

# Using Canny Edge Detection : With Default Parameters

Controls

Choose detector and tune parameters

Detection method

Canny

Auto update (real-time)

**Apply**

Detector Parameters

Lower threshold

100



Deploy

Download Processed Image

# Using Canny Edge Detection : With slightly different parameters

Detector Parameters

Lower threshold  
43

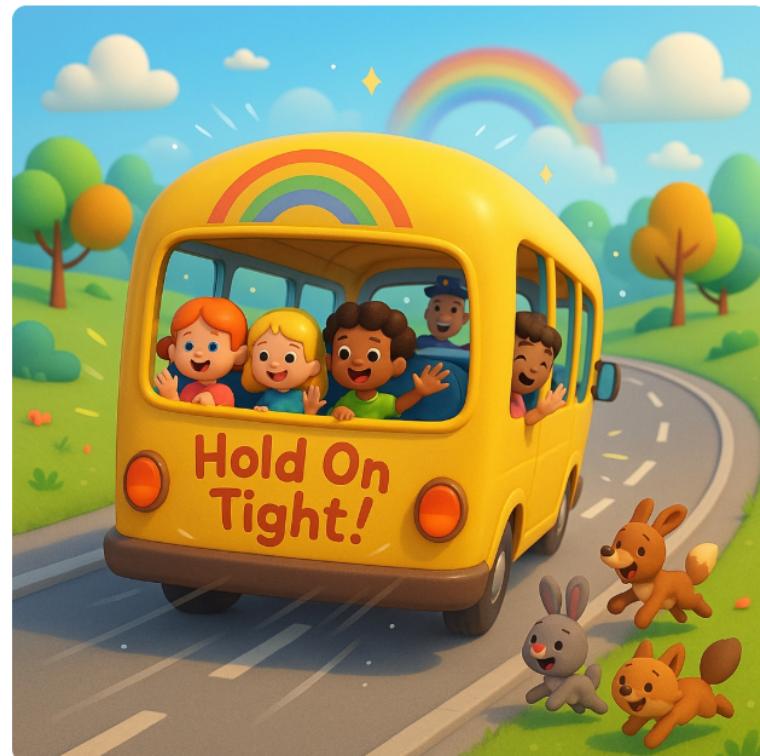
Upper threshold  
84

Gaussian kernel size (odd)  
13

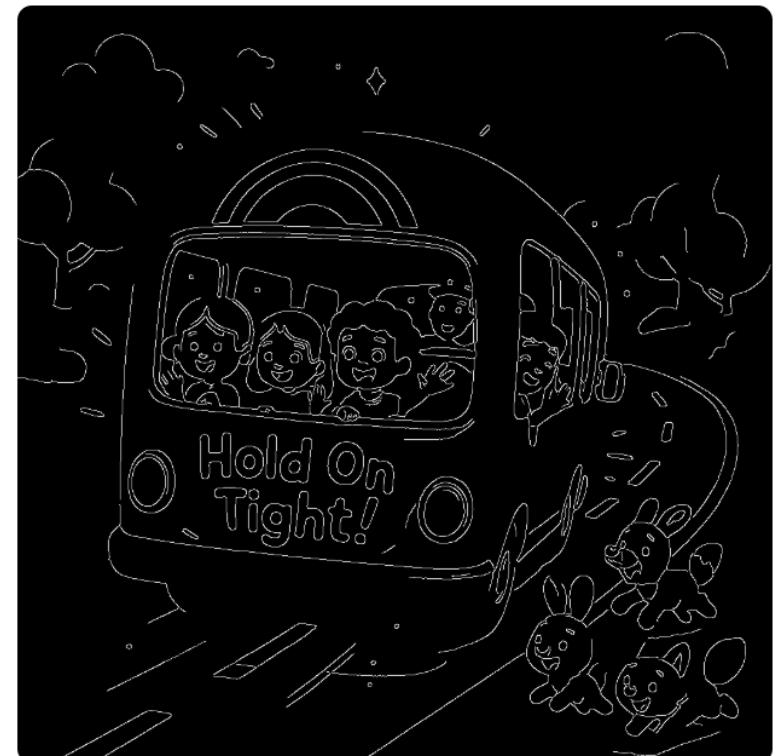
Gaussian sigma  
1.60

**Apply**

**Deploy**



Download Processed Image



## Using Canny Edge Detection : With slightly different parameters

Apply      Deploy      :

Detector Parameters

Lower threshold  
21

Upper threshold  
57

Gaussian kernel size (odd)  
5

Gaussian sigma  
7.10



Download Processed Image

## Using Canny Edge Detection : With slightly different parameters

Apply      Deploy      :

