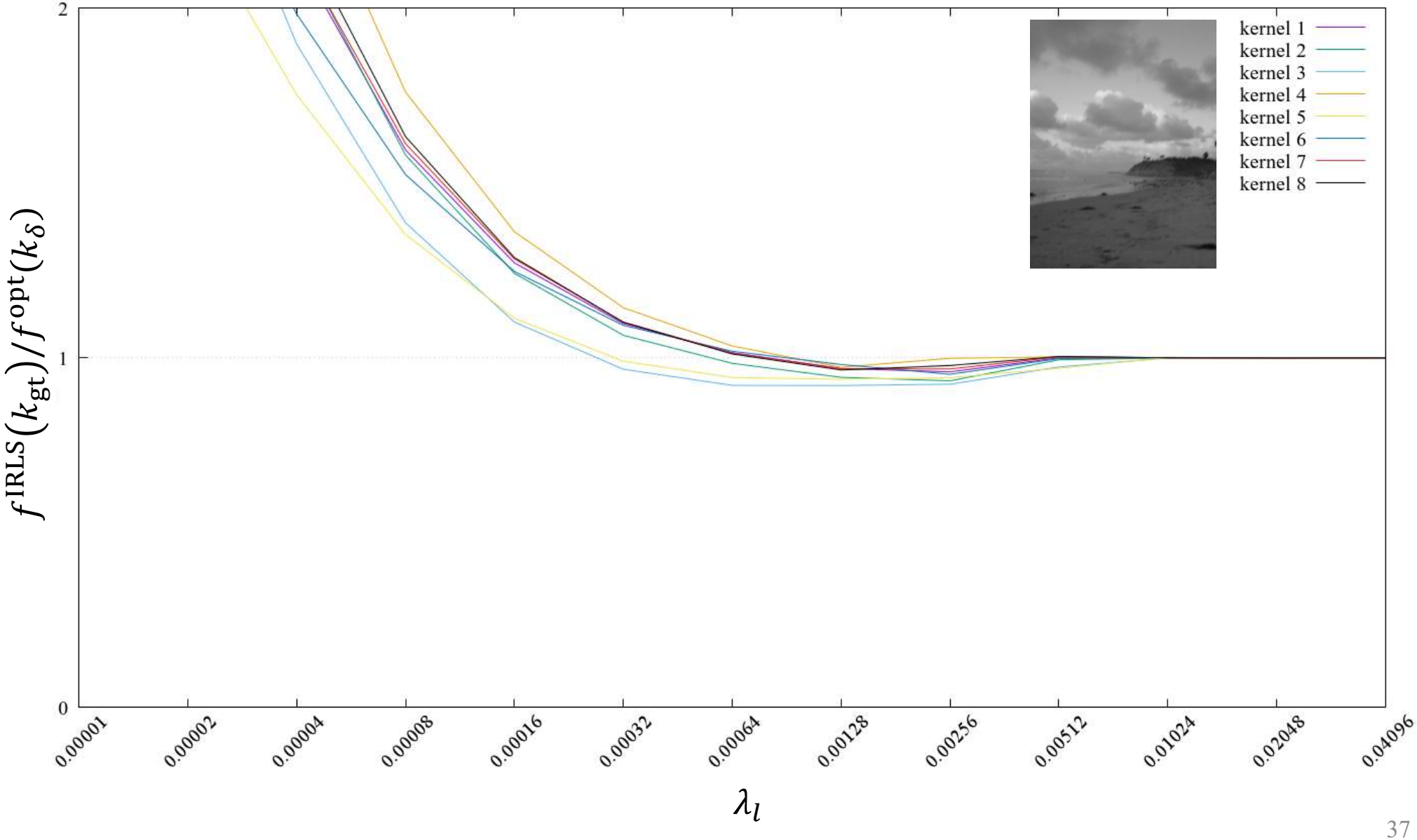
Sun et al.'s dataset - Image 05



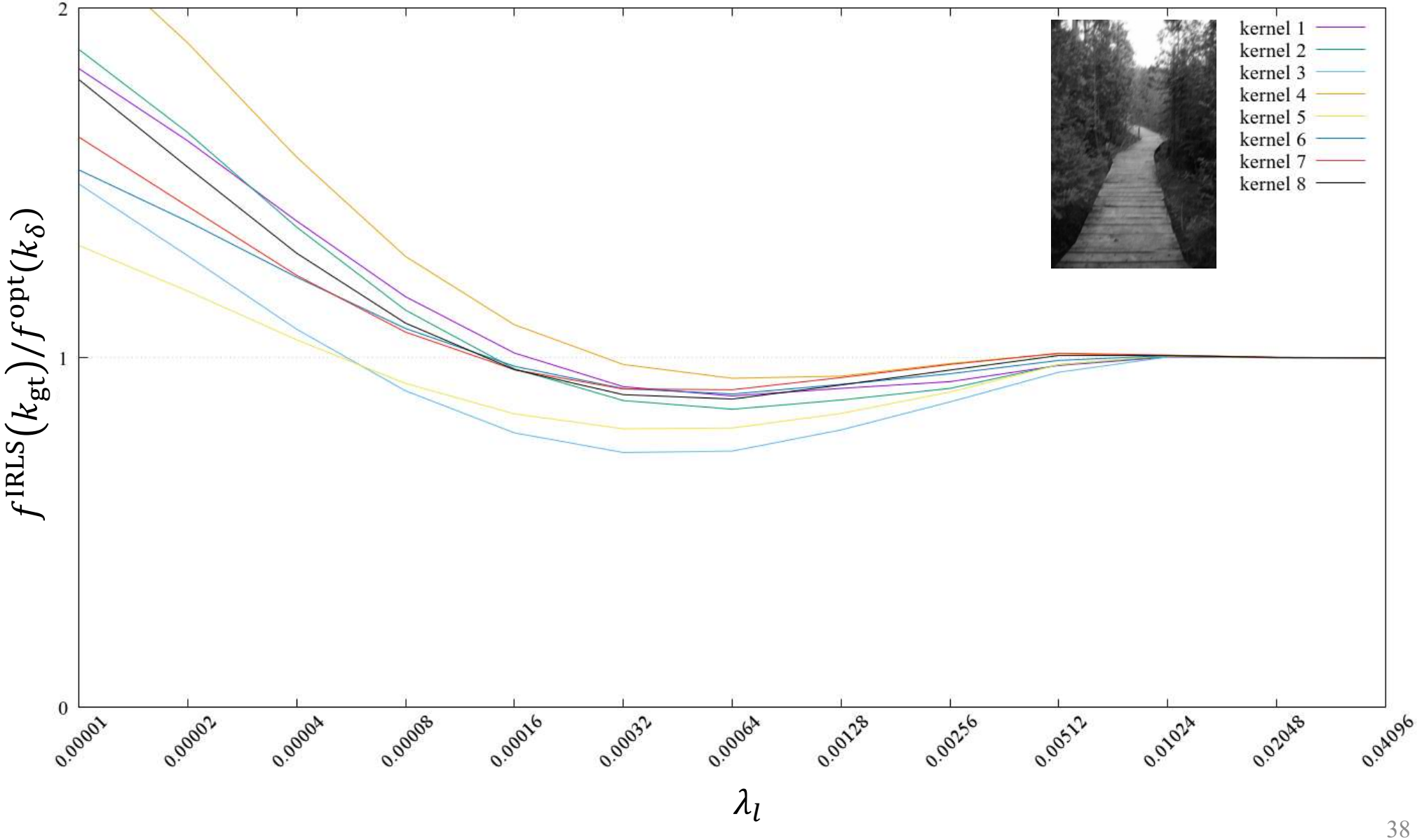
Sun et al.'s dataset - Image 06



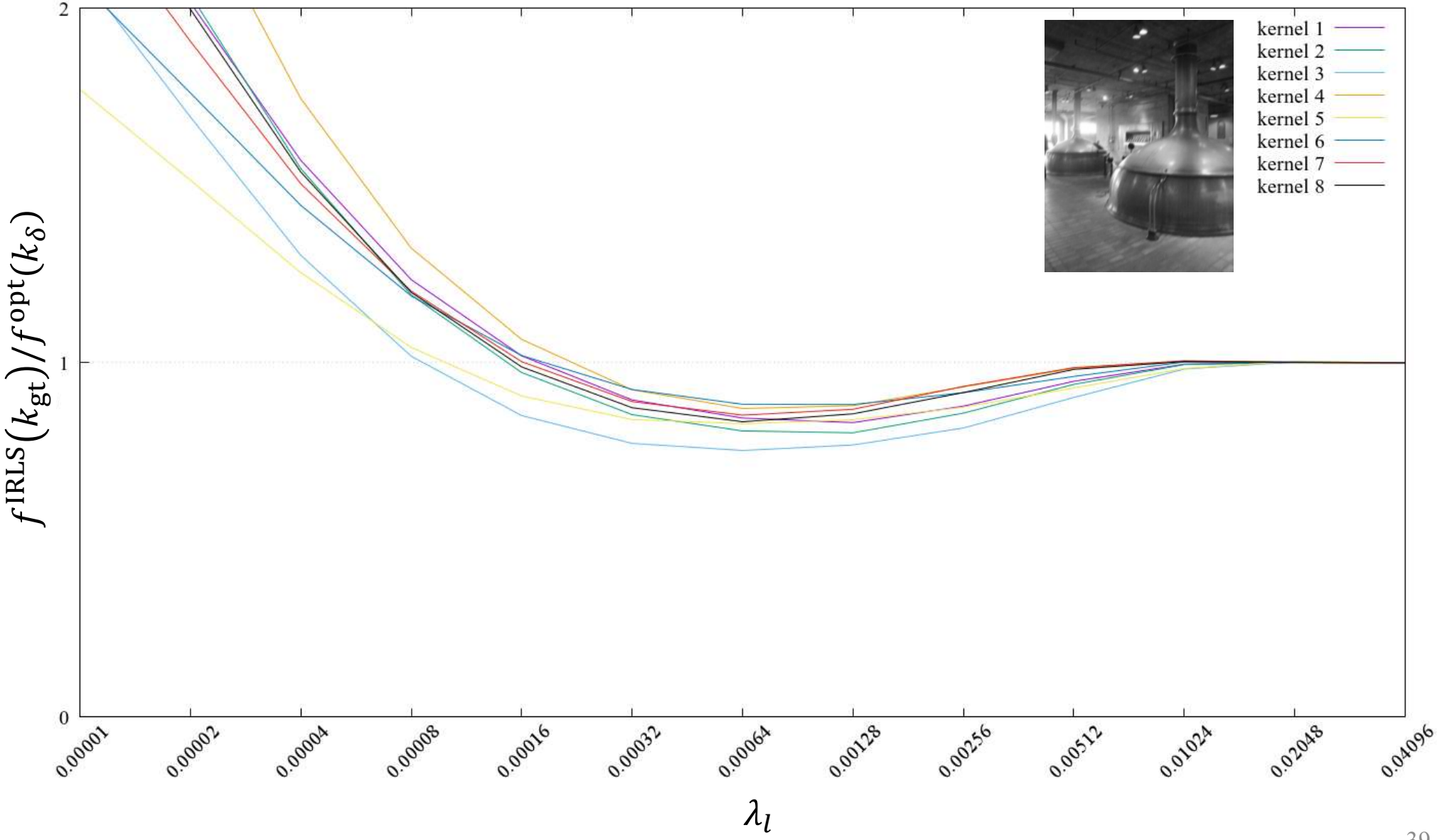
Sun et al.'s dataset - Image 07



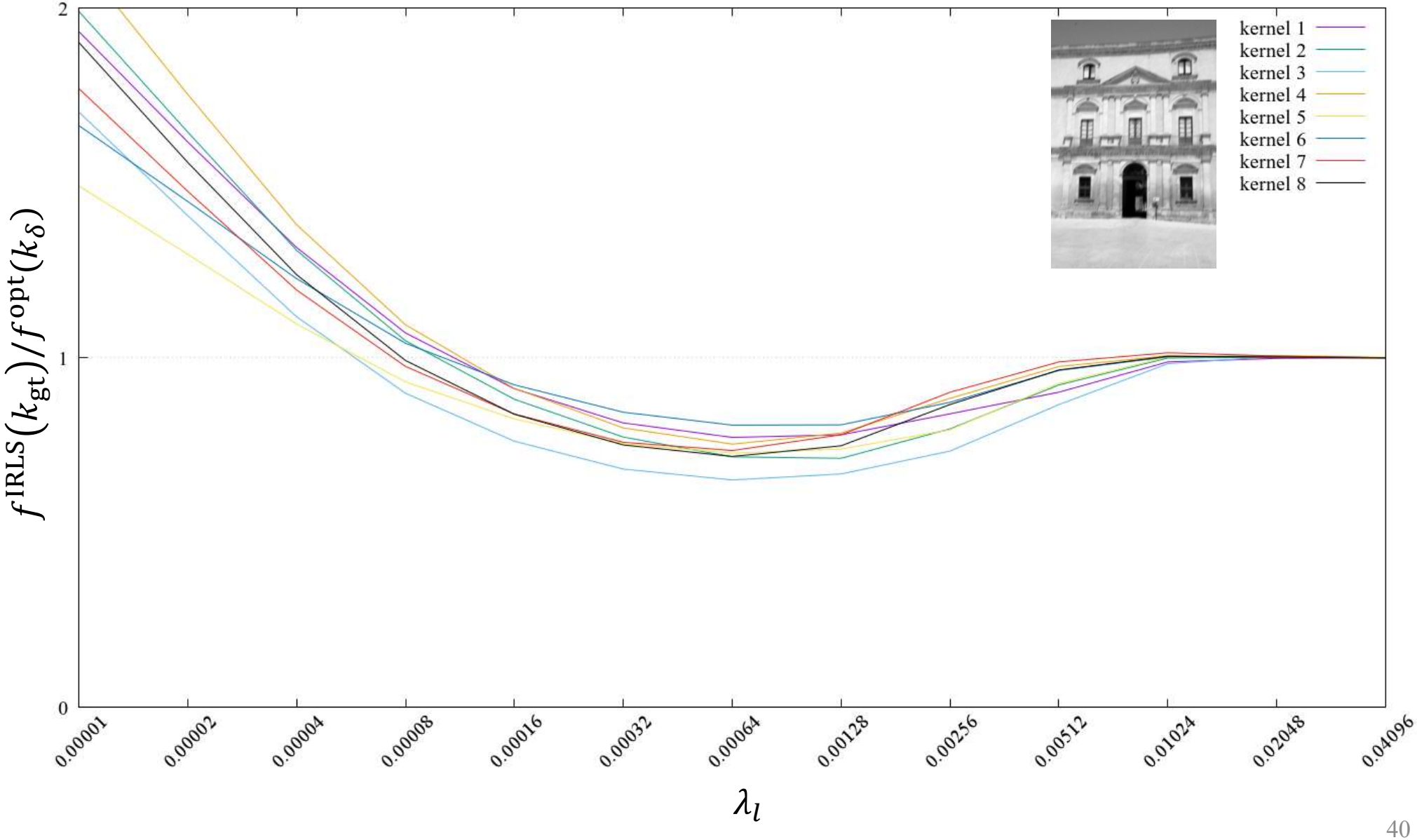
Sun et al.'s dataset - Image 08



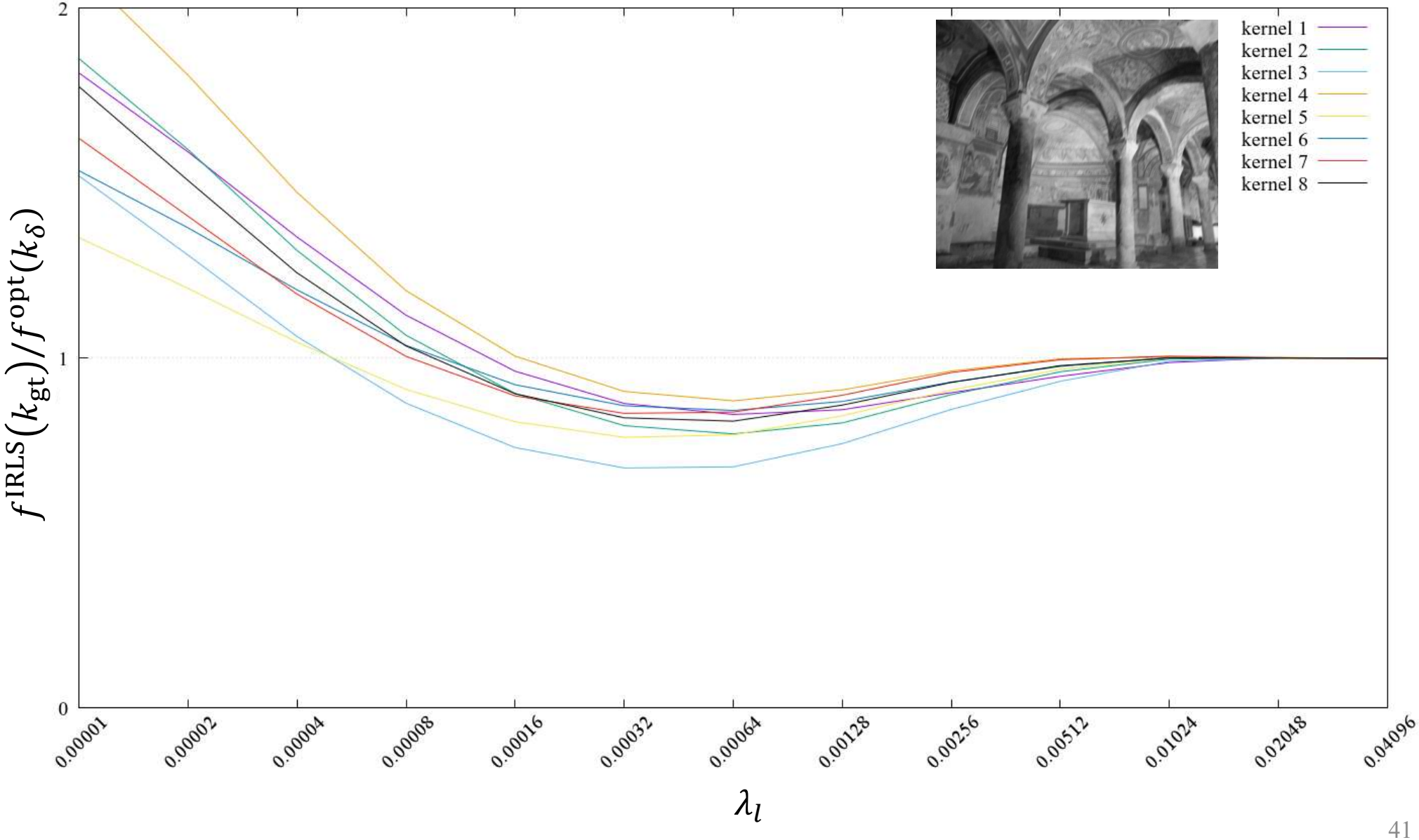
Sun et al.'s dataset - Image 09



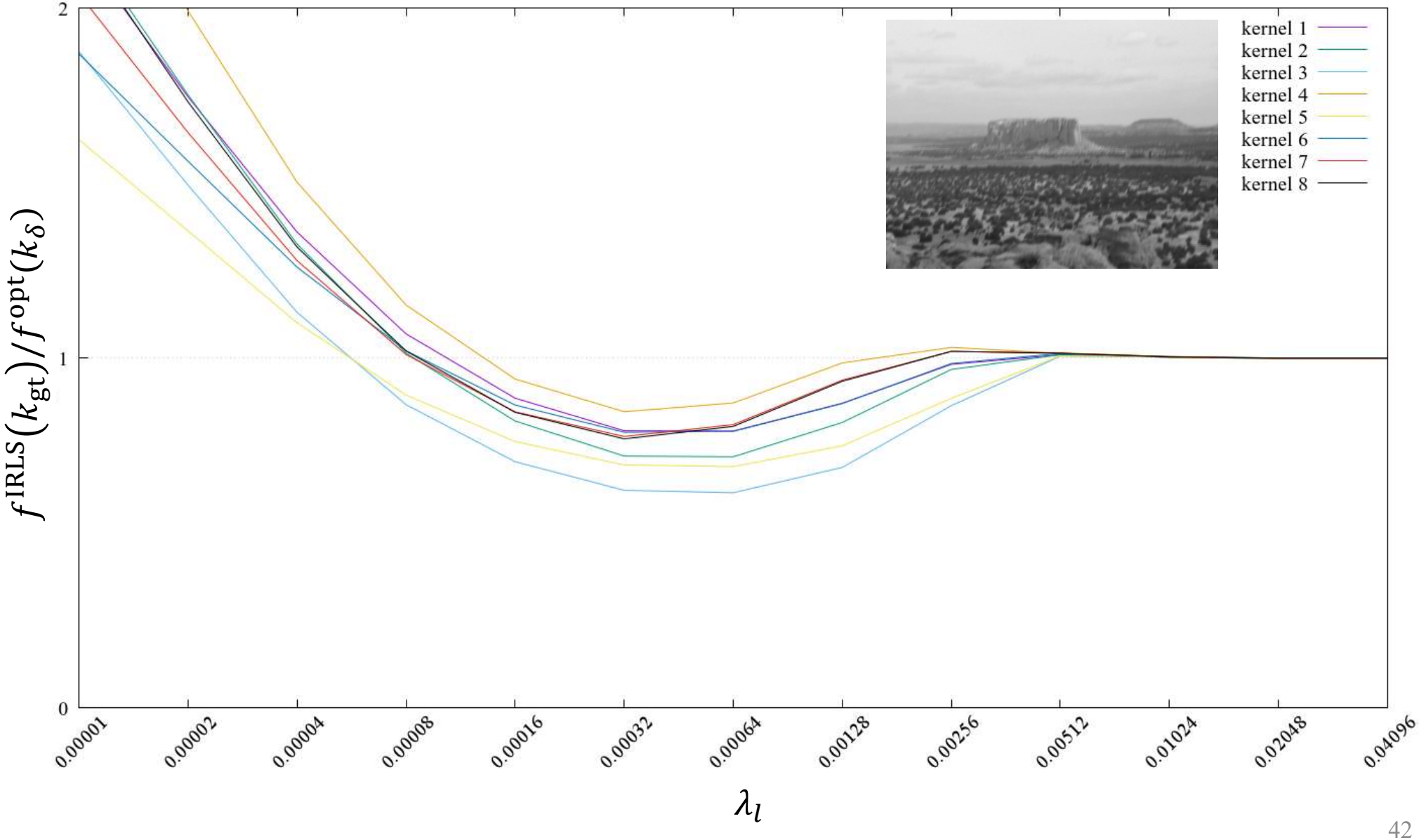
Sun et al.'s dataset - Image 10



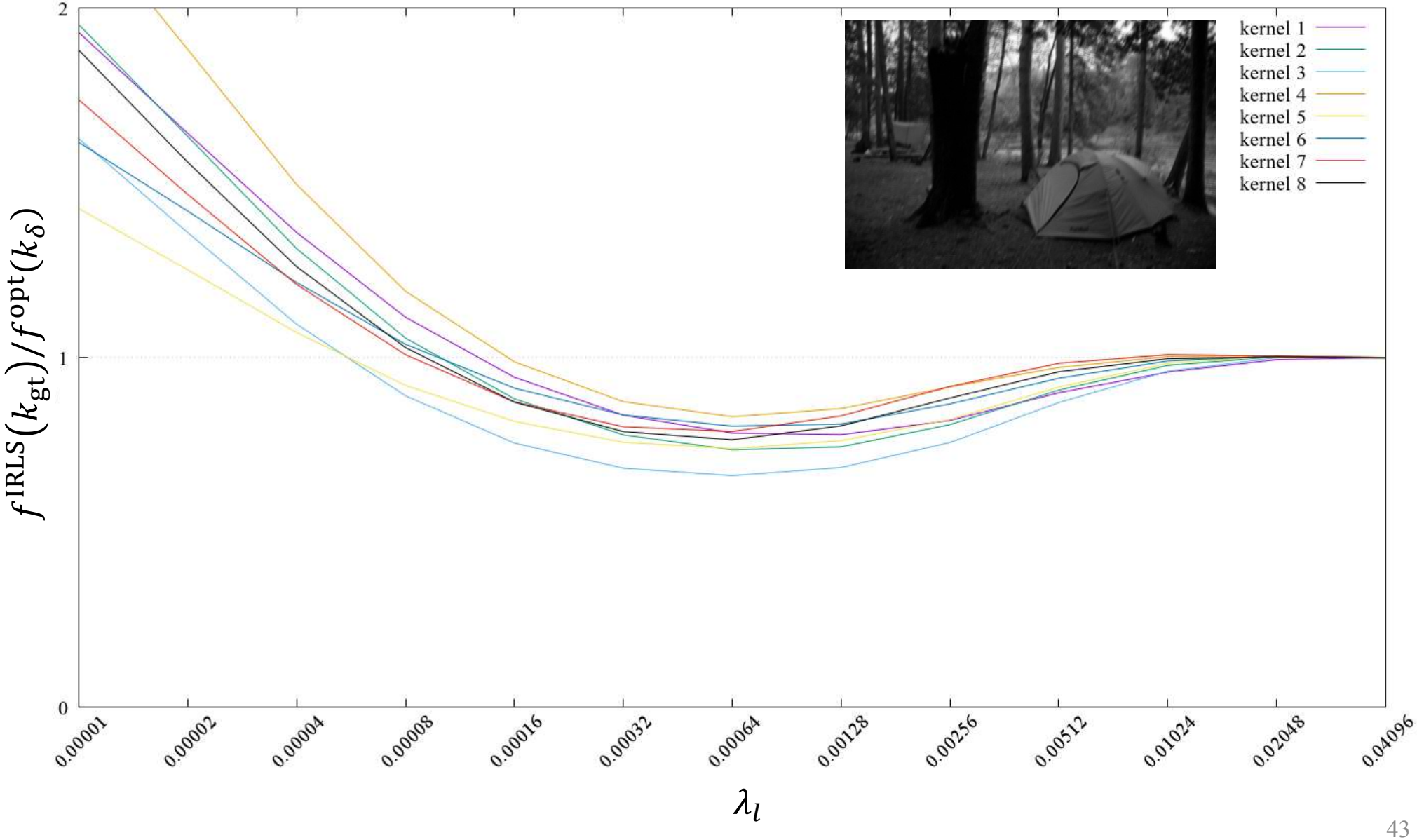
Sun et al.'s dataset - Image 11



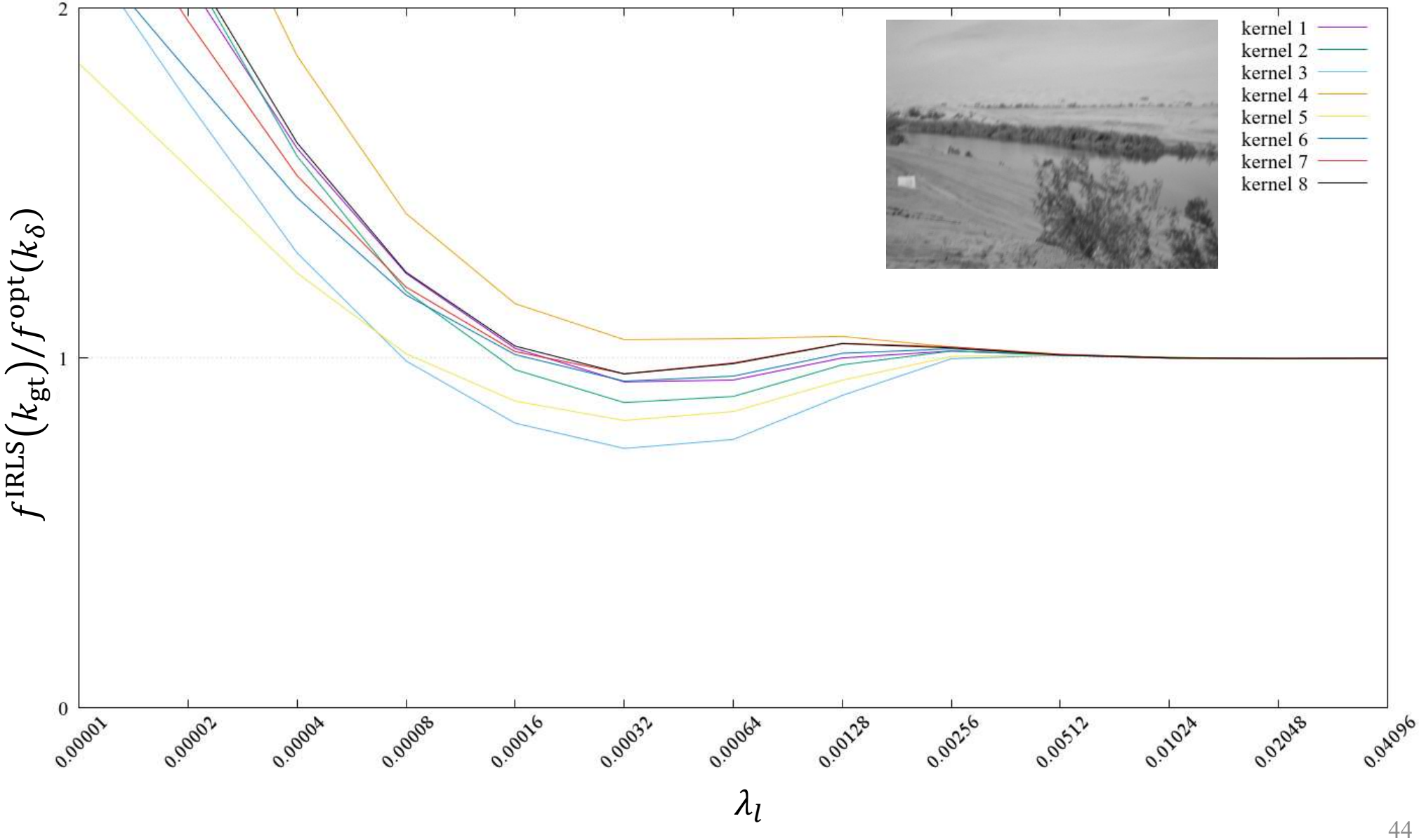
Sun et al.'s dataset - Image 12



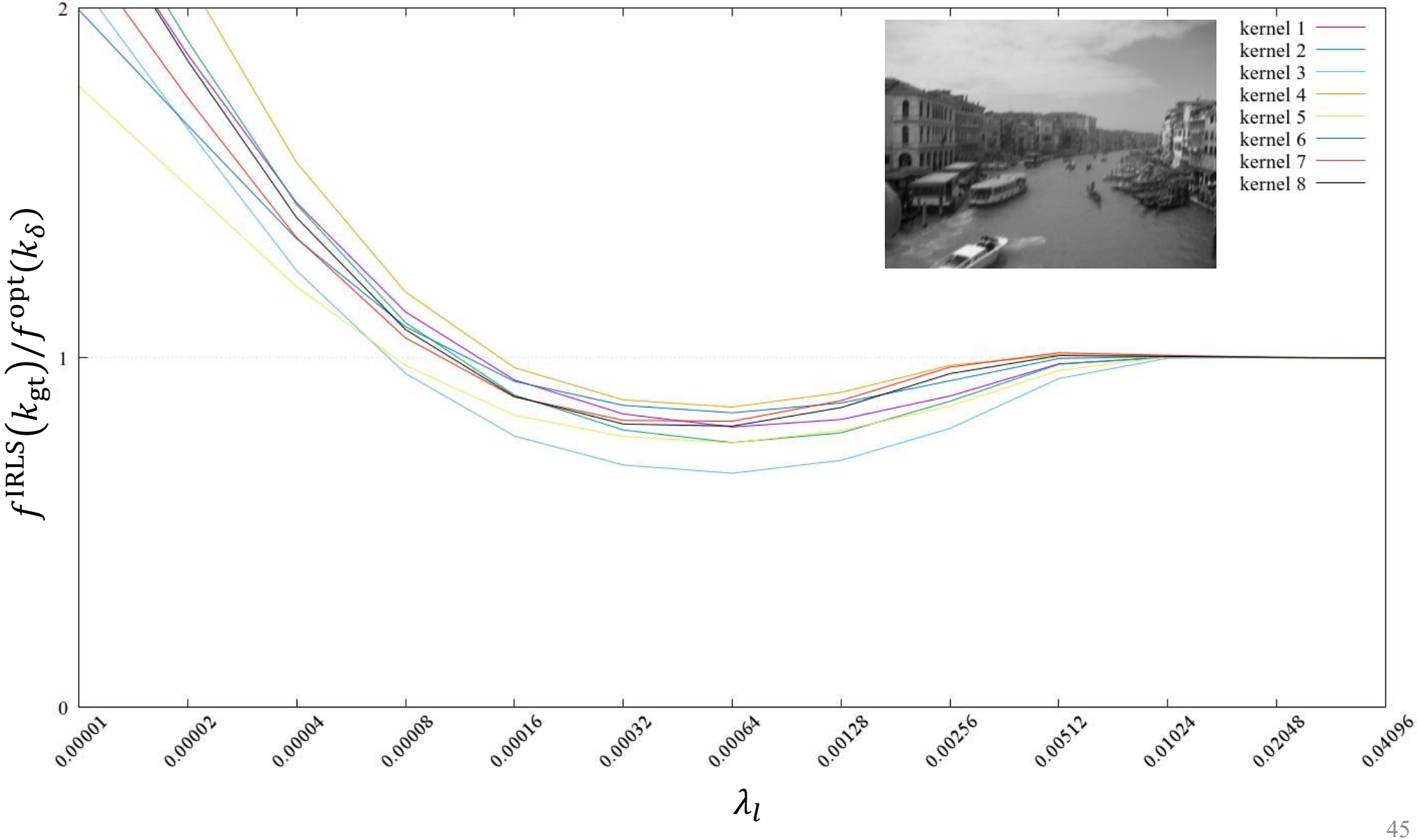
Sun et al.'s dataset - Image 13



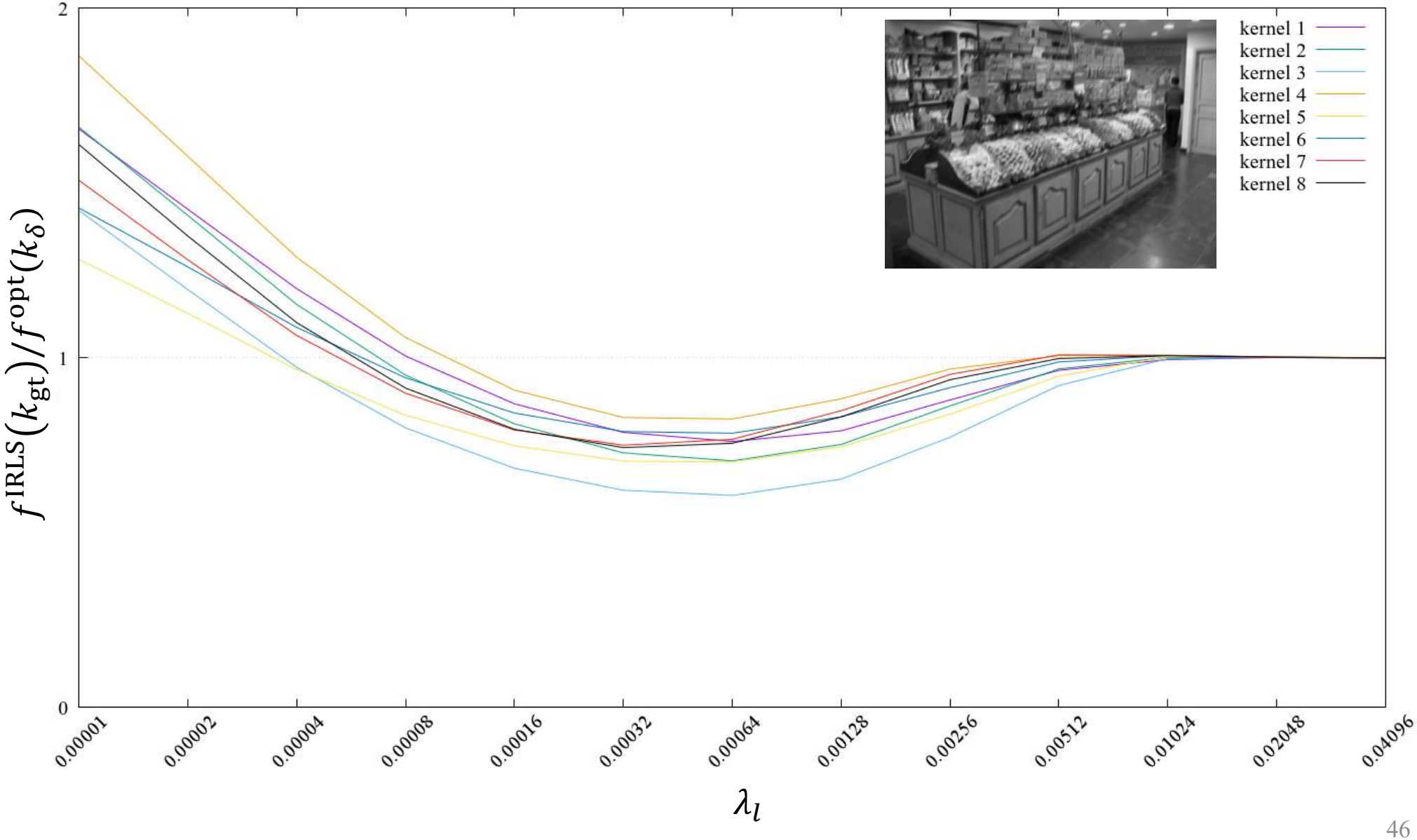
