

Работа 5. Исправление проективных искажений изображений страниц

автор: Лоев В.А.

url: https://mysvn.ru/LOEV V A/loev v a/prj.labs/lab_5/

Задание

1. Нарисовать эталонную "разметку" (четырехугольник изображения страницы) для отсканированного изображения и фотографий (не менее 5 любых).
2. Изготовить эталонную бинаризацию для скана.
3. Запрограммировать совмещение изображений при помощи гомографии с использованием эталонной геометрической разметки.
4. Используя эталонную геометрическую разметку реализовать численную оценку качества бинаризации (из лабораторной 4) для совмещенных изображений (для отдельных изображений и для набора).
5. Реализовать представление результатов выполнения лабораторной (иллюстрации, таблицы, графики и т.д.) и вставить в отчет.
6. Произвести уточнение параметров бинаризации (улучшение качества для набора изображений) и продемонстрировать результаты после уточнения.

Результаты

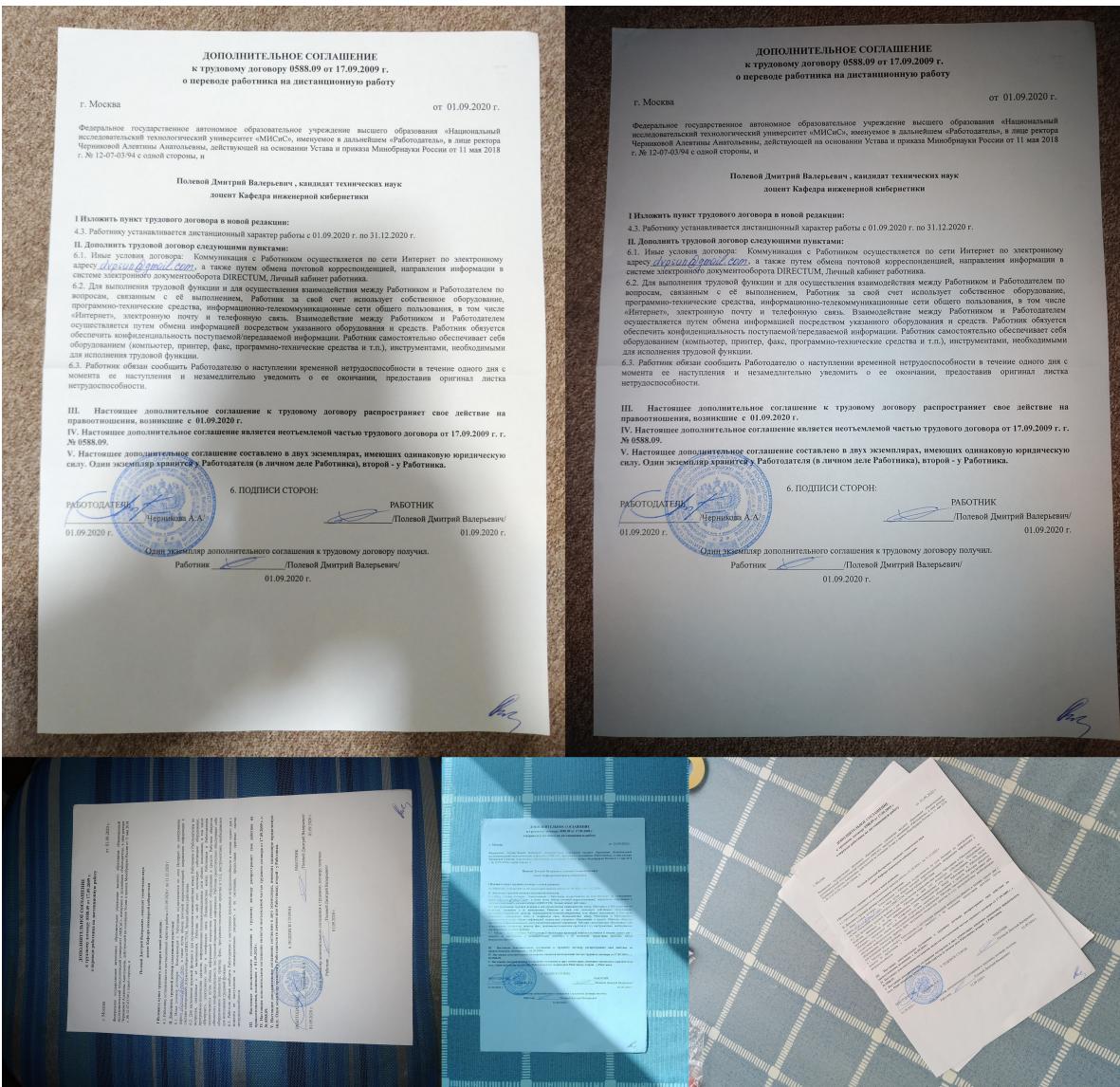


Рис. 1. Исходные изображения I_1, I_2, I_3, I_4, I_5

ДОПОЛНИТЕЛЬНОЕ СОГЛАШЕНИЕ
к трудовому договору №588.09 от 17.09.2009 г.
о переводе работника на дистанционную работу

г. Москва

от 01.09.2020 г.

