OpenCV

faq tags users badges

Hi there! Please sign in help

ALL UNANSWERED Search or ask your questic ASK YOUR QUESTION

0

How to remove small black spots & make digits more clear, complete & sharp in image?

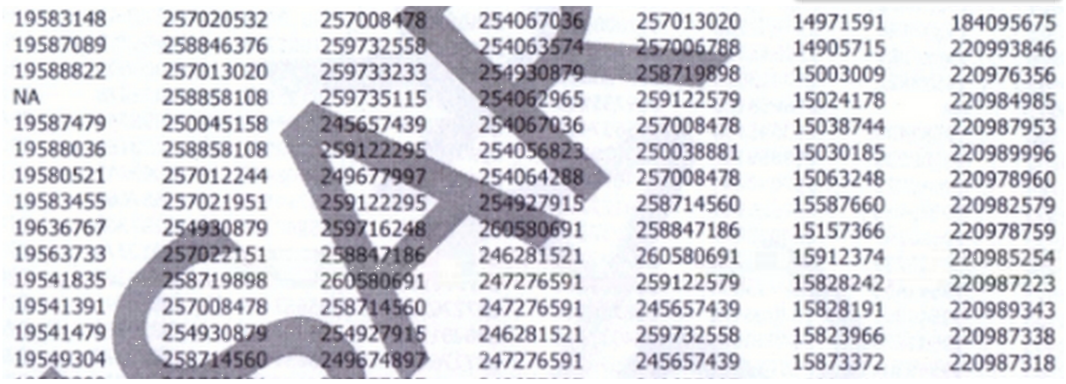
c++ opencv

asked **Jul 22 '19**
Bhavtosh
1 ●1 ●1

updated **Jul 24 '19**
Witek
1146 ●1 ●9 ●19

PS: Numbers are Enrollment IDs that are the combination of course + class + other ids etc

Im trying to write a program to remove a logo from image, clean it before sending it to Ocr program. Here is the input image:



19583148	257020532	257008478	254067036	257013020	14971591	184095675
19587089	258846376	259732558	254063574	257006788	14905715	220993846
19588822	257013020	259733233	254930879	258719898	15003009	220976356
NA	258858108	259735115	254062965	259122579	15024178	220984985
19587479	250045158	245657439	254067036	257008478	15038744	220987953
19588036	258858108	259122295	254056823	250038881	15030185	220989996
19580521	257012244	249677997	254064288	257008478	15063248	220978960
19583455	257021951	259122295	254927915	258714560	15587660	220982579
19636767	254930879	259716248	260580691	258847186	15157366	220978759
19563733	257022151	258847186	246281521	260580691	15912374	220985254
19541835	258719898	260580691	247276591	259122579	15828242	220987223
19541391	257008478	258714560	247276591	245657439	15828191	220989343
19541479	254930879	254927915	246281521	259732558	15823966	220987338
19549304	258714560	249674897	247276591	245657439	15873372	220987318

Im totally new to code in Opencv & C++, I googled and merged below code from various sources so far, It reads a b/w image and removes a mark from it.

```
im = imread(fpath + ".jpg", IMREAD_GRAYSCALE);
// 1. make a copy & approximate the background
bg = im.clone();
// get the structure & apply morphology
kernel2 = getStructuringElement(MORPH_RECT, Size(2 * 5 + 1, 2 * 5 + 1));
morphologyEx(im, bg, CV_MOP_CLOSE, kernel2);
// threshold the difference image
threshold(dif, bw, 0, 255, CV_THRESH_BINARY_INV | CV_THRESH_OTSU);
// threshold the background image so we get dark region
threshold(bg, dark, 0, 255, CV_THRESH_BINARY_INV | CV_THRESH_OTSU);
// extract pixels in the dark region
vector<unsigned char>darkpix(countNonZero(dark));
int index = 0;
for (int r = 0; r < dark.rows; r++)
{
    for (int c = 0; c < dark.cols; c++)
    {
        if (dark.at<unsigned char>(r, c))
        {
            darkpix[index++] = im.at<unsigned char>(r, c);
        }
    }
}
// threshold the dark region so we get the darker pixels inside it
threshold(darkpix, darkpix, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
// paste the extracted darker pixels
index = 0;
for (int r = 0; r < dark.rows; r++)
{
```