Sun et al.'s dataset - Image 14



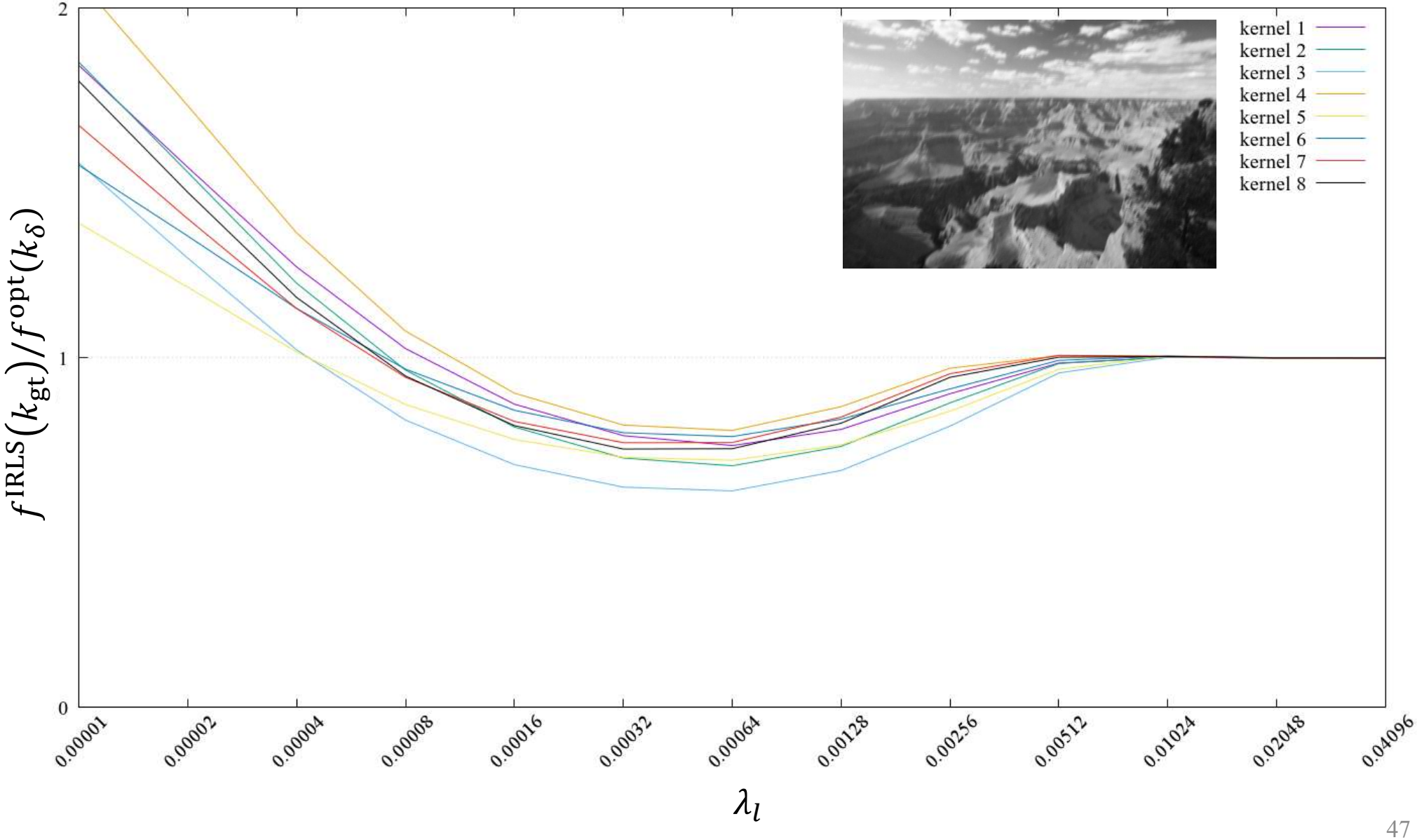
Sun et al.'s dataset - Image 15



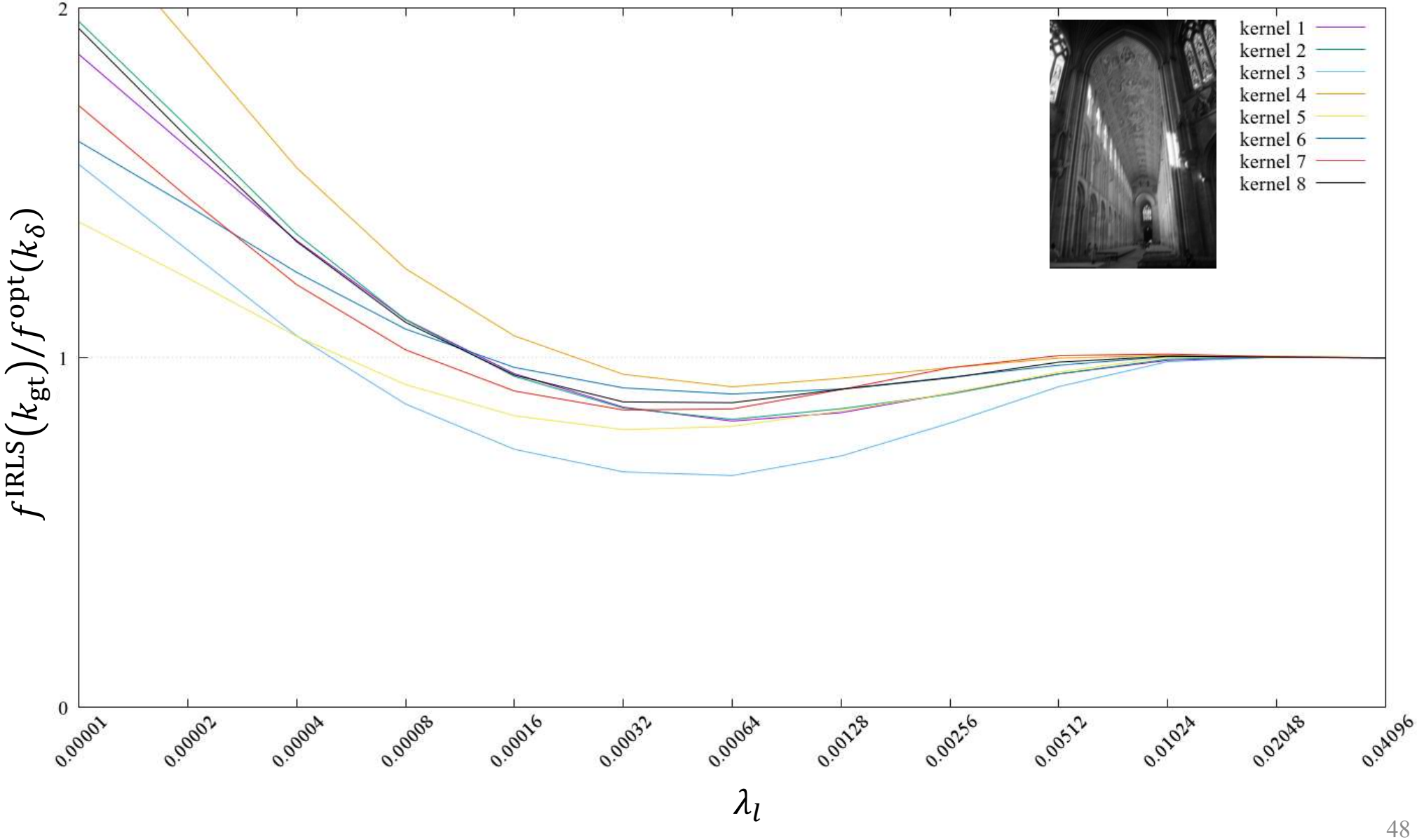
Sun et al.'s dataset - Image 16



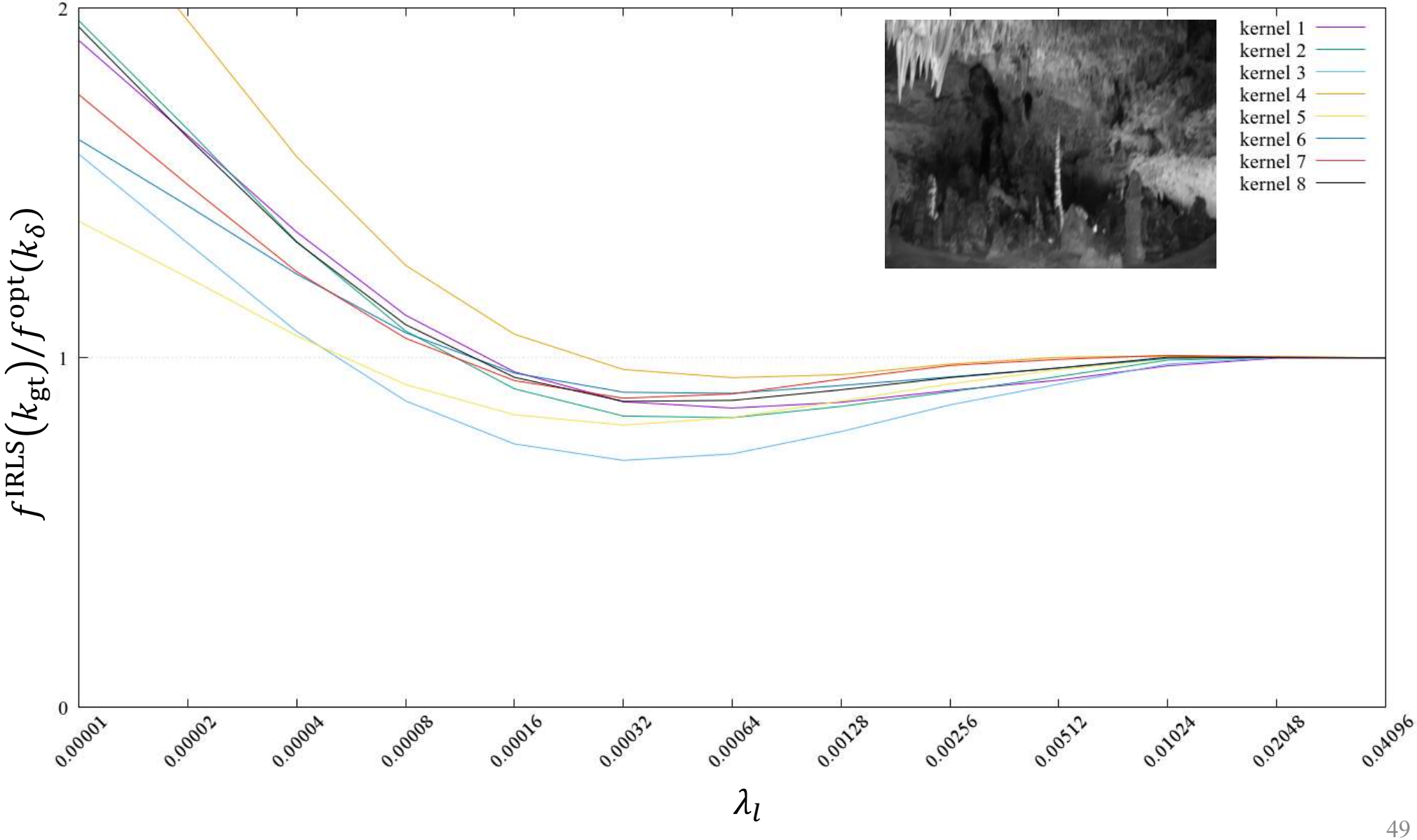
Sun et al.'s dataset - Image 17



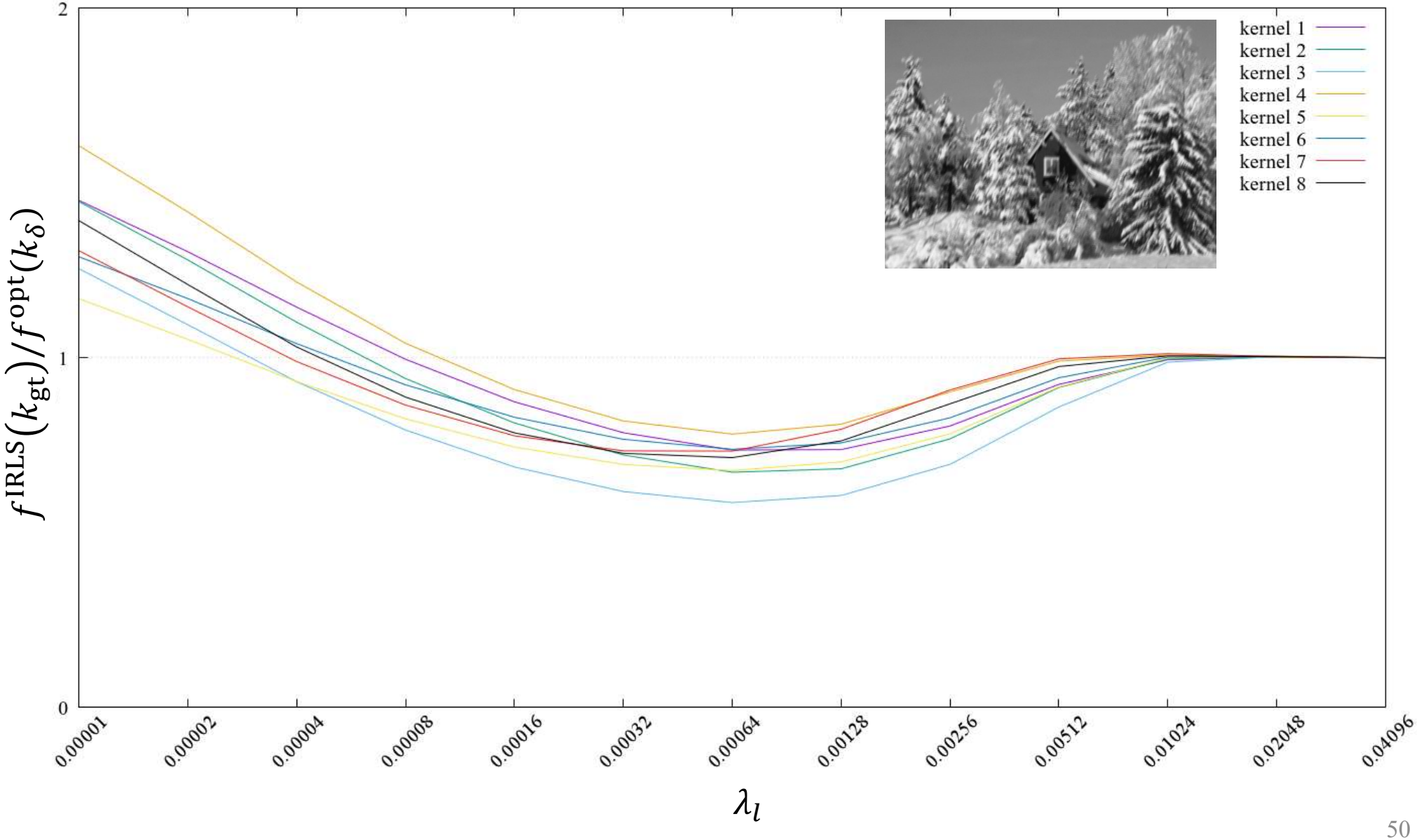
Sun et al.'s dataset - Image 18



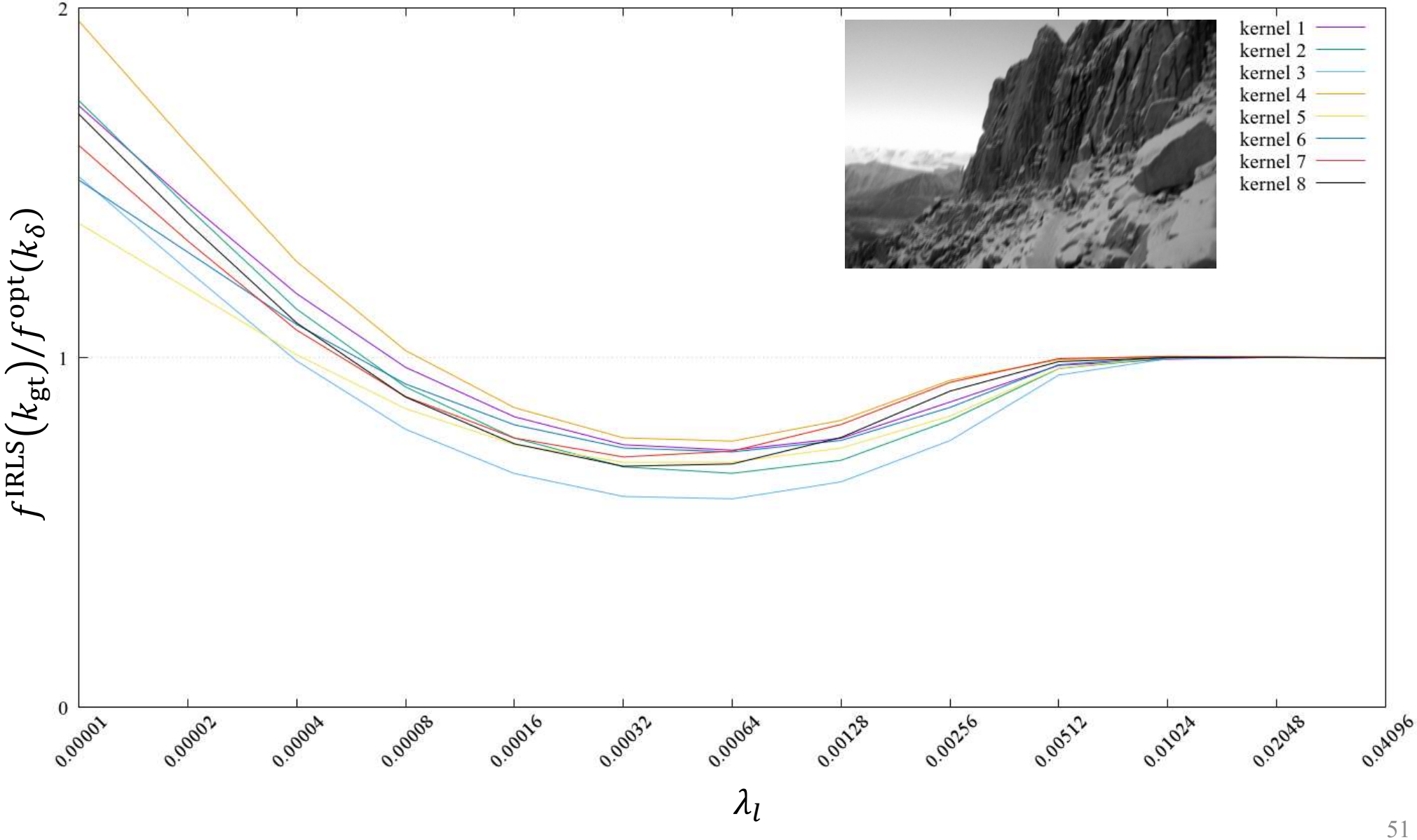
Sun et al.'s dataset - Image 19



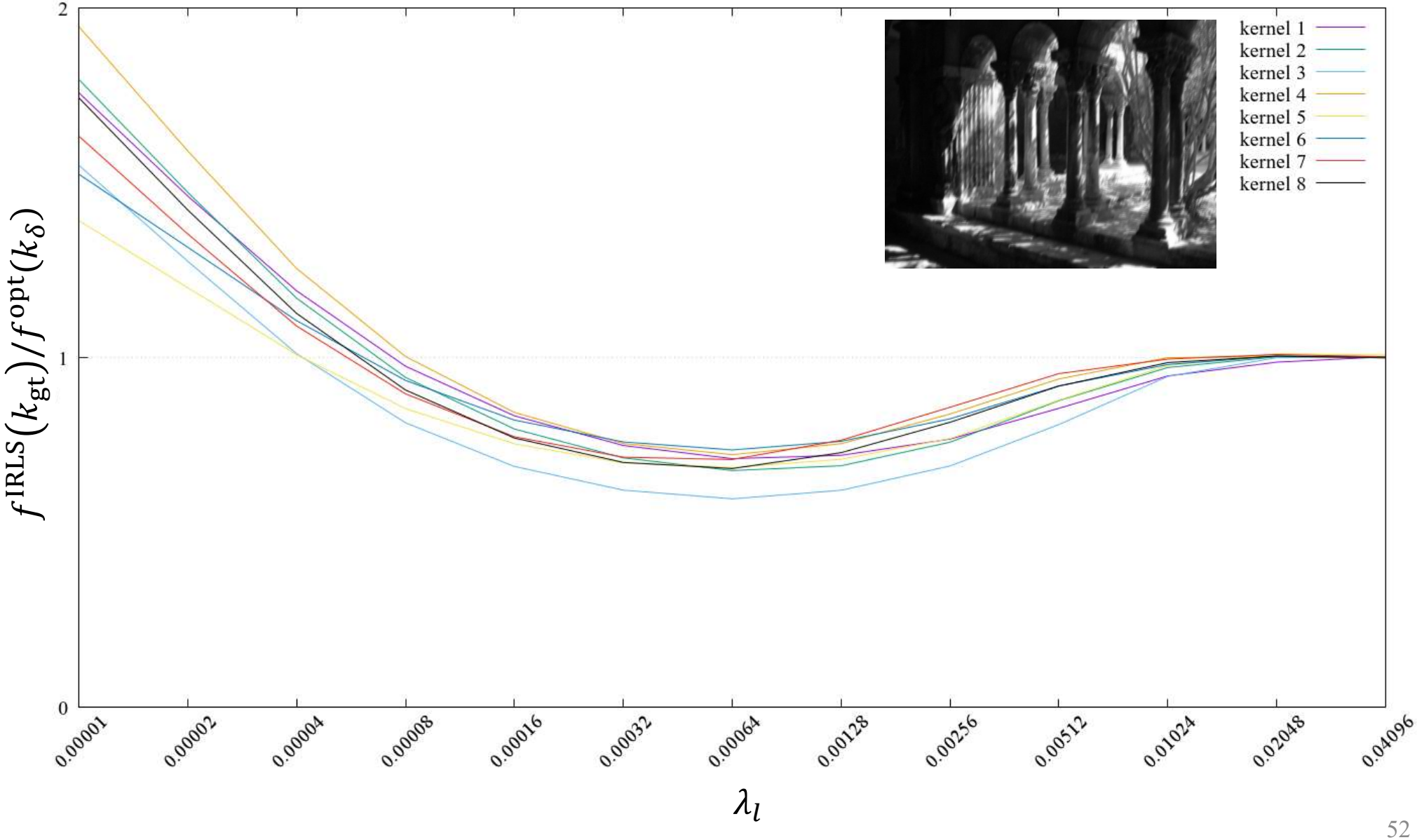
Sun et al.'s dataset - Image 20



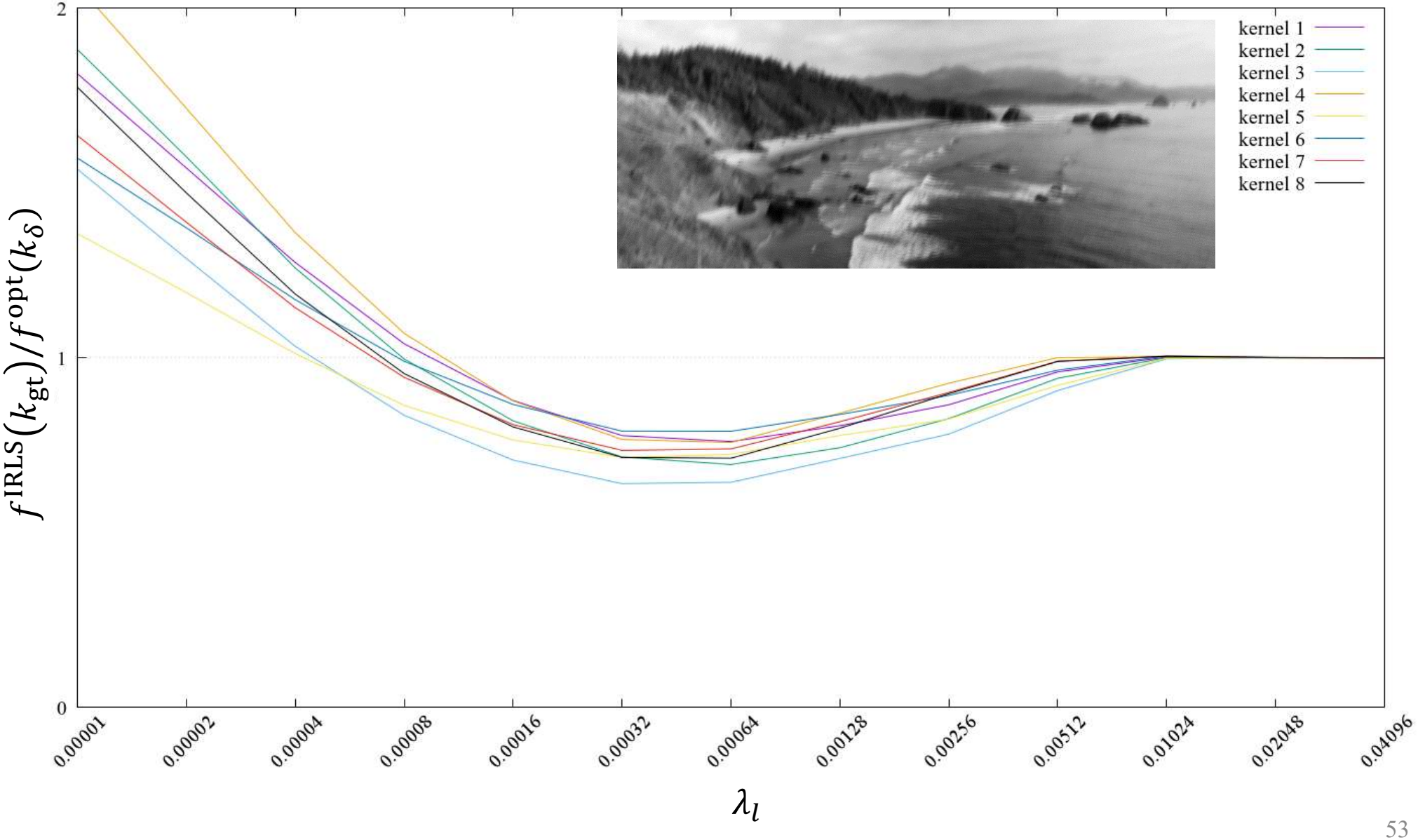
Sun et al.'s dataset - Image 21



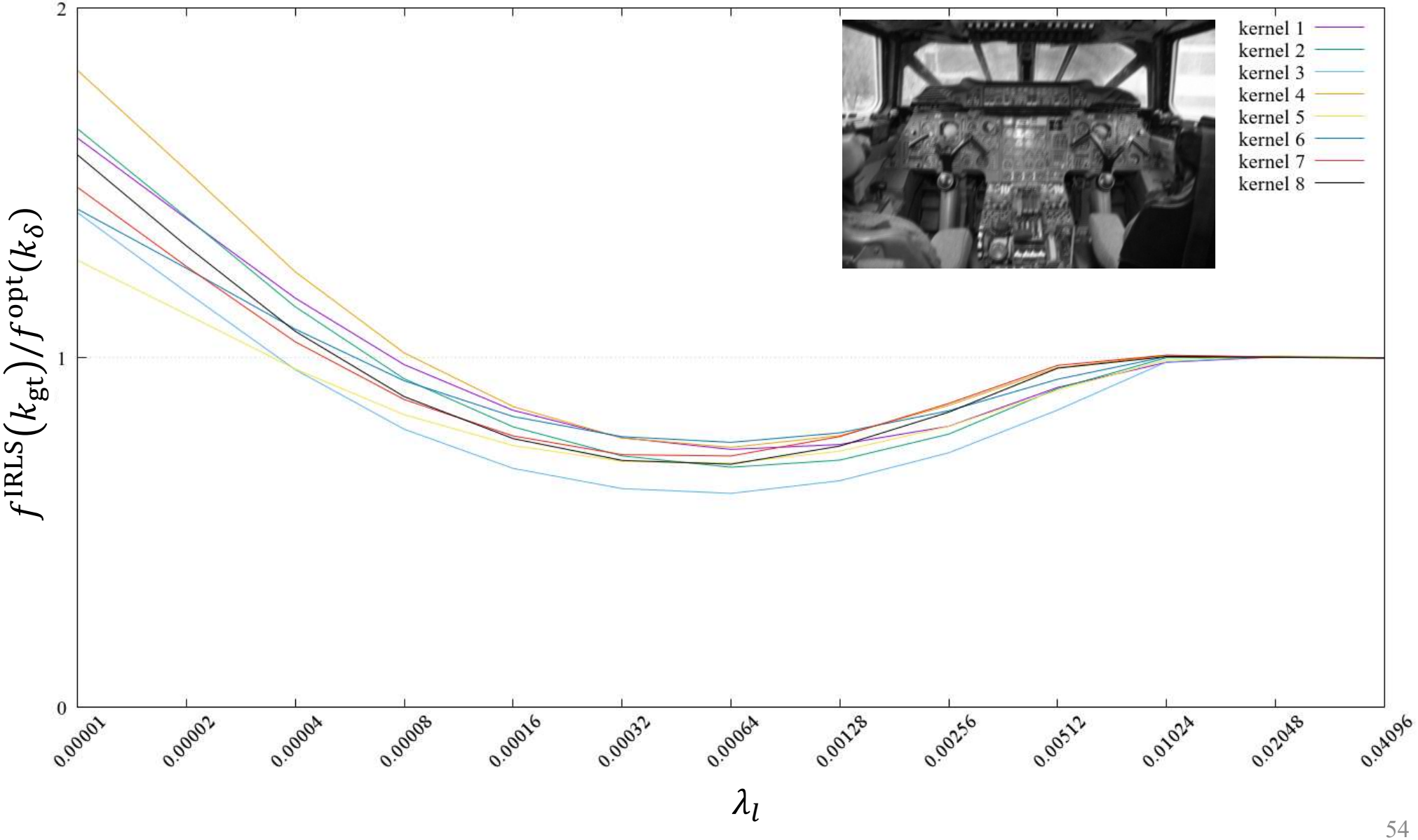
Sun et al.'s dataset - Image 22



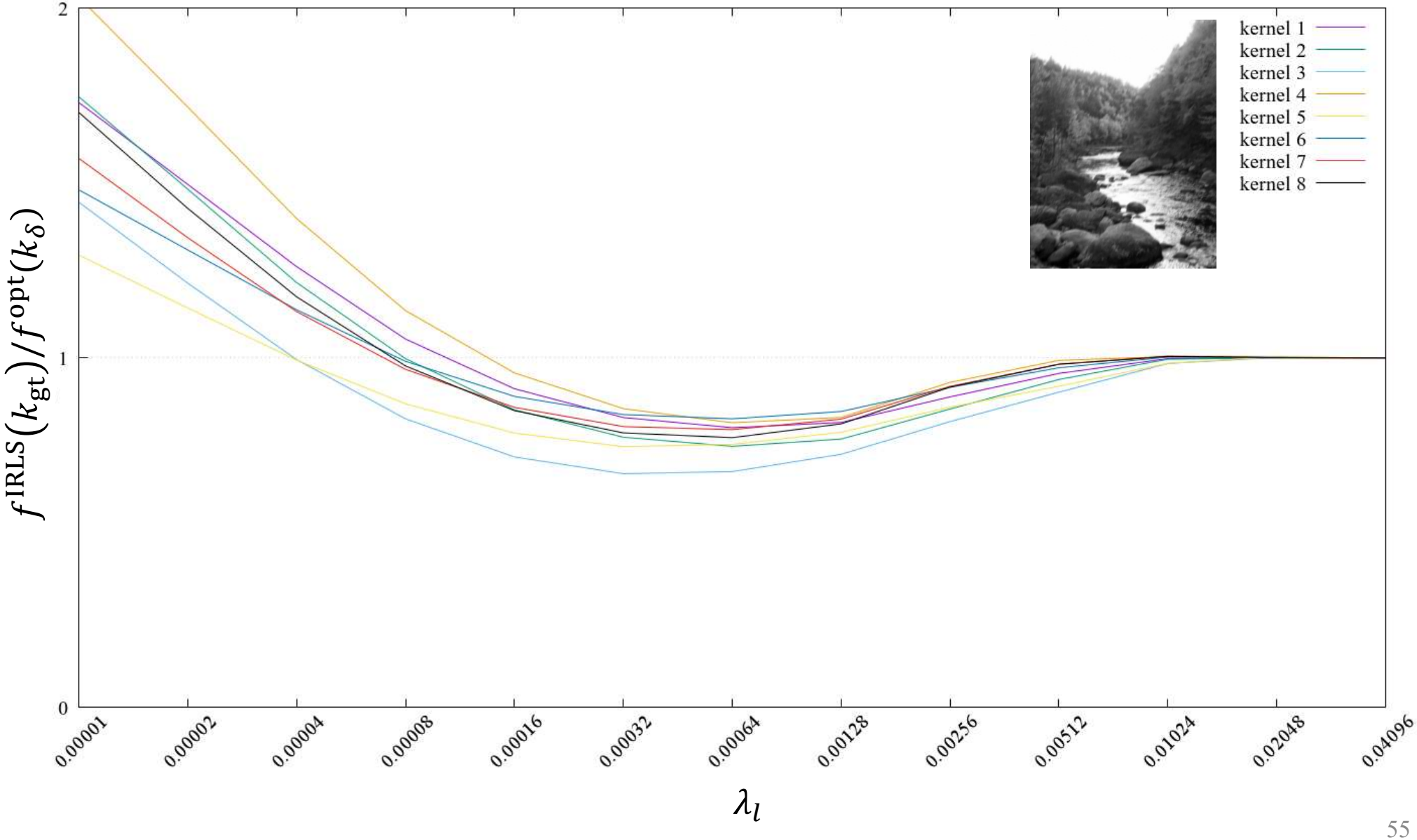