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС», имеющее в дальнейшем «Работодатель», в лице ректора Черниковой Алены Анатольевны, действующей на основании Устава и приказа Минобрнауки России от 11 мая 2018 г. № 12-07-03/94 с одной стороны, и

Полевой Дмитрий Валерьевич, кандидат технических наук
доктор Кандидат технических наук

1 Изложить пункт трудового договора в новой редакции:

4.3. Работнику устанавливается дистанционный характер работы с 01.09.2020 г. по 31.12.2020 г.

II. Дополнить трудовой договор следующими пунктами:

6.1. Иные условия договора. Коммуникация с Работодателем осуществляется по сети Интернет по электронному адресу doksign.direktum.com, а также путем почтовой корреспонденции, направления информации в системе электронной документооборота DIREKTUM, Личный кабинет работника.

6.2. Для выполнения трудовой функции и для осуществления взаимодействия между Работодателем и Работником по вопросам, связанным с ей выполнением, Работник за свой счет использует собственные обстоятельства, предусмотренные законодательством Российской Федерации о труде, в том числе «Интернет», электронную почту и телефонную связь. Взаимодействие между Работодателем и Работником осуществляется путем обмена информацией посредством указанного оборудования и средств связи, что обеспечивает конфиденциальность поступаемой/передаваемой информации. Работник самостоятельно обеспечивает себя оборудованием (компьютер, принтер, факс, программно-технические средства и т.д.), инструментами, необходимыми для исполнения трудовой функции.

6.3. Работник обязан сообщать Работодателю о наступлении временной нетрудоспособности в течение одного дня с момента ее наступления и немедленно уведомить о ее окончании, предоставив оригинал листка нетрудоспособности.

III. Настоящее дополнительное соглашение к трудовому договору распространяет свое действие на правоотношения, возникшие с 01.09.2020 г.

IV. Настоящее дополнительное соглашение является неотъемлемой частью трудового договора от 17.09.2009 г. № 0588.09.

V. Настоящее дополнительное соглашение составлено в двух экземплярах, имеющих одинаковую юридическую силу. Один экземпляр хранится у Работодателя (в личном деле Работника), второй - у Работника.



6. ПОДПИСИ СТОРОН:

РАБОТОДАТЕЛЬ
/Полевой Дмитрий Валерьевич/
01.09.2020 г.

Работник
/Полевой Дмитрий Валерьевич/
01.09.2020 г.

ДОПОЛНИТЕЛЬНОЕ СОГЛАШЕНИЕ
к трудовому договору №588.09 от 17.09.2009 г.
о переводе работника на дистанционную работу

г. Москва

от 01.09.2020 г.

Федеральное государственное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС», имеющее в дальнейшем «Работодатель», в лице ректора Черниковой Алены Анатольевны, действующий на основании Устава и приказа Минобрнауки России от 11 мая 2018 г. № 12-07-03/94 с одной стороны, и

Полевой Дмитрий Валерьевич, кандидат технических наук
доктор Кандидат технических наук

I Изложить пункт трудового договора в новой редакции:

4.3. Работнику устанавливается дистанционный характер работы с 01.09.2020 г. по 31.12.2020 г.

II. Дополнить трудовой договор следующими пунктами:

6.1. Иные условия договора. Коммуникация с Работником осуществляется по сети Интернет по электронному адресу doksign.direktum.com, а также путем обмена почтовой корреспонденцией, направления информации в системе электронной документооборота DIREKTUM, Личный кабинет работника.

6.2. Для выполнения трудовой функции и для осуществления взаимодействия между Работодателем и Работником по вопросам, связанным с ей выполнением, Работник за свой счет использует собственные обстоятельства, предусмотренные законодательством Российской Федерации о труде, в том числе «Интернет», электронную почту и телефонную связь. Взаимодействие между Работодателем и Работником осуществляется путем обмена информацией посредством указанного оборудования и средств связи, что обеспечивает обесценивание конфиденциальности поступаемой/передаваемой информации. Работник самостоятельно обеспечивает себя оборудованием (компьютер, принтер, факс, программно-технические средства и т.д.), инструментами, необходимыми для исполнения трудовой функции.

6.3. Работник обязан сообщать Работодателю о наступлении временной нетрудоспособности в течение одного дня с момента ее наступления и немедленно уведомить о ее окончании, предоставив оригинал листка нетрудоспособности.

