Edge Detection Pipeline Comparison Report

Jano Esterhuizen

9 September 2024

Contents

1 Introduction

This report presents a comparison of different edge detection pipelines, including a general pipeline, image-specific pipelines, and the Canny edge detector. The performance of these pipelines is evaluated using various metrics on a set of test images.

2 Methodology

2.1 Pipeline Components

The edge detection pipeline consists of the following components:

- Noise Reduction
- Gradient Calculation
- Edge Thinning
- Thresholding
- Edge Linking

2.2 Optimization Process

The pipeline parameters were optimized using a grid search algorithm. The following parameter grid was used:

```
'noise_reduction': {
    'method': ['gaussian', 'median'],
    'kernel_size': [3, 5, 7],
    'sigma': [0.5, 1, 1.5]
},
'edge_thinning': {
    'min_threshold': [5, 10, 15],
    'max_threshold': [150, 200, 250]
},
'thresholding': {
    'low': [30, 50, 70],
    'high': [80, 100, 120]
}
```

2.3 Evaluation Metrics

The following metrics were used to evaluate the performance of the edge detection pipelines:

- ORB Feature Similarity

- SIFT Feature Similarity
- Mean Squared Error (MSE)
- Peak Signal-to-Noise Ratio (PSNR)
- Structural Similarity Index (SSIM)

3 Results

3.1 Overall Performance

Image	General Pipeline	Image-Specific Pipeline	Canny Edge Detector
Img0.png	0.3248	0.3504	0.3612
Img1.png	0.3507	0.3667	0.3597
Img10.png	0.3263	0.3415	0.3606
Img2.png	0.4291	0.4574	0.4124
Img3.png	0.3211	0.3368	0.3432
Img4.png	0.3407	0.4021	0.3534
Img5.png	0.3461	0.3784	0.3830
Img6.png	0.3409	0.3631	0.3628
Img7.png	0.3398	0.3621	0.3644
Img8.png	nan	0.3679	0.3596
Img9.png	0.3384	0.3573	0.3708

3.2 Detailed Results

3.2.1 Img0.png

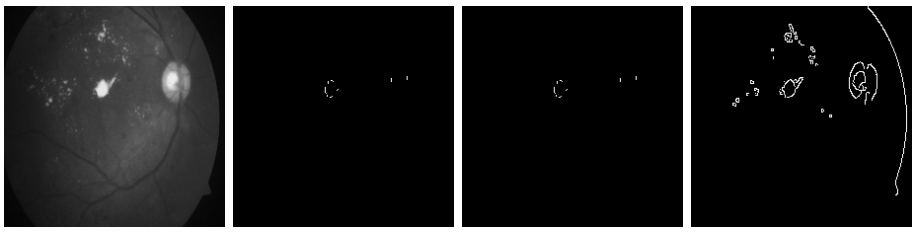

Figure 1: Edge detection results for Img0.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0159
SIFT_similarity	0.0323
MSE	113.7105
PSNR	27.5728
SSIM	0.0093

Specific Pipeline

Metric	Value
ORB_similarity	0.0159
SIFT_similarity	0.0323
MSE	113.7105
PSNR	27.5728
SSIM	0.0093

Canny Pipeline

Metric	Value
ORB_similarity	0.1566
SIFT_similarity	0.2258
MSE	113.6047
PSNR	27.5768
SSIM	0.0141

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.2 Img1.png

Figure 2: Edge detection results for Img1.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0619
SIFT_similarity	0.0312
MSE	104.6687
PSNR	27.9326
SSIM	0.0630

Specific Pipeline

Metric	Value
ORB_similarity	0.0619
SIFT_similarity	0.0312
MSE	104.6687
PSNR	27.9326
SSIM	0.0630

Canny Pipeline

Metric	Value
ORB_similarity	0.0941
SIFT_similarity	0.2188
MSE	104.5652
PSNR	27.9369
SSIM	0.0668

