```
# Install CUDA toolkit (nvcc) on Colab
!apt update
!apt install -y nvidia-cuda-toolkit
!nvcc --version  # Verify nvcc installation
```

```
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2,041 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcupti11.5 amd64 11.5.114~11.5.1-1ubuntu1 [7,696 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libaccinj64-11.5 amd64 11.5.114~11.5.1-1ubuntu1 [845 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxtst6 amd64 2:1.2.3-1build4 [13.4 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxxf86dga1 amd64 2:1.1.5-0ubuntu3 [12.6 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 x11-utils amd64 7.7+5build2 [206 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcub-dev all 1.15.0-3 [217 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcublaslt11 amd64 11.7.4.6~11.5.1-1ubuntu1 [148 MB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcublas11 amd64 11.7.4.6~11.5.1-1ubuntu1 [78.2 MB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcudart11.0 amd64 11.5.117~11.5.1-1ubuntu1 [178 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcufft10 amd64 11.1.1+~10.6.0.107~11.5.1-1ubuntu1 [70.4 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcufftw10 amd64 11.1.1+~10.6.0.107~11.5.1-1ubuntu1 [211 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 libnvidia-compute-510 amd64 525.147.05-0ubuntu2.22.04.1 [7,310
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 libnvidia-compute-495 amd64 510.108.03-0ubuntu0.22.04.1 [7,378
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcuinj64-11.5 amd64 11.5.114~11.5.1-1ubuntu1 [1,004 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcurand10 amd64 11.1.1+~10.2.7.107~11.5.1-1ubuntu1 [41.8 MB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcusolver11 amd64 11.3.2.107~11.5.1-1ubuntu1 [31.3 MB]
Get:23 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcusolvermg11 amd64 11.3.2.107~11.5.1-1ubuntu1 [17.8 MB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libcusparse11 amd64 11.7.0.107~11.5.1-1ubuntu1 [96.2 MB]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libdebuginfod1 amd64 0.186-1ubuntu0.1 [12.8 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 libglx-dev amd64 1.4.0-1 [14.1 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgl-dev amd64 1.4.0-1 [101 kB]
Get:28 http://archive.ubuntu.com/ubuntu jammy/main amd64 libegl-dev amd64 1.4.0-1 [18.0 kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgtk2.0-common all 2.24.33-2ubuntu2.1 [125 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgtk2.0-0 amd64 2.24.33-2ubuntu2.1 [2,038 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgail18 amd64 2.24.33-2ubuntu2.1 [15.9 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgail-common amd64 2.24.33-2ubuntu2.1 [132 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgles1 amd64 1.4.0-1 [11.5 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgles-dev amd64 1.4.0-1 [49.4 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgtk2.0-bin amd64 2.24.33-2ubuntu2.1 [7,936 B]
Get:36 http://archive.ubuntu.com/ubuntu jammy/main amd64 libipt2 amd64 2.0.5-1 [46.4 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppc11 amd64 11.5.1.107~11.5.1-1ubuntu1 [430 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppial11 amd64 11.5.1.107~11.5.1-1ubuntu1 [5,234 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppicc11 amd64 11.5.1.107~11.5.1-1ubuntu1 [2,373 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppidei11 amd64 11.5.1.107~11.5.1-1ubuntu1 [2,587 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppif11 amd64 11.5.1.107~11.5.1-1ubuntu1 [33.8 MB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppig11 amd64 11.5.1.107~11.5.1-1ubuntu1 [14.5 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppim11 amd64 11.5.1.107~11.5.1-1ubuntu1 [3,037 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppist11 amd64 11.5.1.107~11.5.1-1ubuntu1 [13.7 MB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppisu11 amd64 11.5.1.107~11.5.1-1ubuntu1 [177 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnppitc11 amd64 11.5.1.107~11.5.1-1ubuntu1 [1,292 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnpps11 amd64 11.5.1.107~11.5.1-1ubuntu1 [7,116 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvblas11 amd64 11.7.4.6~11.5.1-1ubuntu1 [191 kB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvidia-ml-dev amd64 11.5.50~11.5.1-1ubuntu1 [69.1 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvjpeg11 amd64 11.5.4.107~11.5.1-1ubuntu1 [1,858 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvrtc-builtins11.5 amd64 11.5.119~11.5.1-1ubuntu1 [116 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvrtc11.2 amd64 11.5.119~11.5.1-1ubuntu1 [15.7 MB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 libnvvm4 amd64 11.5.119~11.5.1-1ubuntu1 [8,675 kB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/main amd64 libopengl-dev amd64 1.4.0-1 [3,400 B]
Get:55 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 librsvg2-common amd64 2.52.5+dfsg-3ubuntu0.2 [17.7 kB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/main amd64 libsource-highlight-common all 3.1.9-4.1build2 [64.5 kB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/main amd64 libsource-highlight4v5 amd64 3.1.9-4.1build2 [207 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/main amd64 libvdpau-dev amd64 1.4-3build2 [38.7 kB]
Get:59 http://archive.ubuntu.com/ubuntu jammy/universe amd64 node-html5shiv all 3.7.3+dfsg-4 [13.6 kB]
Get:60 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 nvidia-cuda-toolkit-doc all 11.5.1-1ubuntu1 [6,263 kB]
Get:61 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-8-jre-headless amd64 8u452-ga~us1-0ubuntu1~22.04 [30.8 MB
```

```
%%writefile project.cu
#include <stdlib.h>
#include <cuda_runtime.h>
#include <math.h>
#include <stdio.h>
#pragma pack(push, 1)
typedef struct {
    unsigned short bfType;
    unsigned int bfSize;
    unsigned short bfReserved1;
    unsigned short bfReserved2;
    unsigned int bfOffBits;
} BITMAPFILEHEADER;

typedef struct {
    unsigned int biSize;
```