Links

- Official site
- GitHub
- Wiki
- Documentation

Question Tools

Follow

1 follower

subscribe to rss feed

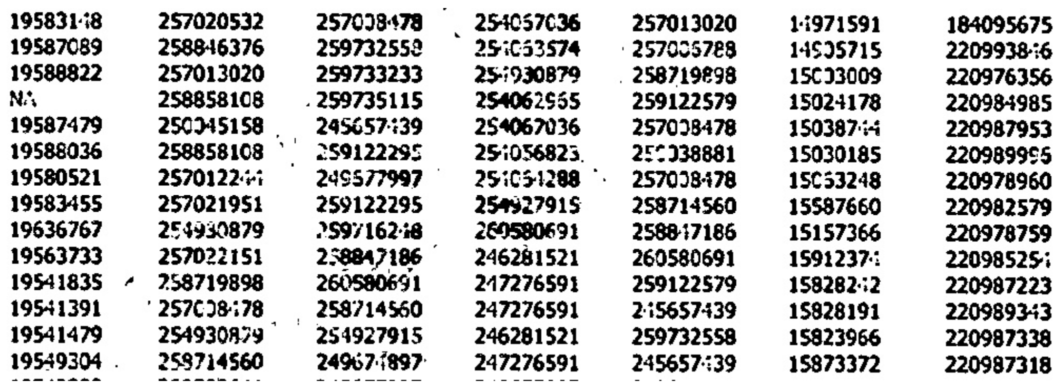
Stats

Asked:	Jul 22 '19
Seen:	1,134 times
Last updated:	Jul 24 '19

Related questions

- OpenCV DescriptorMatcher matches
- Conversion between IplImage and MxArray
- Video On Label OpenCV Qt :: hide cvNamedWindows
- Problems with OpenCV DFT function in C++
- Problems using the math.h class with OpenCV (c++, VS2012)
- How to reduce false positives for face detection
- Area of a single pixel object in OpenCV
- build problems for android_binary_package - Eclipse Indigo, Ubuntu 12.04
- OpenCV for Android (2.4.2): OpenCV Loader imports not resolved
- Can't compile .cu file when including opencv.hpp

```
        for (int c = 0; c < dark.cols; c++)
        {
            if (dark.at<unsigned char>(r, c))
            {
                bw.at<unsigned char>(r, c) = darkpix[index++];
            }
        }
}
// Clean image to make more readable and clear
adaptiveThreshold(bw, dst, 75, CV_ADAPTIVE_THRESH_MEAN_C, CV_THRESH_BINARY, 3, -15);
image_out = bw - dst;
imshow("Final", image_out);
```



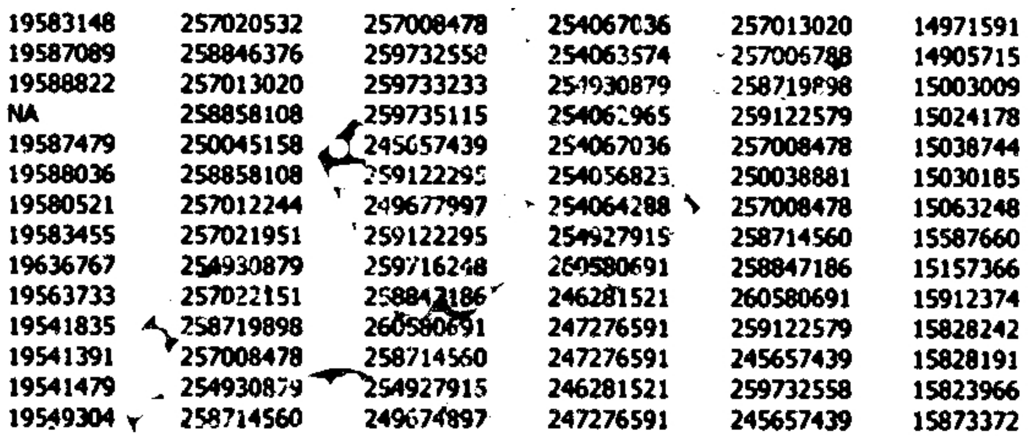
19583148	257020532	257008478	254067036	257013020	14971591	184095675
19587089	258846376	259732558	254063574	257006788	14905715	220993846
19588822	257013020	259733233	254930879	258719898	15003009	220976356
NA	258858108	259735115	254062965	259122579	15024178	220984985
19587479	250045158	245657439	254067036	257008478	15038744	220987953
19588036	258858108	259122295	254056823	250038881	15030185	220989995
19580521	257012244	249577997	254064288	257008478	15063248	220978960
19583455	257021951	259122295	254927915	258714560	15587660	220982579
19636767	254930879	259716248	260580691	258847186	15157366	220978759
19563733	257022151	258847186	246281521	260580691	15912374	22098525[illegible]
19541835	258719898	260580691	247276591	259122579	15828242	220987223
19541391	257008478	258714560	247276591	245657439	15828191	220989343
19541479	254930879	254927915	246281521	259732558	15823966	220987338
19549304	258714560	249674897	247276591	245657439	15873372	220987318

