

计算机视觉实验报告

16340220 王培钰 电子政务

1. 图像分割进行A4纸矫正

(1) 将图像从rgb空间转为灰度空间

```
void ImageSegmentation::rgb2gray() {
    grayImg.resize(Img._width, Img._height, 1, 1, 0);
    cimg_forXY(grayImg, x, y) {
        double R = Img(x,y,0);
        double G = Img(x,y,1);
        double B = Img(x,y,2);
        double Gray = (R * 299 + G * 587 + B * 114 + 500) / 1000;
        grayImg(x,y) = Gray;
    }
}
```

(2) 高斯滤波

```
void ImageSegmentation::Gauss_blur() {
    blurImg = grayImg.get_blur(guassain_blur);
}
```

(3) 求阈值

1. 迭代法求阈值

```
//迭代法求阈值
void ImageSegmentation::get_thres_iteration() {
    CImg<int> image = blurImg;
    CImg<int> hist = image.histogram(256, 0, 255);
    int size = blurImg.size();
    cimg_forX(hist, i) {
        threshold += i*hist(i);
    }
    threshold /= size;
    int threshold_new;
    while (true) {
        int t1 = 0, t2 = 0;
        int num1 = 0, num2 = 0;
        // 计算小于等于阈值threshold的灰度平均值t1以及大于阈值的t2
        cimg_forX(hist, i) {
```

```

        if (i <= threshold) {
            t1 += i * hist(i);
            num1 += hist(i);
        }
        else {
            t2 += i * hist(i);
            num2 += hist(i);
        }
    }
    if (num1 == 0 || num2 == 0)
        continue;
    t1 /= num1;
    t2 /= num2;
    threshold_new = (t1 + t2) / 2;
    // 若两个阈值相等，则返回阈值threshold，否则更新阈值继续循环
    if (threshold == threshold_new) break;
    else threshold = threshold_new;
}
cout << "threshold = " << threshold << endl;
}

```

2.OSTU法求阈值

```

//OSTU法求阈值
void ImageSegmentation::get_thres_ostu() {
    //定义类间方差
    double variance = 0.0;
    CImg<int> image = blurImg;
    CImg<int> hist = image.histogram(256, 0, 255);
    int size = blurImg.size();
    for (int i = 0; i < 256; i++) {
        //定义前景图，背景图的像素点所占比例以及平均灰度
        double p1 = 0.0, p2 = 0.0, g1 = 0.0, g2 = 0.0;
        cimg_forX(hist, j) {
            if (j <= i) {
                p1 += hist(j);
                g1 += j*hist(j);
            }
            else {
                p2 += hist(j);
                g2 += j*hist(j);
            }
        }
        if (p1 == 0 || p2 == 0)
            continue;
        g1 /= p1;
        p1 /= size;
        g2 /= p2;
        p2 /= size;
        double temp_variance = p1 * p2 * (g1 - g2) * (g1 - g2);
        if (variance < temp_variance) {
            variance = temp_variance;
        }
    }
}

```

```

        threshold = i;
    }
}
cout << "threshold = " << threshold << endl;
}

```

(4)按阈值进行图像分割

```

void ImageSegmentation::Segmentation() {
    segImg.resize(Img._width, Img._height, 1, 1, 0);
    cimg_forXY(blurImg, x, y) {
        if (blurImg(x,y) > threshold) {
            segImg(x,y) = 0;
        }
        else
            segImg(x,y) = 255;
    }
    //segImg.display();
}

```

(5) 对分割后的图像进行梯度检测

PS:如果此时直接进行矫正或者膨胀后膨胀效果不好，所以我们检测一步梯度（采用sobel算子）

```

void ImageSegmentation::gradDecton() {
    gradImg.resize(segImg._width, segImg._height, 1, 1, 0);
    CImg_3x3(I, double);
    cimg_for3x3(segImg, x, y, 0, 0, I, double) {
        const double ix = (Inn + 2 * Icn + Ipn) - (Ipp + 2 * Icp + Inp);
        const double iy = (Inp + 2 * Inc + Inn) - (Ipp + 2 * Ipc + Ipn);
        double grad = sqrt(ix * ix + iy * iy);
        if (grad > 255) grad = 255;
        if (grad < 0 ) grad = 0;
        gradImg(x, y) = grad;
    }
    gradImg.display();
}

```

(6) 霍夫变换

1. 统计霍夫空间点

```

void ImageSegmentation::Hough_Statistics() {
    //double maxDistance = sqrt(Img._width*Img._width + Img._height*Img._height);
    double w = Img._width;
    double h = Img._height;
    double center_x = w/2;
    double center_y = h/2;
    double hough_h = ((sqrt(2.0) * (double)(h>w?h:w)) / 2.0);
}