Sun et al.'s dataset - Image 23



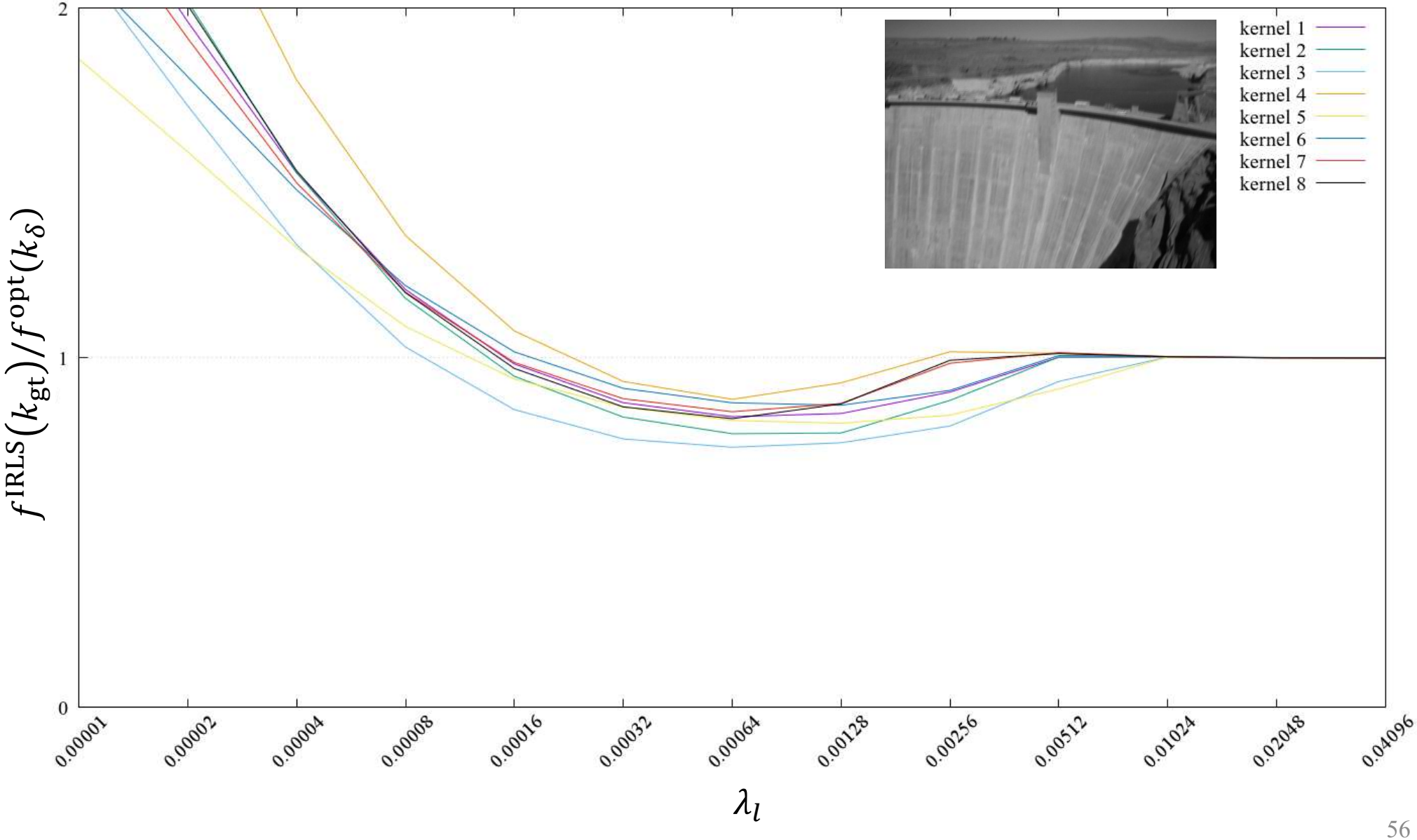
Sun et al.'s dataset - Image 24



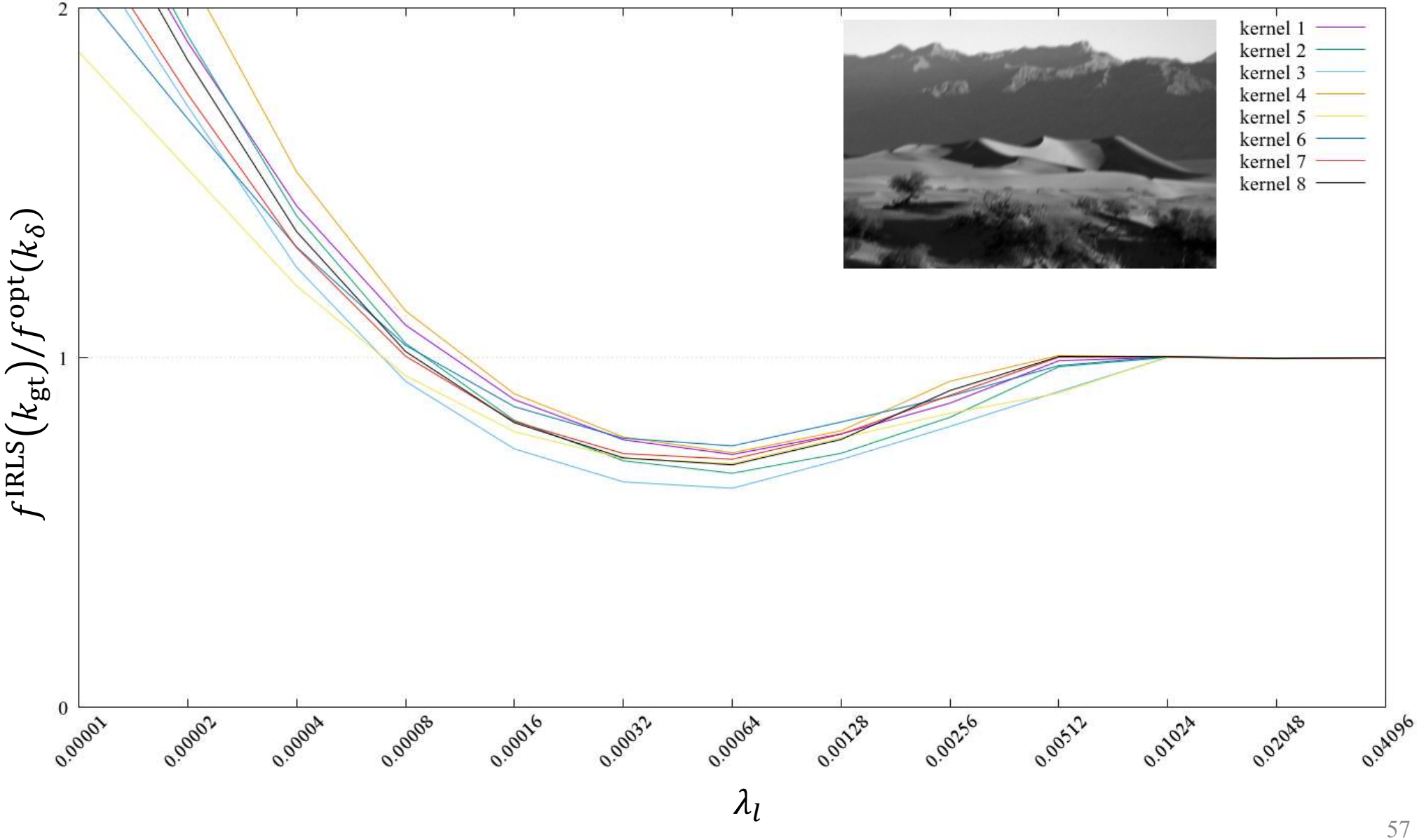
Sun et al.'s dataset - Image 25



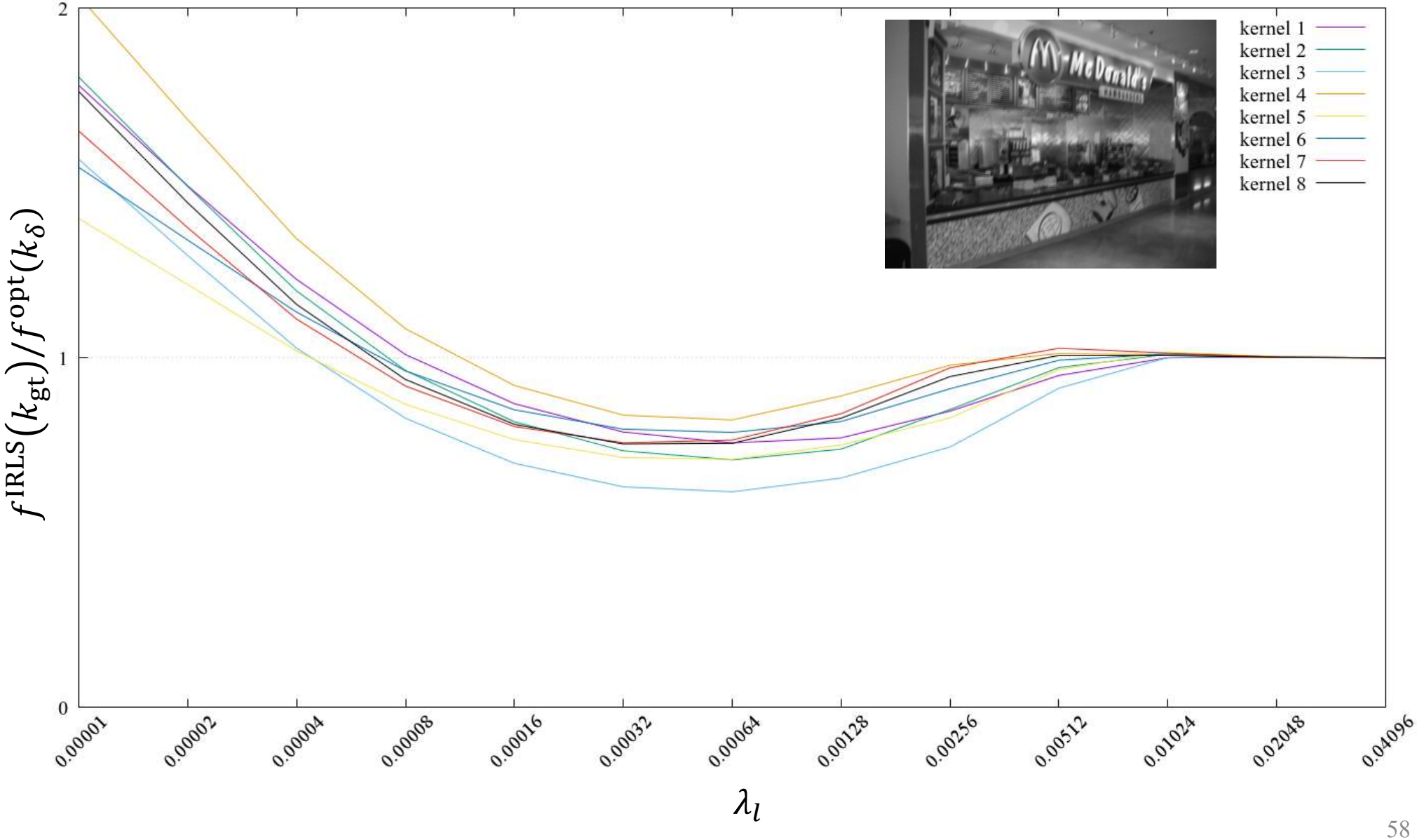
Sun et al.'s dataset - Image 26



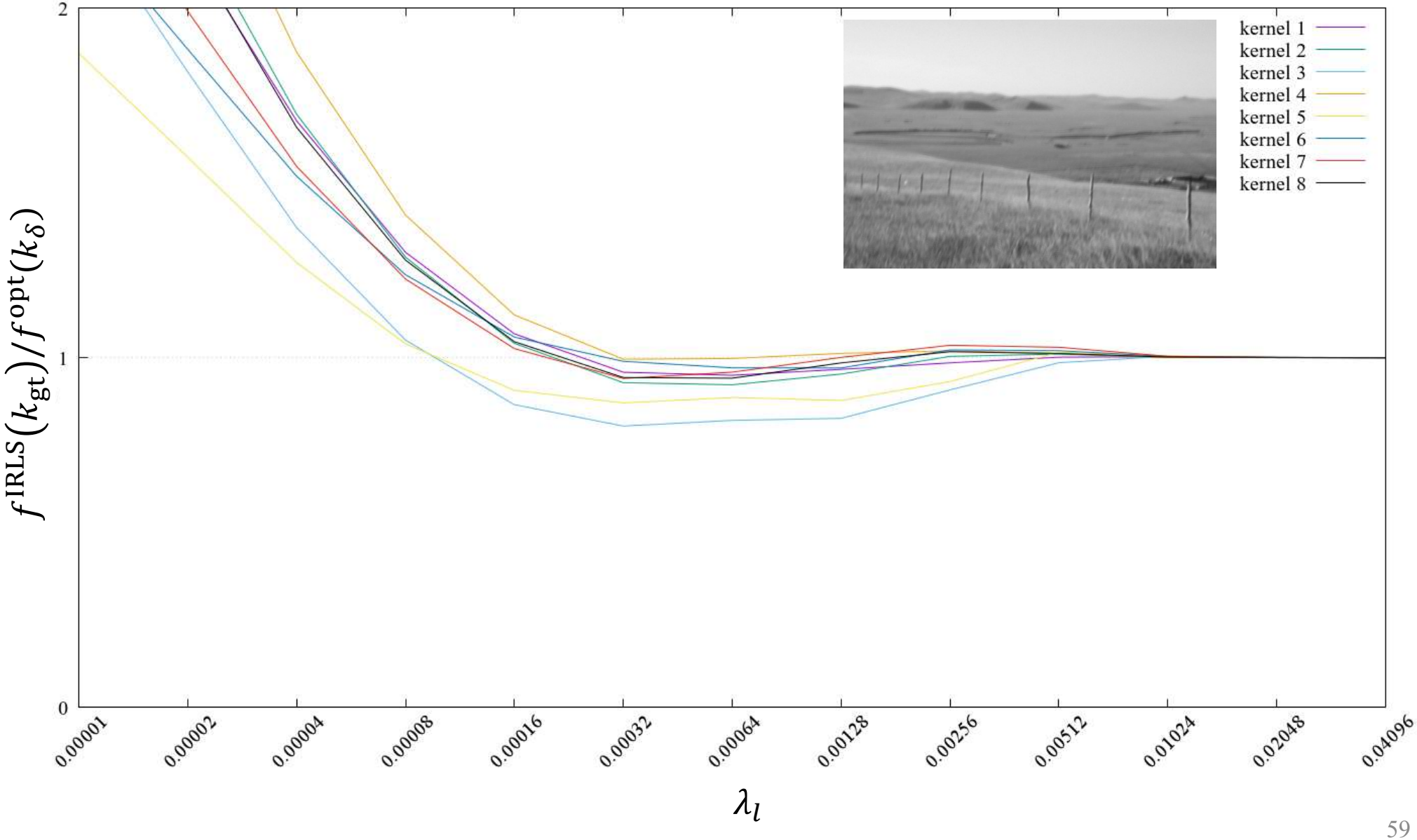
Sun et al.'s dataset - Image 27



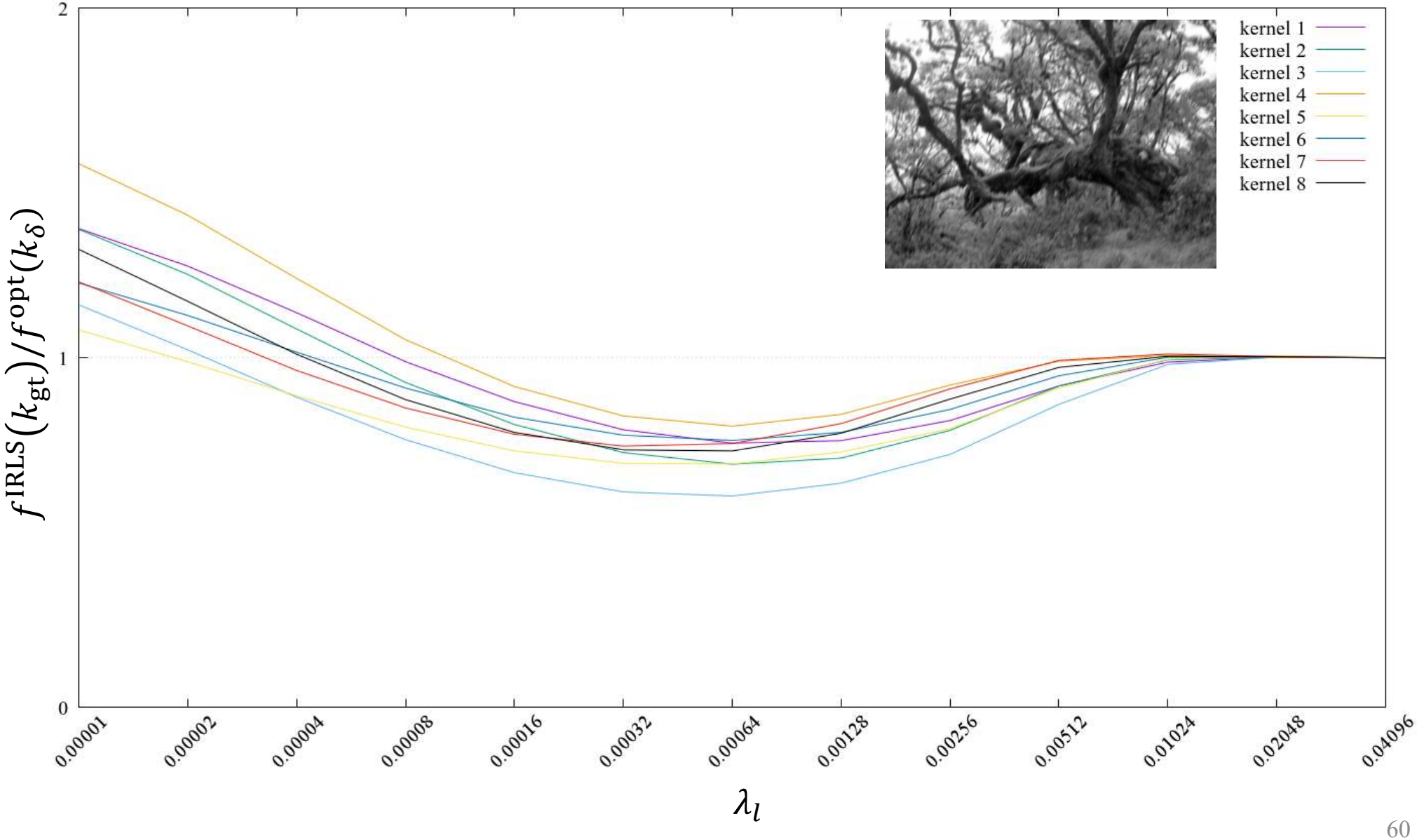
Sun et al.'s dataset - Image 28



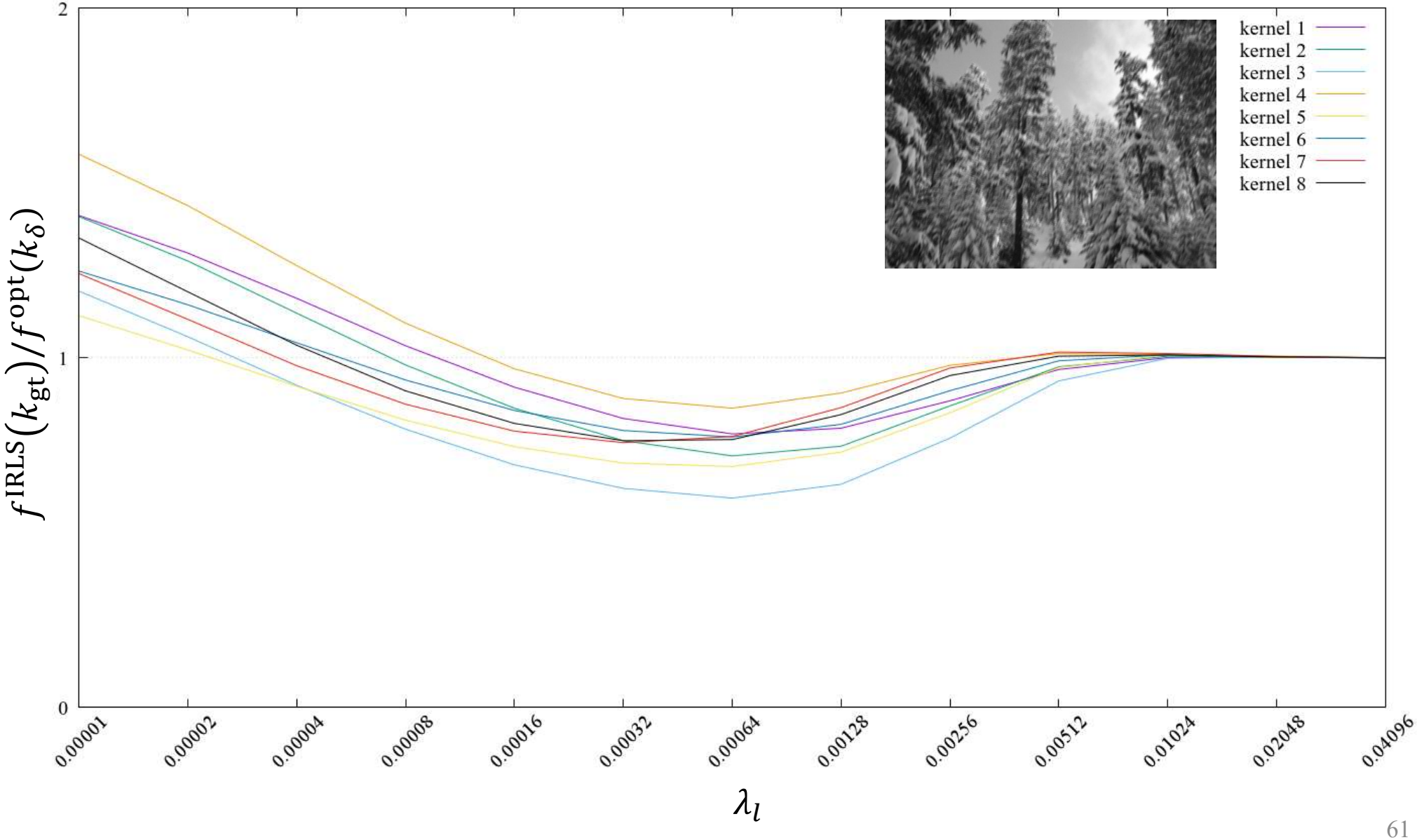
Sun et al.'s dataset - Image 29



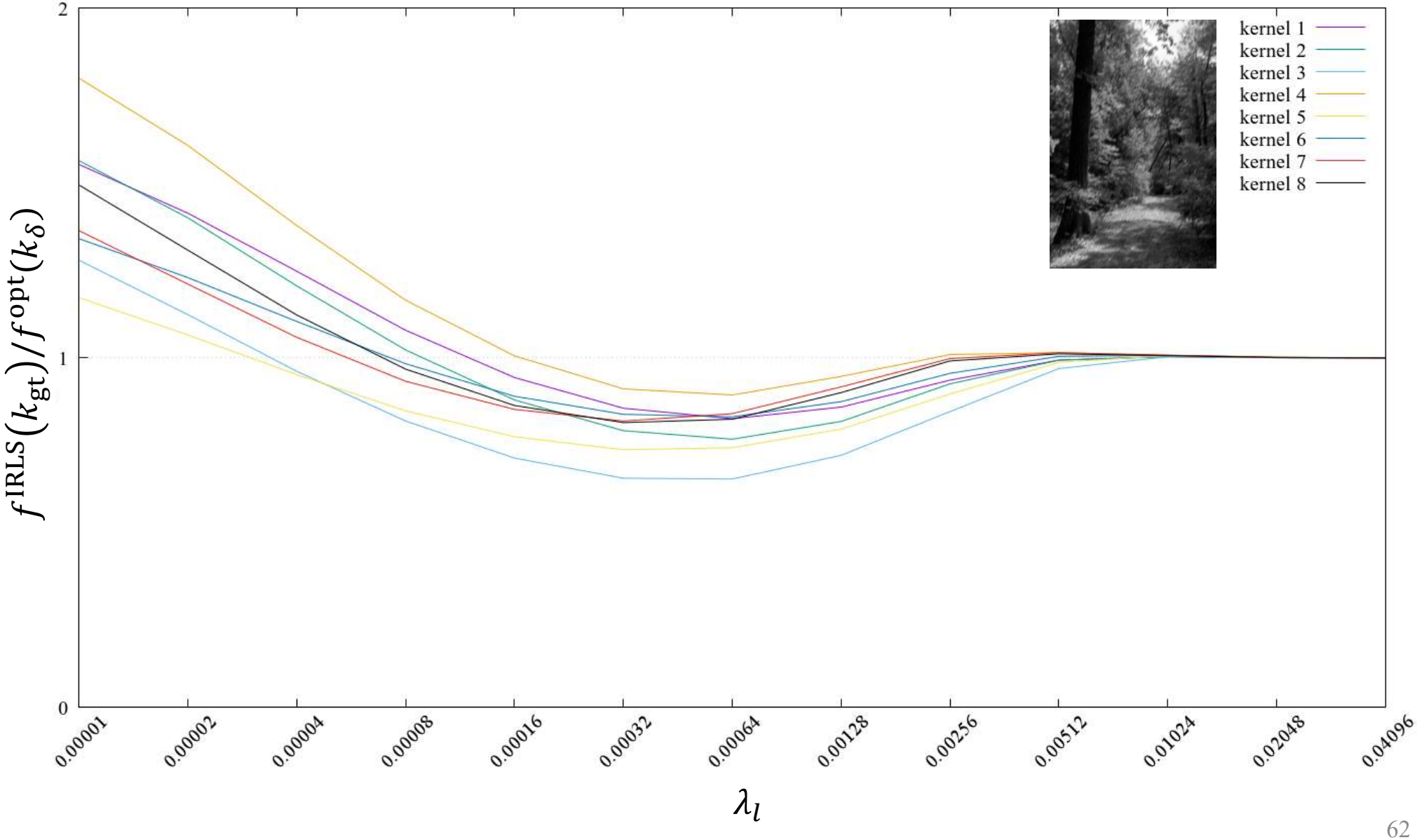
Sun et al.'s dataset - Image 30



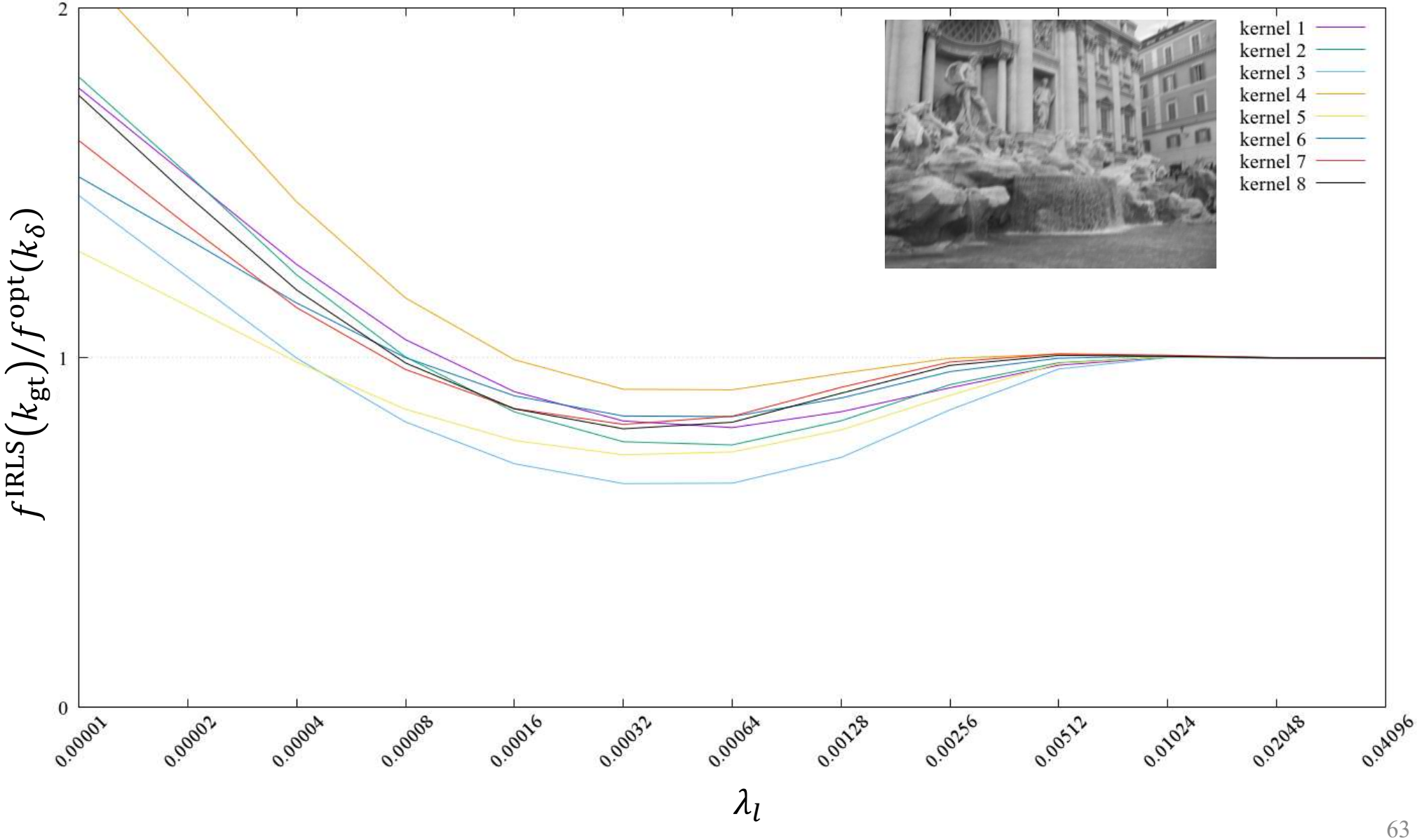
Sun et al.'s dataset - Image 31



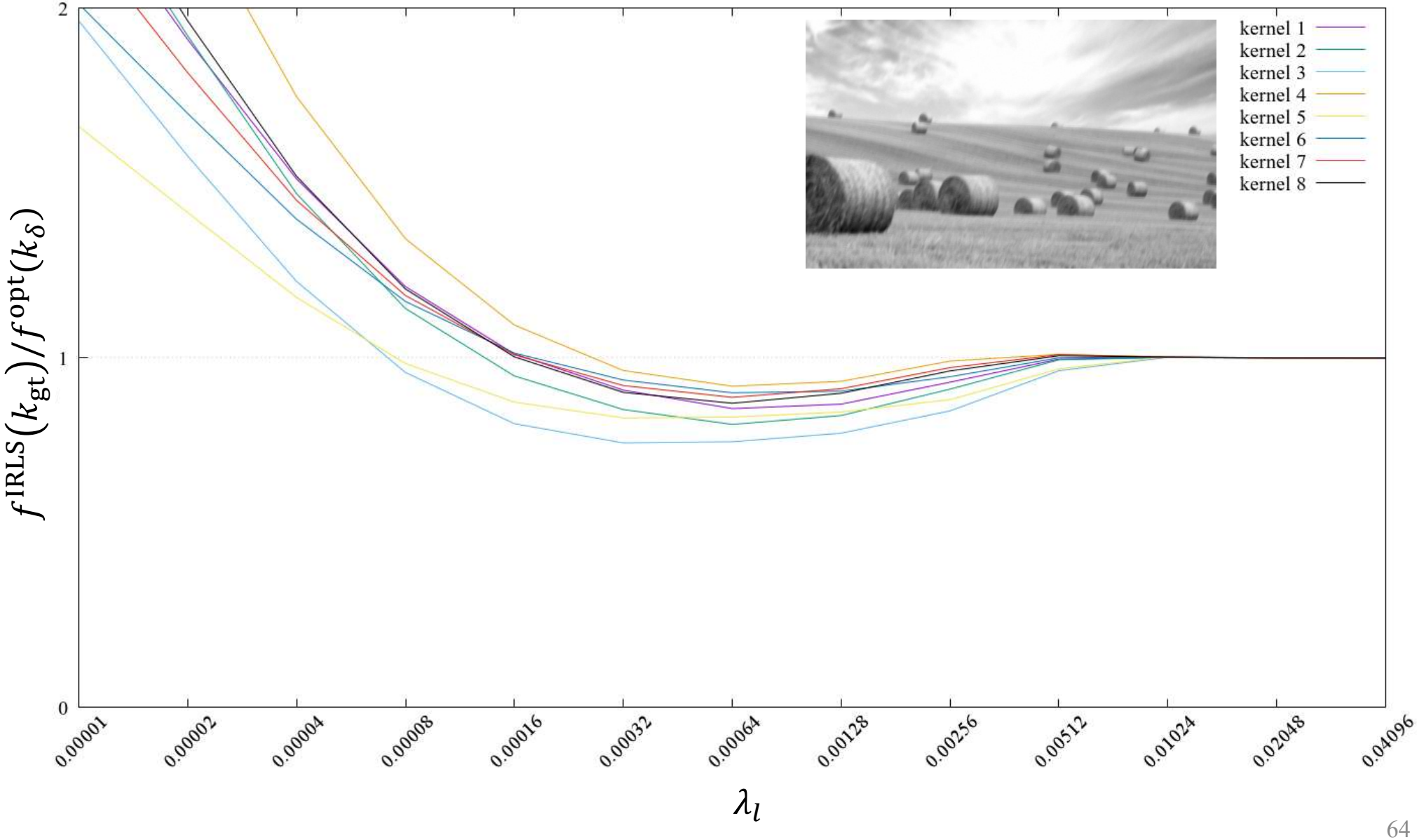
Sun et al.'s dataset - Image 32