Detector Parameters

Lower threshold  
21

Upper threshold  
99

Gaussian kernel size (odd)  
7

Gaussian sigma  
0.00



Download Processed Image

# Using Laplacian Edge Detection : With default parameters

« Deploy

**Controls**

Choose detector and tune parameters

Detection method

Laplacian

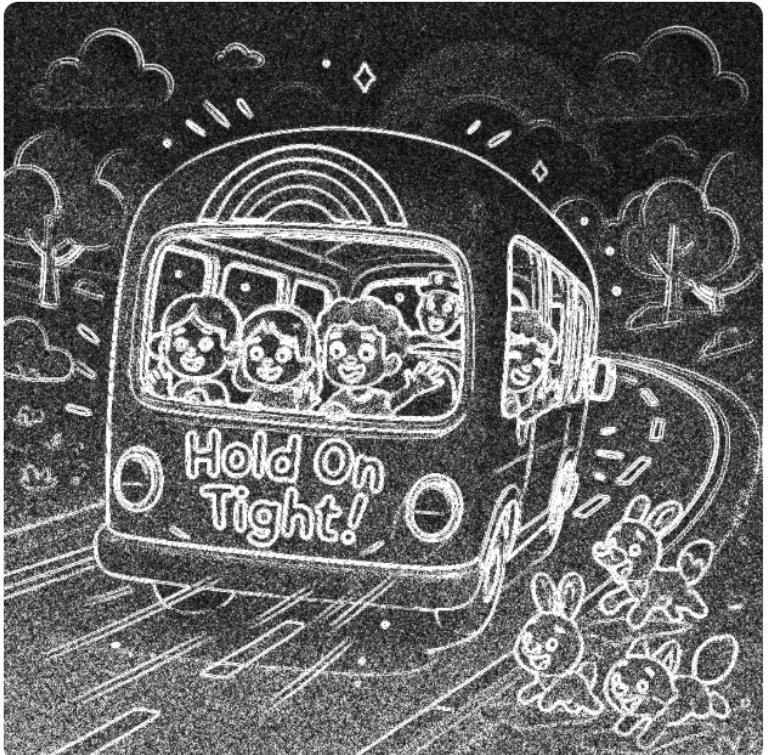
Auto update (real-time)

**Apply**

Detector Parameters

Kernel size (odd)

5



Download Processed Image

## Using Laplacian Edge Detection : With slightly different parameters

Detector Parameters

Kernel size (odd)

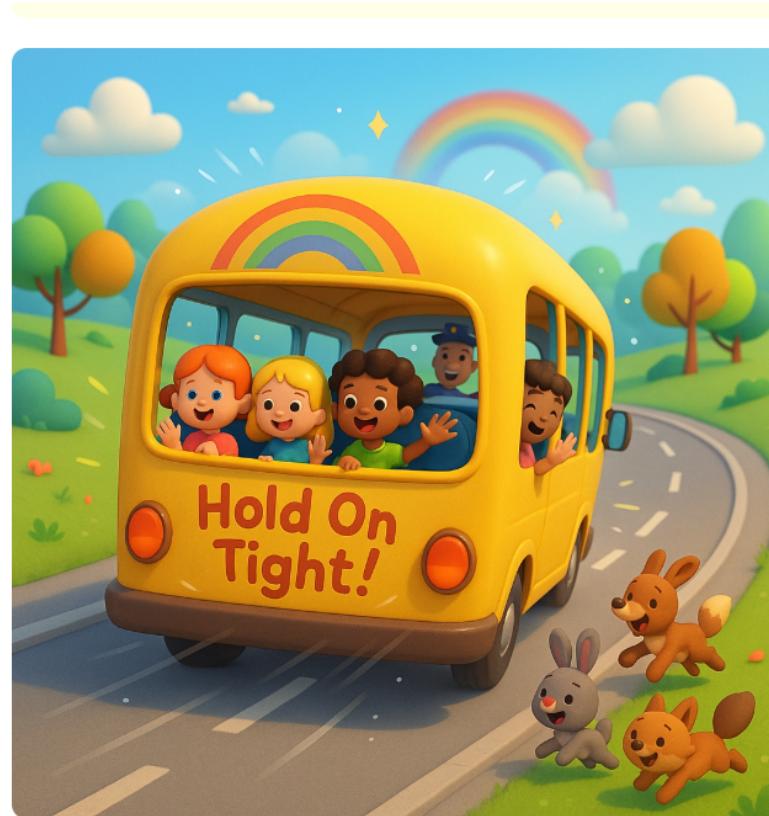
3

Scale

2.40

Output

Invert output (white edges on black)



Deploy

## Using Laplacian Edge Detection : With slightly different parameters

Detector Parameters

Kernel size (odd)  5

Scale  2.40

Output

Invert output (white edges on black)