With the above output image, now Im badly stuck with 2 open queries:

1. Remove small black spots (black color) & noise
2. And make numbers, for example 0, 2, 6, 8, 9 etc & text NA, more complete, sharp & readable

Please advise & suggest, thanks a lot...

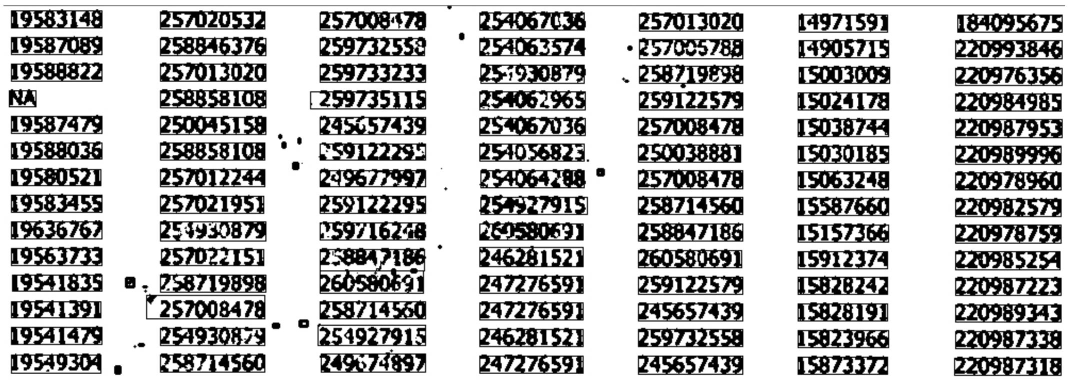
Im very new to Opencv but somehow Im able to load and display image, please refer this image for more clarity on what I mean:



19583148	257020532	257008478	254067036	257013020	14971591
19587089	258846376	259732558	254063574	257006788	14905715
19588822	257013020	259733233	254930879	258719898	15003009
NA	258858108	259735115	254062965	259122579	15024178
19587479	250045158	245657439	254067036	257008478	15038744
19588036	258858108	259122295	254056823	250038881	15030185
19580521	257012244	249677997	254064288	257008478	15063248
19583455	257021951	259122295	254927915	258714560	15587660
19636767	254930879	259716248	260580691	258847186	15157366
19563733	257022151	258842186	246281521	260580691	15912374
19541835	258719898	260580691	247276591	259122579	15828242
19541391	257008478	258714560	247276591	245657439	15828191
19541479	254930879	254927915	246281521	259732558	15823966
19549304	258714560	249674897	247276591	245657439	15873372

Going ahead with the help of getStructuringElement, morphologyEx & threshold functions, I could try little image processing.

The last sample code I tried before posting this query was boundingRect and findContours & below is what i got in result, it looks like some rectangles are including spots & noise AND I dont know how to move ahead from here...