```

```

houghImg.resize(180, hough_h * 2, 1, 1, 0);
cimg_forXY(gradImg, x, y) {
    if (gradImg(x,y) != 0) {
        cimg_forX(houghImg, angle) {
            double _angle = (double)PI*angle / 180.0f;
            int polar = (int)((double)x - center_x)*cos(_angle) + ((double)y -
center_y)*sin(_angle)) + hough_h);
            //cout << polar << endl;
            houghImg(angle, polar) += 1;
        }
    }
}
//houghImg.display();
}

```

2. 获取对应检测出的直线

```

void ImageSegmentation::GetLine() {
    resultImg = Img;
    //剔除掉可能出现的重合线，方法是取9x9空间内的霍夫最大值
    int hough_h = houghImg._height;
    //int hough_w = houghImg._width;
    int img_h = Img._height;
    int img_w = Img._width;
    const int y_min = 0;
    const int y_max = Img._height - 1;
    const int x_min = 0;
    const int x_max = Img._width - 1;
    cimg_forXY(houghImg, angle, polar) {
        if (houghImg(angle, polar) >= Min_thres) {
            int max = houghImg(angle, polar);
            for(int ly=-DIFF;ly<=DIFF;ly++) {
                for(int lx=-DIFF;lx<=DIFF;lx++) {
                    if( (ly+polar>=0 && ly+polar<houghImg._height) && (lx+angle>=0 &&
lx+angle<houghImg._width) ) {
                        if( (int)houghImg(angle + lx, polar + ly ) > max ) {
                            max = houghImg(angle + lx, polar + ly );
                            ly = lx = DIFF + 1;
                        }
                    }
                }
            }
            if (max > (int)houghImg(angle, polar) )
                continue;
            peaks.push_back(pair< pair<int, int>, int >(pair<int, int>(angle, polar),
houghImg(angle, polar)));
        }
    }
    sort(peaks.begin(), peaks.end(), [](const pair< pair<int, int>, int > &a, const pair<
pair<int, int>, int > &b) -> int {return a.second > b.second ;});
    for (int i = 0; lines.size() != 4; i++) {
        int angle = peaks[i].first.first;
    }
}

```

```

        int polar = peaks[i].first.second;
        //cout << angle << endl << polar << endl;
        int x1, y1, x2, y2;
        x1 = y1 = x2 = y2 = 0;
        double _angle = (double)PI*angle / 180.0f;
        if(angle >= 45 && angle <= 135) {
            x1 = 0;
            y1 = ((double)(polar-(hough_h/2)) - ((x1 - (img_w/2) ) * cos(_angle))) / sin(_angle)
+ (img_h / 2);
            x2 = img_w;
            y2 = ((double)(polar-(hough_h/2)) - ((x2 - (img_w/2) ) * cos(_angle))) / sin(_angle)
+ (img_h / 2);
        }
        else {
            y1 = 0;
            x1 = ((double)(polar-(hough_h/2)) - ((y1 - (img_h/2) ) * sin(_angle))) / cos(_angle)
+ (img_w / 2);
            y2 = img_h;
            x2 = ((double)(polar-(hough_h/2)) - ((y2 - (img_h/2) ) * sin(_angle))) / cos(_angle)
+ (img_w / 2);
        }
        //if
        bool flag = true;
        for (int k = 0; k < lines.size(); k++) {
            if (distance(lines[k].first.first - x1, lines[k].first.second - y1) < 100 &&
distance(lines[k].second.first - x2, lines[k].second.second - y2) < 100) {
                flag = false;
                break;
            }
        }
        if (flag == true) {
            lines.push_back(pair< pair<int, int>, pair<int, int> >(pair<int, int>(x1, y1),
pair<int, int>(x2, y2)));
        }
    }
    for (int i = 0; i < lines.size(); i++) {
        cout << lines[i].first.first << ", " << lines[i].first.second << " .. " <<
lines[i].second.first << ", " << lines[i].second.second << endl;
        resultImg.draw_line(lines[i].first.first, lines[i].first.second, lines[i].second.first,
lines[i].second.second, Red);
    }
    //resultImg.draw_line(200, 3458, 2500, 3459, Red);
    //resultImg.display();
}

```

3. 获得四个角点

```

void ImageSegmentation::GetVertexs() {
    for (int i = 0; i < lines.size(); i++) {
        double k0, b0;
        if (lines[i].first.first == lines[i].second.first) {
            k0 = DBL_MAX;