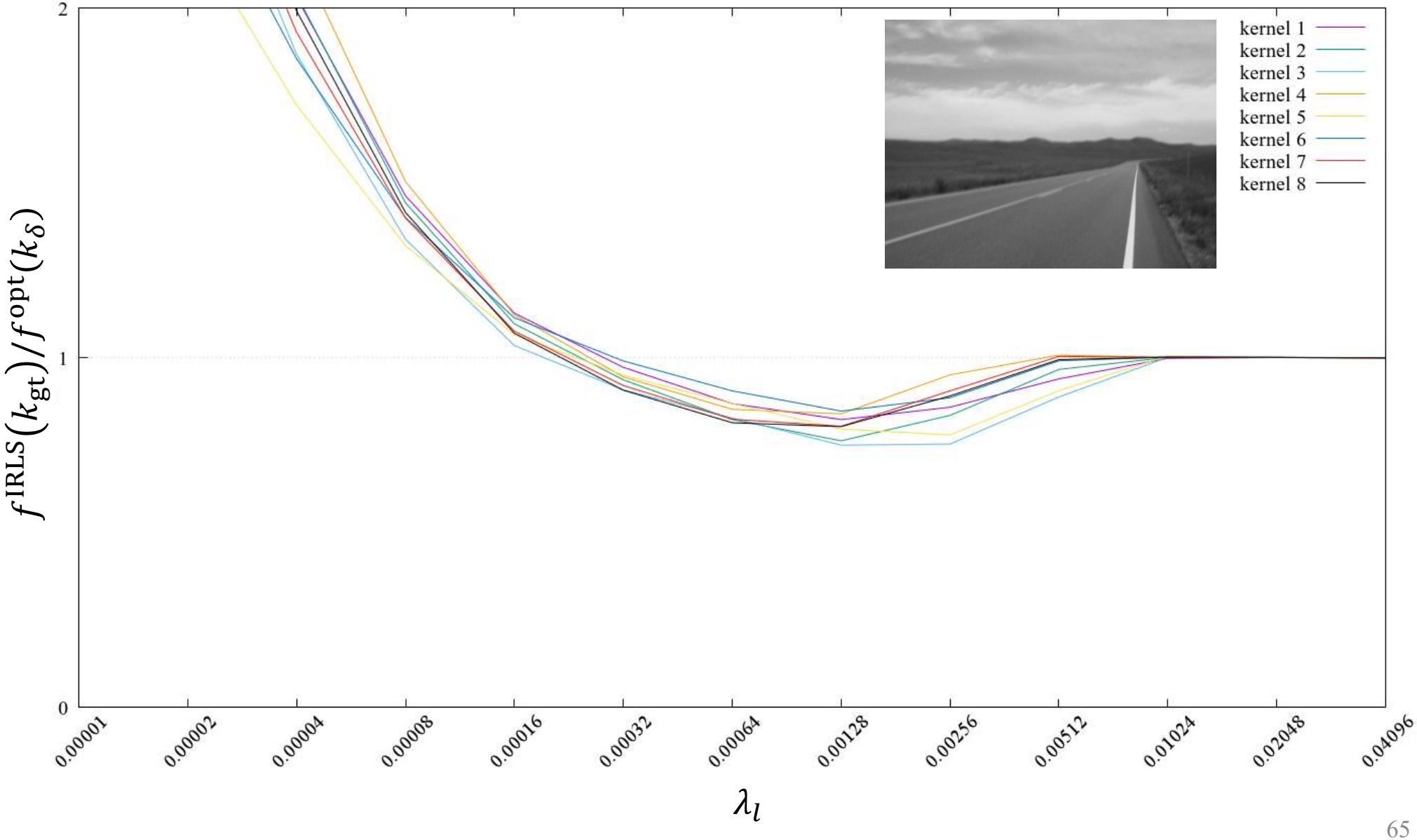
Sun et al.'s dataset - Image 33



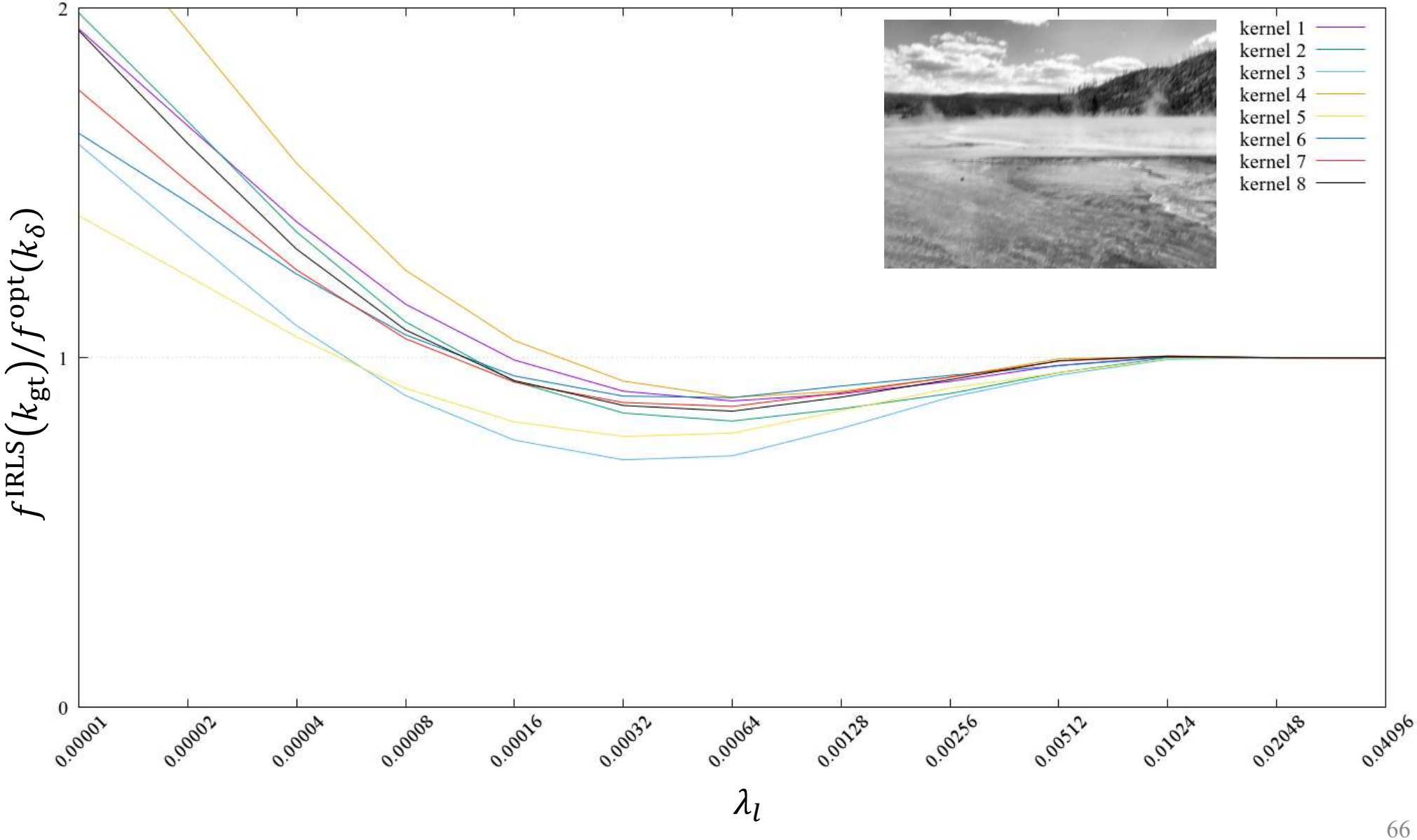
Sun et al.'s dataset - Image 34



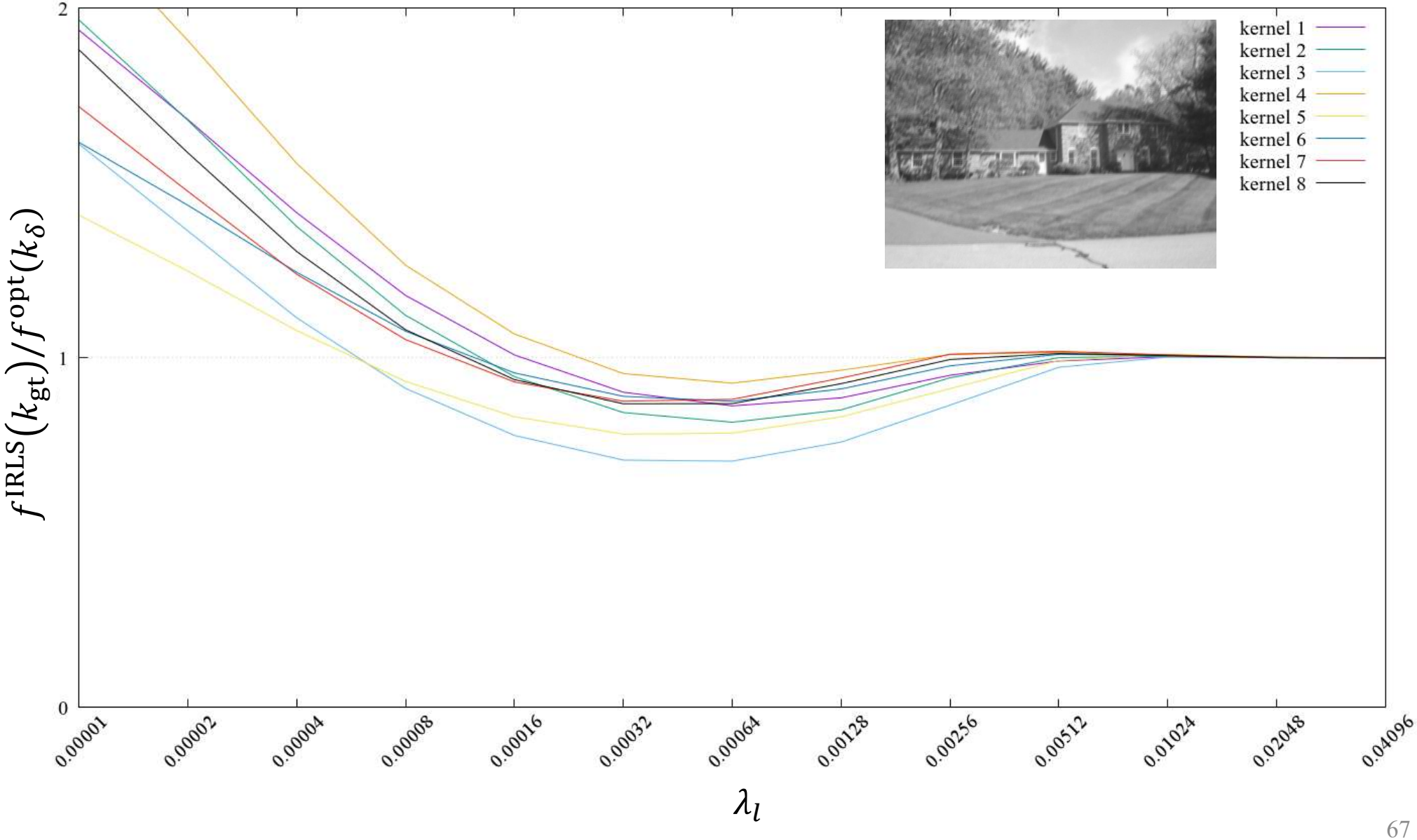
Sun et al.'s dataset - Image 35



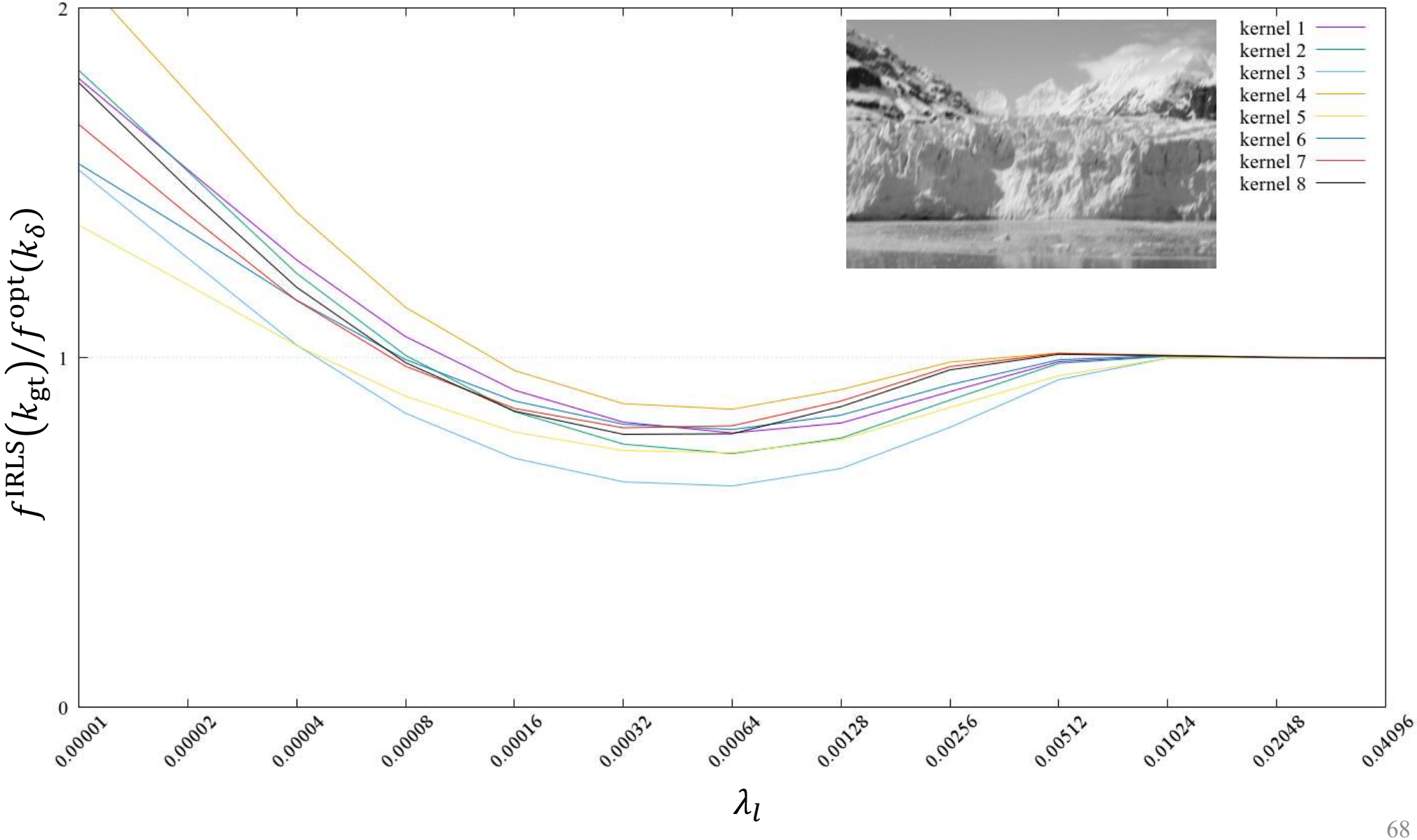
Sun et al.'s dataset - Image 36



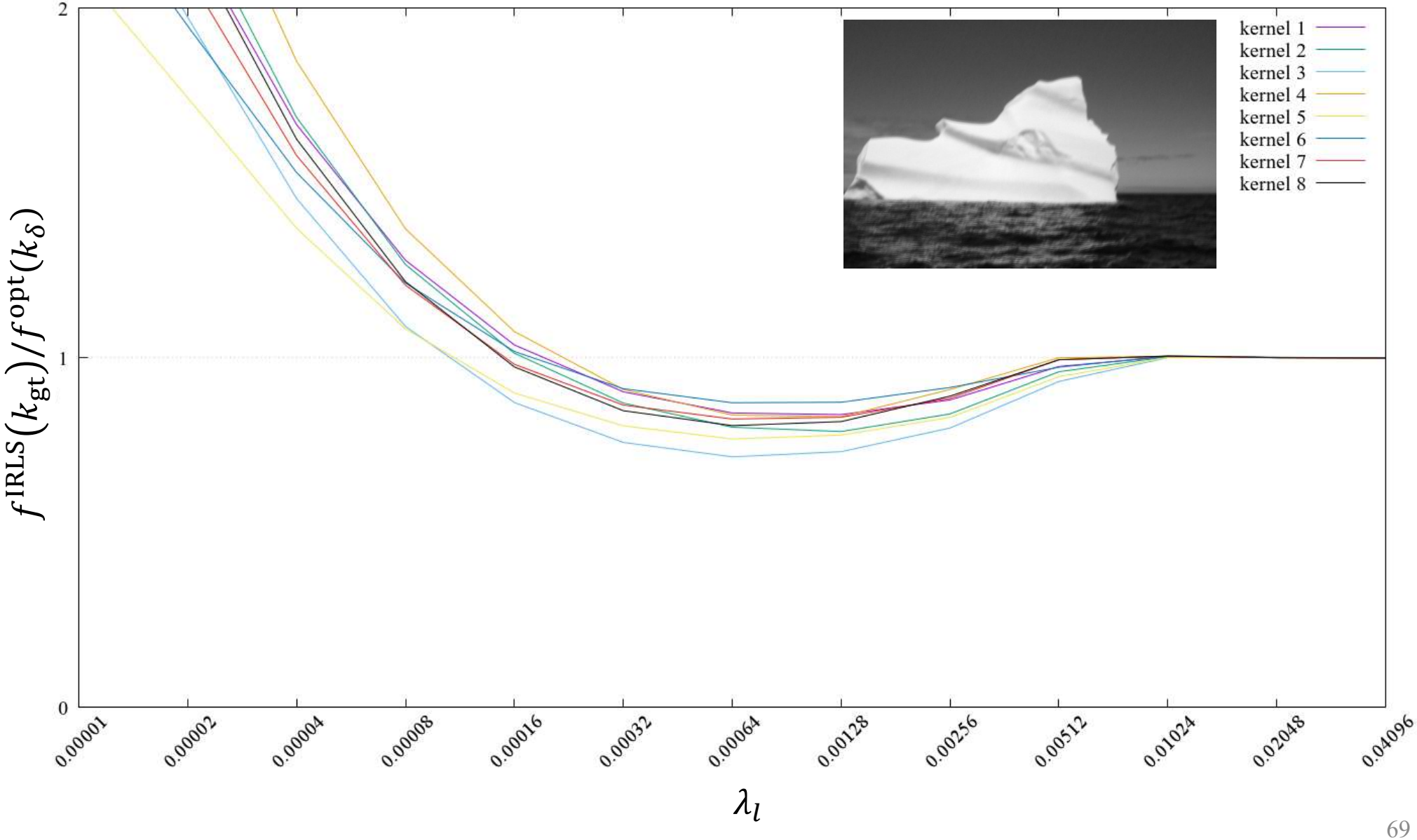
Sun et al.'s dataset - Image 37



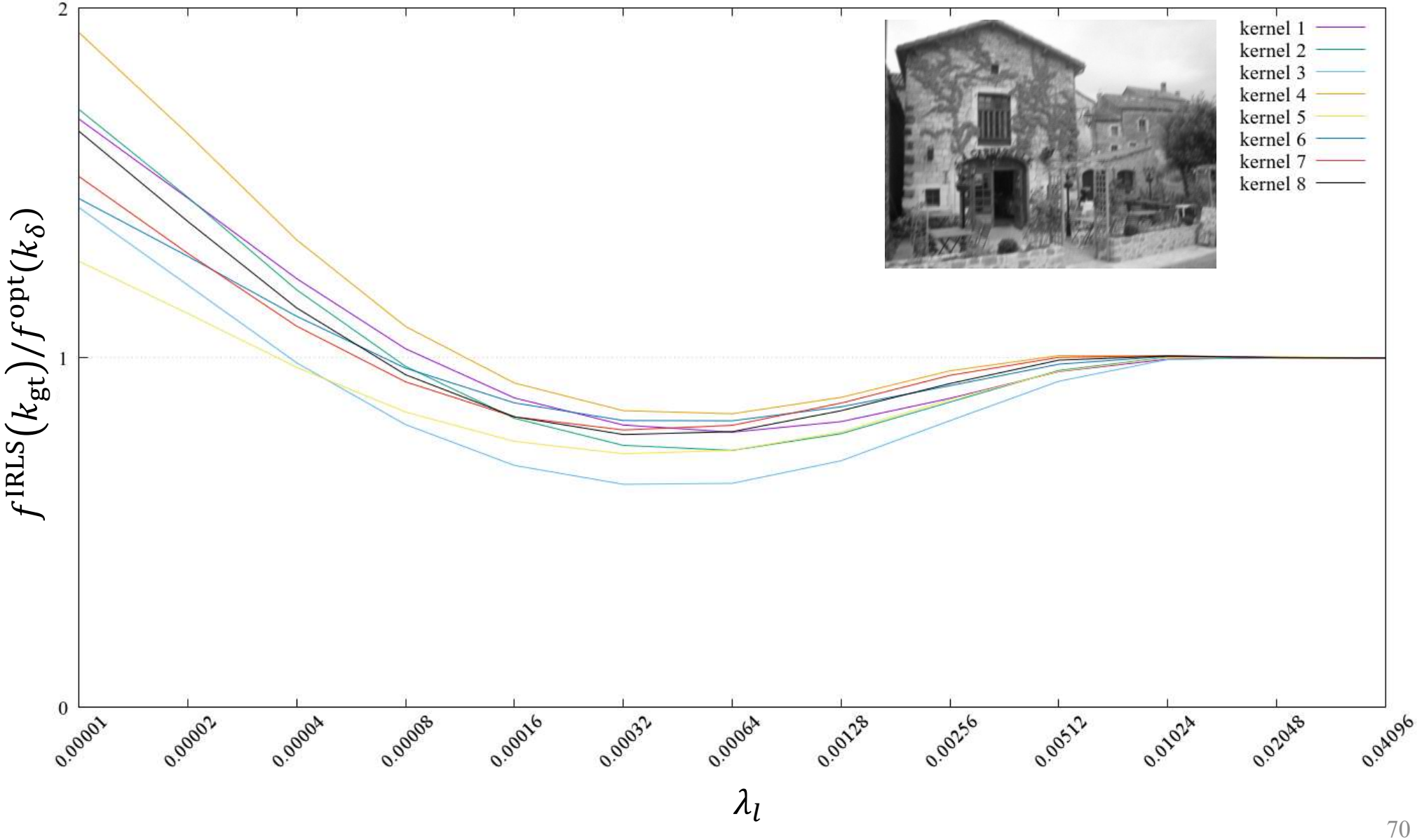
Sun et al.'s dataset - Image 38



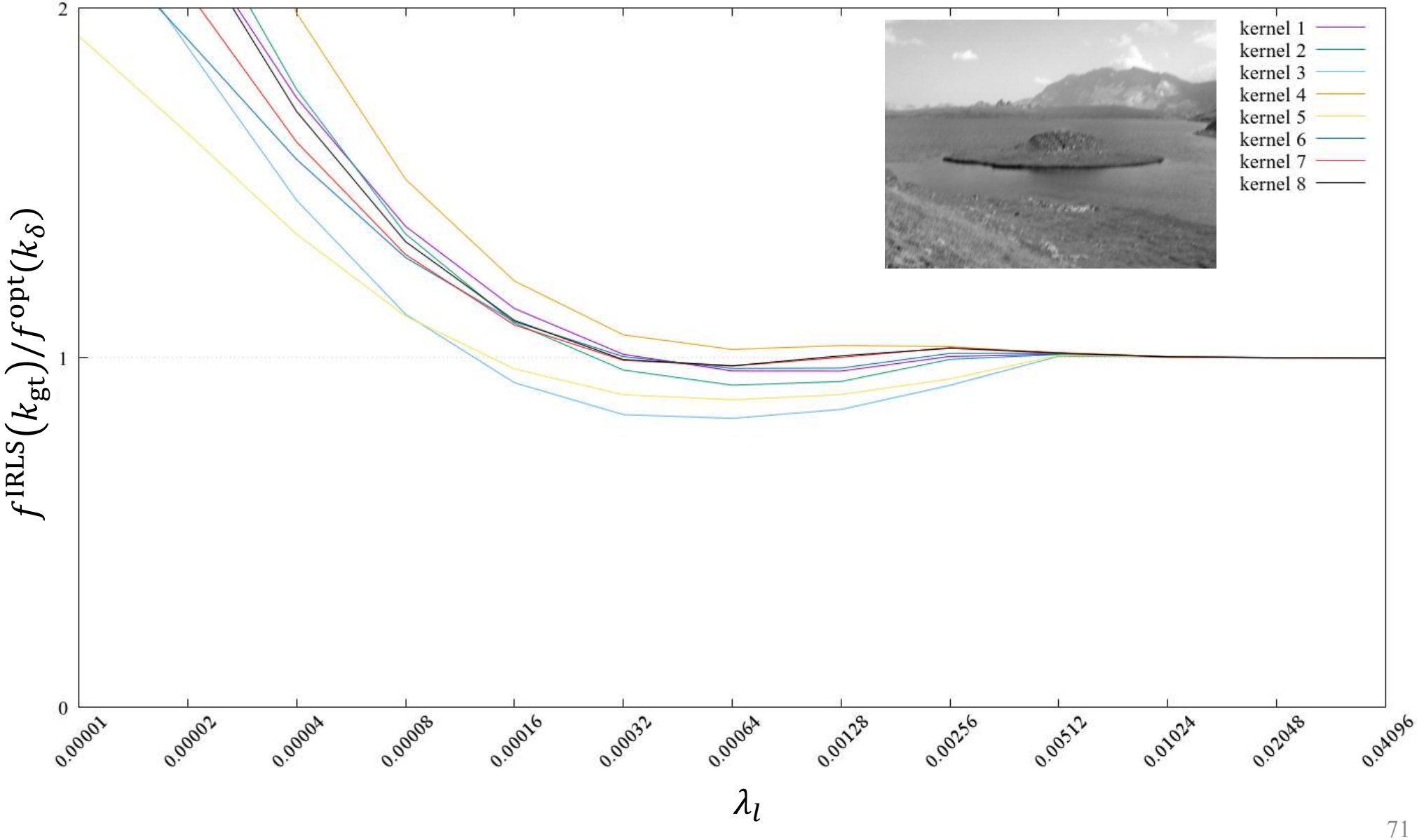
Sun et al.'s dataset - Image 39



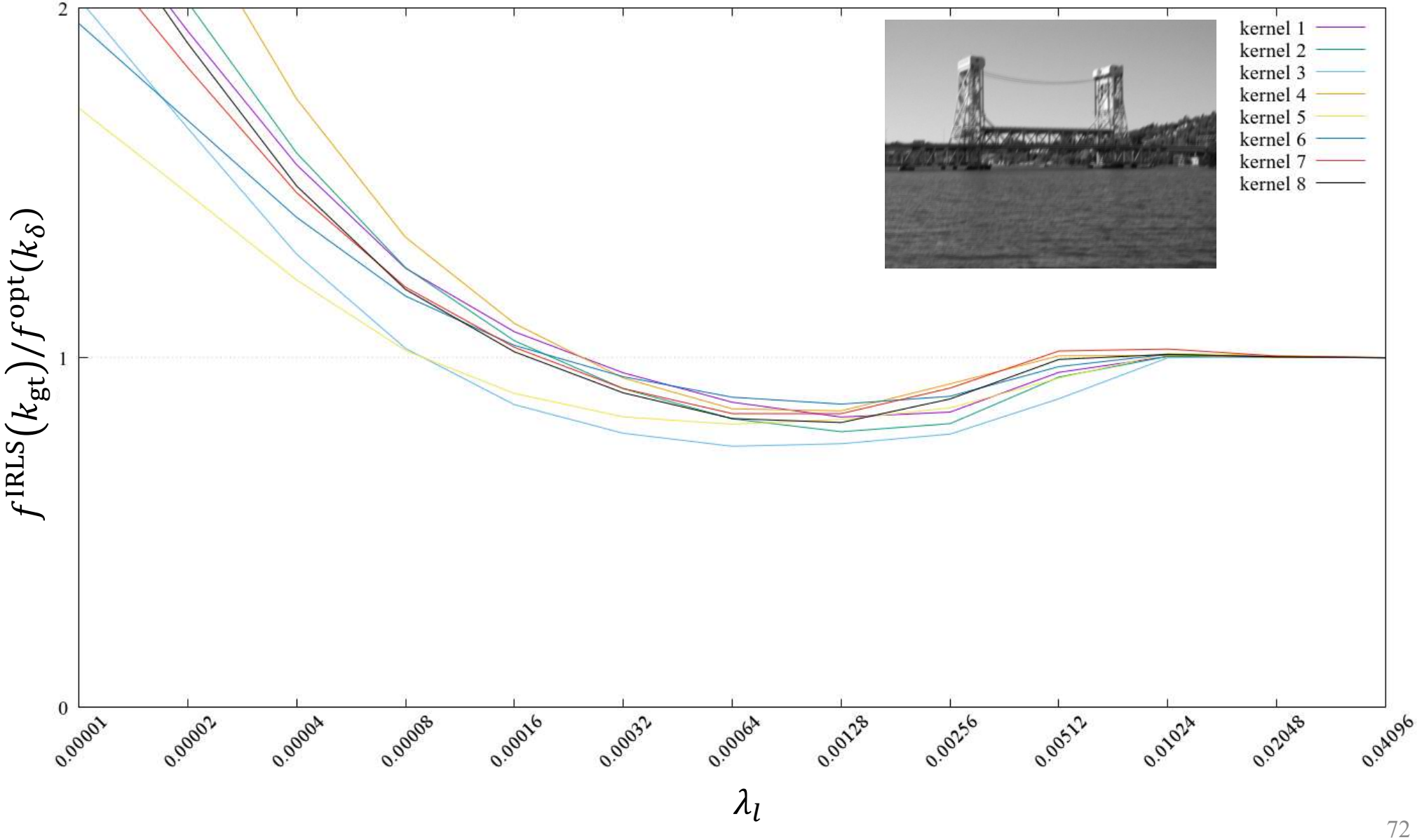
Sun et al.'s dataset - Image 40



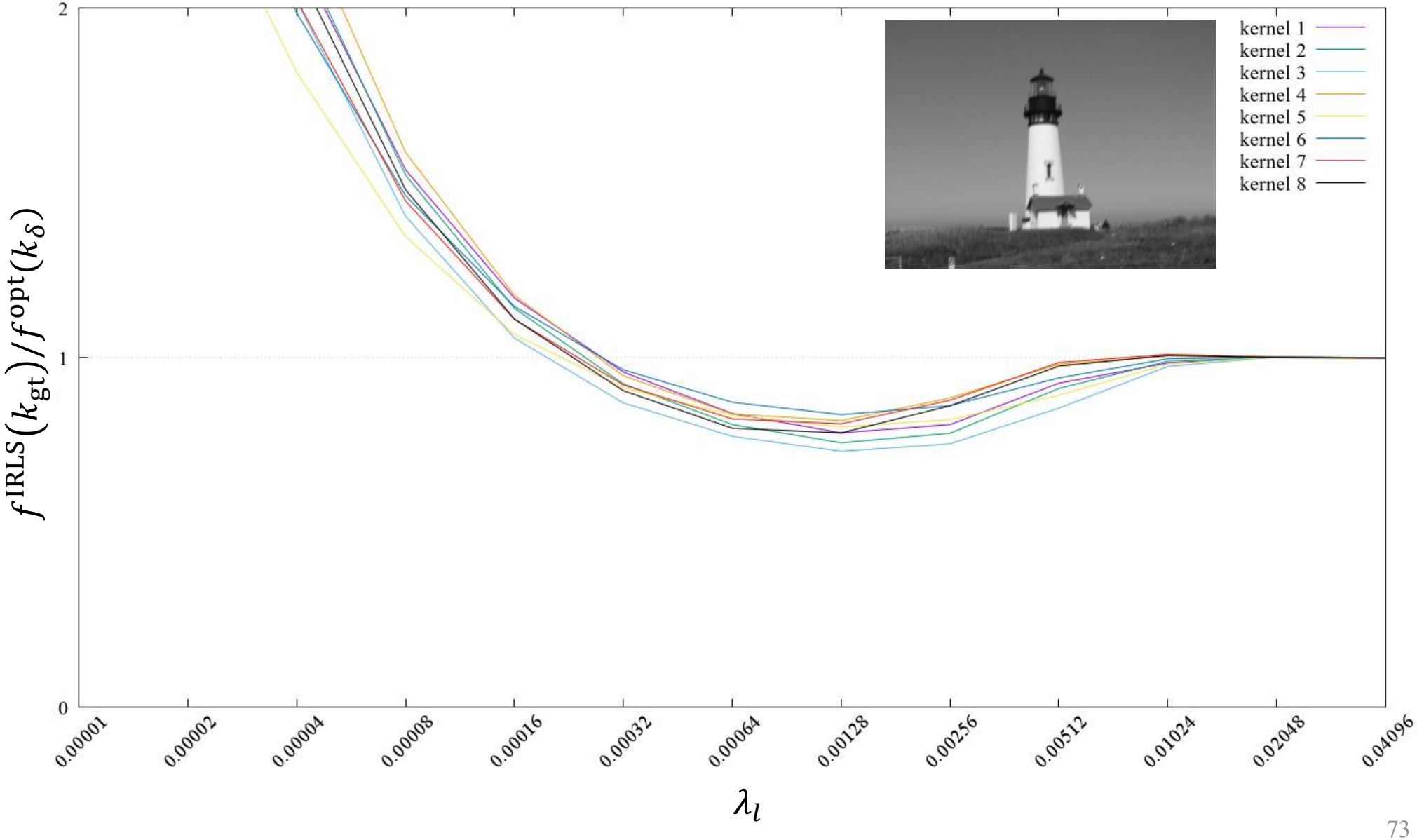
Sun et al.'s dataset - Image 41