III. Настоящее дополнительное соглашение к трудовому договору распространяет свое действие на правоотношения, возникшие с 01.09.2020 г.

IV. Настоящее дополнительное соглашение является неотъемлемой частью трудового договора от 17.09.2009 г. № 0588.09.

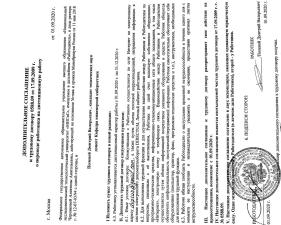
V. Настоящее дополнительное соглашение составлено в двух экземплярах, имеющих одинаковую юридическую силу. Один экземпляр хранится у Работодателя (в личном деле Работника), второй - у Работника.



6. ПОДПИСИ СТОРОН:

РАБОТОДАТЕЛЬ
/Полевой Дмитрий Валерьевич/
01.09.2020 г.

Работник
/Полевой Дмитрий Валерьевич/
01.09.2020 г.



h

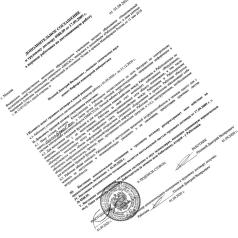


Рис. 2. Эталонная бинаризация E_1, E_2, E_3, E_4, E_5 полученная с помощью гомографии

Процесс бинаризации $F(I_i) = B_i$

1. Быстрый алгоритм локальной бинаризации с гауссовым окном

Параметры:

- Окно = (91, 91)
- Уровень шума = 20
- Пороговое значение = 0.9

2. Медианный фильтр

Параметры:

- Окно = (3, 3)

3. Выделение компонент связности

Параметры:

- Общая площадь компоненты > 15 пикселей

ДОПОЛНИТЕЛЬНОЕ СОГЛАШЕНИЕ
к трудовому договору 0588.09 от 17.09.2009 г.
о переводе работника на дистанционную работу

г. Москва

Федеральное государственное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС», имеющее в запасе № «Работодатель», в лице ректора Черновской Алены Анатольевны, действующей на основании Устава и приказа Минобрнауки России от 11 мая 2018 г. № 12-07-03 94 с одной стороны, и

Полевой Дмитрий Валерьевич, кандидат технических наук
доктор Кафедры инженерной кибернетики

от 01.09.2020 г.

г. Москва

ДОПОЛНИТЕЛЬНОЕ СОГЛАШЕНИЕ
к трудовому договору 0588.09 от 17.09.2009 г.
о переводе работника на дистанционную работу

от 01.09.2020 г.

Федеральное государственное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС», имеющее в запасе № «Работодатель», в лице ректора Черновской Алены Анатольевны, действующей на основании Устава и приказа Минобрнауки России от 11 мая 2018 г. № 12-07-03 94 с одной стороны, и

Полевой Дмитрий Валерьевич, кандидат технических наук
доктор Кафедры инженерной кибернетики

I Изложить пункт трудового договора в новой редакции:

4.3 Работнику устанавливается дистанционный характер работы с 01.09.2020 г. по 31.12.2020 г.

II. Дополнить трудовой договор следующими пунктами:

- 6.1. Иные условия договора: Коммуникация с Работником осуществляется по сети Интернет по электронному адресу Dmitriy_Polzhev@misu.ru, а также путем обмена почтовой корреспонденцией, направления информации в системе электронного документооборота DIRECTUM, Личный кабинет работника.
- 6.2. Для выполнения трудовых функций и для осуществления взаимодействия между Работником и Работодателем по вопросам, связанным с выполнением трудовых функций, Работник и Работодатель за свой счет используют собственное оборудование, программно-технические средства, информационно-телекоммуникационные сети общего пользования, в том числе «Интернет», электронную почту и телефонную связь. Взаимодействие между Работником и Работодателем осуществляется путем обмена информацией посредством указанного оборудования и средств Работника общается обменом текстовой информацией посредством передачи информации Работником самостоятельно обеспечивает себя оборудованием (компьютер, принтер, факс, программно-технические средства и т.д.), инструментами, необходимыми для исполнения трудовой функции.
- 6.3. Работник обязан сообщать Работодателю о наступлении временной нетрудоспособности в течение одного дня с момента ее наступления и немедленно уведомлять о ее окончании, предоставив оригинал листка нетрудоспособности.

III. Настоящее дополнительное соглашение к трудовому договору распространяет свое действие на правоотношения, возникшие с 01.09.2020 г.

IV. Настоящее дополнительное соглашение является неотъемлемой частью трудового договора от 17.09.2009 г. № 0588.09 с одной стороны, и

V. Настоящее дополнительное соглашение составлено в двух экземплярах, имеющих одинаковую юридическую силу. Один экземпляр хранится у Работодателя (в личном деле Работника), второй - у Работника.