```

```

        b0 = lines[i].first.first;
    }
    else {
        k0 = (double) (lines[i].first.second - lines[i].second.second) /
(lines[i].first.first - lines[i].second.first);
        b0 = (double) (lines[i].first.second * lines[i].second.first -
lines[i].second.second * lines[i].first.first) / (lines[i].second.first - lines[i].first.first);
    }
    for (int j = i + 1; j < lines.size(); j++) {
        double k1, b1;
        if (lines[j].first.first == lines[j].second.first) {
            k1 = DBL_MAX;
            b1 = lines[j].first.first;
        }
        else {
            k1 = (double) (lines[j].first.second - lines[j].second.second) /
(lines[j].first.first - lines[j].second.first);
            b1 = (double) (lines[j].first.second * lines[j].second.first -
lines[j].second.second * lines[j].first.first) / (lines[j].second.first - lines[j].first.first);
        }
        if (k0 == k1)
            continue;
        if (k0 == DBL_MAX) {
            int _x = b0, _y = k1 * b0 + b1;
            if (_x >= 0 && _x < Img._width && _y >= 0 && _y < Img._height)
                vertex.push_back(make_pair(_x, _y));
            continue;
        }
        if (k1 == DBL_MAX) {
            int _x = b1, _y = k0 * b1 + b0;
            if (_x >= 0 && _x < Img._width && _y >= 0 && _y < Img._height)
                vertex.push_back(make_pair(_x, _y));
            continue;
        }
        int _x = (b0 - b1) / (k1 - k0);
        int _y = (k0 * b1 - k1 * b0) / (k0 - k1);
        if (_x >= 0 && _x < Img._width && _y >= 0 && _y < Img._height)
            vertex.push_back(make_pair(_x, _y));
    }
}
for (int i = 0; i < vertex.size(); i++) {
    cout << vertex[i].first << " ... " << vertex[i].second << endl;
    resultImg.draw_circle(vertex[i].first, vertex[i].second, 50, Red);
}
resultImg.display();
resultImg.save("result_a.bmp");
}

```

(7) 对获得的角点进行排序:

```

void ImageSegmentation::orderVertexs() {

```

```

        sort(vertex.begin(), vertex.end(), [](const pair<int, int> &a, const pair<int, int> &b)->
int {return distance(a.first, a.second) < distance(b.first, b.second);});
        double w = distance(vertex[0].first - vertex[1].first, vertex[0].second - vertex[1].second);
        double h = distance(vertex[0].first - vertex[2].first, vertex[0].second - vertex[2].second);
        //纸张是横向的
        if (vertex[1].first < vertex[2].first && h > w) {
            swap(vertex[1], vertex[2]);
            swap(vertex[2], vertex[3]);
            vertex.push_back(vertex[0]);
            vertex.erase(vertex.begin());
        }
        //纸张是竖向的
        else {
            swap(vertex[2], vertex[3]);
        }
    }
}

```

(8) 得到用于矫正A4纸的特征矩阵

```

void ImageSegmentation::calcMatrix() {
    double x0 = vertex[0].first, x1 = vertex[1].first, x2 = vertex[2].first, x3 = vertex[3].first;
    double y0 = vertex[0].second, y1 = vertex[1].second, y2 = vertex[2].second, y3 =
vertex[3].second;
    double dx3 = x0 - x1 + x2 - x3;
    double dy3 = y0 - y1 + y2 - y3;
    if (fabs(dx3) < 10e-5 && fabs(dy3) < 10e-5) {
        M[0] = x1 - x0, M[1] = y1 - y0, M[2] = 0;
        M[3] = x2 - x1, M[4] = y2 - y1, M[5] = 0;
        M[6] = x0, M[7] = y0, M[8] = 1;
    }
    else {
        double dx1 = x1 - x2, dx2 = x3 - x2, dy1 = y1 - y2, dy2 = y3 - y2;
        double det = dx1 * dy2 - dx2 * dy1;
        double a13 = (dx3 * dy2 - dx2 * dy3) / det;
        double a23 = (dx1 * dy3 - dx3 * dy1) / det;
        M[0] = x1 - x0 + a13 * x1, M[1] = y1 - y0 + a13 * y1, M[2] = a13;
        M[3] = x3 - x0 + a23 * x3, M[4] = y3 - y0 + a23 * y3, M[5] = a23;
        M[6] = x0, M[7] = y0, M[8] = 1;
    }
}
}

```

(9) 进行A4纸矫正

```

void ImageSegmentation::warping() {
    resultImg = Img;
    double P[3];
    resultImg.resize(1050, 1485); // 标准A4纸比例
    double width = resultImg.width(), height = resultImg.height();
    cimg_forXY(resultImg, x, y) {

```

```

        double _x = x / width, _y = y / height;
        double denominator = M[2] * _x + M[5] * _y + M[8];
        double tx = (M[0] * _x + M[3] * _y + M[6]) / denominator;
        double ty = (M[1] * _x + M[4] * _y + M[7]) / denominator;
        /*Interpolation(Img, tx, ty, P);
        resultImg(x,y,0) = P[0];
        resultImg(x,y,1) = P[1];
        resultImg(x,y,2) = P[2];
        //cout << P[0] << endl; */
        cimg_forC(resultImg, c) {
            resultImg(x,y,c) = Img((int)tx, (int)ty, c);
        }
    }
    resultImg.display();
    resultImg.save("result_b.bmp");
}