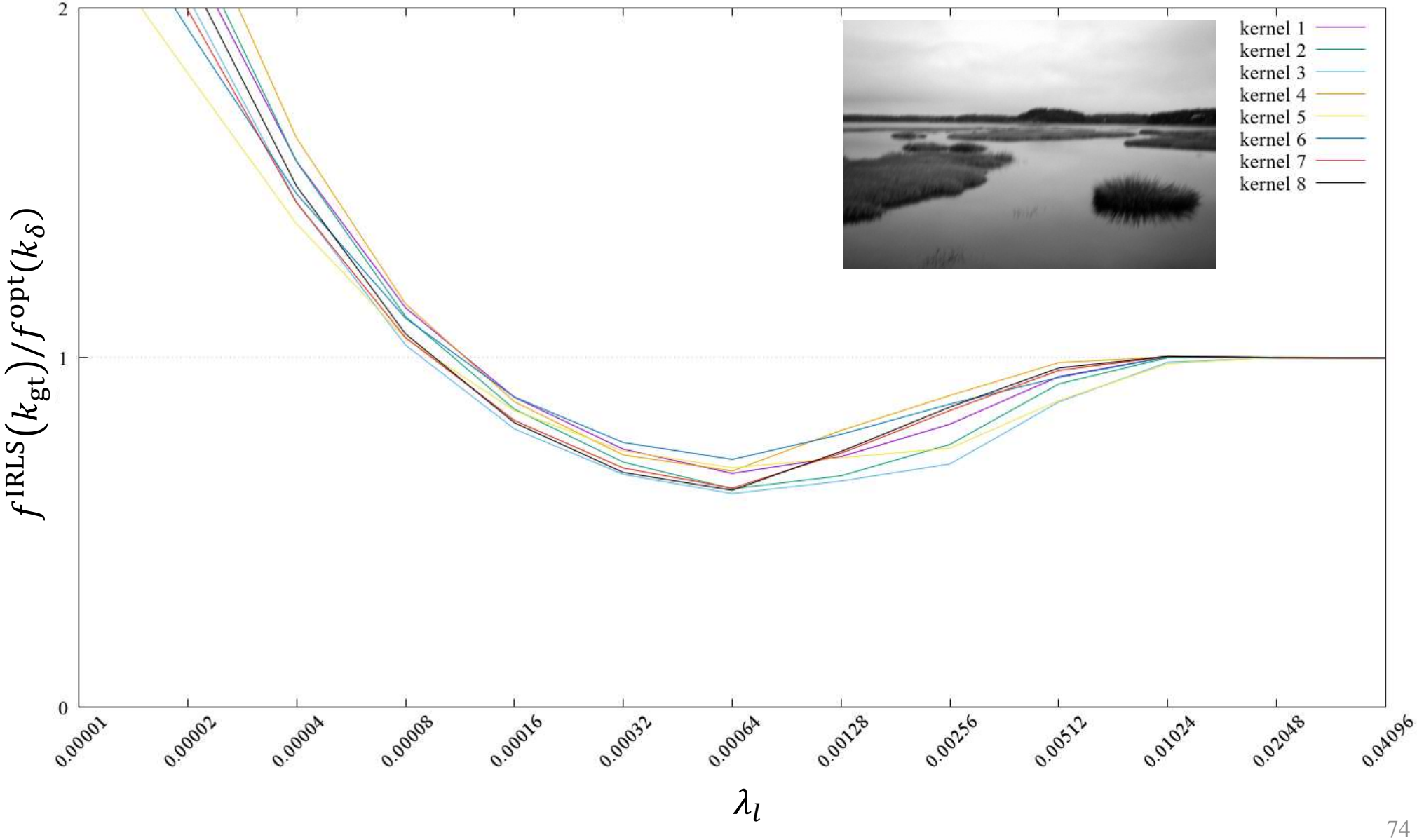
Sun et al.'s dataset - Image 42



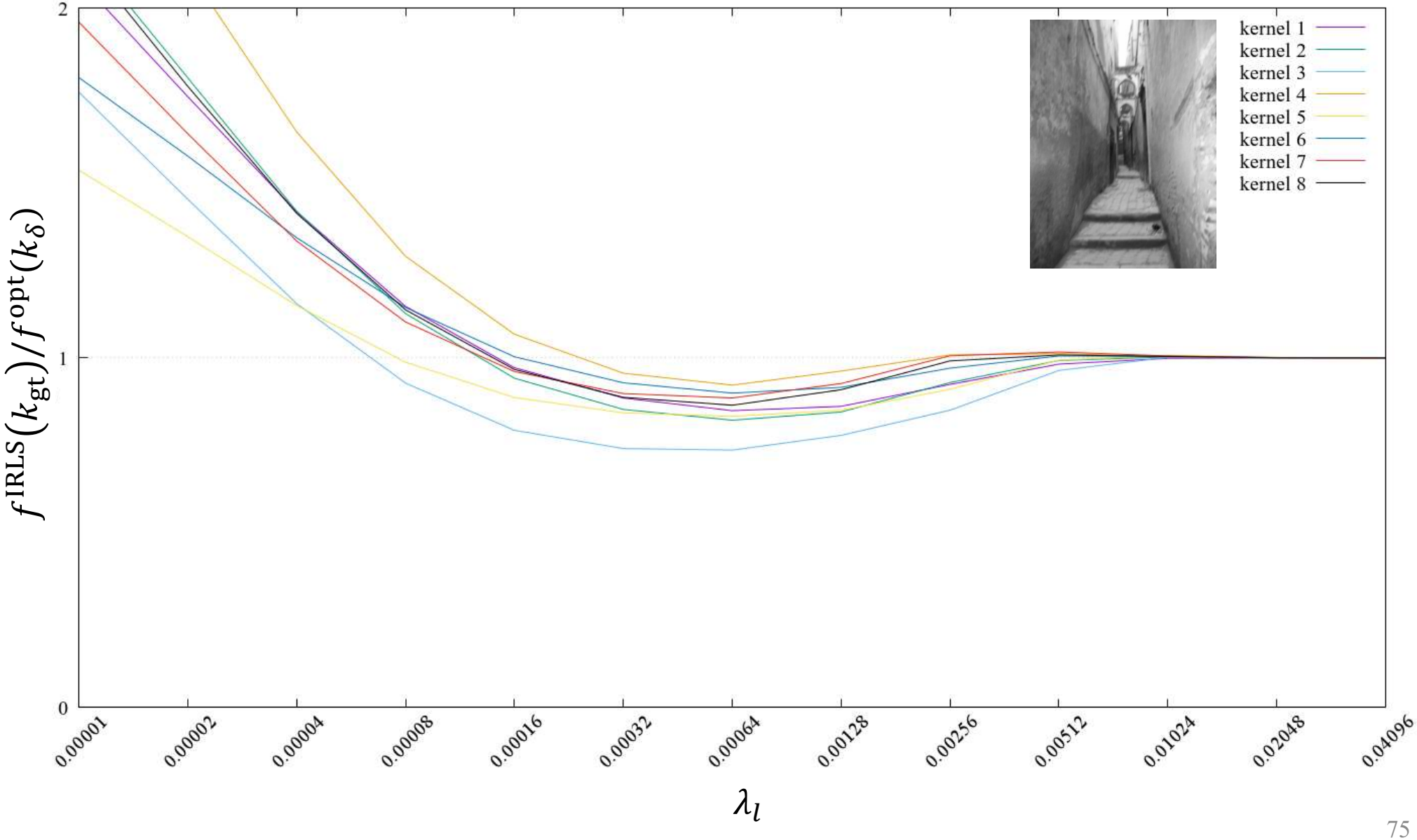
Sun et al.'s dataset - Image 43



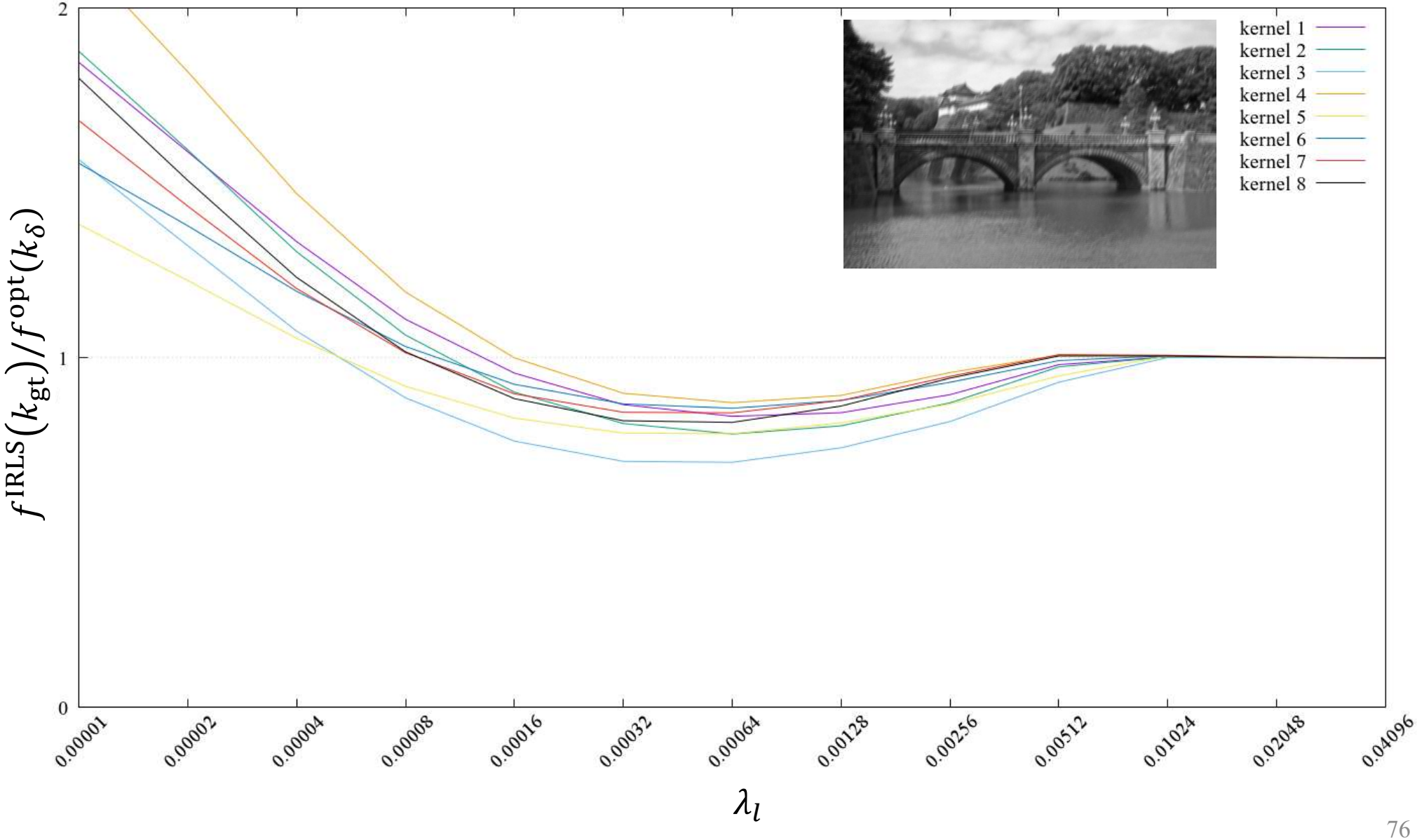
Sun et al.'s dataset - Image 44



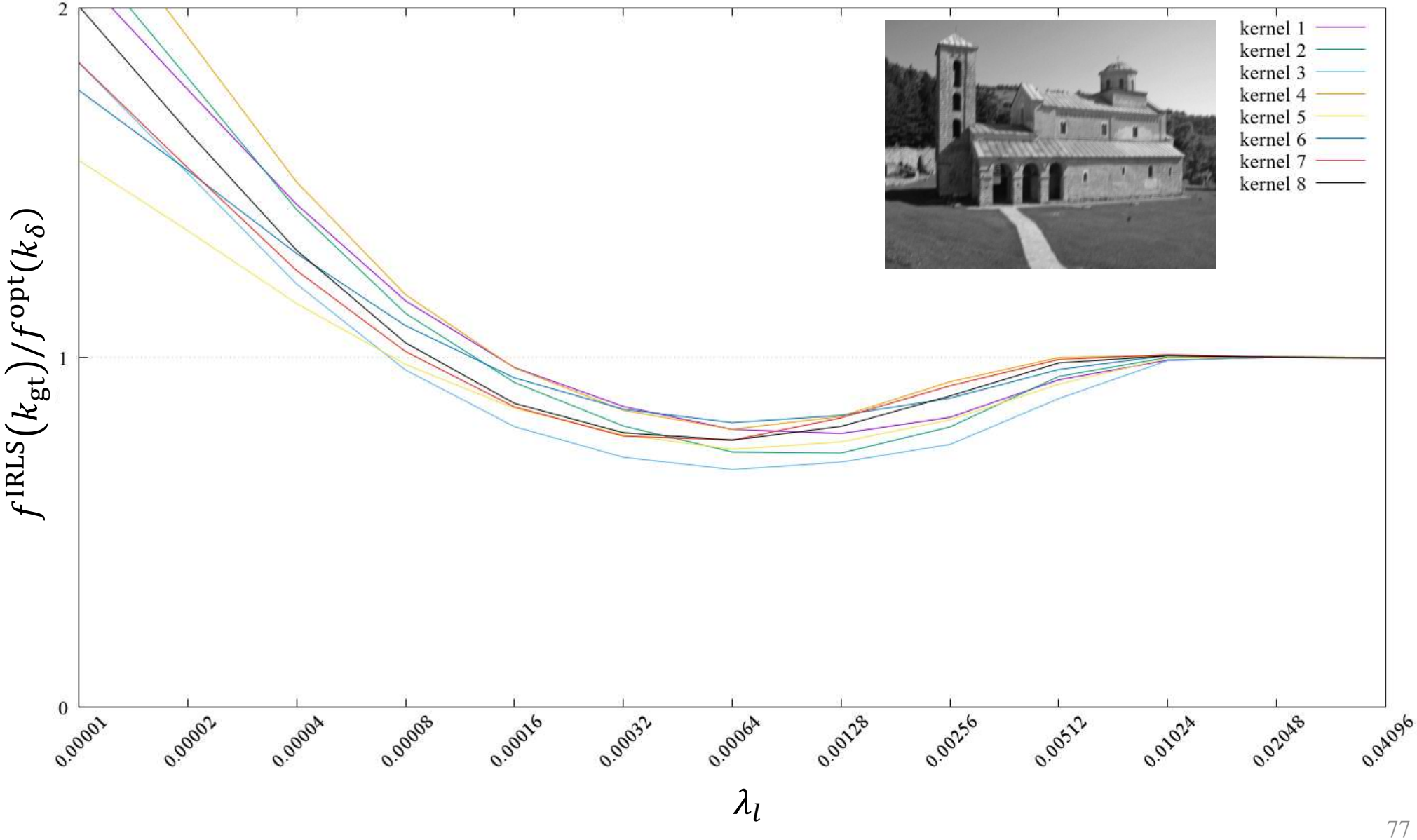
Sun et al.'s dataset - Image 45



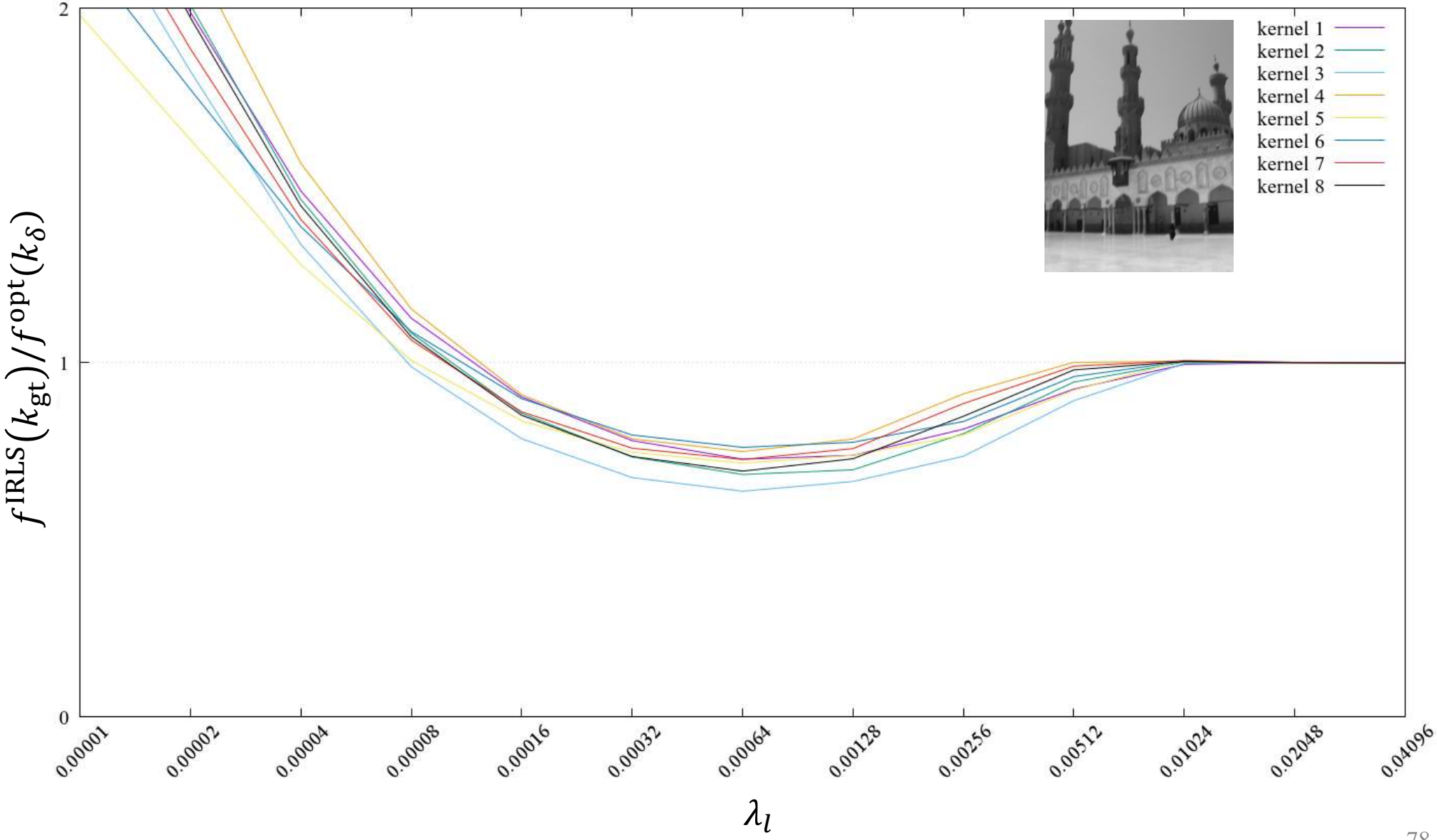
Sun et al.'s dataset - Image 46



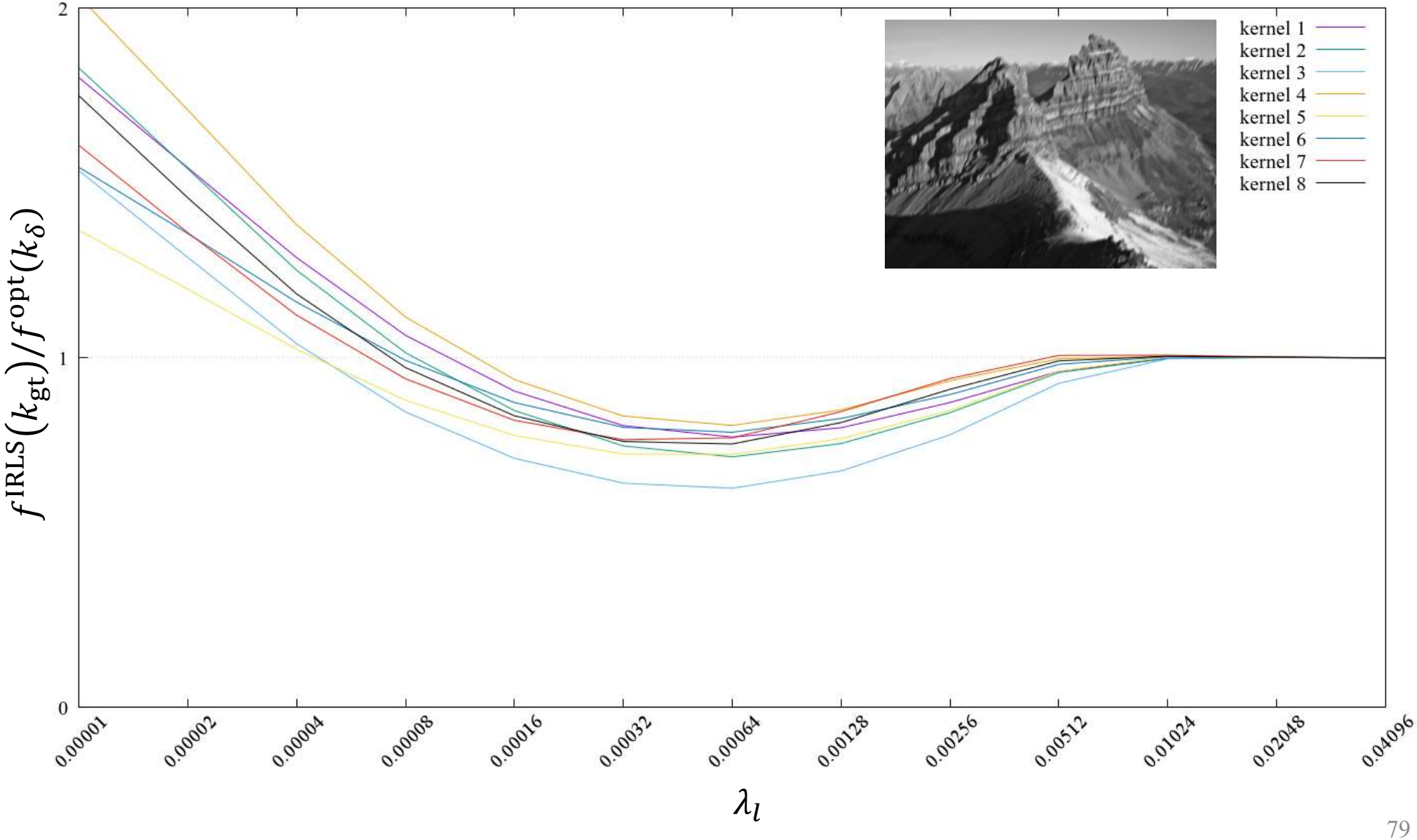
Sun et al.'s dataset - Image 47



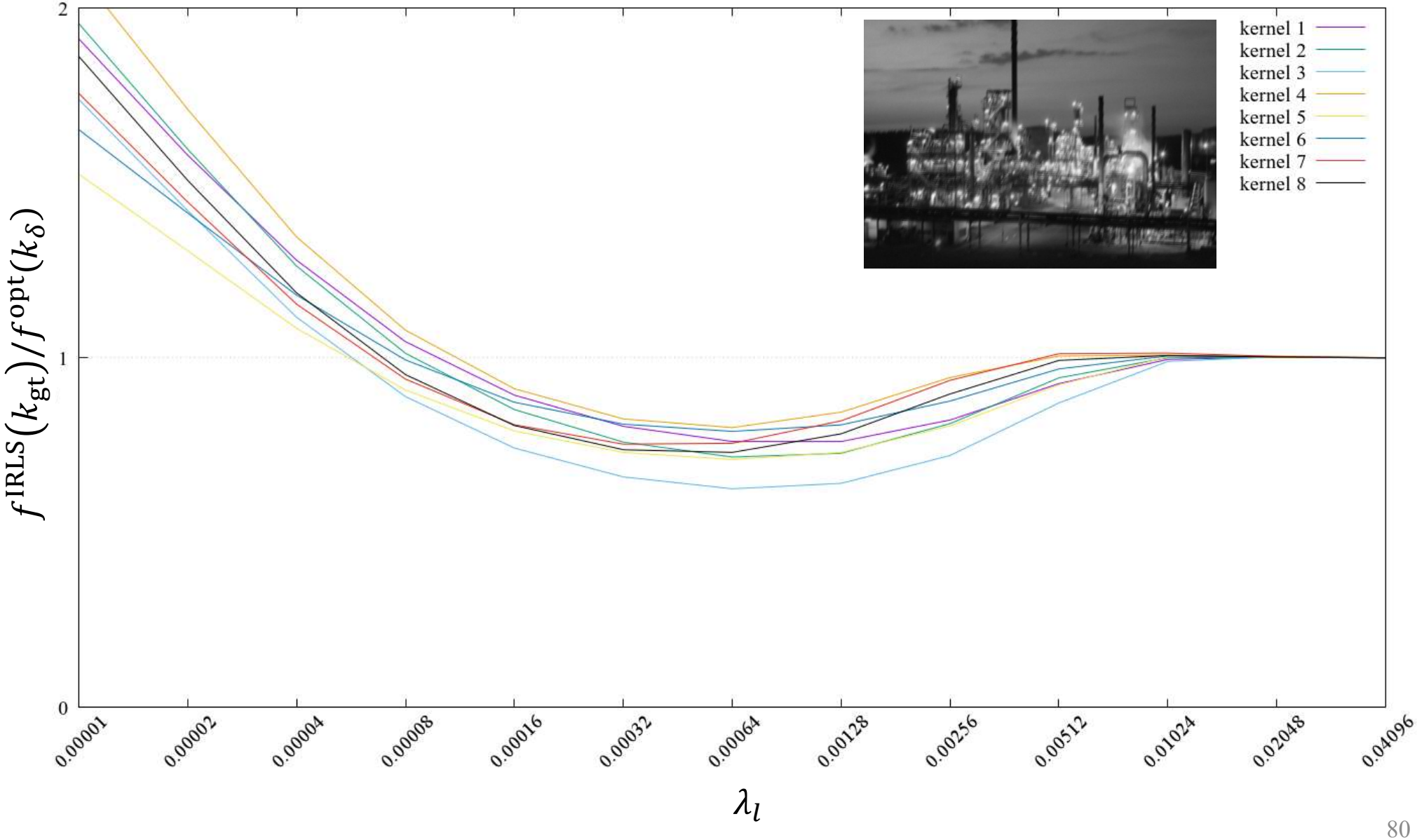
Sun et al.'s dataset - Image 48



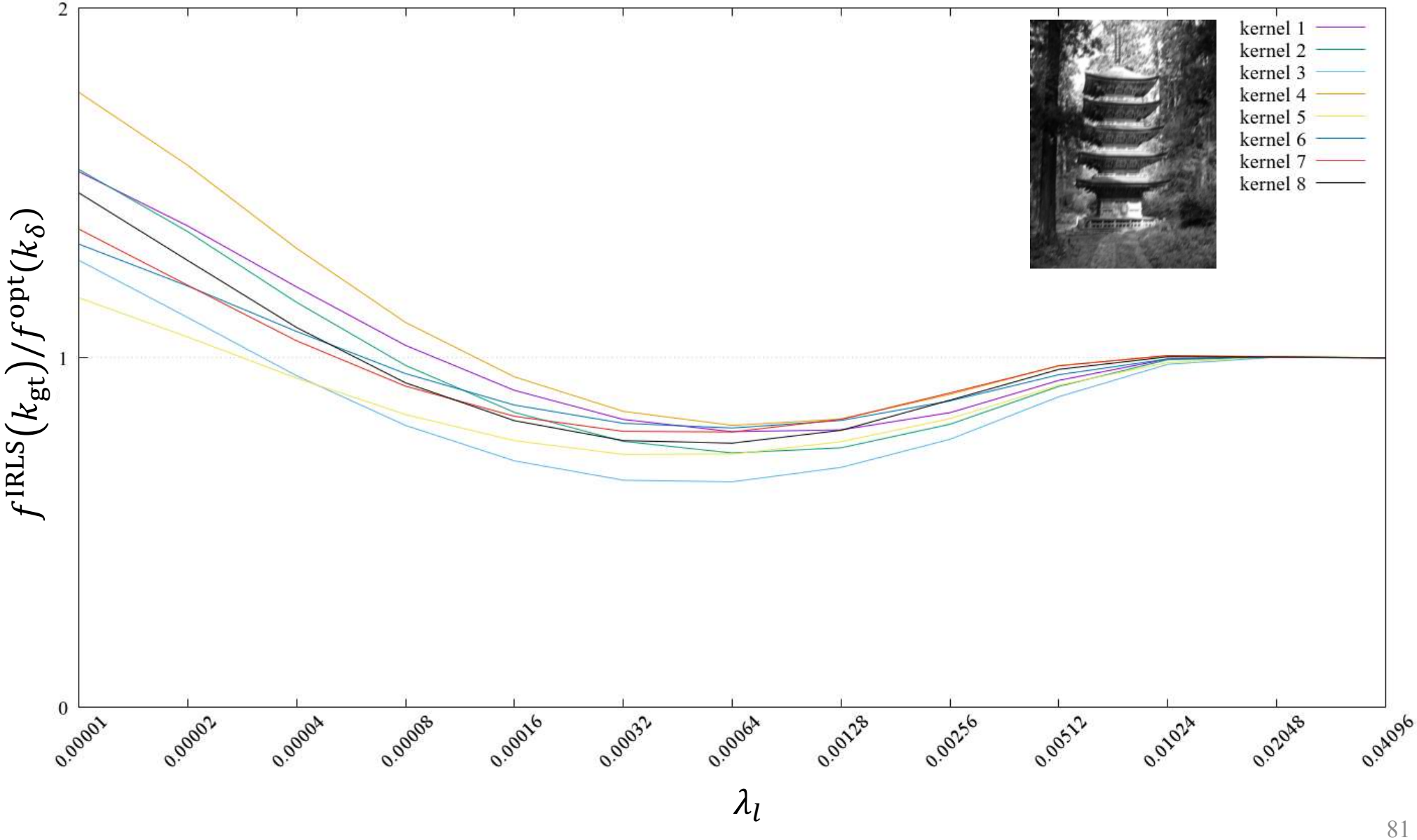
Sun et al.'s dataset - Image 49



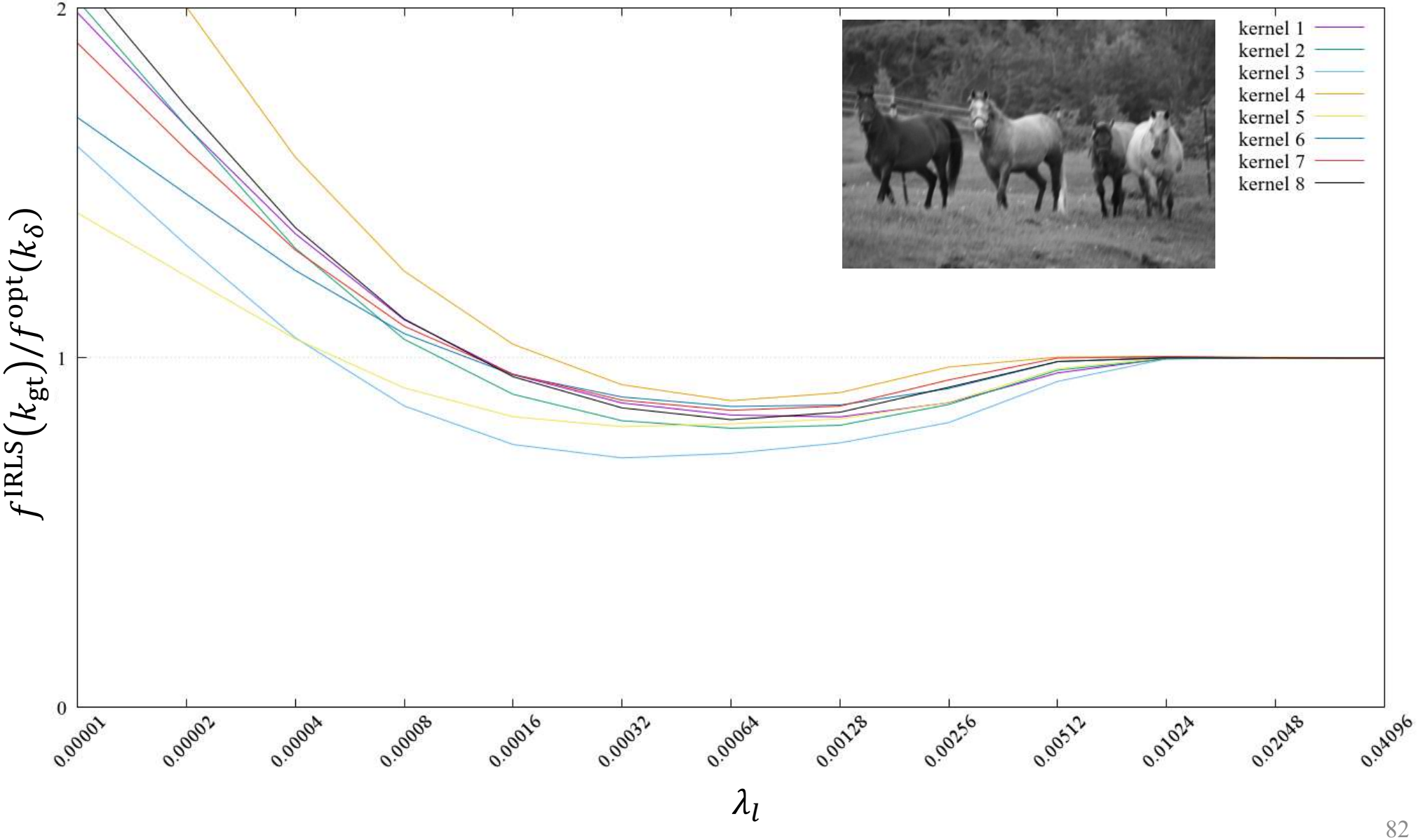
Sun et al.'s dataset - Image 50



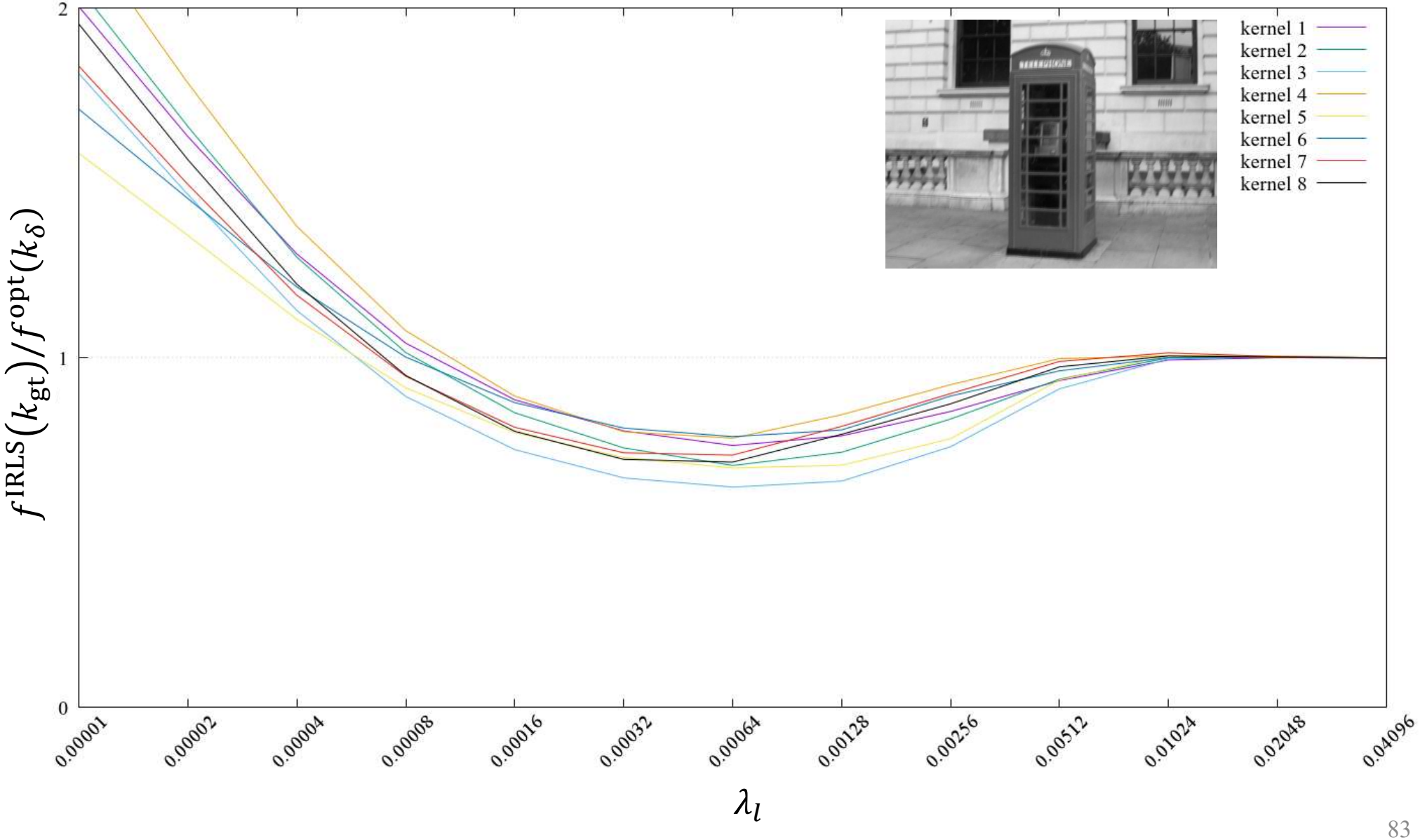