Deploy

Download Processed Image

## Using Laplacian Edge Detection : With slightly different parameters

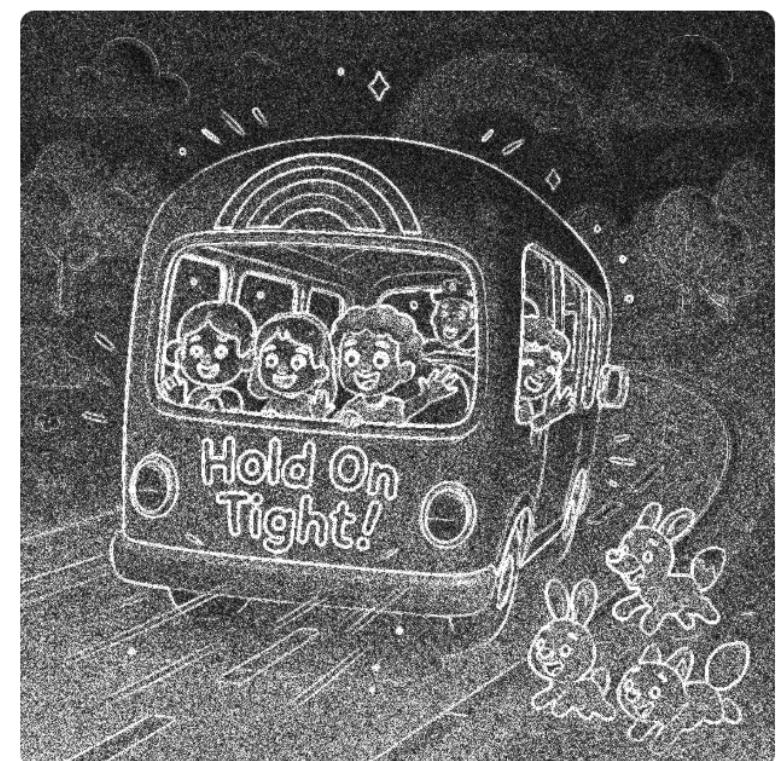
Detector Parameters

Kernel size (odd)  
3

Scale  
4.50

Output

Invert output (white edges on black)



[Download Processed Image](#)

Deploy

## Using Laplacian Edge Detection : With slightly different parameters

Detector Parameters

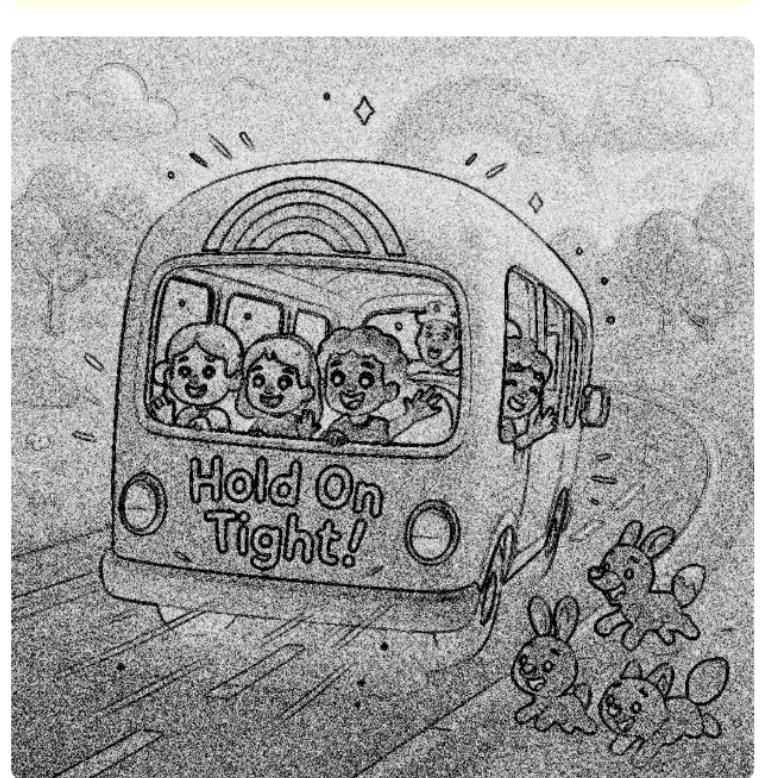
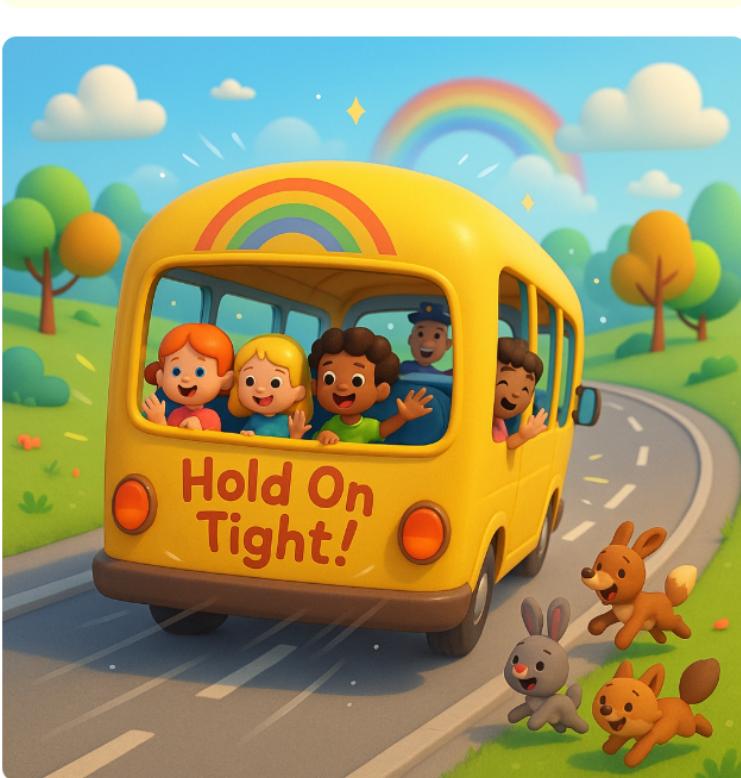
Kernel size (odd)  
3