6. ПОДПИСИ СТОРОН:



РАБОТОДАТЕЛЬ
Черновик А.А.
01.09.2020 г.

РАБОТНИК
/Полевой Дмитрий Валерьевич/
01.09.2020 г.

Один экземпляр дополнительного соглашения к трудовому договору получила:
Работник /Полевой Дмитрий Валерьевич/
01.09.2020 г.

I Изложить пункт трудового договора в новой редакции:

4.3. Работнику устанавливается дистанционный характер работы с 01.09.2020 г. по 31.12.2020 г.

II. Дополнить трудовой договор следующими пунктами:

- 6.1. Иные условия договора: Коммуникация с Работником осуществляется по сети Интернет по электронному адресу Dmitriy_Polzhev@misu.ru, а также путем обмена почтовой корреспонденцией, направления информации в системе электронного документооборота DIRECTUM, Личный кабинет работника.
- 6.2. Для выполнения трудовых функций и для осуществления взаимодействия между Работником и Работодателем по вопросам, связанным с выполнением трудовых функций, Работник и Работодатель за свой счет используют собственное оборудование, программно-технические средства, информационно-телекоммуникационные сети общего пользования, в том числе «Интернет», электронную почту и телефонную связь. Взаимодействие между Работником и Работодателем осуществляется путем обмена информацией посредством указанного оборудования и средств Работника общается обменом текстовой информацией посредством передачи информации Работником самостоятельно обеспечивает себя оборудованием (компьютер, принтер, факс, программно-технические средства и т.д.), инструментами, необходимыми для исполнения трудовой функции.
- 6.3. Работник обязан сообщать Работодателю о наступлении временной нетрудоспособности в течение одного дня с момента ее наступления и немедленно уведомлять о ее окончании, предоставив оригинал листка нетрудоспособности.

III. Настоящее дополнительное соглашение к трудовому договору распространяет свое действие на правоотношения, возникшие с 01.09.2020 г.

IV. Настоящее дополнительное соглашение является неотъемлемой частью трудового договора от 17.09.2009 г. № 0588.09 с одной стороны, и

V. Настоящее дополнительное соглашение составлено в двух экземплярах, имеющих одинаковую юридическую силу. Один экземпляр хранится у Работодателя (в личном деле Работника), второй - у Работника.

6 ПОДПИСИ СТОРОН

Р Т О
Черновик А.А.
01.09.2020 г.

РАБОТНИК
/Полевой Дмитрий Валерьевич/
01.09.2020 г.

один экземпляр дополнительного соглашения к трудовому договору получил

Работник /Полевой Дмитрий Валерьевич/
01.09.2020 г.