Sun et al.'s dataset - Image 51



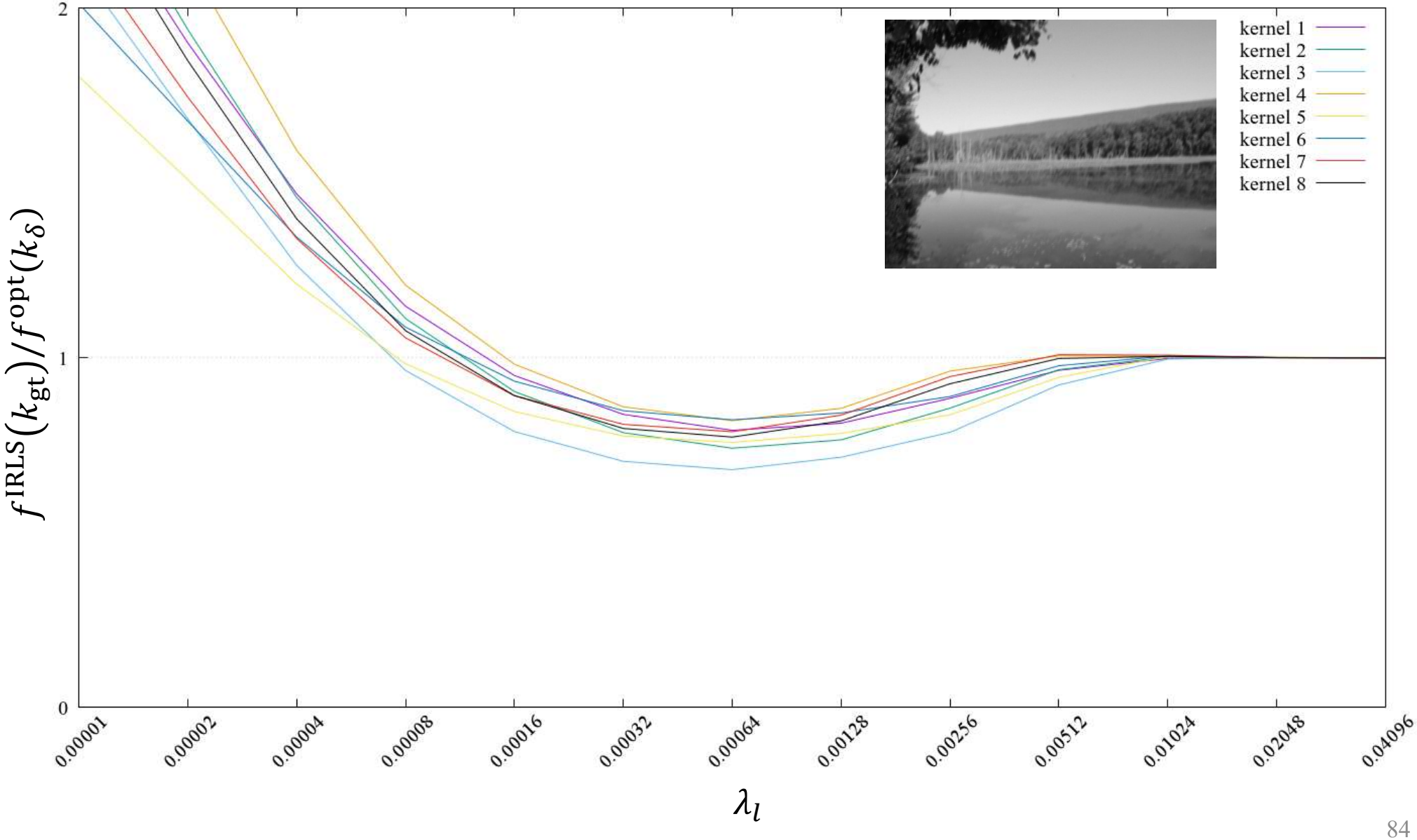
Sun et al.'s dataset - Image 52



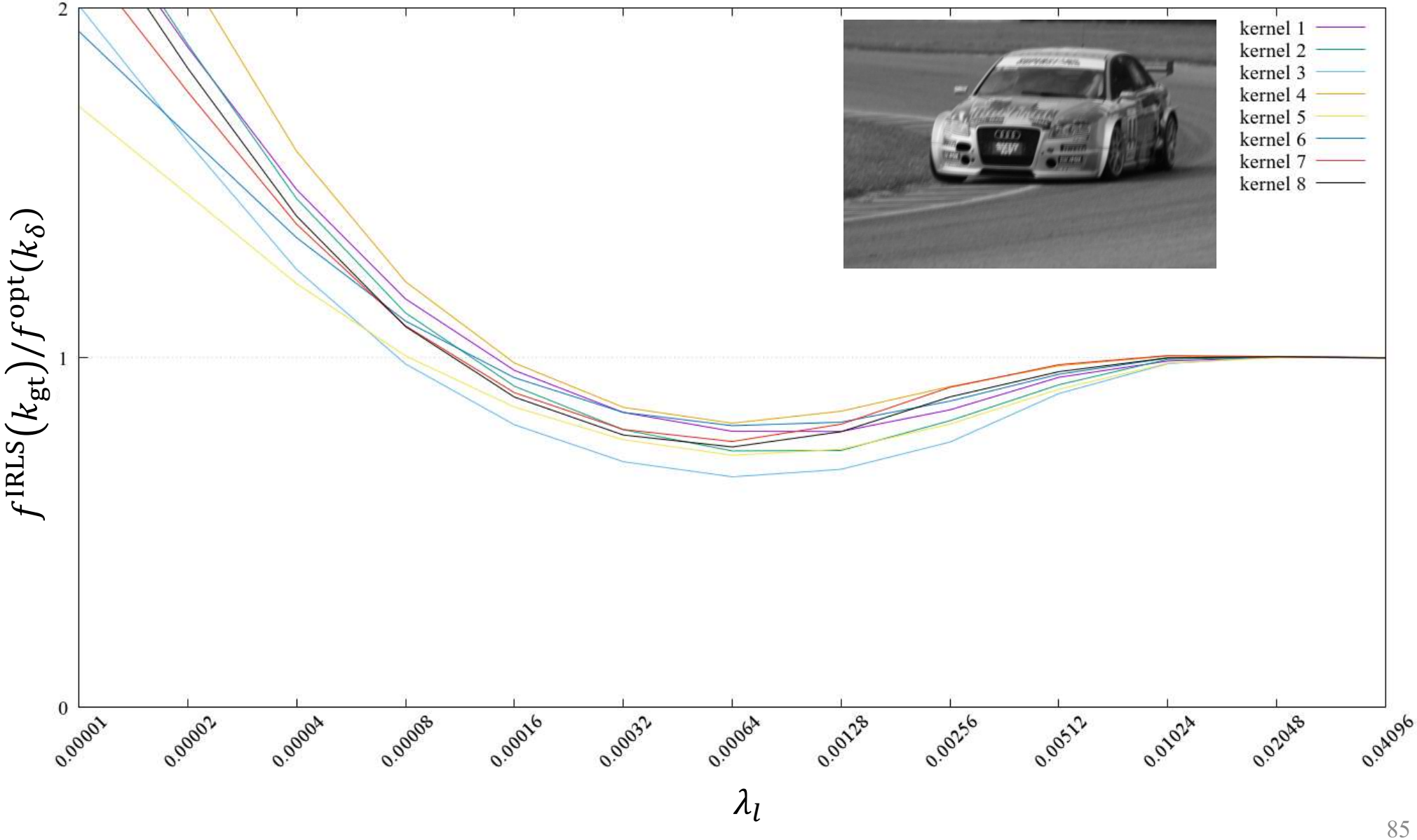
Sun et al.'s dataset - Image 53



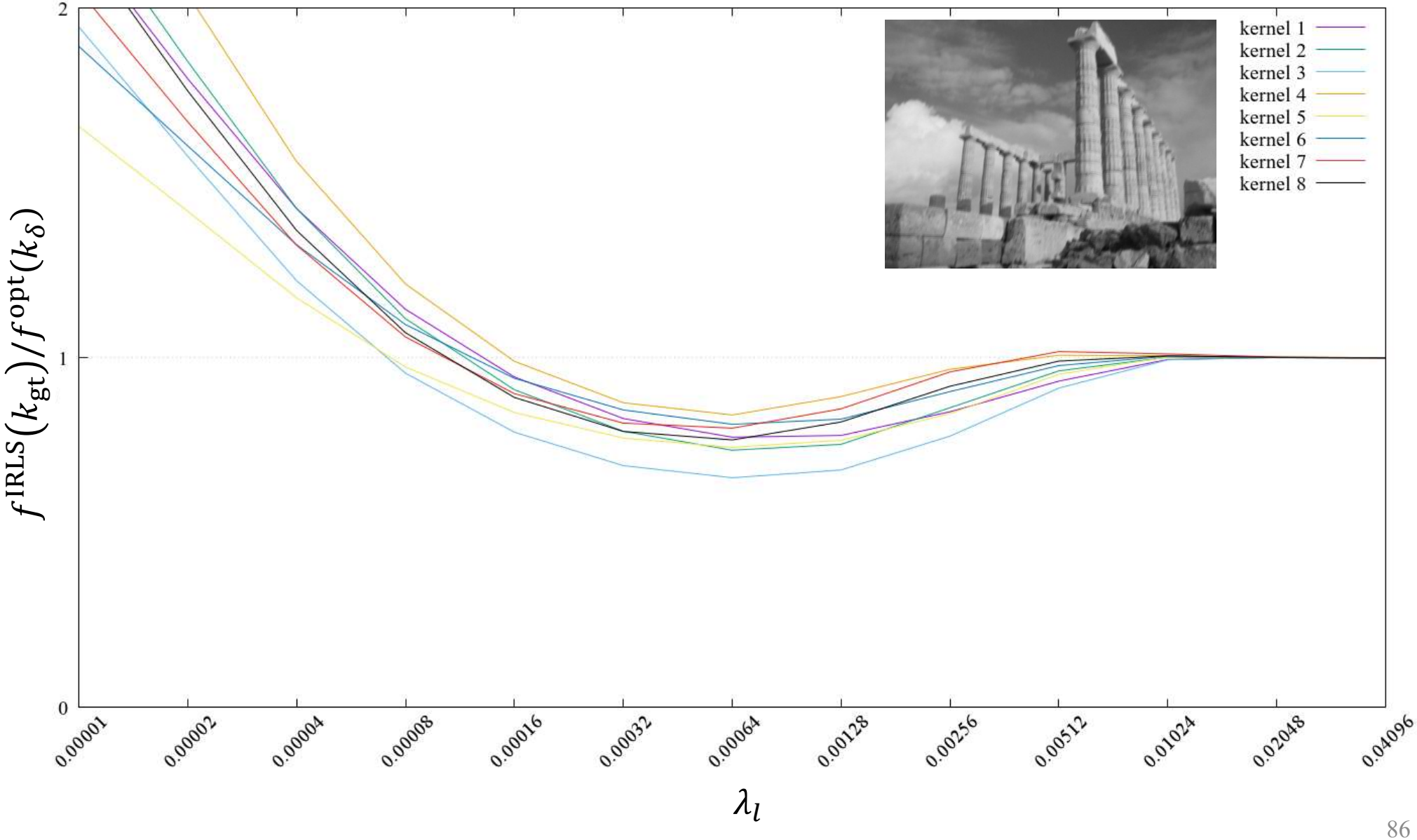
Sun et al.'s dataset - Image 54



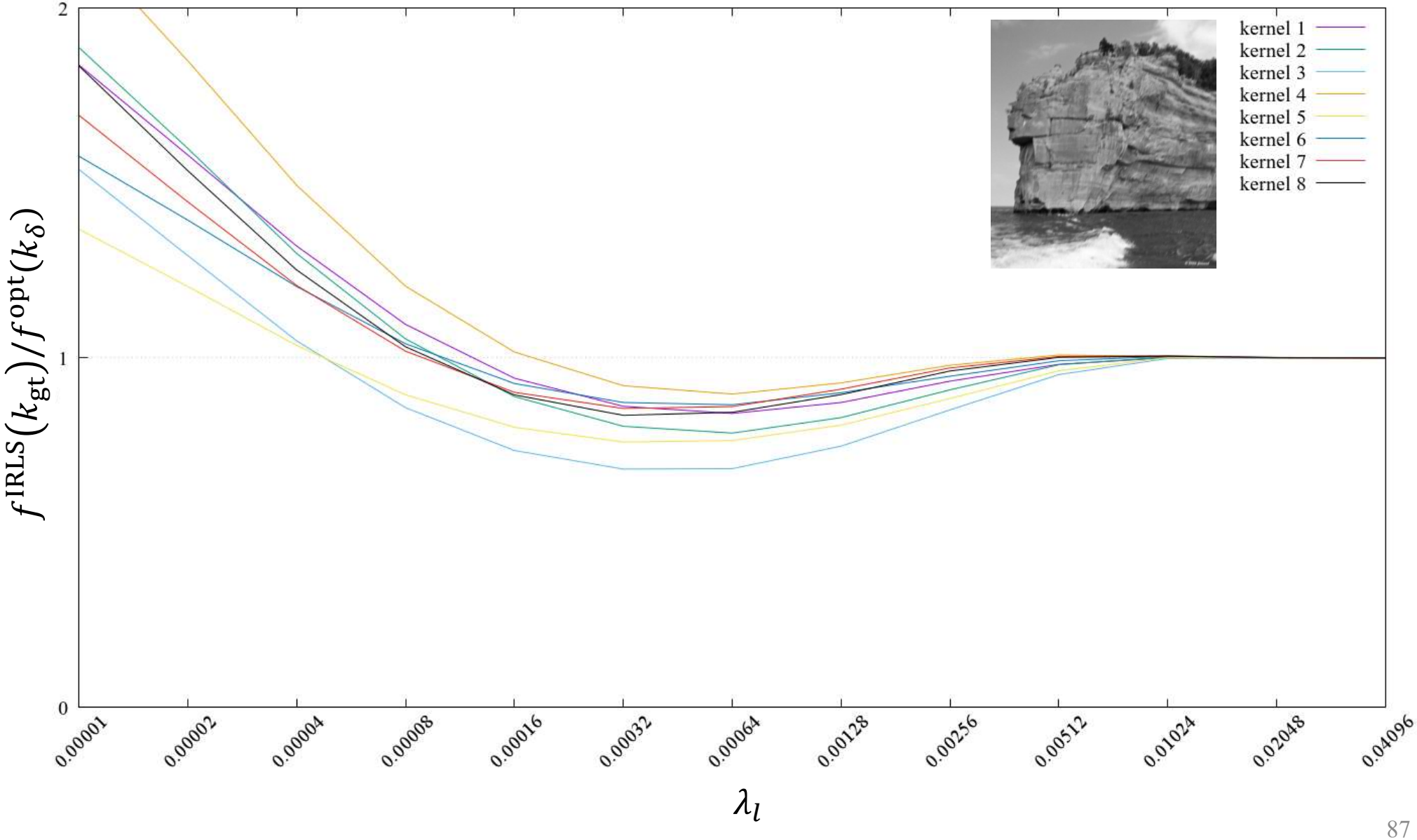
Sun et al.'s dataset - Image 55



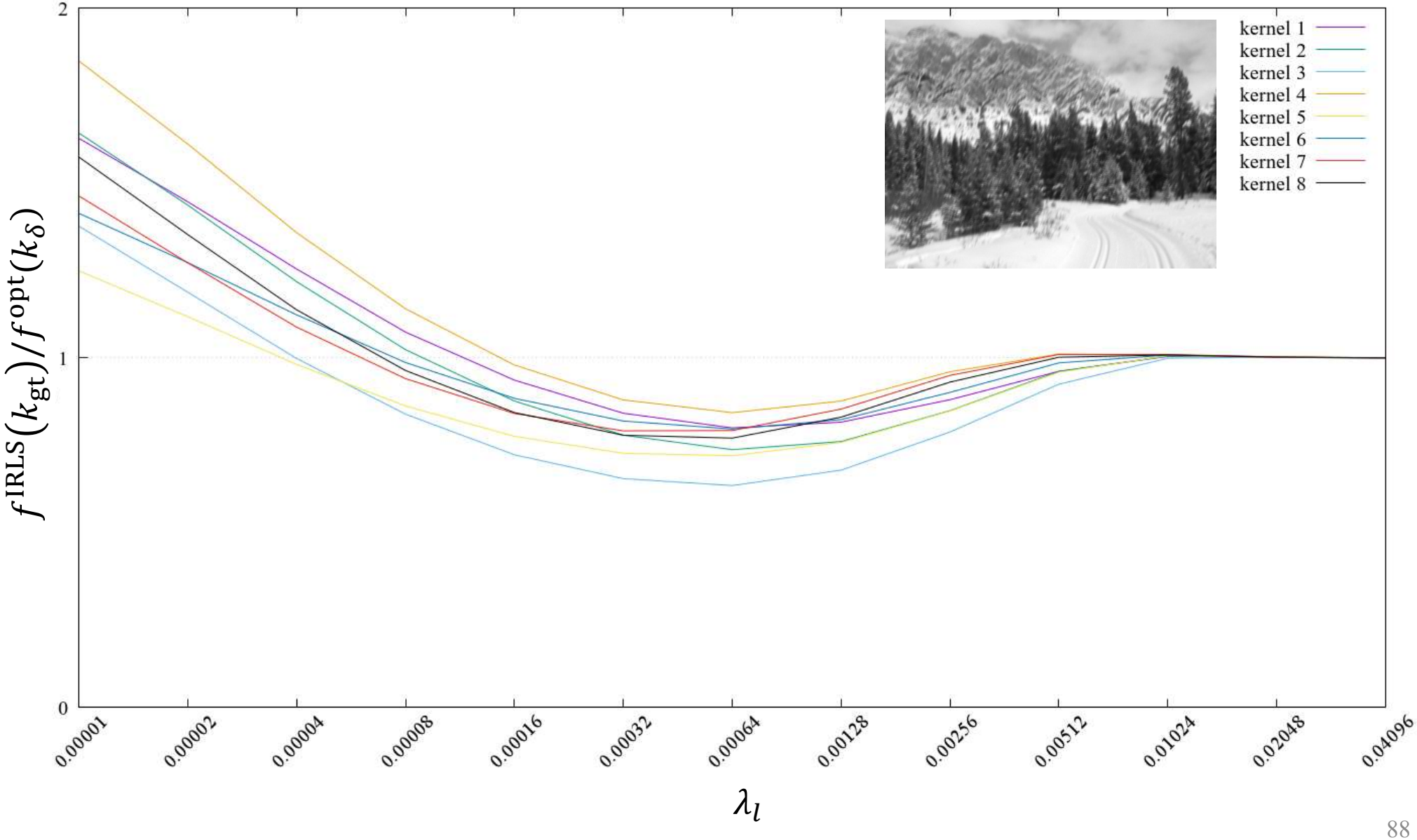
Sun et al.'s dataset - Image 56



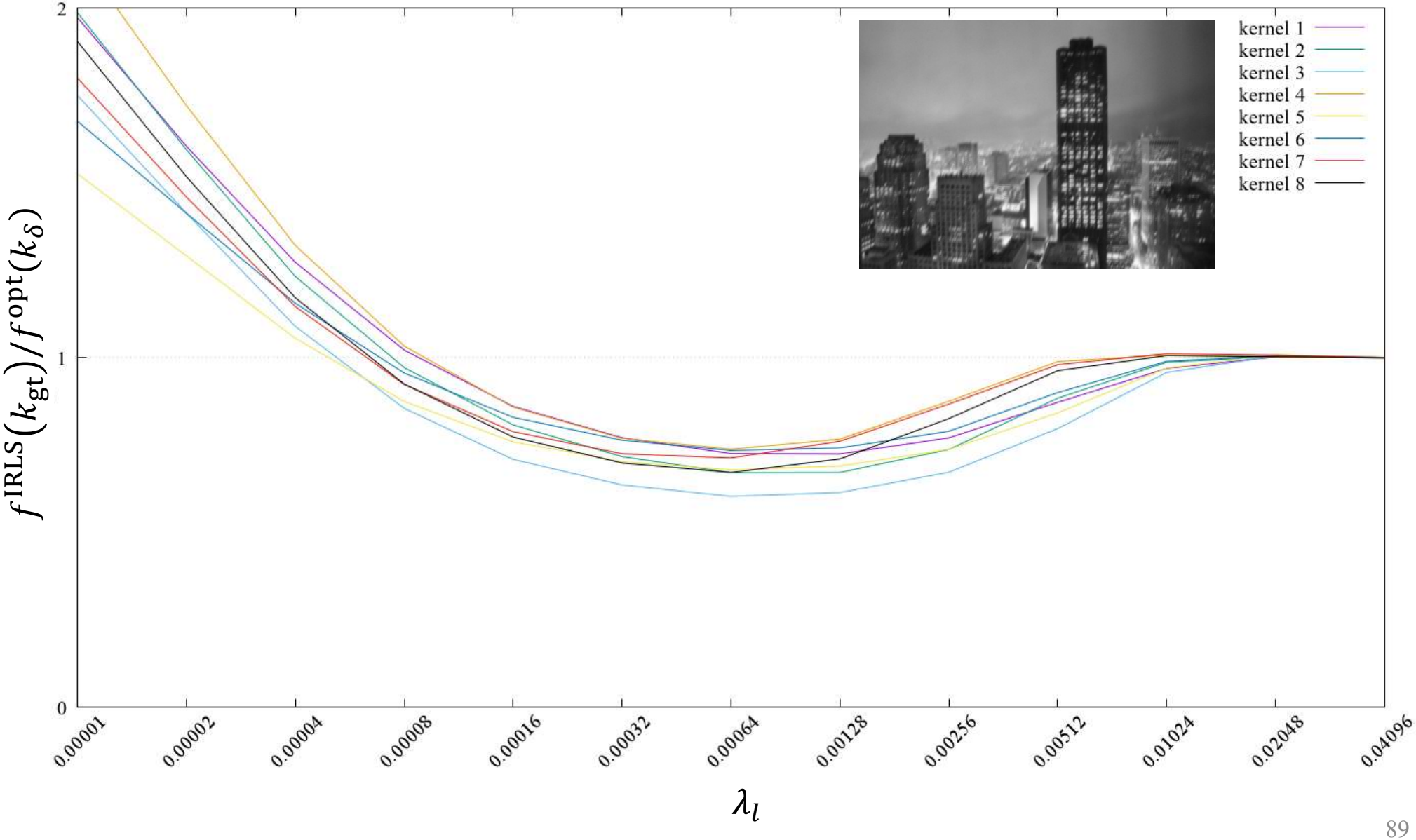
Sun et al.'s dataset - Image 57



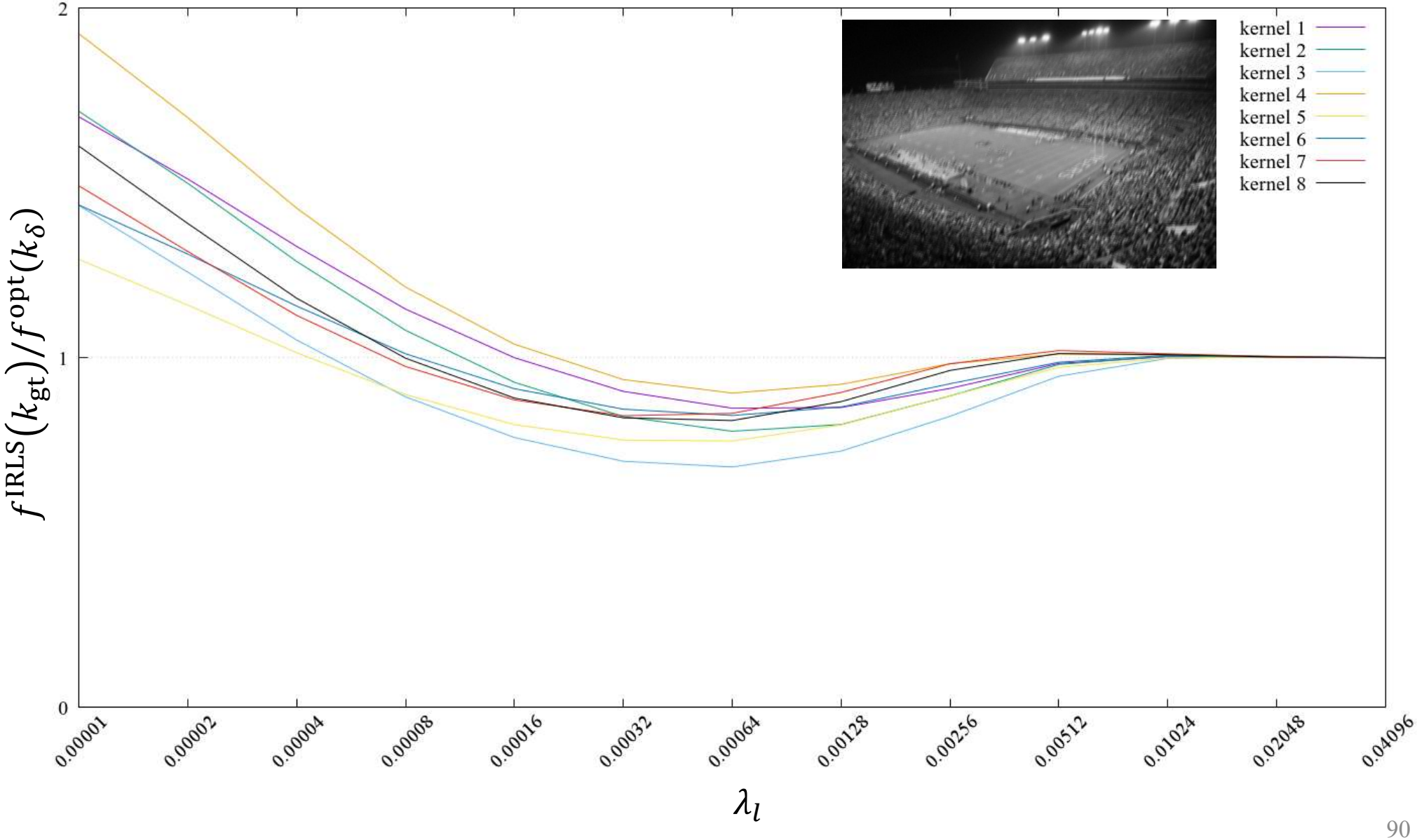
Sun et al.'s dataset - Image 58



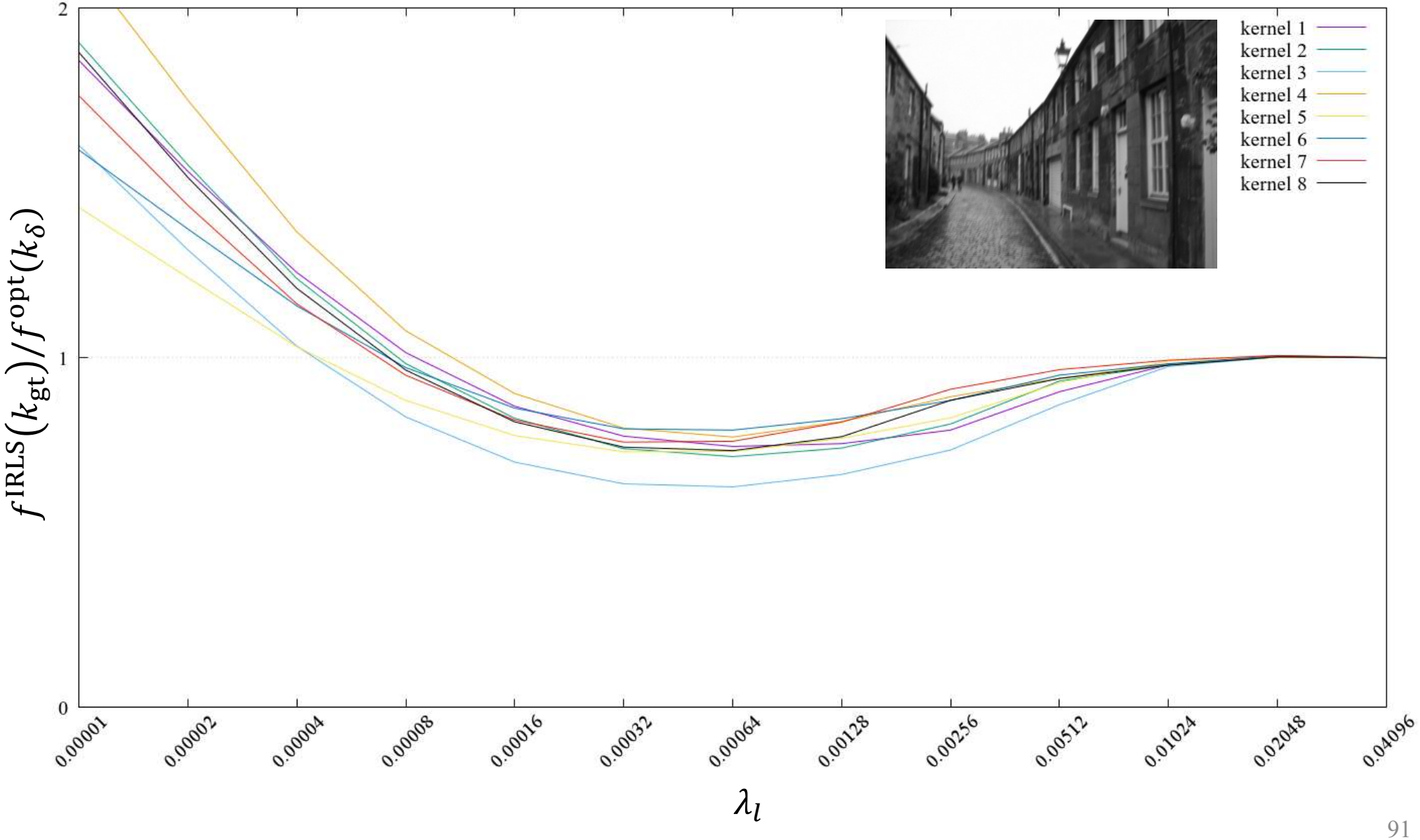
Sun et al.'s dataset - Image 59



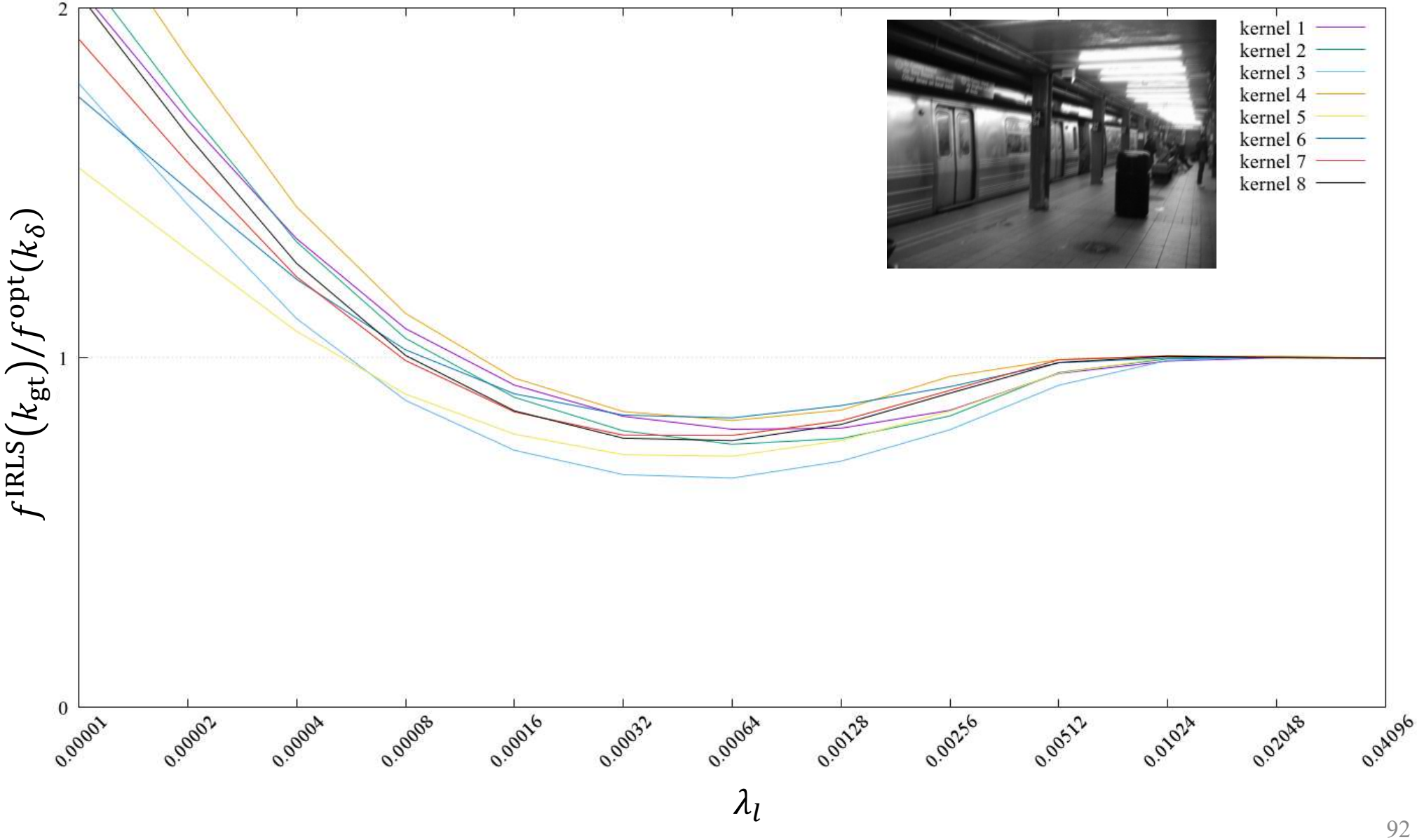