What can I help you build?

```
        int biWidth;
        int biHeight;
        unsigned short biPlanes;
        unsigned short biBitCount;
        unsigned int biCompression;
        unsigned int biSizeImage;
        int biXPelsPerMeter;
        int biYPelsPerMeter;
        unsigned int biClrUsed;
        unsigned int biClrImportant;
    } BITMAPINFOHEADER;
    #pragma pack(pop)

    unsigned char* loadBMP(const char* filename, int *width, int *height, int *rowSize) {
        FILE* f = fopen(filename, "rb");
        if (!f) return NULL;

        BITMAPFILEHEADER fileHeader;
        fread(&fileHeader, sizeof(fileHeader), 1, f);
        if (fileHeader.bfType != 0x4D42) {
            fclose(f);
            return NULL;
        }

        BITMAPINFOHEADER infoHeader;
        fread(&infoHeader, sizeof(infoHeader), 1, f);
        if (infoHeader.biBitCount != 24) {
            fclose(f);
            return NULL;
        }

        *width = infoHeader.biWidth;
        *height = infoHeader.biHeight;
        *rowSize = (*width * 3 + 3) & (~3);

        fseek(f, fileHeader.bfOffBits, SEEK_SET);
        unsigned char* data = (unsigned char*)malloc(*rowSize * (*height));
        fread(data, 1, *rowSize * (*height), f);
        fclose(f);
        return data;
    }

    void writeBMP(const char* filename, unsigned char* data, int width, int height) {
        FILE* f = fopen(filename, "wb");
        if (!f) return;

        int rowSize = (width * 3 + 3) & (~3);
        int dataSize = rowSize * height;

        BITMAPFILEHEADER fileHeader = {0};
        BITMAPINFOHEADER infoHeader = {0};

        fileHeader.bfType = 0x4D42;
        fileHeader.bfSize = sizeof(fileHeader) + sizeof(infoHeader) + dataSize;
        fileHeader.bfOffBits = sizeof(fileHeader) + sizeof(infoHeader);

        infoHeader.biSize = sizeof(infoHeader);
        infoHeader.biWidth = width;
        infoHeader.biHeight = height;
        infoHeader.biPlanes = 1;
        infoHeader.biBitCount = 24;
        infoHeader.biSizeImage = dataSize;

        fwrite(&fileHeader, sizeof(fileHeader), 1, f);
        fwrite(&infoHeader, sizeof(infoHeader), 1, f);
        fwrite(data, 1, dataSize, f);
        fclose(f);
    }

    __global__ void maskKernel(unsigned char* gray, unsigned char* result, int width, int height) {
        int x = blockIdx.x * blockDim.x + threadIdx.x;
        int y = blockIdx.y * blockDim.y + threadIdx.y;
        int mask[3][3] = {
            {-1, -1, -1},
            {-1,  8, -1},
            {-1, -1, -1}
        };
    if (x > 0 && y > 0 && x < width - 1 && y < height - 1) {
        int sum = 0;
        for (int j = -1; j <= 1; j++)
            for (int i = -1; i <= 1; i++)
                sum += gray[(y+j)*width + (x+i)] * mask[j+1][i+1];

        sum = abs(sum);
        if (sum > 255) sum = 255;
        result[y * width + x] = (unsigned char)sum;
    }
}

void rgbToGray(unsigned char* img, unsigned char* gray, int width, int height, int rowSize) {
    for (int y = 0; y < height; y++) {
        unsigned char* row = img + y * rowSize;
        for (int x = 0; x < width; x++) {
            unsigned char b = row[3*x + 0];
            unsigned char g = row[3*x + 1];
            unsigned char r = row[3*x + 2];
            gray[y * width + x] = (unsigned char)(0.299 * r + 0.587 * g + 0.114 * b);
        }
    }
}

void grayToRGB(unsigned char* gray, unsigned char* img, int width, int height, int rowSize) {
    for (int y = 0; y < height; y++) {
        unsigned char* row = img + y * rowSize;
        for (int x = 0; x < width; x++) {
            unsigned char val = gray[y * width + x];
            row[3*x + 0] = val;
            row[3*x + 1] = val;
            row[3*x + 2] = val;
        }
    }
}

int main(int argc, char** argv) {
    if (argc < 3) {
        printf("Usage: ./object_detect input.bmp output.bmp\n");
        return 1;
    }

    int width, height, rowSize;
    unsigned char* img = loadBMP(argv[1], &width, &height, &rowSize);
    if (!img) {
        printf("Failed to load image or unsupported format\n");
        return 1;
    }

    int size = width * height;
    unsigned char* gray = (unsigned char*)malloc(size);
    unsigned char* result = (unsigned char*)malloc(size);

    rgbToGray(img, gray, width, height, rowSize);

    unsigned char *d_gray, *d_result;
    cudaMalloc((void**)&d_gray, size);
    cudaMalloc((void**)&d_result, size);
    cudaMemcpy(d_gray, gray, size, cudaMemcpyHostToDevice);

    dim3 block(16, 16);
    dim3 grid((width + 15)/16, (height + 15)/16);
    maskKernel<<<grid, block>>>(d_gray, d_result, width, height);
    cudaDeviceSynchronize();

    cudaMemcpy(result, d_result, size, cudaMemcpyDeviceToHost);

    // Optional: Print first 10x10 matrix for inspection
    printf("--- Object Detection Output Matrix (Top-left 10x10) ---\n");
    for (int y = 0; y < 10 && y < height; y++) {
        for (int x = 0; x < 10 && x < width; x++) {
            printf("%3d ", result[y * width + x]);
        }
        printf("\n");
    }
    grayToRGB(result, img, width, height, rowSize);
    writeBMP(argv[2], img, width, height);

    cudaFree(d_gray);
    cudaFree(d_result);
    free(img);
    free(gray);
    free(result);

    printf("Object detection completed. Output saved to %s\n", argv[2]);
    return 0;
}
```

```
Overwriting project.cu
```

```
!nvcc -arch=sm_75 -o object_detect project.cu
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Cat4.jpg

- **Cat4.jpg**(image/jpeg) - 178795 bytes, last modified: 7/1/2025 - 100% done