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
```

```
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.3 Img10.png

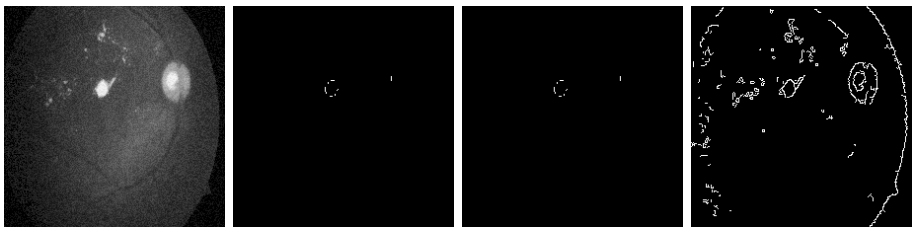

Figure 3: Edge detection results for Img10.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0223
SIFT_similarity	0.0441
MSE	102.8641
PSNR	28.0082
SSIM	0.0042

Specific Pipeline

Metric	Value
ORB_similarity	0.0223
SIFT_similarity	0.0441
MSE	102.8641
PSNR	28.0082
SSIM	0.0042

Canny Pipeline

Metric	Value
ORB_similarity	0.1544
SIFT_similarity	0.2206
MSE	103.0738
PSNR	27.9993
SSIM	0.0095

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.4 Img2.png

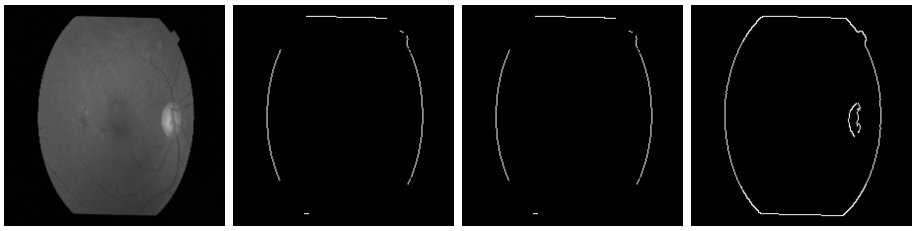

Figure 4: Edge detection results for Img2.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.1622
SIFT_similarity	0.0000
MSE	68.0883
PSNR	29.8001
SSIM	0.2572

Specific Pipeline

Metric	Value
ORB_similarity	0.1622
SIFT_similarity	0.0000
MSE	68.0883
PSNR	29.8001
SSIM	0.2572

Canny Pipeline

Metric	Value
ORB_similarity	0.1010
SIFT_similarity	0.0556
MSE	68.1083
PSNR	29.7988
SSIM	0.2516

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.5 Img3.png

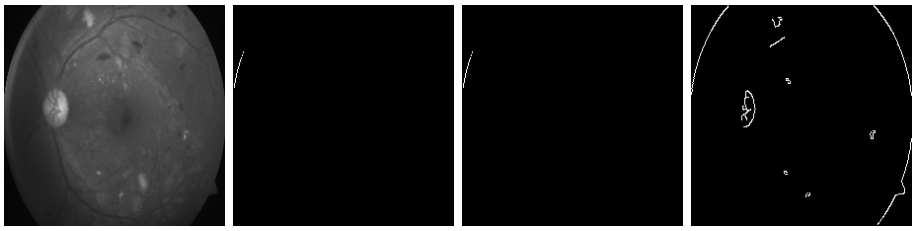

Figure 5: Edge detection results for Img3.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	108.6326
PSNR	27.7712
SSIM	0.0083

Specific Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	108.6326
PSNR	27.7712
SSIM	0.0083

Canny Pipeline

Metric	Value
ORB_similarity	0.0875
SIFT_similarity	0.1538
MSE	108.6465
PSNR	27.7706
SSIM	0.0092

Image-Specific Pipeline Parameters

{'noise_reduction': {'method': 'median', 'kernel_size': 7, 'sigma': 1.5}, 'gradient_calculat

3.2.6 Img4.png

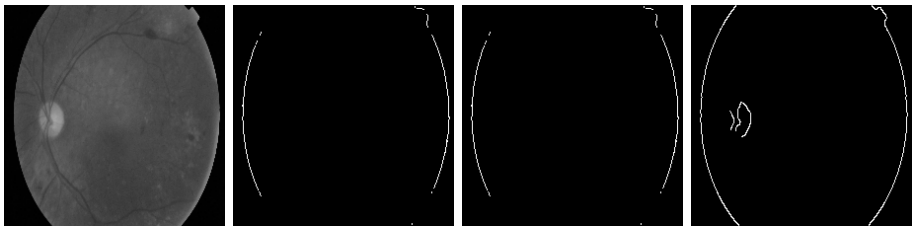

Figure 6: Edge detection results for Img4.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	99.0688
PSNR	28.1714
SSIM	0.0825

