```cpp
#include <iostream>

#include <opencv2/core.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/opencv.hpp>

using namespace cv;

void filter(Mat input, Mat& result) {
  Size s = input.size();
  long h, w;
  long sum;

  std::cout << "Input type was : " << input.type() << std::endl;
  uint8_t out[s.height][s.width];

  std::cout << s.height << " " << s.width << std::endl;

  for (w=0; w<s.width; w++){
    for(h=0; h<s.height; h++){
      sum = 0;
      std::vector<uint8_t> median;

      for (int _x = -1; _x < 2; ++_x)
      {
        for (int _y = -1; _y < 2; ++_y)
        {
          int idx_y = h + _y;
          int idx_x = w + _x;

          if (idx_x < 0 || idx_x > s.width)
            break;

          if (idx_y < 0 || idx_y > s.height)
            break;

          median.push_back(input.at<uint8_t>(idx_y,idx_x));
        }
      }
      std::sort(std::begin(median), std::end(median));

      for (auto it = median.begin(); it != median.end(); ++it) {
        sum = median.at(median.size()/2);
      }
      out[h][w]=(uint8_t)sum;
    }
  }
  result = Mat(s.height, s.width, CV_8U, out); //or maybe CV_8UC1?
  Size _s = result.size();
  std::cout << "done: " << _s.height << " " << _s.width << std::endl;
}


int main() {
  // Read the image (in BGR)
    Mat img = imread("fordgt_test.png", IMREAD_COLOR);
    if(img.empty())
```

```
    {
        std::cout << "Could not read the image: " << std::endl;
        return 1;
    }

    // Split the image into 3 new images for blue, green and red.
    std::cout << "Splitting channels: " << std::endl;
  Mat bands[3];
  split(img, bands);


  Mat bandsFiltered[3];
  filter(bands[0],bandsFiltered[0]);
  filter(bands[1],bandsFiltered[1]);
  filter(bands[2],bandsFiltered[2]);

    // Display the image until q is pressed
    std::cout << "Displaying result: " << std::endl;
    imshow("Display window", bands[0]);
    waitKey(0); // Wait for a keystroke in the window
    imshow("Display window", bands[1]);
    waitKey(0); // Wait for a keystroke in the window
    imshow("Display window", bands[2]);
    waitKey(0); // Wait for a keystroke in the window
    imshow("Display window", bandsFiltered[0]);
    waitKey(0); // Wait for a keystroke in the window
    imshow("Display window", bandsFiltered[1]);
    waitKey(0); // Wait for a keystroke in the window
    imshow("Display window", bandsFiltered[2]);
    waitKey(0); // Wait for a keystroke in the window
    return 0;
}

```