```
Saving Cat4.jpg to Cat4 (1).jpg
```

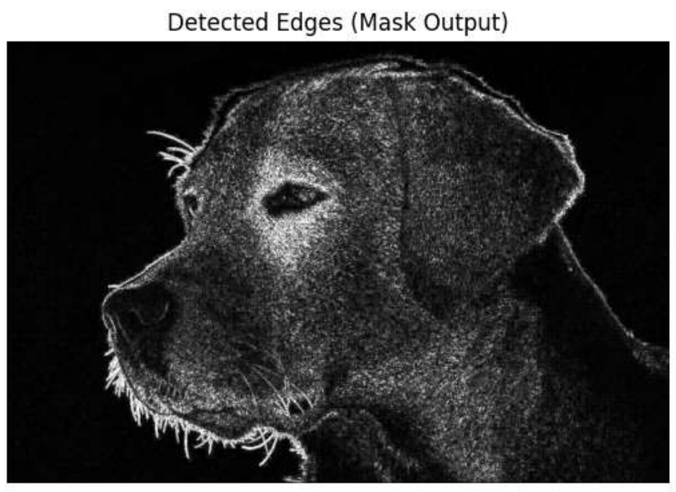
```
from PIL import Image

# Replace with the correct uploaded filename
Image.open("Cat4 (1).jpg").convert("RGB").save("input.bmp")
```

```
!ls -lh output.bmp
```

```
-rw-r--r-- 1 root root 2.8M Jul  2 23:48 output.bmp
```

```
# Step 5: Visualize
from PIL import Image
import matplotlib.pyplot as plt
img = Image.open("output.bmp").convert("L")
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.title("Detected Edges (Mask Output)")
plt.show()
```



```
!./object_detect input.bmp output.bmp
```

```
--- Object Detection Output Matrix (Top-left 10x10) ---
   0   0   0   0   0   0   0   0   0   0
   0   1   3   3   1   2   4   4   3   4
   0   2   3   3   0   0   1   4   2   2
   0   2   4   1   2   3   3   1   5   4
   0   1   4   1   5   3   3   8   7   3
   0   1   7   1   5   1   2   6  10   1
   0   8   8   4   2   3   5  11   7   5
   0   0   6   1   1   6   0  10   6   3
   0   2   1   2   2   1   2   9   6   2
   0   2   1   3   3   1   2   9   7   3
Object detection completed. Output saved to output.bmp
```

```
from google.colab import files
files.download("output.bmp")
```