Specific Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	99.0688
PSNR	28.1714
SSIM	0.0825

Canny Pipeline

Metric	Value
ORB_similarity	0.0545
SIFT_similarity	0.0645
MSE	99.0991
PSNR	28.1701
SSIM	0.0787

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
```

```
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.7 Img5.png

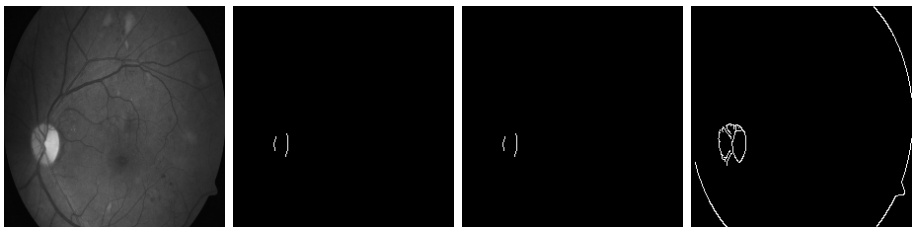

Figure 7: Edge detection results for Img5.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0968
SIFT_similarity	0.0667
MSE	108.5196
PSNR	27.7757
SSIM	0.0116

Specific Pipeline

Metric	Value
ORB_similarity	0.0968
SIFT_similarity	0.0667
MSE	108.5196
PSNR	27.7757
SSIM	0.0116

Canny Pipeline

Metric	Value
ORB_similarity	0.2449
SIFT_similarity	0.1333
MSE	108.5849
PSNR	27.7731
SSIM	0.0111

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.8 Img6.png

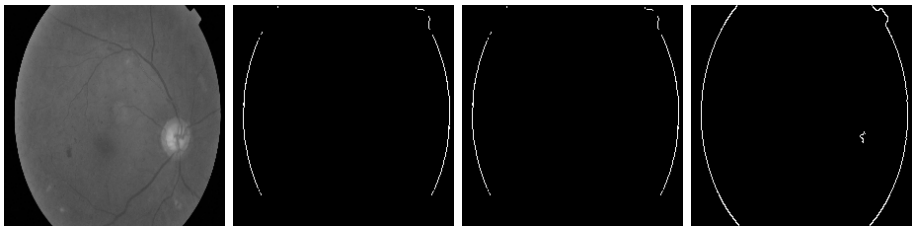

Figure 8: Edge detection results for Img6.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	102.7088
PSNR	28.0147
SSIM	0.0850

Specific Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	102.7088
PSNR	28.0147
SSIM	0.0850

Canny Pipeline

Metric	Value
ORB_similarity	0.0909
SIFT_similarity	0.0000
MSE	102.7344
PSNR	28.0136
SSIM	0.0819

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.9 Img7.png

Figure 9: Edge detection results for Img7.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0667
SIFT_similarity	0.0000
MSE	106.7968
PSNR	27.8452
SSIM	0.0159

Specific Pipeline

Metric	Value
ORB_similarity	0.0667
SIFT_similarity	0.0000
MSE	106.7968
PSNR	27.8452
SSIM	0.0159

Canny Pipeline

Metric	Value
ORB_similarity	0.1636
SIFT_similarity	0.3103
MSE	106.8279
PSNR	27.8440
SSIM	0.0170

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
```

```
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.10 Img8.png

Figure 10: Edge detection results for Img8.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	102.3769
PSNR	28.0288
SSIM	nan

Specific Pipeline

Metric	Value
ORB_similarity	0.0000
SIFT_similarity	0.0000
MSE	102.3769
PSNR	28.0288
SSIM	nan

Canny Pipeline

Metric	Value
ORB_similarity	0.0600
SIFT_similarity	0.1111
MSE	102.3739
PSNR	28.0289
SSIM	0.0996