Рис. 4. Сравнение результатов бинаризации B_2 и B'_2

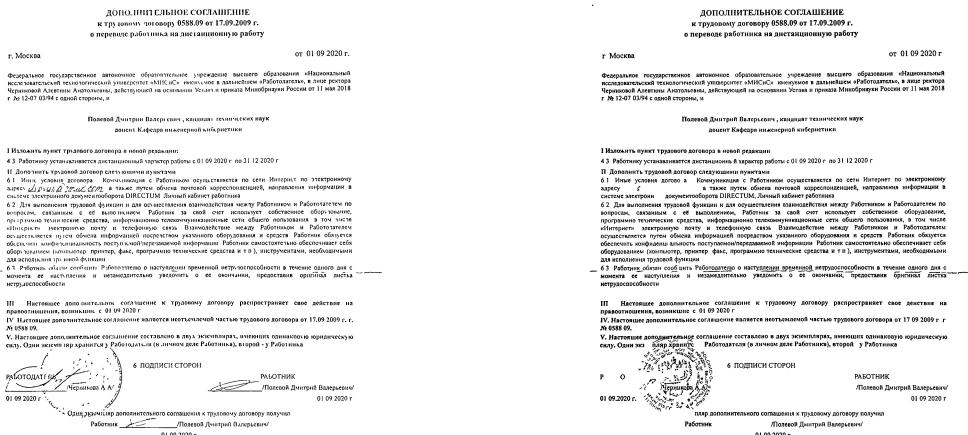


Рис. 5. Сравнение результатов бинаризации B_3 и B'_3

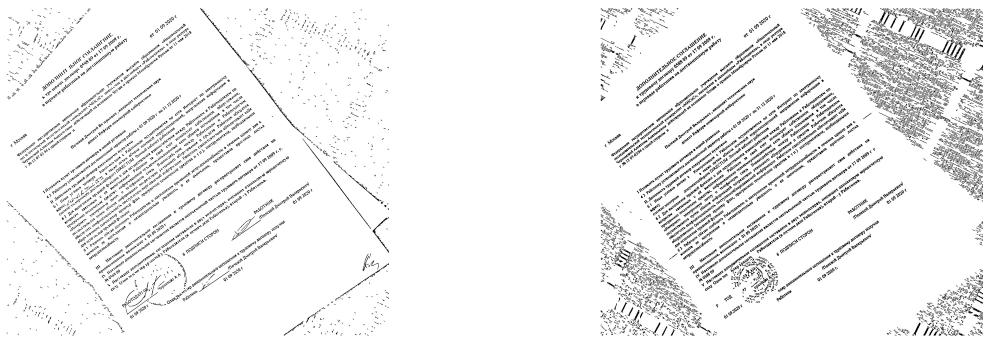


Рис. 6. Сравнение результатов бинаризации B_4 и B'_4

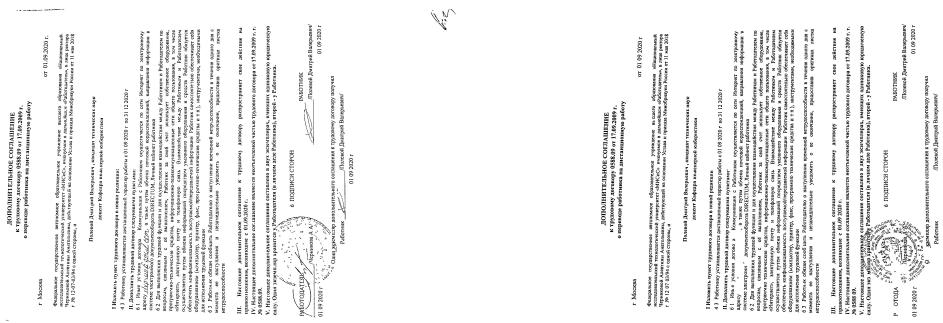


Рис. 7. Сравнение результатов бинаризации B_5 и B'_5

Численные оценки

TP - черный пиксель стоит как на эталоне

FP - черный пиксель стоит на месте белого пикселя в эталоне

FN - белый пиксель стоит как на эталоне

TN - белый пиксель стоит на месте черного пикселя в эталоне

$$BA \text{ (Balanced accuracy)} = \frac{\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right)}{2} \in [0,1]$$

$$F_1 \text{ (F1-score)} = \frac{2TP}{2TP+FP+FN} \in [0,1]$$

Фото	TP	FP	FN	TN	BA	F1
B_1	64557	232602	259511	5.59333e+06	0.579641	0.207837
B_2	76218	227278	278985	5.84352e+06	0.588569	0.23142
B_3	40740	135648	160123	3.25689e+06	0.58142	0.215984
B_4	34082	85555	156737	3.50363e+06	0.577386	0.219561
B_5	50349	151628	195271	3.81875e+06	0.583399	0.224975
Среднее для набора	53189.2	166542.2	210125.4	4.403224e+06	0.582083	0.2199554

Таблица 1. Оценки для бинаризации F

Фото	TP	FP	FN	TN	BA	F1
B'_1	85135	212024	341136	5.51171e+06	0.581339	0.235365
B'_2	90337	213159	338079	5.78443e+06	0.587661	0.246852
B'_3	51062	125326	209274	3.20774e+06	0.579269	0.233841
B'_4	43841	75796	364880	3.29548e+06	0.542391	0.165952
B'_5	58776	143201	228616	3.78541e+06	0.584032	0.240211
Среднее для набора	65830.2	153901.2	296397	4.316954e+06	0.5749384	0.2244442

Таблица 2. Оценки для бинаризации F'

Фото	F	F'	Разница
1	0.579641	0.581339	+0.3%
2	0.588569	0.587661	-0.15%
3	0.58142	0.579269	-0.37%
4	0.577386	0.542391	-6.1%
5	0.583399	0.584032	+0.11%
Среднее для набора	0.582083	0.5749384	-1.23%

Таблица 3. Сравнение BA-оценки для бинаризаций F и F'

Фото	F	F'	Разница
1	0.207837	0.235365	+13.2%
2	0.23142	0.246852	+6.67%
3	0.215984	0.233841	+8.27%
4	0.219561	0.165952	-24.41%
5	0.224975	0.240211	+6.77%
Среднее для набора	0.2199554	0.2244442	+2.04%

Таблица 4. Сравнение F_1 -оценки для бинаризаций F и F'