```

(10) 封装成类

```

class ImageSegmentation
{
private:
    //源图
    CImg<double> Img;
    //灰度图
    CImg<double> grayImg;
    //高斯平滑后的图像
    CImg<double> blurImg;
    //用于图像分割的阈值
    int threshold;
    //图像分割后的图片
    CImg<double> segImg;
    //梯度图像
    CImg<double> gradImg;
    //霍夫空间图像
    CImg<double> houghImg;
    CImg<double> resultImg;
    vector< pair< pair<int, int>, int > > peaks;
    //直线点集
    vector< pair< pair<int, int>, pair<int, int> > > lines;
    //四个角点
    vector< pair<int, int> > vertex;
    double M[9];

private:
    void rgb2gray();
    void Gauss_blur();
    //迭代法求阈值
    void get_thres_iteration();
    //OSTU法求阈值
    void get_thres_ostu();

```



```
void get_thres(string type_c);
void Segmentation();
void gradDectection();
void Hough_Statistics();
void GetLine();
void GetVertexs();
void orderVertexs();
void calcMatrix();
void warping();

public:
    ImageSegmentation();
    ImageSegmentation(const char* filename);
    ~ImageSegmentation();
    void correct_process(string type_c);
};
```

(11) 效果展示:

1.jpg

申请人需要说明：

(1) 选择“理论创新研究”类别的申请人的代表性成果应以高水平学术论文为主；选择“应用、技术与工程研究”类别的申请人的代表性成果应以科研成果及其应用成果(专利)、技术开发与工程应用成果(建、上、装料)、人才培养与应用研究成果(公式外推与体育学科)为主。

(2) 代表性学术成果请按以下相应要求填写：

- ◆ 论文：按全部作者排序(注明学号、通讯作者、共同第一作者、共同通讯作者)。论文题目、刊物名称(注明 ISSN 号)、刊物类别、出版年份、卷号、期号、起止页码、第一作者单位列出。
- ◆ 著作：按全部作者排序。著作名称、出版单位、出版年份、总字数、本人撰写字数列出。
- ◆ 研究报告：按全部作者排序。报告题目、报告时间、服务对象、总字数、本人撰写字数列出。
- ◆ 决策咨询成果/研究成果：按全部作者排序。咨询内容、咨询时间、咨询对象、总字数、本人撰写字数列出。
- ◆ 应用型科技成果：按全部作者排序。成果名称、发布时间、奖励名称(注明奖励级别)、颁奖部门、获奖等级、本人承担的工作列出。
- ◆ 国家标准/行业标准/医疗行业标准/技术标准：按全部作者排序。标准名称、标准号、发布日期、起草部门列出。
- ◆ 国家专利：按全部专利发明人排序。专利名称、专利号、专利主项目、专利类型、授权时间、专利权人列出。
- ◆ 软件著作权：按全部著作人排序。软件名称、登记号、发表时间列出。
- ◆ 国家新药证书：按全部作者排序。药品名称、证书编号、持有人、颁发时间列出。
- ◆ 药物临床试验批件：按申请人排序。药物名称、批件号、批准时间列出。
- ◆ 医疗器械注册证书：按全部作者排序。产品名称、批件号、持有人、批准时间、有效期列出。
- ◆ (公共管理与体育学科) 按申请人参加获奖：按获奖时间、指导学号(专业、学号、姓名)、奖项名称(注明奖项级别)、获奖等级、本人承担的工作列出。
- ◆ 其他：按全部作者排序。成果名称、发表时间、内容、影响列出。

(3) “说明”栏请简要列出成果标点和影响力情况。例如刊物类别、收录引用及转载情况、奖励情况等可以证明成果影响力的内容，不超过 20 字。代表性论文刊例类别，文科根据学校所在地城市的《中山大学人文社会科学重要期刊目录或原则》(编号及(2014 年修订版))按 A、B 类核心期刊填写，理工医科填写 Thomson Reuters JCR-SCI 一区、二区、三区、四区期刊收录，或中国科学院文献情报中心 JCR-SCI 一区、二区、三区、四区期刊收录，或 ESI 高引用论文或热点论文，或 EI 收录，或 ISTP 收录。