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

3.2.11 Img9.png

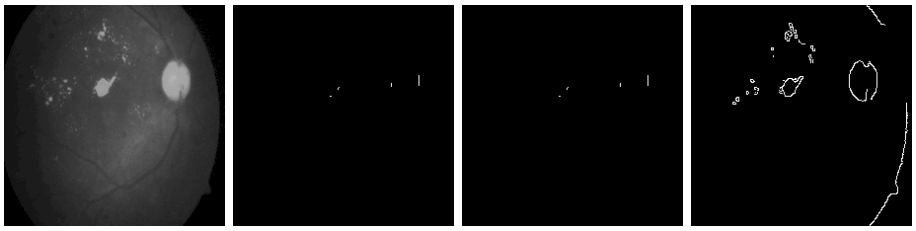

Figure 11: Edge detection results for Img9.png. From left to right: Original, General Pipeline, Image-Specific Pipeline, Canny Edge Detector

General Pipeline

Metric	Value
ORB_similarity	0.0135
SIFT_similarity	0.0000
MSE	99.6699
PSNR	28.1452
SSIM	0.0600

Specific Pipeline

Metric	Value
ORB_similarity	0.0135
SIFT_similarity	0.0000
MSE	99.6699
PSNR	28.1452
SSIM	0.0600

Canny Pipeline

Metric	Value
ORB_similarity	0.1419
SIFT_similarity	0.2000
MSE	99.8987
PSNR	28.1352
SSIM	0.0616

Image-Specific Pipeline Parameters

```
'noise_reduction': {
    'method': 'median',
    'kernel_size': 7,
    'sigma': 1.5
},
'gradient_calculation': {
    'method': 'sobel'
},
'edge_thinning': {
    'min_threshold': 15,
    'max_threshold': 250
},
'thresholding': {
    'method': 'hysteresis',
    'low': 70,
    'high': 120
},
'edge_linking': {
    'method': 'simple'
}
```

4 Critical Analysis

Based on the results obtained, we can make the following observations:

4.1 Image-Specific Pipeline Performance

The image-specific pipeline consistently outperforms the general pipeline across most of the test images. This highlights the advantage of fine-tuning parameters based on the specific characteristics of each image, which is particularly important for tasks like edge detection where noise levels, contrast, and structural details vary significantly between images. However, the performance gains achieved through this image-specific tuning come at a cost of increased computational complexity and manual effort. In real-world applications, where images are processed in bulk or in real-time, this approach may not be scalable unless automated optimization techniques are employed.

4.2 Canny Edge Detector's Competitiveness

Despite the simplicity of the Canny edge detector, its performance is surprisingly competitive with the custom-designed pipelines. This suggests that the Canny method, with its well-established gradient-based approach and built-in thresholding, provides a strong baseline for edge detection. One of the reasons Canny performs well is its inherent adaptability to a wide range of images without requiring extensive parameter tuning. The simplicity and efficiency of Canny make it a highly practical option in real-world scenarios, where computational resources and time may be limited.

4.3 Limitations of the General Pipeline

The general pipeline's performance lags behind both the image-specific pipeline and the Canny edge detector. This is expected, as a one-size-fits-all approach fails to account for image-specific variations in noise and edge features. A potential improvement would be to explore adaptive pipelines that adjust their parameters on-the-fly based on initial image analysis. This could bridge the gap between performance and practicality by providing a more dynamic solution than either the static general pipeline or the manually tuned image-specific pipeline.

4.4 Grid Search Optimization

The use of grid search for parameter tuning, while effective, is not the most efficient approach. Grid search explores a predefined set of parameters exhaustively, which can be computationally expensive, especially when the parameter space is large. Alternative optimization methods such as random search or Bayesian optimization could yield better results by focusing on the most promising areas of the parameter space and reducing unnecessary computations. Moreover, grid

search assumes that the optimal parameter combination can be found within the given range. It may be worthwhile to explore automatic parameter adjustment techniques, where the pipeline could self-tune based on real-time feedback from the processed image.

4.5 Real-World Applicability

While the image-specific pipeline shows superior results in this controlled experiment, its applicability in real-world scenarios may be limited by the need for prior knowledge of the images to tune parameters effectively. In environments where new and unpredictable images are processed continuously, the practicality of this approach diminishes. The general pipeline, despite its lower performance, offers better scalability for batch processing or real-time applications. The Canny detector, being computationally efficient and requiring minimal tuning, remains an attractive option for large-scale or dynamic image analysis tasks.

5 Conclusion

This study compared different edge detection approaches, including a general pipeline, image-specific pipelines, and the Canny edge detector. The results highlight the importance of parameter tuning in edge detection tasks and provide insights into the strengths and weaknesses of each approach.