Текст программы

```
#include <opencv2/opencv.hpp>
#include <fstream>
#include "json.hpp"

using namespace std;
using namespace cv;
```

```

using json = nlohmann::json;

Mat findDifference(Mat etalon, Mat example) {

    Mat stats = Mat::zeros(Size(2, 3), CV_32F);

    //TP, FP, FN, TN
    for (int i = 0; i < etalon.cols; ++i) {
        for (int j = 0; j < etalon.rows; ++j) {
            if (etalon.at<uchar>(j,i) == 0) {

                if (example.at<uchar>(j,i) == 0) {
                    stats.at<float>(0, 0) += (float)1;
                }
                else {
                    stats.at<float>(0, 1) += (float)1;;
                }
            }
            else {

                if (example.at<uchar>(j, i) == 255) {
                    stats.at<float>(1, 1) += (float)1;
                }
                else {
                    stats.at<float>(1, 0) += (float)1;
                }
            }
        }
    }

    //balanced accuracy
    float sensitivity = stats.at<float>(0, 0) / (stats.at<float>(0, 0) +
    stats.at<float>(1, 0));
    float specificity = stats.at<float>(1, 1) / (stats.at<float>(1, 1) +
    stats.at<float>(0, 1));
    stats.at<float>(2, 0) = (sensitivity + specificity) / 2;

    //f_1 score
    float precision = stats.at<float>(0, 0) / (stats.at<float>(0, 0) +
    stats.at<float>(0, 1));
    float recall = stats.at<float>(0, 0) / (stats.at<float>(0, 0) +
    stats.at<float>(1, 0));
    stats.at<float>(2, 1) = 2 * precision * recall / (precision + recall);

    return stats;
}

Mat showDifference(Mat etalon, Mat example) {

    Mat diff_img = example.clone();
    cvtColor(diff_img, diff_img, COLOR_GRAY2BGR);

    for (int i = 0; i < etalon.cols; ++i) {
        for (int j = 0; j < etalon.rows; ++j) {
            if (etalon.at<uchar>(j, i) == 0) {

                if (example.at<uchar>(j, i) != 0) {

```

```

                diff_img.at<Vec3b>(j, i) = { 0, 0, 255 };
            }
        }
    else {

        if (example.at<uchar>(j, i) != 255) {
            diff_img.at<Vec3b>(j, i) = { 0, 255, 0 };
        }
    }
}
return diff_img;
}

void getStats(Mat stat) {
    cout << "TP: " << stat.at<float>(0, 0) << endl;
    cout << "FP: " << stat.at<float>(0, 1) << endl;
    cout << "FN: " << stat.at<float>(1, 0) << endl;
    cout << "TN: " << stat.at<float>(1, 1) << endl;
    cout << "Balanced accuracy: " << stat.at<float>(2, 0) << endl;
    cout << "F1 score: " << stat.at<float>(2, 1) << endl;
}

Mat getEtalonHomography(Mat scan, Mat photo, json scan_borders, json
photo_borders) {

    //scan borders
    string file_name_scan = scan_borders["_via_image_id_list"][0];

    vector<Point> scan_points;
    for (int i = 0; i < scan_borders["_via_img_metadata"][file_name_scan]
["regions"].size(); i++) {
        for (int j = 0; j < scan_borders["_via_img_metadata"][file_name_scan]
["regions"][i]["shape_attributes"]["all_points_x"].size(); j++) {

            int x = scan_borders["_via_img_metadata"][file_name_scan]["regions"]
[i]["shape_attributes"]["all_points_x"][j];
            int y = scan_borders["_via_img_metadata"][file_name_scan]["regions"]
[i]["shape_attributes"]["all_points_y"][j];

            scan_points.push_back(Point(x, y));
        }
    }

    //photo borders
    string file_name_photo = photo_borders["_via_image_id_list"][0];

    vector<Point> example_points;
    for (int i = 0; i < photo_borders["_via_img_metadata"][file_name_photo]
["regions"].size(); i++) {
        for (int j = 0; j < photo_borders["_via_img_metadata"][file_name_photo]
["regions"][i]["shape_attributes"]["all_points_x"].size(); j++) {

            int x = photo_borders["_via_img_metadata"][file_name_photo]
["regions"][i]["shape_attributes"]["all_points_x"][j];
            int y = photo_borders["_via_img_metadata"][file_name_photo]
["regions"][i]["shape_attributes"]["all_points_y"][j];
        }
    }
}