Sun et al.'s dataset - Image 60



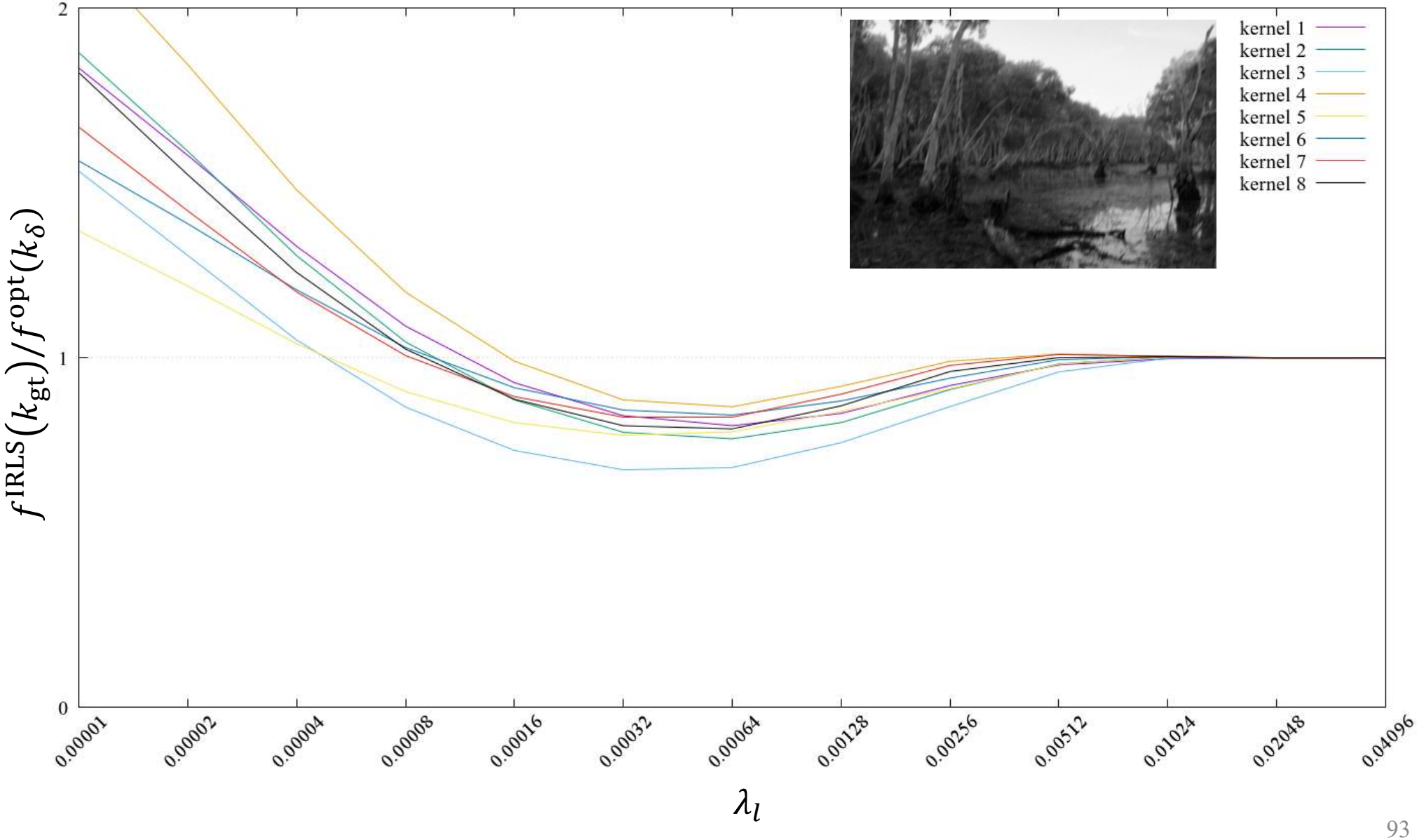
Sun et al.'s dataset - Image 61



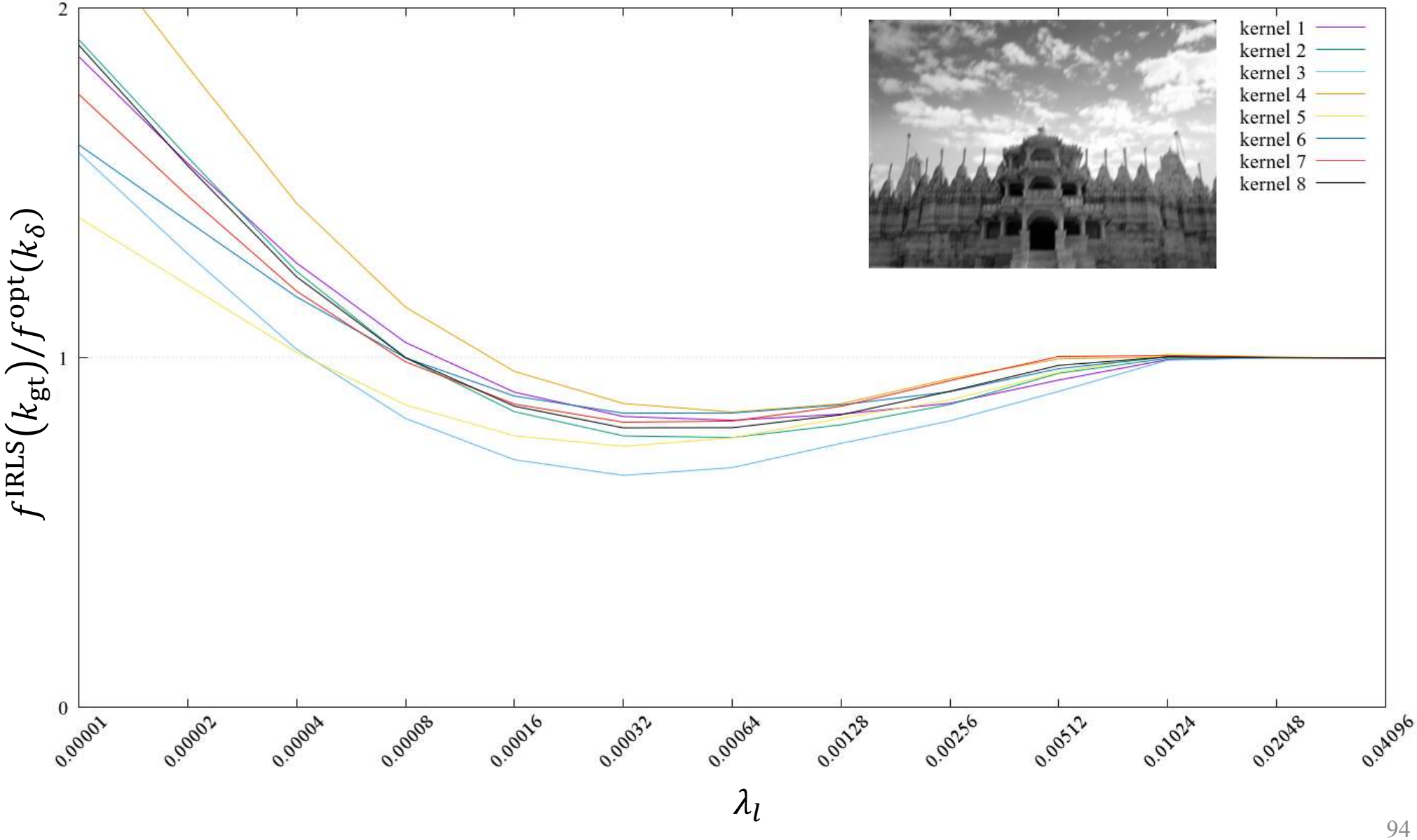
Sun et al.'s dataset - Image 62



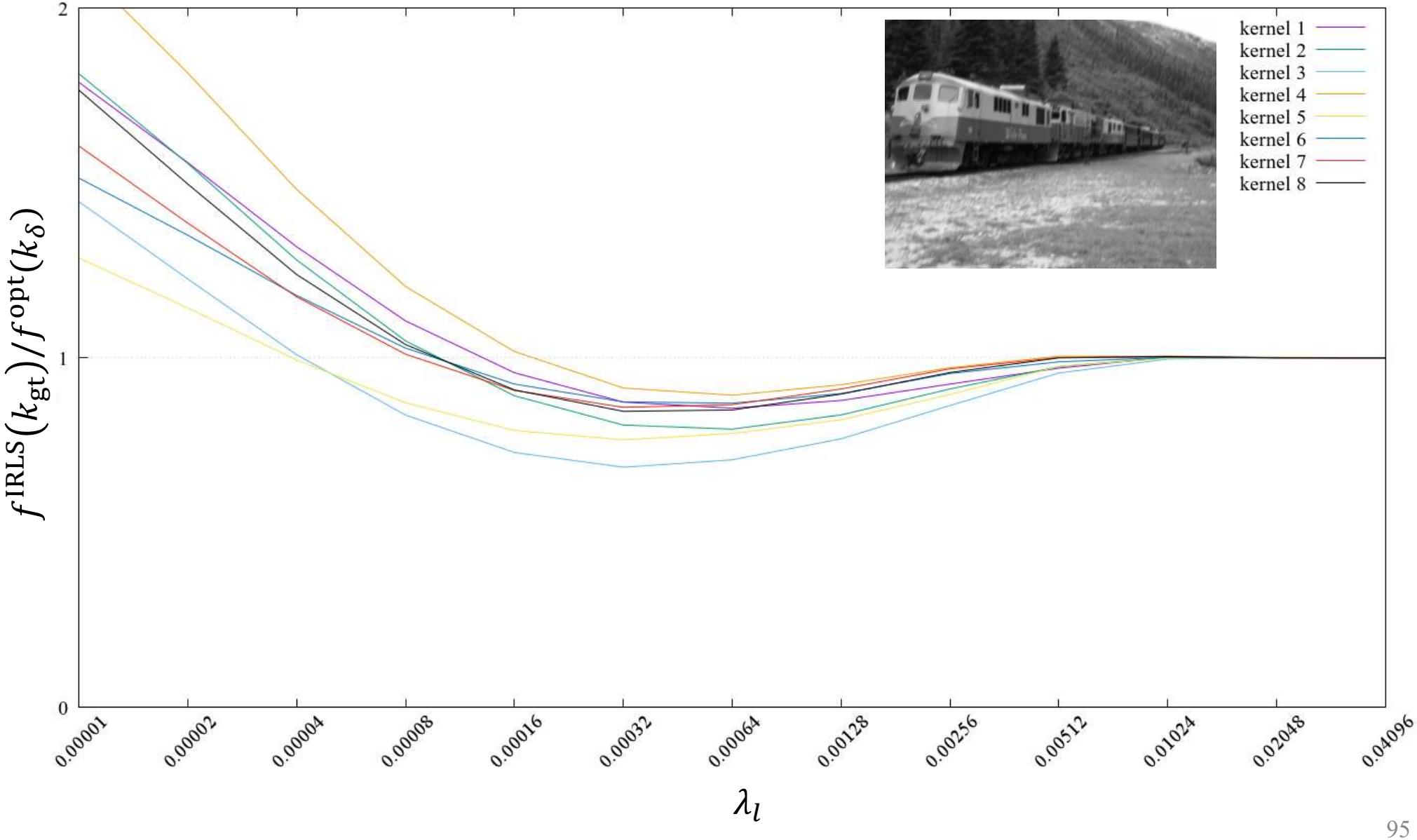
Sun et al.'s dataset - Image 63



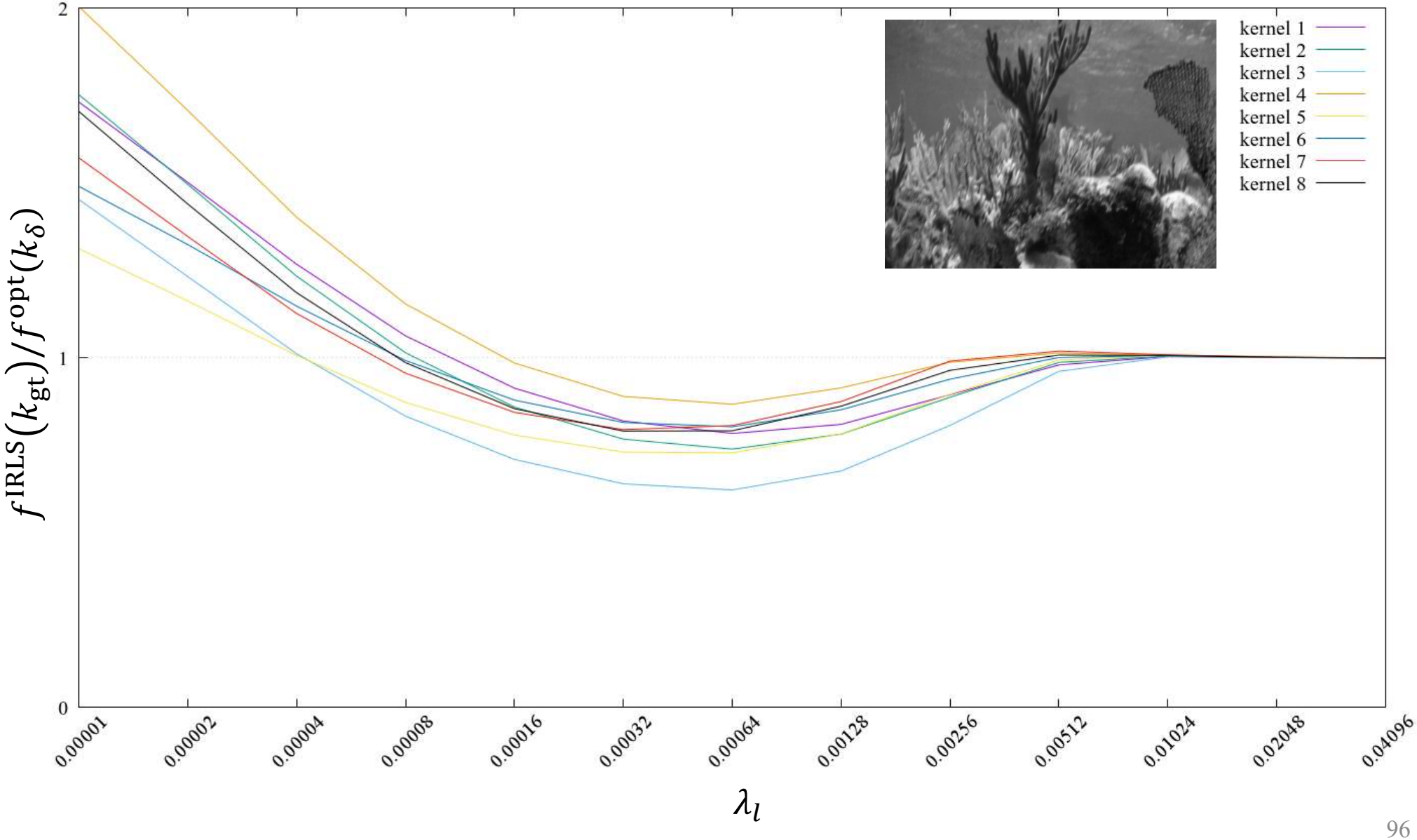
Sun et al.'s dataset - Image 64



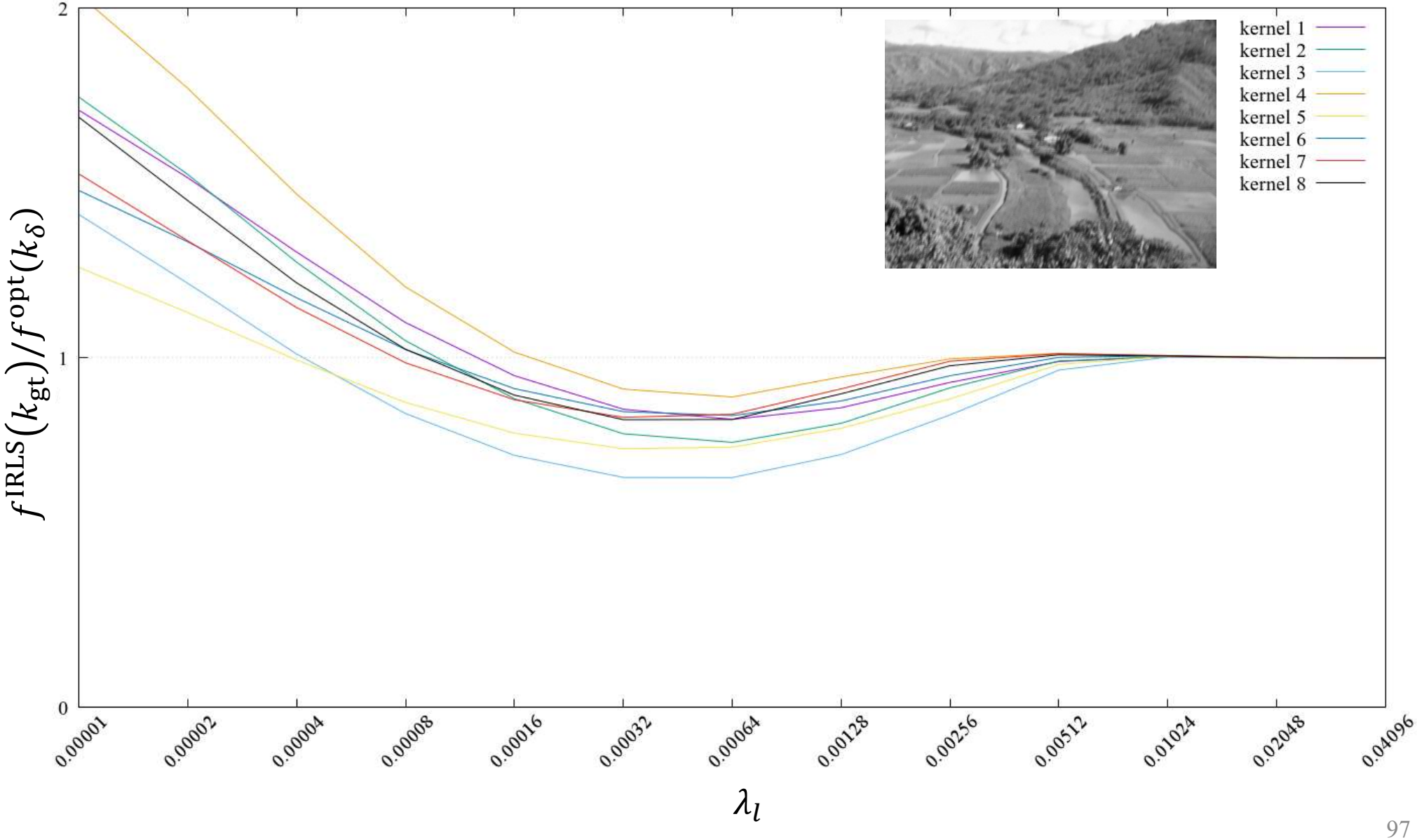
Sun et al.'s dataset - Image 65



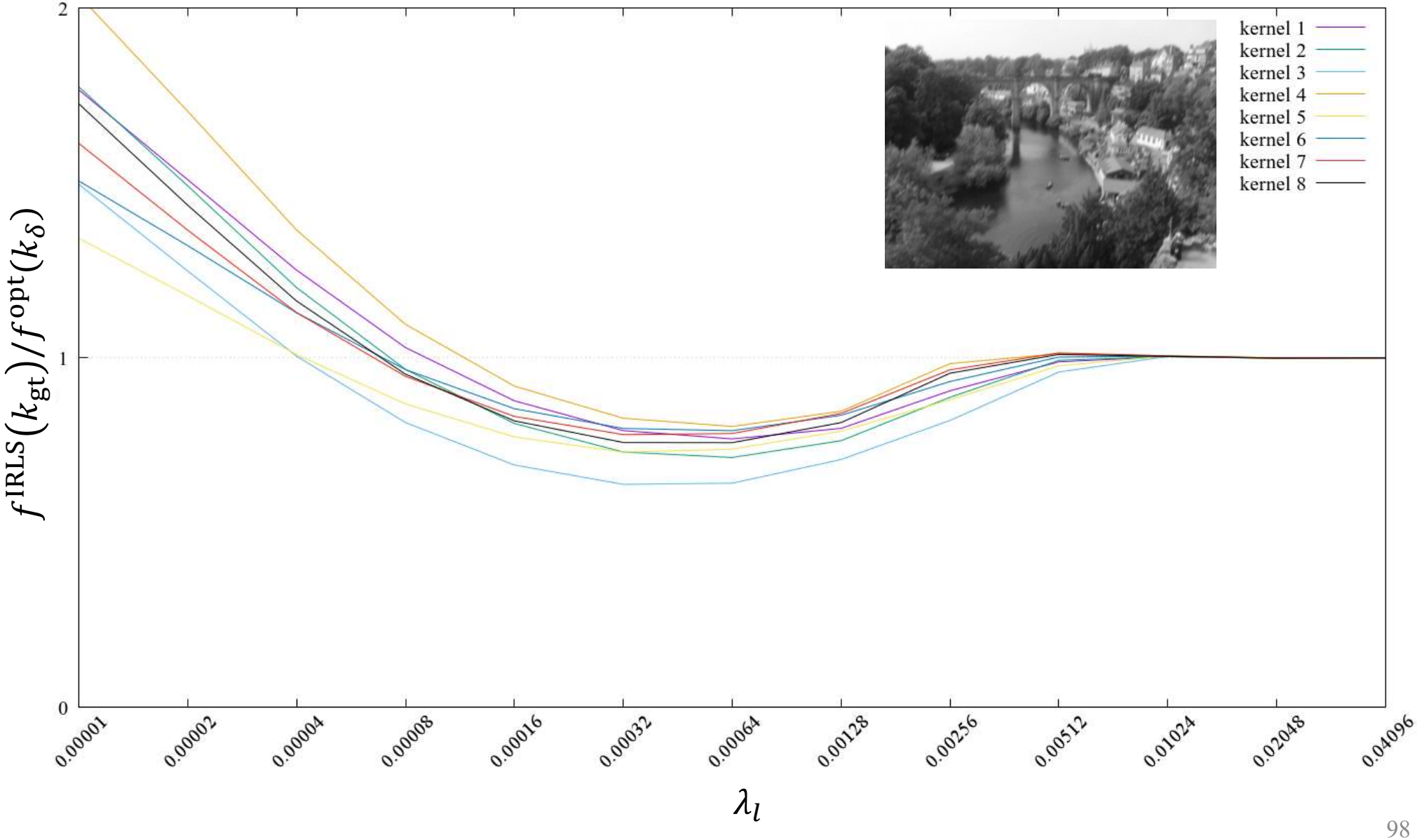
Sun et al.'s dataset - Image 66



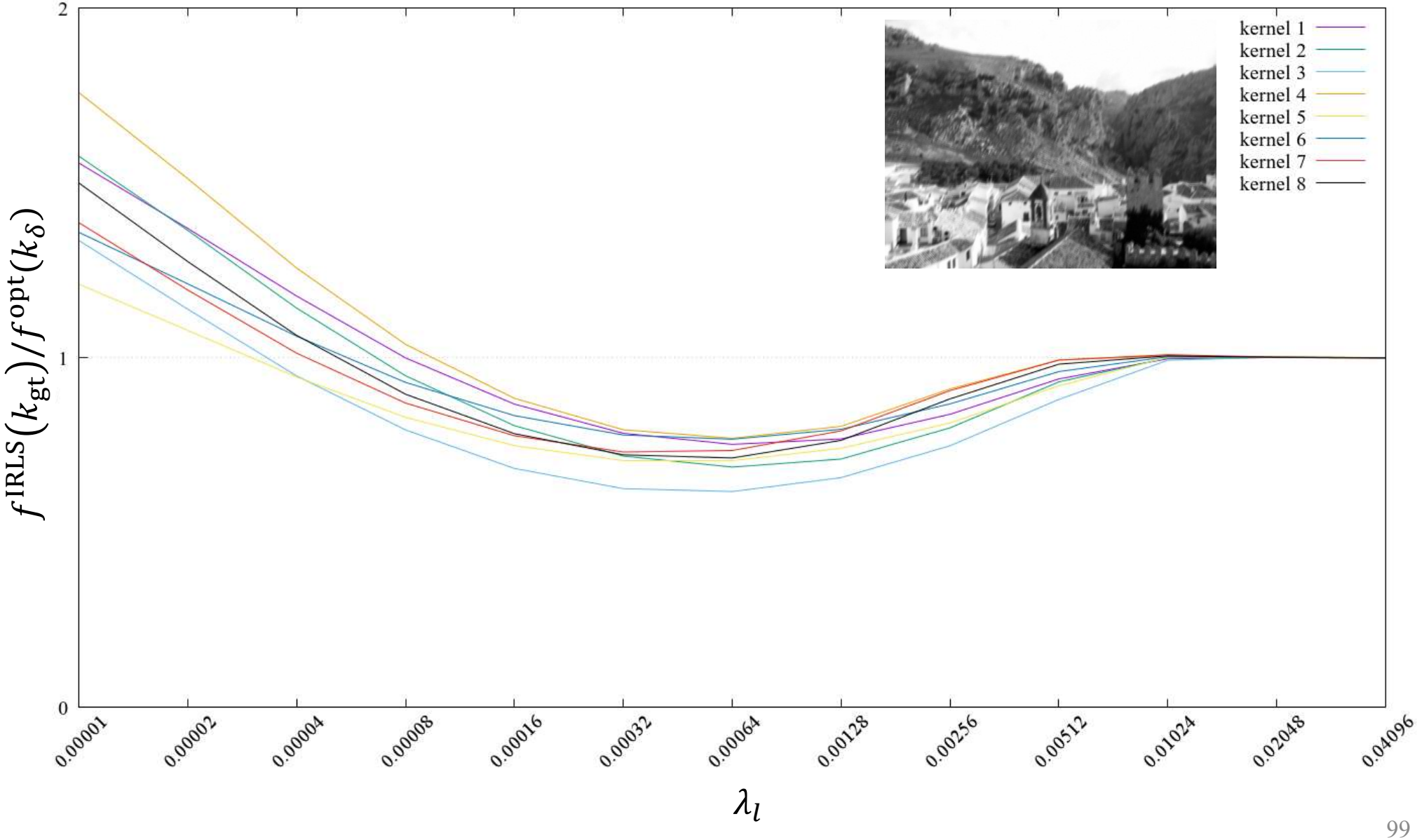
Sun et al.'s dataset - Image 67



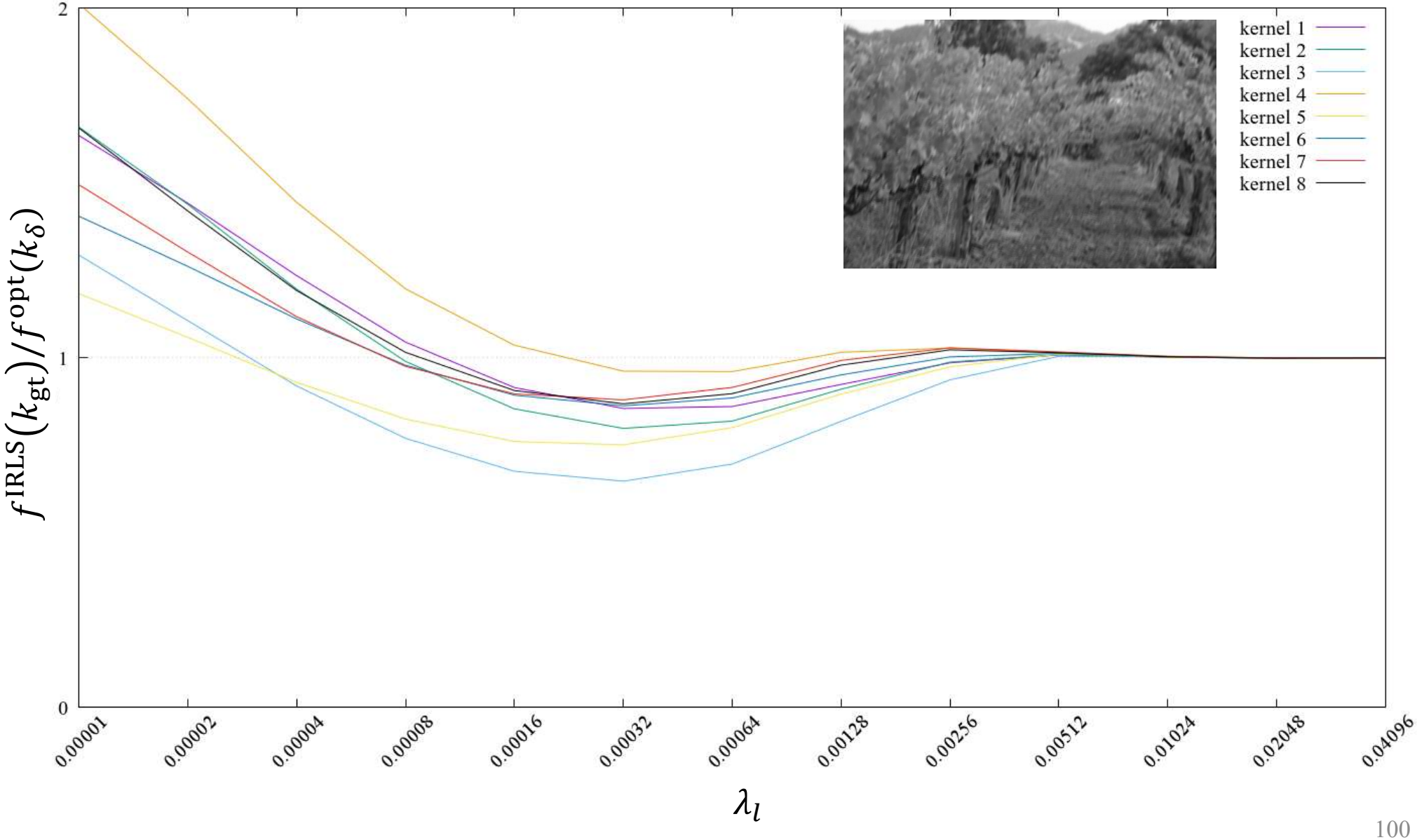
Sun et al.'s dataset - Image 68



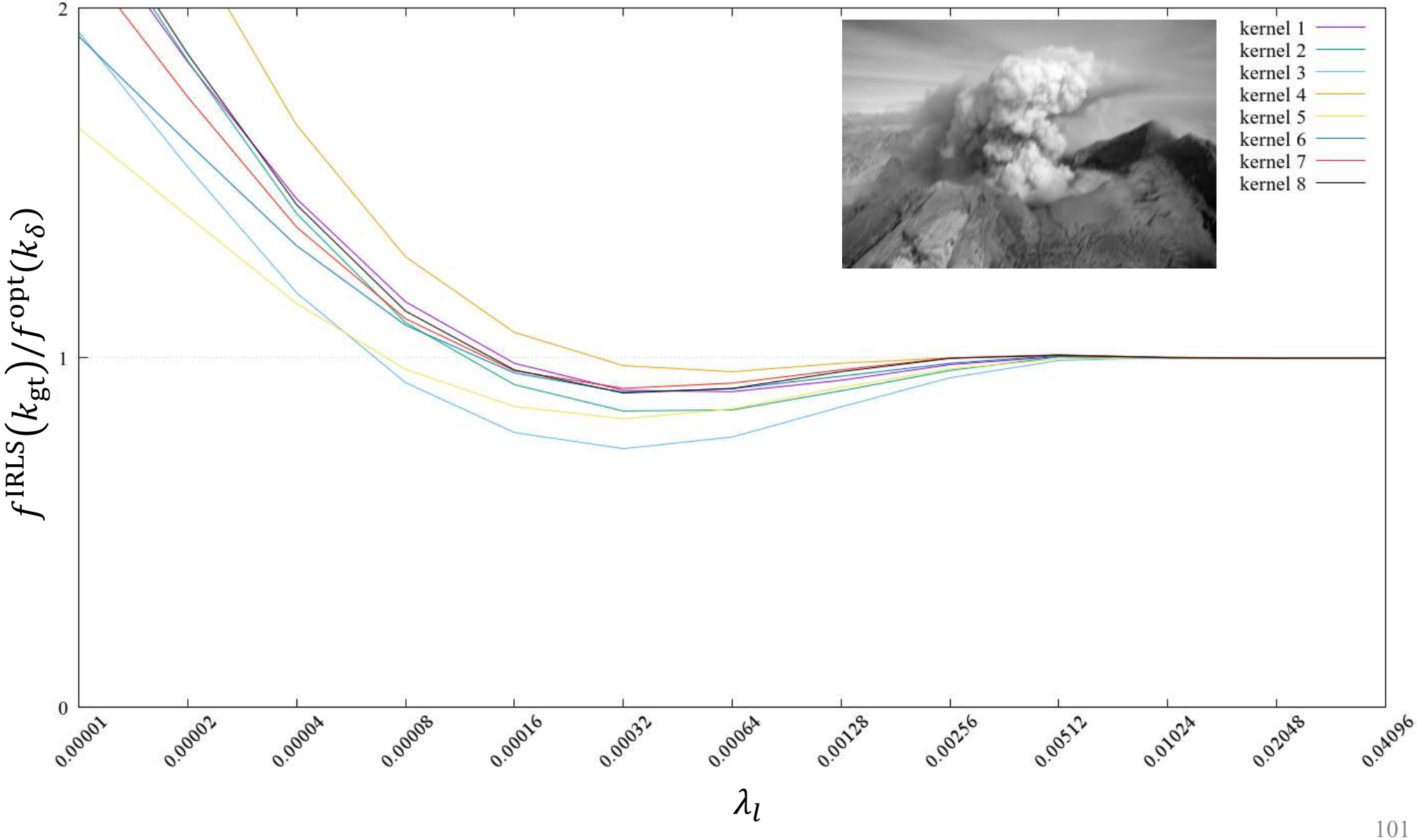
Sun et al.'s dataset - Image 69