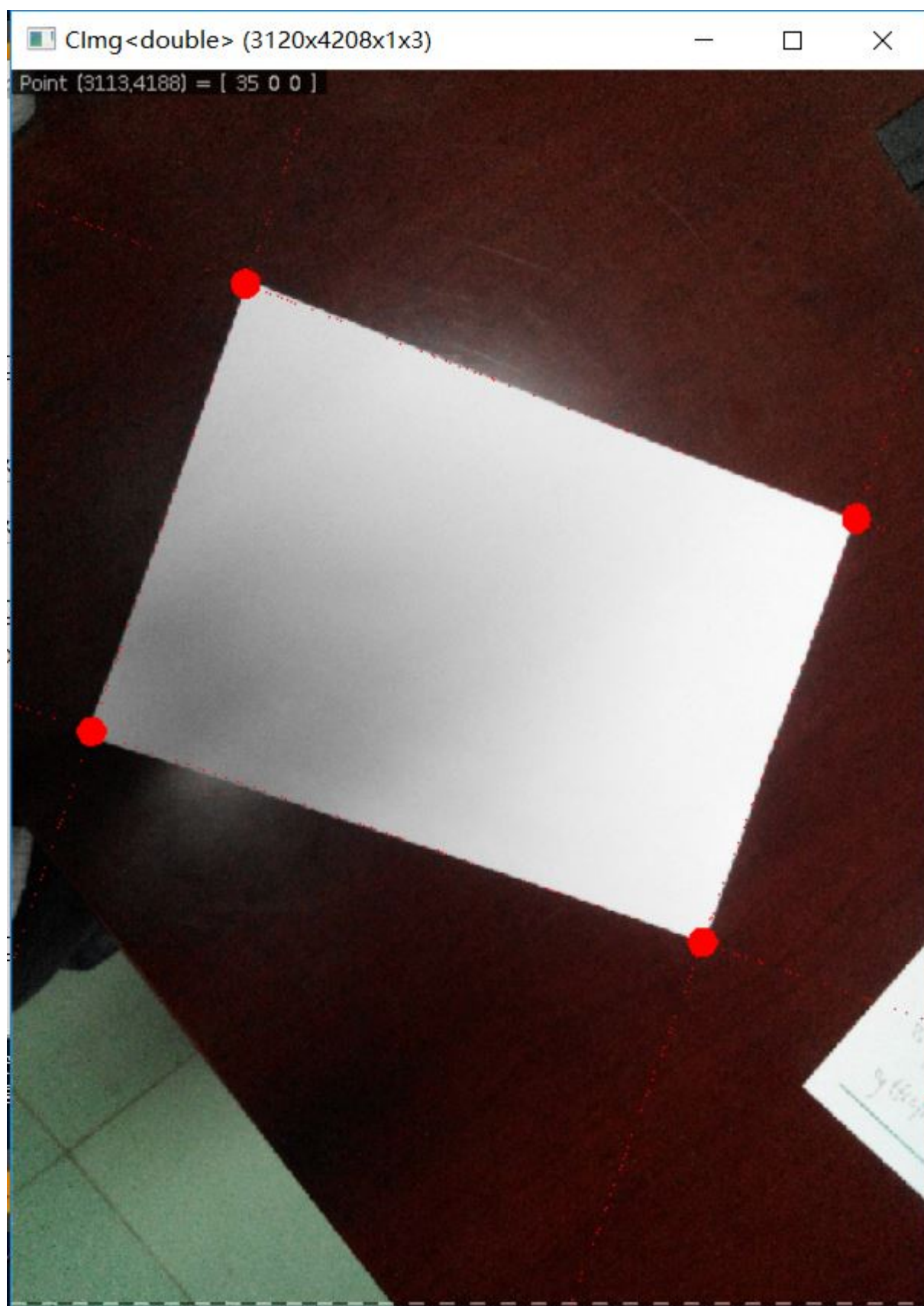
申请人填表说明:

(1) 选择“理论创新研究”类别的申请人的代表性成果须以高水平学术论文为主;选择“应用、技术与工程研究”类别的申请人的代表性成果须以决策研究及其应用成果(文科)、技术开发与工程应用成果(理、工、医科)、人才培养与应用研究成果(公共外语与体育学科)为主。

(2) 代表性学术成果请按以下相应要求填写:

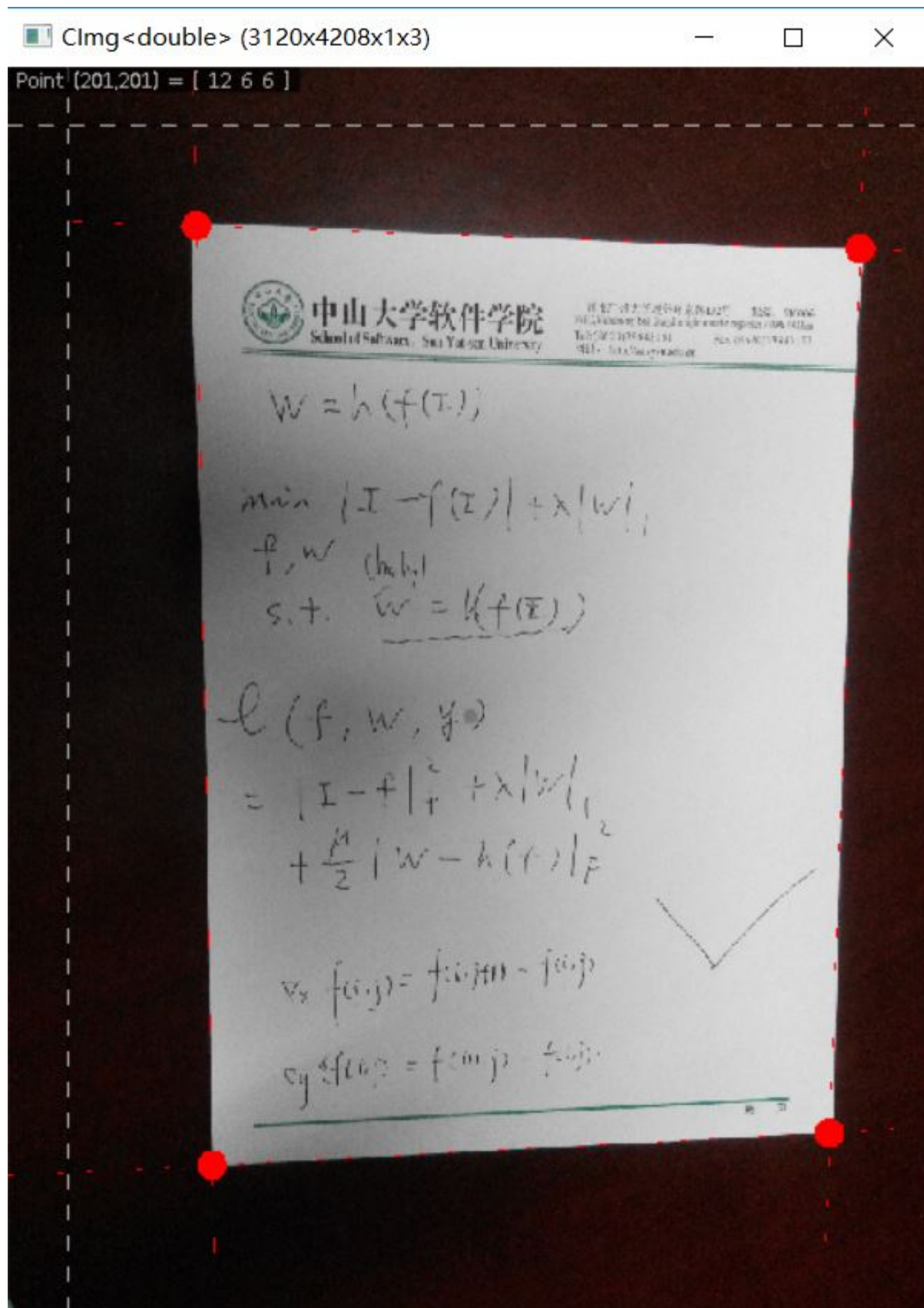
- ◆ 论文:按全部作者排序(注明学生、通讯作者、共同第一作者、共同通讯作者)、论文题目、刊物名称(注明 ISSN 号)、刊物类别、出版日期、卷号、期号、起止页码、第一作者单位列出。
- ◆ 著作:按全部作者排序、著作名称、出版单位、出版时间、总字数、本人撰写字数列出。
- ◆ 研究报告:按全部作者排序、报告题目、报告时间、服务对象、总字数、本人撰写字数列出。
- ◆ 决策咨询成果/研究成果:按全部作者排序、咨询内容、咨询时间、咨询对象、总字数、本人撰写字数列出。
- ◆ 应用性科技成果:按全部作者排序、成果名称、发布时间、奖项名称(注明奖项级别)、授奖部门、奖项等级、本人承担的工作列出。
- ◆ 国家标准/行业标准/医疗行业标准/技术标准:按全部作者排序、标准名称、标准号、发布日期、起草部门列出。
- ◆ 国家专利:按全部专利发明人排序、专利名称、专利号、专利申请日、专利类型、授权时间、专利权人列出。
- ◆ 软件著作权:按全部著作权人排序、软件名称、登记号、发表时间列出。
- ◆ 国家新药证书:按全部作者排序、药品名称、证书编号、持有人、颁布时间列出。
- ◆ 药物临床试验批件:按申请人排序、药物名称、批件号、获批时间列出。
- ◆ 医疗器械注册证书:按全部作者排序、产品名称、证书编号、持有人、批准时间、有效期列出。
- ◆ (公共外语与体育学科)指导学生参加竞赛获奖:按竞赛时间、指导学生数(专业、年级、姓名)、奖项名称(注明奖项级别)、奖项等级、本人承担的工作列出。
- ◆ 其他:按全部参与者排序、成果名称、发表时间、内容、影响列出。

(3)“说明”一栏请简要列出成果特点和影响力情况,例如刊物类别、收录引用及转载情况、奖励情况等可以证明成果影响力的内容,不超过 20 字。代表性论著的刊物类别,文科根据学校社科处颁布的《中山大学人文社会科学重要期刊目录原则(试行)》(2014 年修订版)按 A、B 重要核心期刊填写,理工医科可填写 Thomson Reuters JCR-SCI 一区、二区、三区、四区期刊收录,或中国科学院文献情报中心 JCR-SCI 一区、二区、三区、四区期刊收录,或 ESI 高引用论文或热点论文,或 EI 收录,或 ISTP 收录。