```

```

        example_points.push_back(Point(x, y));
    }

}

//get homography
Mat scan_inv = 255 - scan;
Mat homography = findHomography(scan_points, example_points);
Mat res_H(photo.size(), CV_8U);
warpPerspective(scan_inv, res_H, homography, photo.size());
res_H = 255 - res_H;

return res_H;
}

Mat binarizeImage(Mat src) {

//binarization
Mat gb;
GaussianBlur(src, gb, Size(71, 71), 0, 0);
Mat m = cv::abs(gb - src);
Mat m_gb;
GaussianBlur(m, m_gb, Size(71, 71), 0, 0);

Mat bin = Mat::zeros(src.size(), CV_32F);

for (int i = 0; i < bin.cols; ++i) {
    for (int j = 0; j < bin.rows; ++j) {
        bin.at<float>(j, i) = (float)
            (gb.at<uchar>(j, i) - src.at<uchar>(j, i)) / (m_gb.at<uchar>(j,
i) + 60) < 0.15 ? 1 : 0;
    }
}

bin.convertTo(bin, CV_8U, 255);

//filtering
Mat f_2;
medianBlur(bin, f_2, 3);

//connected components
bitwise_not(f_2, f_2);

Mat labelImage, statsImage, centroidsImage;
int nLabels = connectedComponentsWithStats(f_2, labelImage, statsImage,
centroidsImage, 8);
cout << nLabels << endl;

Mat mask = Mat::zeros(f_2.size(), CV_8UC1);
for (int i = 1; i < statsImage.rows; ++i) {

    if ((statsImage.at<int>(Point(CC_STAT_AREA, i)) > 20) &
(statsImage.at<int>(Point(CC_STAT_AREA, i)) < 750) &
        (statsImage.at<int>(Point(CC_STAT_HEIGHT, i)) > 4) &
(statsImage.at<int>(Point(CC_STAT_HEIGHT, i)) < 200) &
        (statsImage.at<int>(Point(CC_STAT_WIDTH, i)) > 10) &
(statsImage.at<int>(Point(CC_STAT_WIDTH, i)) < 100) &
        (statsImage.at<int>(Point(CC_STAT_ASPECT_RATIO, i)) > 0.5) &
(statsImage.at<int>(Point(CC_STAT_ASPECT_RATIO, i)) < 2.0) &
        (statsImage.at<int>(Point(CC_STAT_CENTROID_X, i)) > 0) &
(statsImage.at<int>(Point(CC_STAT_CENTROID_Y, i)) > 0) &
        (statsImage.at<int>(Point(CC_STAT_CENTROID_X, i)) < photo.cols) &
(statsImage.at<int>(Point(CC_STAT_CENTROID_Y, i)) < photo.rows))
}
}
```

```

        (statsImage.at<int>(Point(CC_STAT_WIDTH, i)) > 3) &
        (statsImage.at<int>(Point(CC_STAT_WIDTH, i)) < 80))

    {
        mask += (labelImage == i);
    }
}

return mask;
}

int main() {

//load scan with borders
Mat scan_image = imread("scan.png", IMREAD_GRAYSCALE);
ifstream scan_borders_stream("scan_borders.json");
json scan_borders;
scan_borders_stream >> scan_borders;

//load photos
Mat photo_1 = imread("P_20210309_064128_vHDR_Auto.jpg", IMREAD_GRAYSCALE);
Mat photo_2 = imread("P_20210309_064141_vHDR_Auto.jpg", IMREAD_GRAYSCALE);
Mat photo_3 = imread("P_20210410_170813_vHDR_On.jpg", IMREAD_GRAYSCALE);
Mat photo_4 = imread("P_20210410_171055_vHDR_On.jpg", IMREAD_GRAYSCALE);
Mat photo_5 = imread("P_20210410_171334_vHDR_On.jpg", IMREAD_GRAYSCALE);

//load borders
ifstream photo_1_borders_stream("P_20210309_064128_vHDR_Auto.json");
ifstream photo_2_borders_stream("P_20210309_064141_vHDR_Auto.json");
ifstream photo_3_borders_stream("P_20210410_170813_vHDR_On.json");
ifstream photo_4_borders_stream("P_20210410_171055_vHDR_On.json");
ifstream photo_5_borders_stream("P_20210410_171334_vHDR_On.json");

json photo_1_borders;
json photo_2_borders;
json photo_3_borders;
json photo_4_borders;
json photo_5_borders;

photo_1_borders_stream >> photo_1_borders;
photo_2_borders_stream >> photo_2_borders;
photo_3_borders_stream >> photo_3_borders;
photo_4_borders_stream >> photo_4_borders;
photo_5_borders_stream >> photo_5_borders;

//Get etalon gor photos
Mat etalon_photo_1 = getEtalonHomography(scan_image, photo_1, scan_borders,
photo_1_borders);
Mat etalon_photo_2 = getEtalonHomography(scan_image, photo_2, scan_borders,
photo_2_borders);
Mat etalon_photo_3 = getEtalonHomography(scan_image, photo_3, scan_borders,
photo_3_borders);
Mat etalon_photo_4 = getEtalonHomography(scan_image, photo_4, scan_borders,
photo_4_borders);
Mat etalon_photo_5 = getEtalonHomography(scan_image, photo_5, scan_borders,
photo_5_borders);
}