Scale  
4.50

Output

Invert output (white edges on black)

Deploy ⋮



Download Processed Image

# Using Sobel Edge Detection : In X-Direction

**Controls**

Choose detector and tune parameters

Detection method

Sobel

Auto update (real-time)

**Apply**

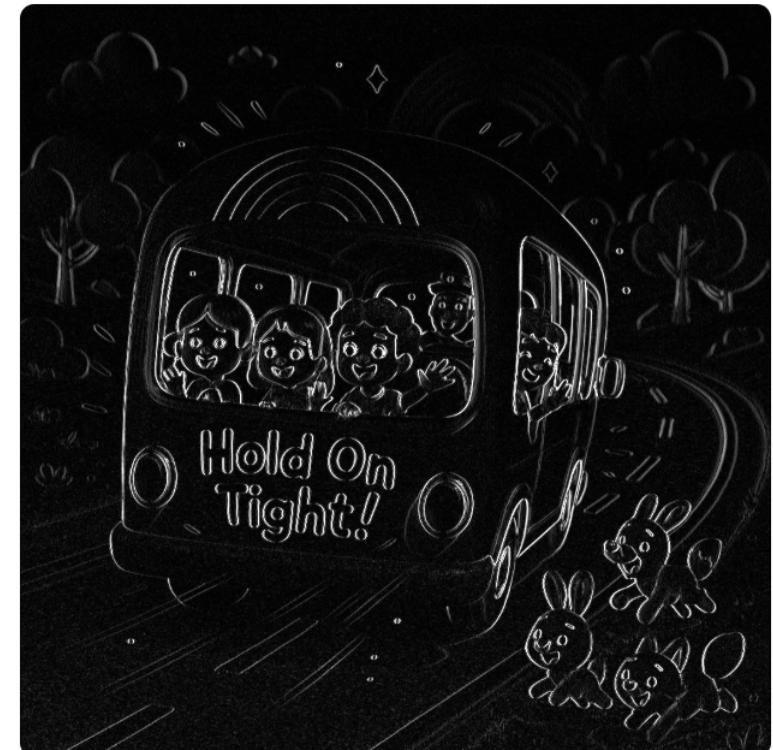
Detector Parameters

Kernel size (odd)

3



Download Processed Image



Deploy

## Using Sobel Edge Detection : In X-Direction with different kernel size

Auto update (real-time) Deploy ⋮

Detector Parameters

Kernel size (odd)

Gradient X ?

Gradient Y ?

Combine X & Y magnitude

Apply



Download Processed Image

## Using Sobel Edge Detection : In Y-Direction

Auto update (real-time) Deploy ⋮

**Apply**

Detector Parameters

Kernel size (odd)

Gradient X?

Gradient Y?

Combine X & Y magnitude



[Download Processed Image](#)

## Using Sobel Edge Detection : In Y-Direction with diff kernel size

Auto update (real-time)  Deploy ⋮

**Apply**

Detector Parameters

Kernel size (odd)  
3

Gradient X ?  
0

Gradient Y ?  
1

Combine X & Y magnitude

Download Processed Image

# Using Sobel Edge Detection : In Both XY-Direction

Auto update (real-time)  Deploy ⋮

**Apply**

Detector Parameters

Kernel size (odd)  
3

Gradient X ?  
1

Gradient Y ?  
1

Combine X & Y magnitude



Download Processed Image

## Using Sobel Edge Detection : In Both XY-Direction with diff kernel size

Auto update (real-time)  Deploy ⋮

Apply

Detector Parameters

Kernel size (odd)

Gradient X?

Gradient Y?

Combine X & Y magnitude



Download Processed Image