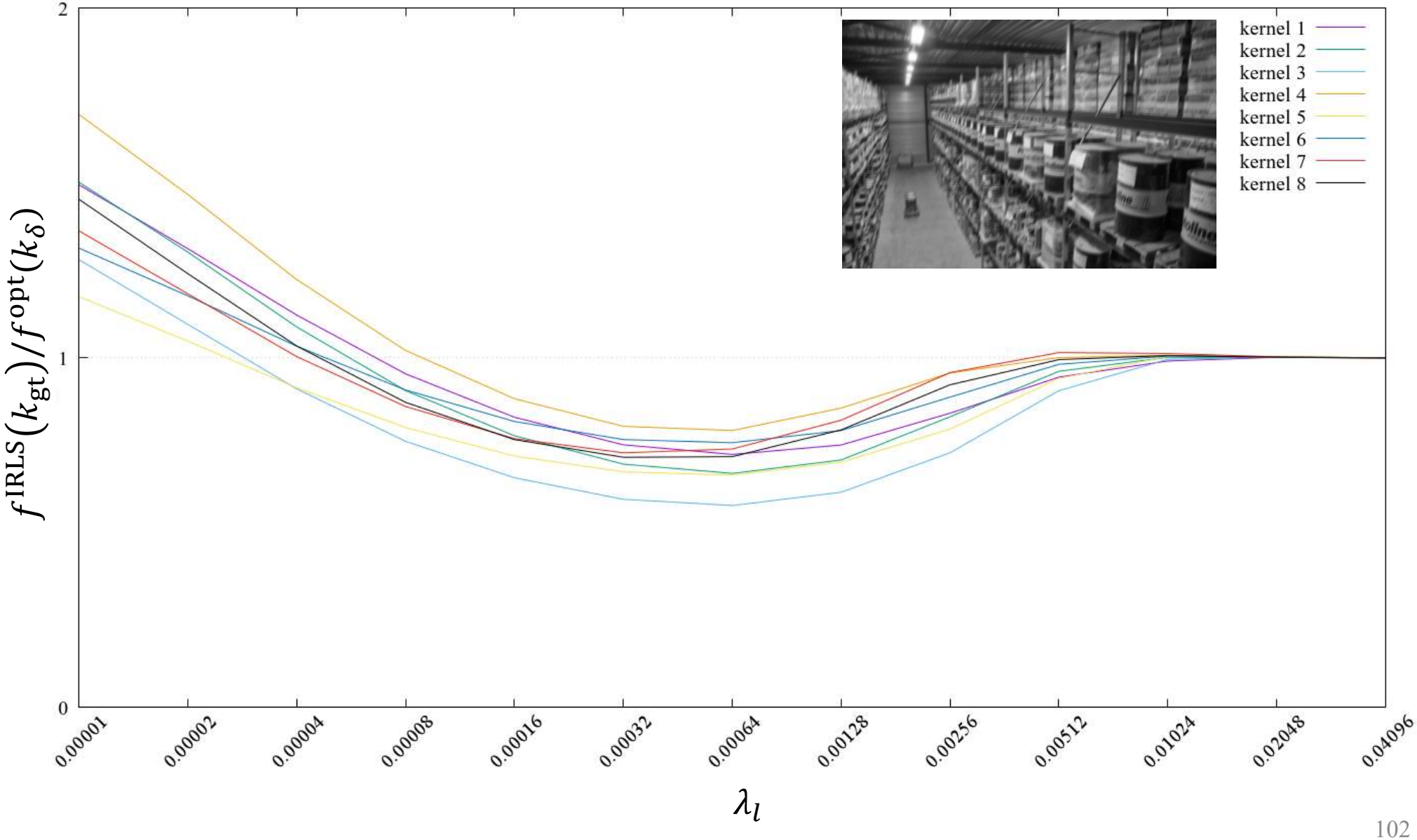
Sun et al.'s dataset - Image 70



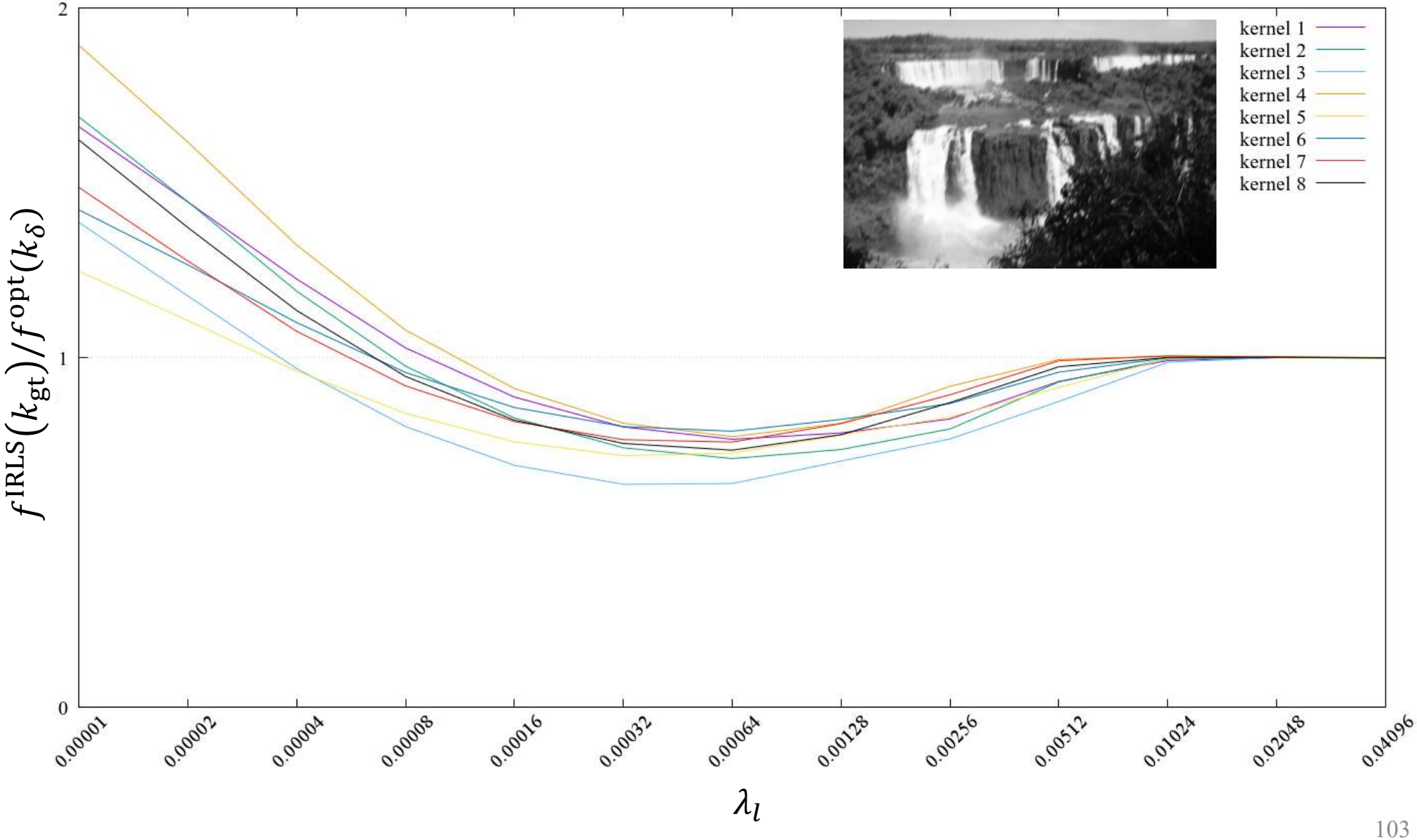
Sun et al.'s dataset - Image 71



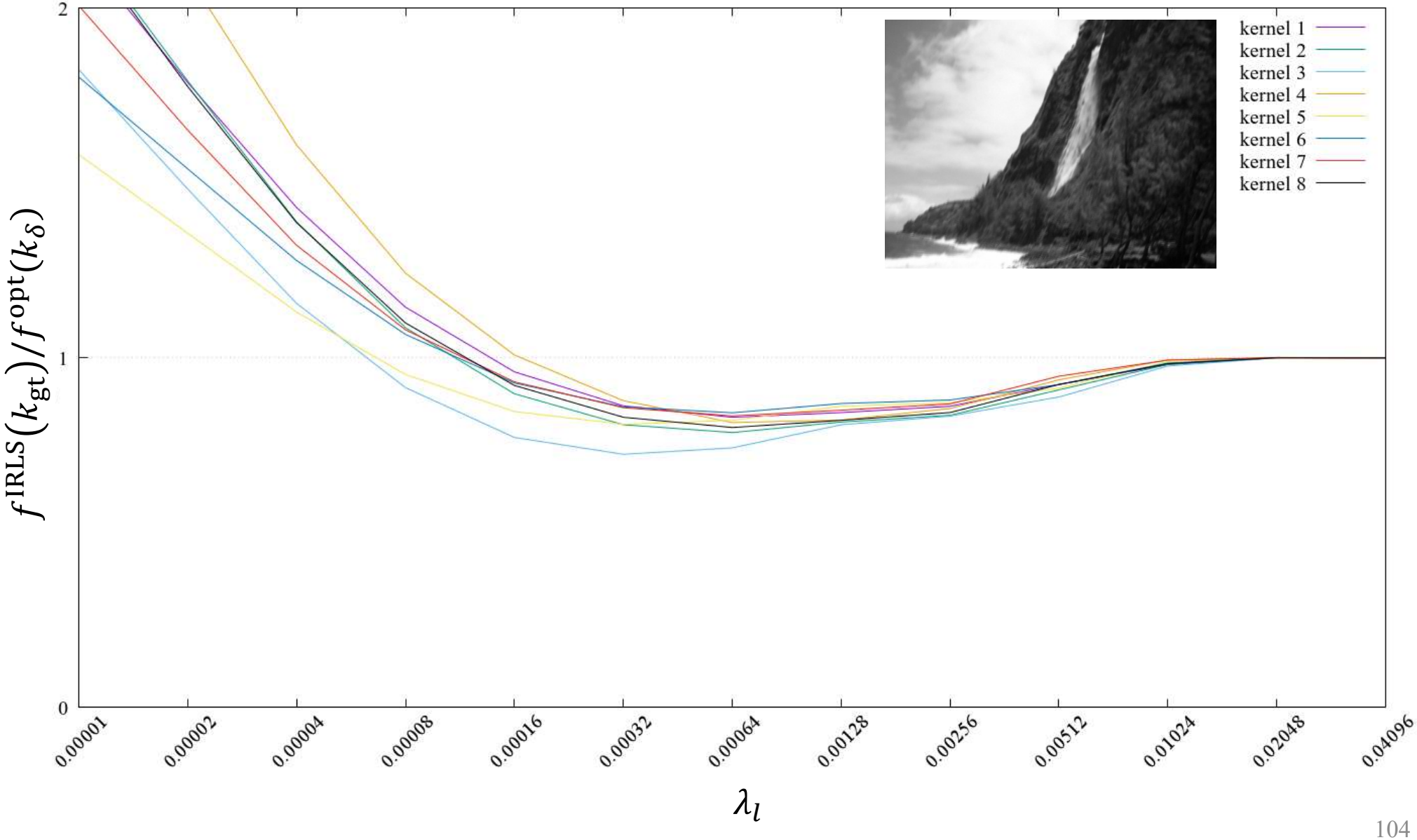
Sun et al.'s dataset - Image 72



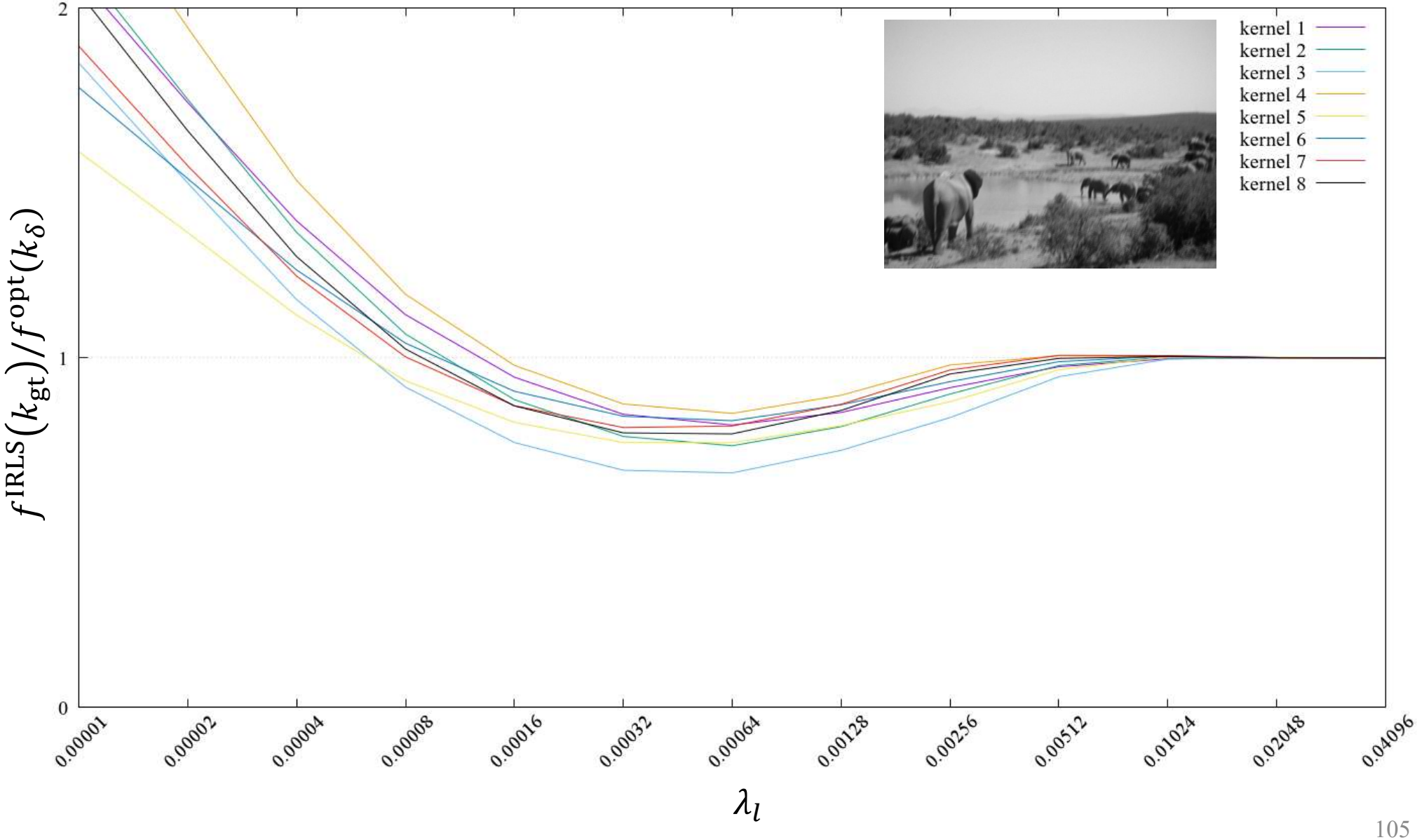
Sun et al.'s dataset - Image 73



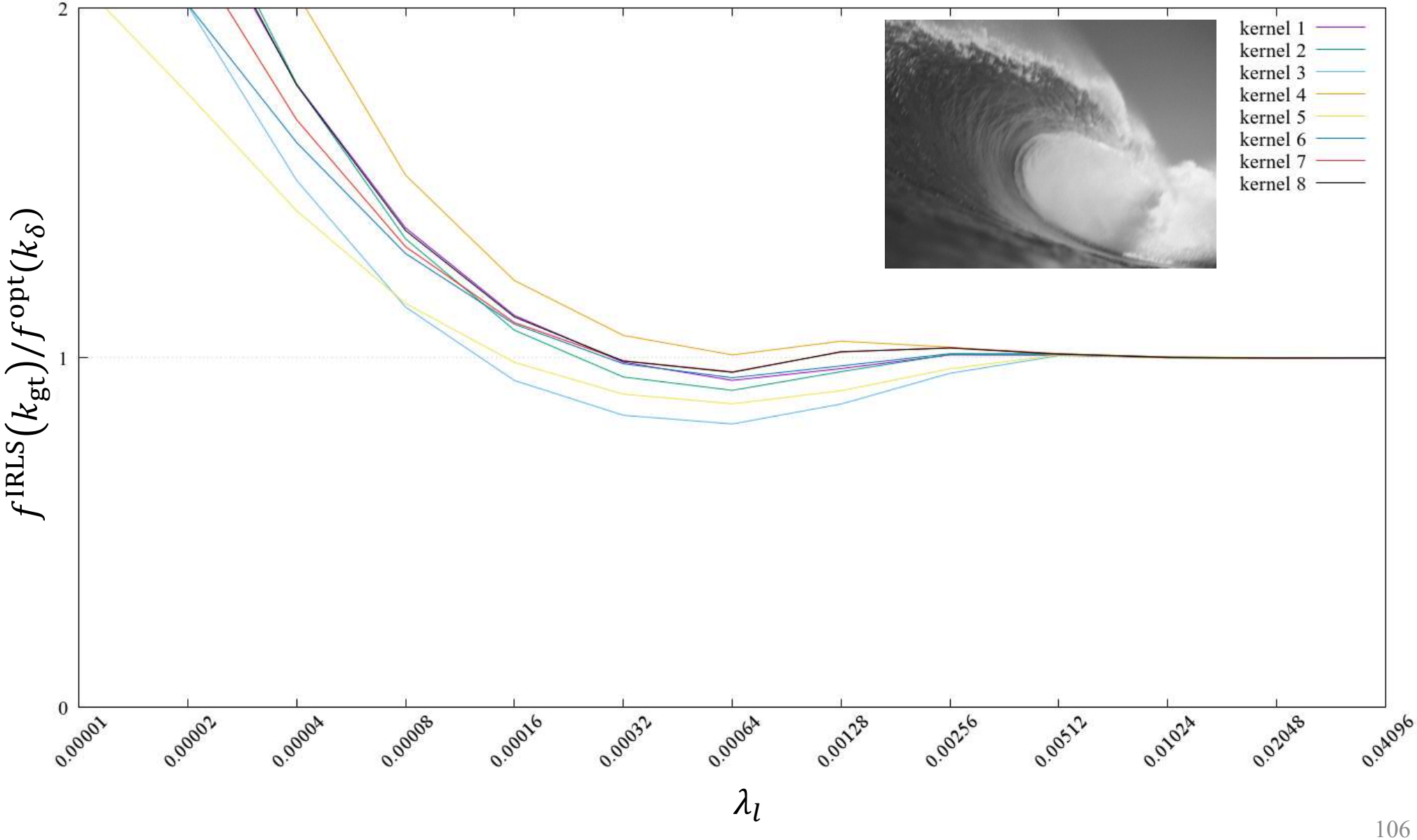
Sun et al.'s dataset - Image 74



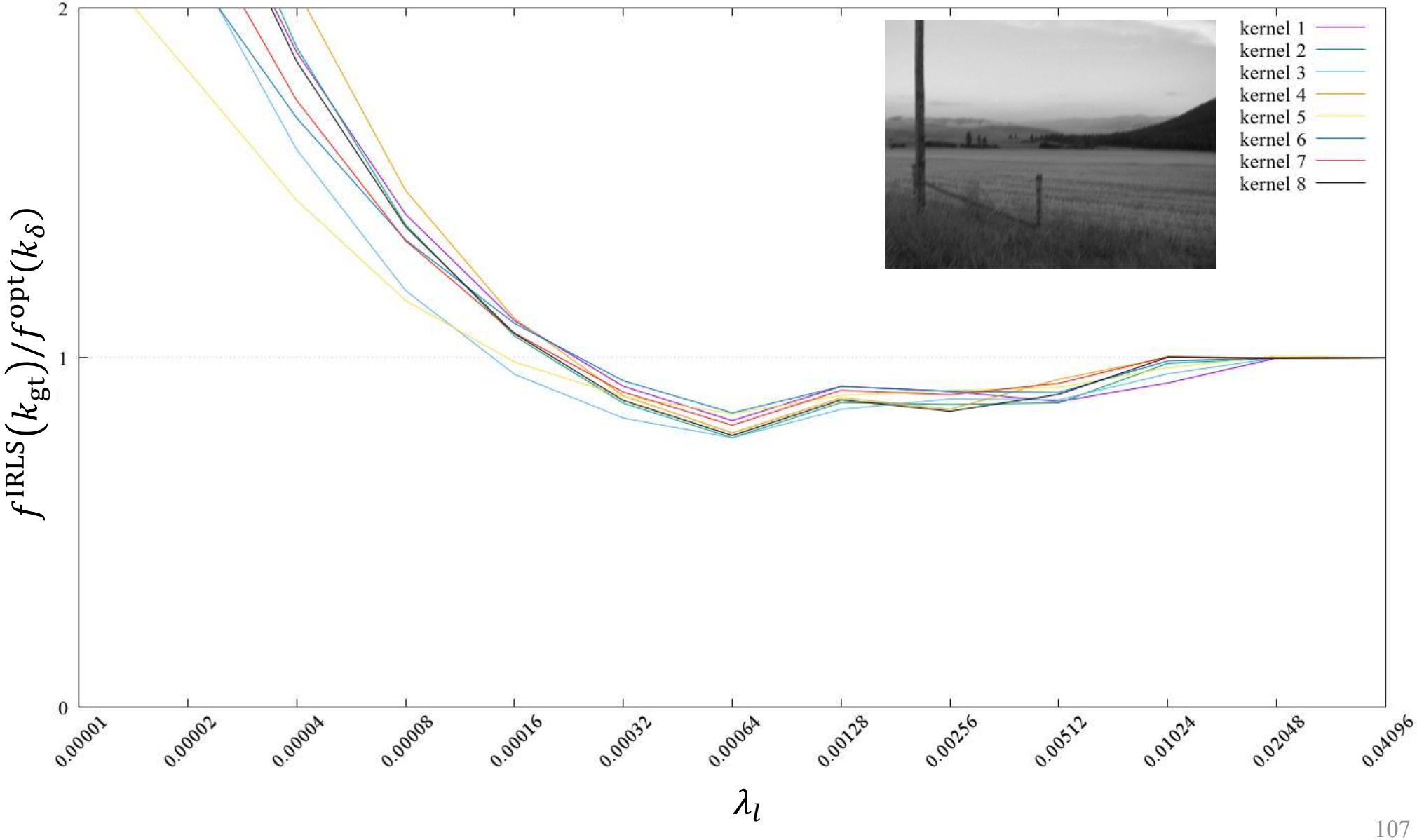
Sun et al.'s dataset - Image 75



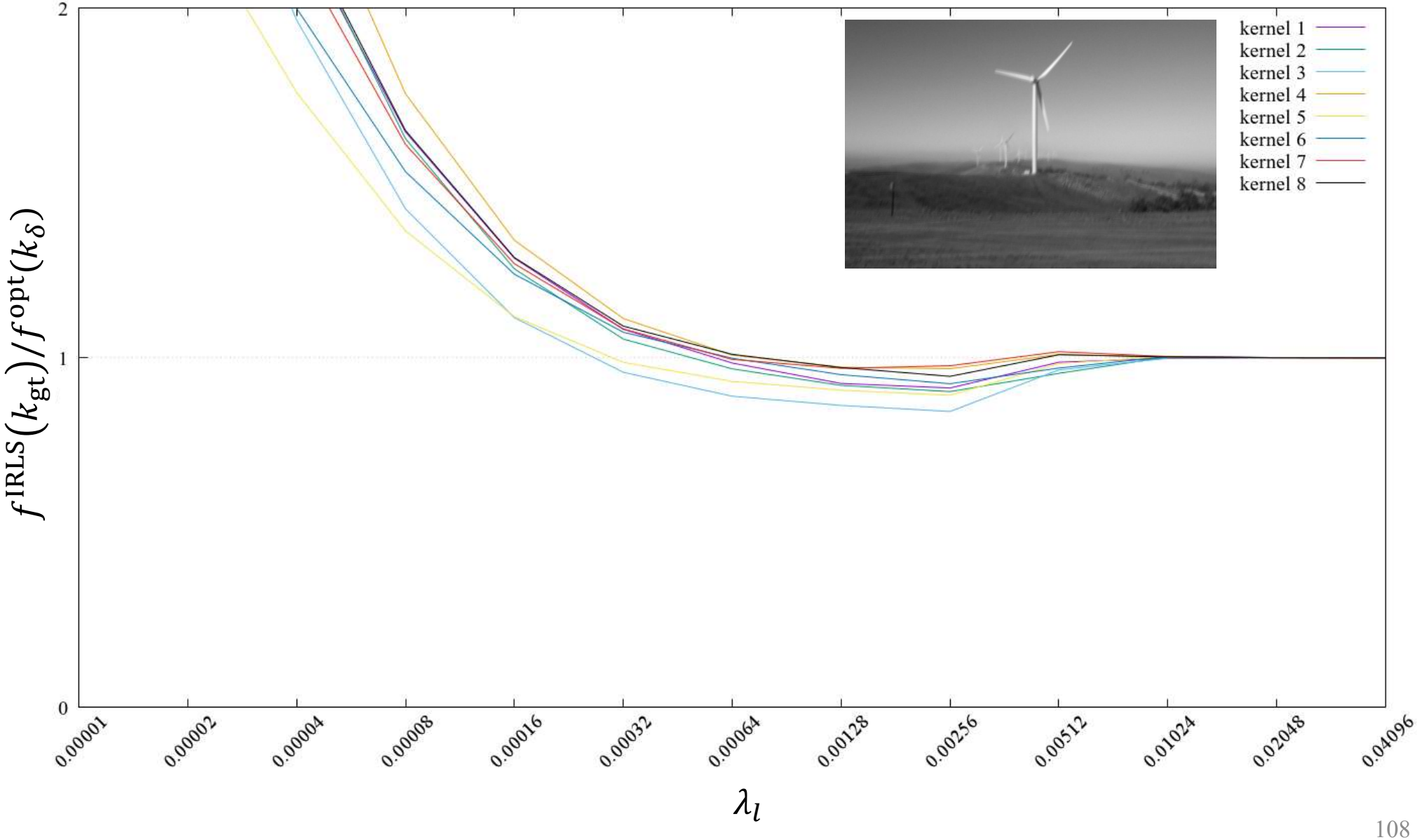
Sun et al.'s dataset - Image 76



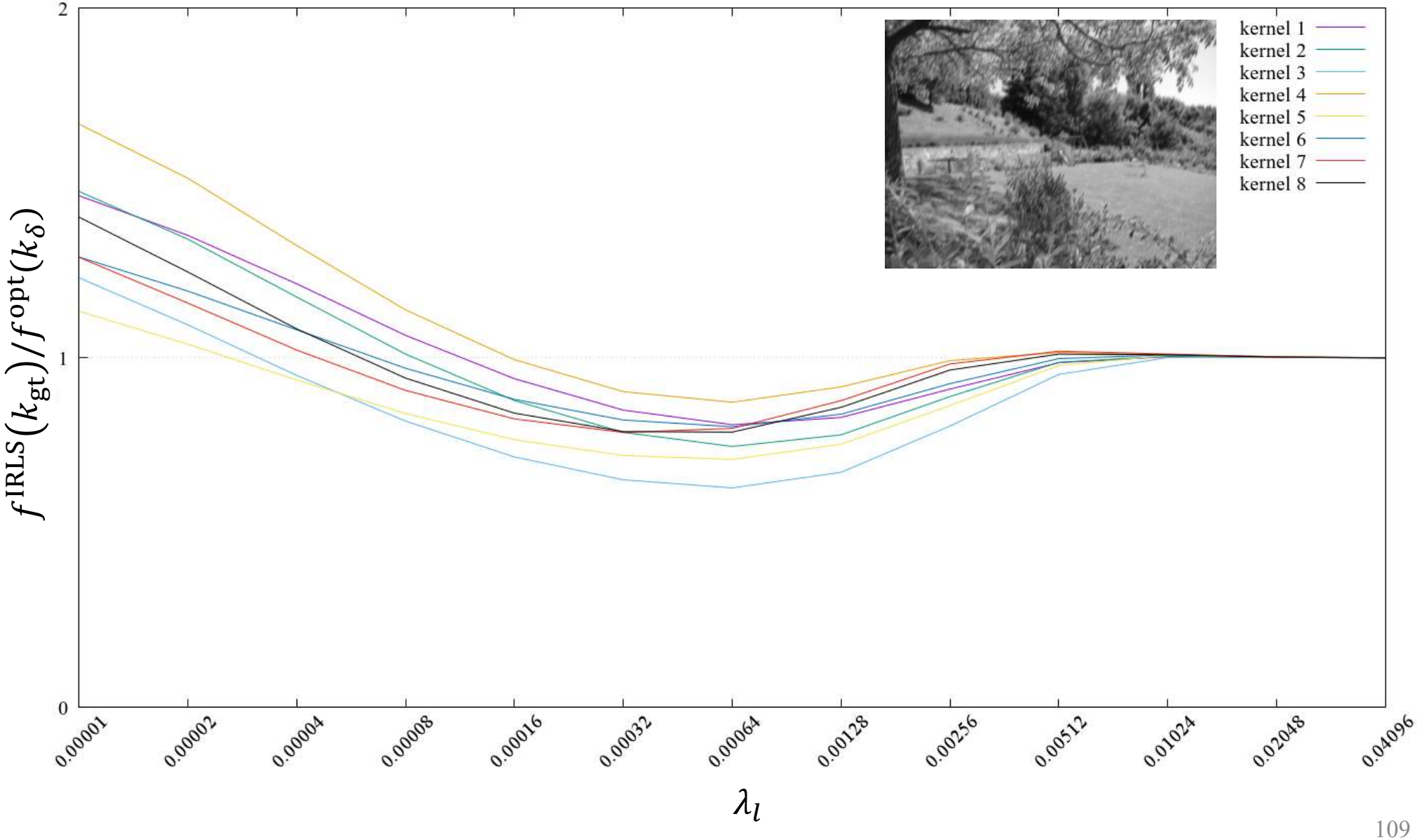
Sun et al.'s dataset - Image 77



Sun et al.'s dataset - Image 78



Sun et al.'s dataset - Image 79



Sun et al.'s dataset - Image 80