```

```

imwrite("etalon_photo_1.png", etalon_photo_1);
imwrite("etalon_photo_2.png", etalon_photo_2);
imwrite("etalon_photo_3.png", etalon_photo_3);
imwrite("etalon_photo_4.png", etalon_photo_4);
imwrite("etalon_photo_5.png", etalon_photo_5);

//Define roi
Rect2d roi_1 = { 250, 150, 2050, 3000 };
Rect2d roi_2 = { 203, 100, 2100, 3060 };
Rect2d roi_3 = { 420, 630, 1590, 2260 };
Rect2d roi_4 = { 900, 200, 2100, 1800 };
Rect2d roi_5 = { 565, 340, 2480, 1700 };

Mat roi_img_1 = photo_1(roi_1);
Mat roi_img_2 = photo_2(roi_2);
Mat roi_img_3 = photo_3(roi_3);
Mat roi_img_4 = photo_4(roi_4);
Mat roi_img_5 = photo_5(roi_5);

//binarization
Mat res_bin_1 = binarizeImage(roi_img_1);
Mat res_bin_2 = binarizeImage(roi_img_2);
Mat res_bin_3 = binarizeImage(roi_img_3);
Mat res_bin_4 = binarizeImage(roi_img_4);
Mat res_bin_5 = binarizeImage(roi_img_5);

Mat photo_bin_1 = Mat::zeros(photo_1.size(), CV_8U);
Mat photo_bin_2 = Mat::zeros(photo_2.size(), CV_8U);
Mat photo_bin_3 = Mat::zeros(photo_3.size(), CV_8U);
Mat photo_bin_4 = Mat::zeros(photo_4.size(), CV_8U);
Mat photo_bin_5 = Mat::zeros(photo_5.size(), CV_8U);

Mat photo_bin_roi_1 = photo_bin_1(roi_1);
Mat photo_bin_roi_2 = photo_bin_2(roi_2);
Mat photo_bin_roi_3 = photo_bin_3(roi_3);
Mat photo_bin_roi_4 = photo_bin_4(roi_4);
Mat photo_bin_roi_5 = photo_bin_5(roi_5);

photo_bin_roi_1 += res_bin_1;
photo_bin_roi_2 += res_bin_2;
photo_bin_roi_3 += res_bin_3;
photo_bin_roi_4 += res_bin_4;
photo_bin_roi_5 += res_bin_5;

bitwise_not(photo_bin_1, photo_bin_1);
bitwise_not(photo_bin_2, photo_bin_2);
bitwise_not(photo_bin_3, photo_bin_3);
bitwise_not(photo_bin_4, photo_bin_4);
bitwise_not(photo_bin_5, photo_bin_5);

//Get statistics
Mat stats_photo_1 = findDifference(etalon_photo_1(roi_1),
photo_bin_1(roi_1));

```

```

    Mat stats_photo_2 = findDifference(etalon_photo_2(roi_2),
photo_bin_2(roi_2));
    Mat stats_photo_3 = findDifference(etalon_photo_3(roi_3),
photo_bin_3(roi_3));
    Mat stats_photo_4 = findDifference(etalon_photo_4(roi_4),
photo_bin_4(roi_4));
    Mat stats_photo_5 = findDifference(etalon_photo_5(roi_5),
photo_bin_5(roi_5));

    getStats(stats_photo_1);
    getStats(stats_photo_2);
    getStats(stats_photo_3);
    getStats(stats_photo_4);
    getStats(stats_photo_5);

//Diff images
Mat diff_1 = showDifference(etalon_photo_1, photo_bin_1);
Mat diff_2 = showDifference(etalon_photo_2, photo_bin_2);
Mat diff_3 = showDifference(etalon_photo_3, photo_bin_3);
Mat diff_4 = showDifference(etalon_photo_4, photo_bin_4);
Mat diff_5 = showDifference(etalon_photo_5, photo_bin_5);

//imwrite("res_bin_1.png", photo_bin_1);
//imwrite("res_bin_2.png", photo_bin_2);
//imwrite("res_bin_3.png", photo_bin_3);
//imwrite("res_bin_4.png", photo_bin_4);
//imwrite("res_bin_5.png", photo_bin_5);

//imwrite("diff_1.png", diff_1);
//imwrite("diff_2.png", diff_2);
//imwrite("diff_3.png", diff_3);
//imwrite("diff_4.png", diff_4);
//imwrite("diff_5.png", diff_5);

imwrite("res_bin_1_v2.png", photo_bin_1);
imwrite("res_bin_2_v2.png", photo_bin_2);
imwrite("res_bin_3_v2.png", photo_bin_3);
imwrite("res_bin_4_v2.png", photo_bin_4);
imwrite("res_bin_5_v2.png", photo_bin_5);

imwrite("diff_1_v2.png", diff_1);
imwrite("diff_2_v2.png", diff_2);
imwrite("diff_3_v2.png", diff_3);
imwrite("diff_4_v2.png", diff_4);
imwrite("diff_5_v2.png", diff_5);

waitKey(0);
return 0;
}

```