19583148	257020532	257008478	254067036	257013020	14971591	184095675
19587089	258846376	259732558	254063574	257005788	14905715	220993846
19588822	257013020	259733233	254930879	258719898	15003009	220976356
NA	258858108	259735115	254062965	259122579	15024178	220984985
19587479	250045158	245657439	254067036	257008478	15038744	220987953
19588036	258858108	259122295	254056823	250038881	15030185	220989996
19580521	257012244	249677997	254064288	257008478	15063248	220978960
19583455	257021951	259122295	254927915	258714560	15587660	220982579
19636767	254930879	259716248	260580691	258847186	15157366	220978759
19563733	257022151	258847186	246281521	260580691	15912374	220985254
19541835	258719898	260580691	247276591	259122579	15828242	220987223
19541391	257008478	258714560	247276591	245657439	15828191	220989343
19541479	254930879	254927915	246281521	259732558	15823966	220987338
19549304	258714560	249674897	247276591	245657439	15873372	220987318

Im badly stuck on below issues: 1. Clean black spots, noise and anything else except ID numbers 2. Make numbers like 0, 2, 6, 8, 9 more thick, complete and sharp

Kindly help out with some code sample, link or article which can help me.

Thanks in advance...

Comments

2 I merged your two questions into one.
The last image looks promising. How about removing small spots based on their contour area, axis length and/or axis ratio? Can you increase resolution of the input image?
Witek (**Jul 24 '19**)

1 Hi Witek, thanks for merging and now post has more info.... I confess, being new to opencv its hard for me to understand either code-samples or try any code myself but im trying to sort it out...
To answer yr query, increasing resolution of input image will dither the digits more but in what way inout resolution will help when it is good enough to create this output image?
Somehow I just want to (1). clear these spots and (2). make digits more clear and complete.
Bhavtosh (**Jul 25 '19**)

1 Higher resolution might make the digits easier to separate - they will not blend together so much. Also it should improve thresholded results as transitions between black and white will be smoother. This should improve the shape of the digits,which could be important for OCR at the later stage. I played a little with your problem, and I must admit, I was not able to get a clear image. It is not going to be easy, especially if the big gray watermark is going to be different (brighter or darker) in different images. Perhaps using simple template matching would be easier? That would also solve the OCR problem.
Witek (**Jul 25 '19**)

1 Making a image patch template will be another added task here and that too with so many numbers :) Right now watermark is not a problem because it is removed but yes left few spots and took out edges of few numbers.... Is it not possible to play with contours in some more ways?
Bhavtosh (**Jul 25 '19**)

1 Since you enclosed the numbers in rectangles, you already removed the small spots, I guess? Increasing the resolution should help you achieve point 2.
Witek (**Jul 26 '19**)

1 I only used rects just to help others understand & visualize the problem, yes i agree with yr 2nd suggestion, zooming the image a bit does help in clarity BUT finding and filling the edges still to be solved + these spots also
Bhavtosh (**Jul 26 '19**)

1 You can remove the spots by their properties, that is find contours and their bounding rectangle and remove all these contours that are singular - ones that do not have a neighbor of similar size in close vicinity. This not likely to work in 100% cases as there always might be a spot that is similar to a number in terms of bounding box size and position AND some numbers might be broken into two small parts that could be treated as spots.
Witek (**Jul 26 '19**)

1 In my opinion it will be extremely difficult to design an algorithm based on thresholding and morphology that will provide 100% accurate results. There is a very thin line between filling edges and removing spots - these are opposite requirements - on one head you want to enhance small, thin lines of numbers and on the other you want to remove small spots that are noise. I don't think it is possible to separate these two classes as they overlap sometimes. I will repeat my suggestion of using template matching that might solve all your problems. It is quite easy and quick to try. Or perhaps use a deep network like Yolo - this will require much more effort to deploy, but might work better and faster.
Witek (**Jul 26 '19**)

I will explore Yolo, thanks.
Bhavtosh (**Jul 26 '19**)

add a comment

Login/Signup to Answer

about | faq | help | privacy policy | terms of service
Powered by Askbot version 0.10.2