3.jpg





$$w = h(f(I))$$

$$\min_{f, w} |I - f(I)| + \lambda |w|,$$

$$f, w \text{ (待求)}$$

$$\text{s.t. } \underline{w} = h(f(I))$$

$$\ell(f, w, y)$$

$$= |I - f|^2 + \lambda |w| + \frac{\mu}{2} |w - h(f)|^2$$

$$\nabla_x f(i, j) = f(i, j+1) - f(i, j)$$

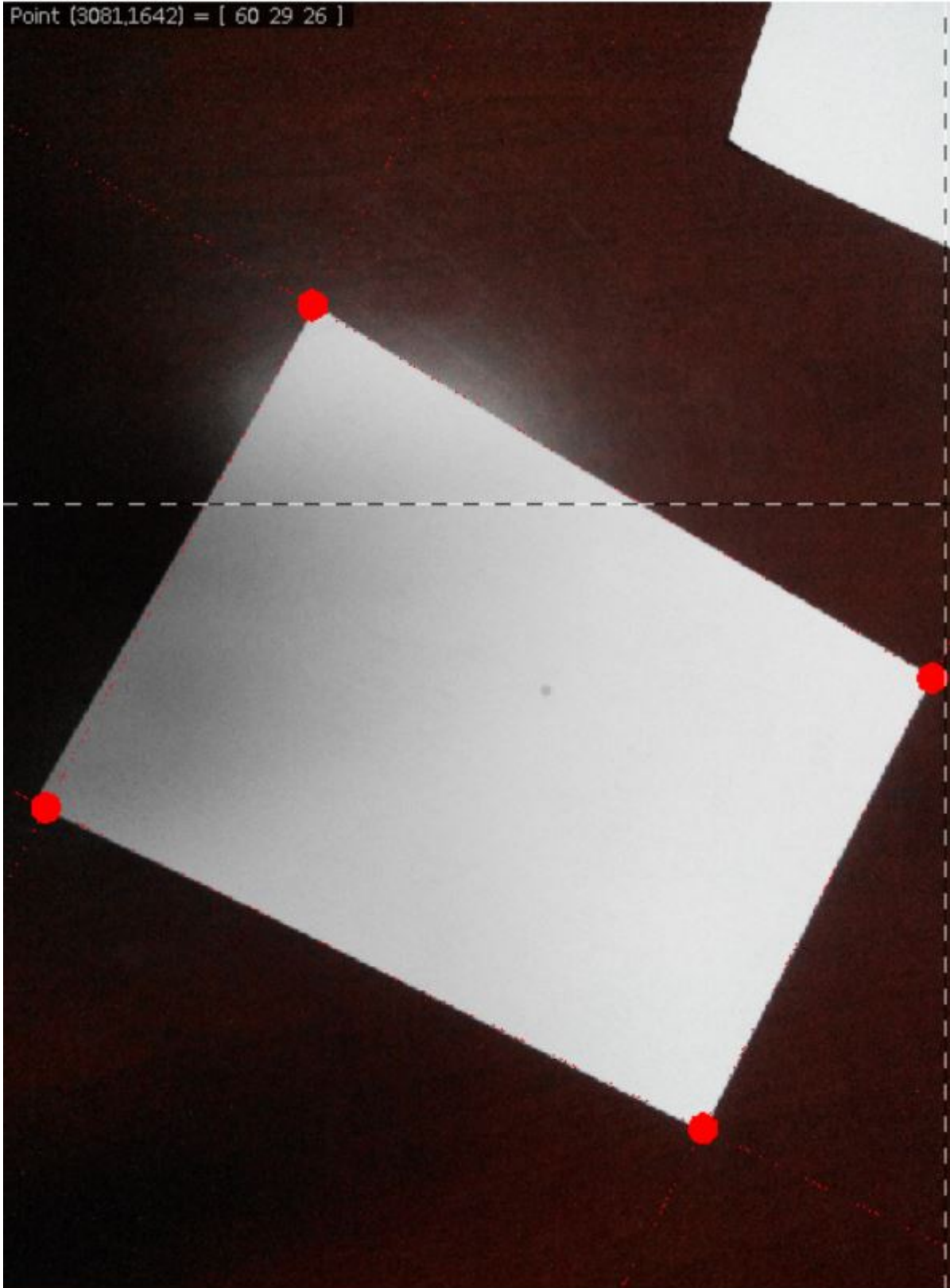
$$\nabla_y f(i, j) = f(i, j) - f(i, j-1)$$

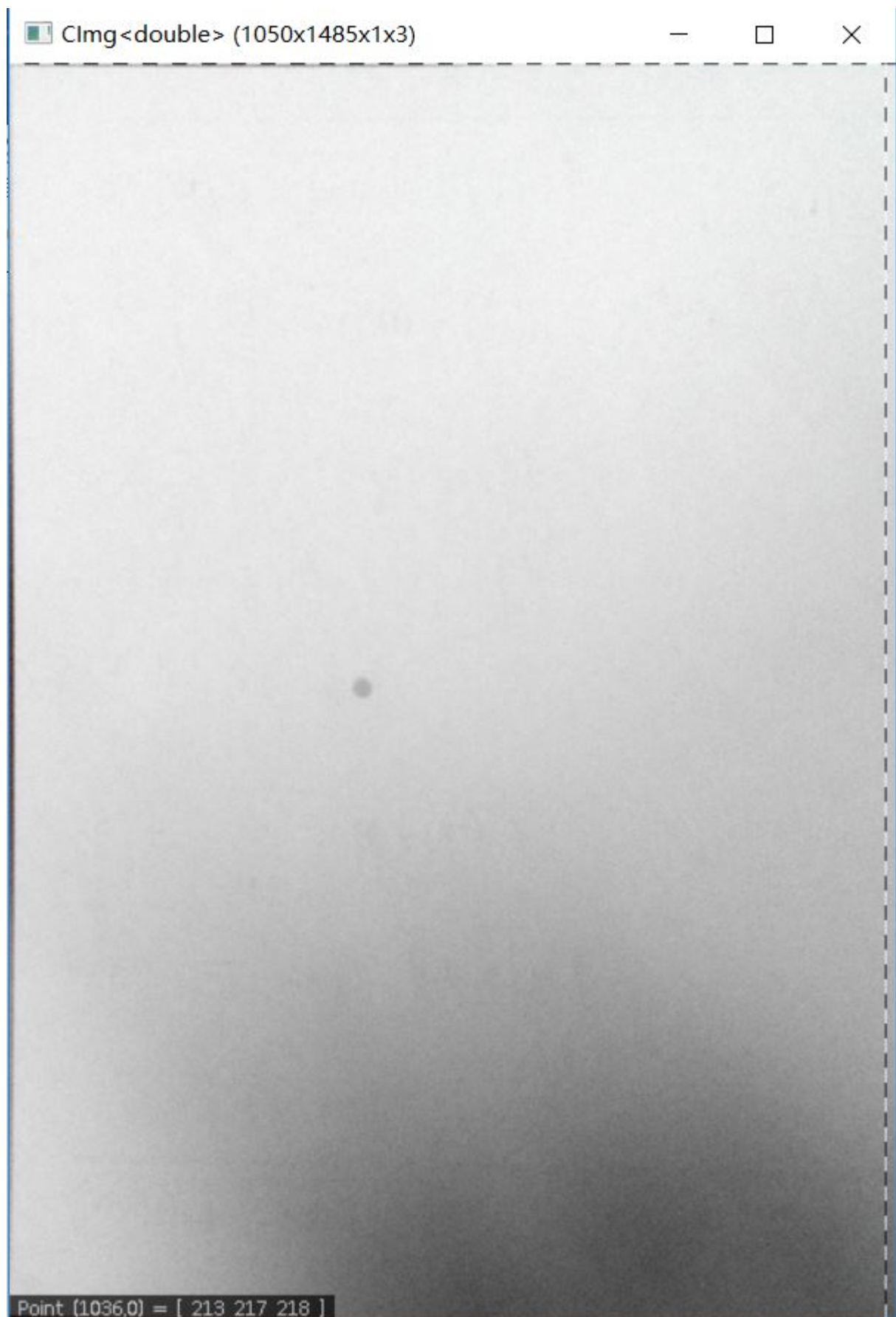


CImg<double> (3120x4208x1x3)



Point (3081,1642) = [60 29 26]





5.jpg

1) 五官 分类 (分类)

耳

0 脸型 (分类)

+ 照片



6

发

2) 根据人脸识别结果 (人脸识别) 识别



人脸识别结果 (识别)



3) 怎么识别 (识别) 识别?

识别时的条件: 人脸 背景

<1> 五官分类 (分不准)

萌

a. 脸型: (分类)

+ 照片



6.

李强

✓ 很多人说开始的原因在哪? 调研

李强

不同人有不同需求 (李强)



✓ 怎么设计? (软件 素材)

市场已有的软件?

规范 也素材

中山大学软件学院本科毕业论文 指导协议

- 一、导师负责全面指导學生本科毕业论文（以下简称论文）的撰写过程，除此之外，还应评估和指导學生确定论文的研究方向，指导修读专业课程，并为学生解答专业方向发展等相关问题。
- 二、导师应及时与学生沟通，帮助学生确定论文的题目，指导学生做好开题报告，还应要求学生在论文开题上能产生一定意义或价值。
- 三、导师应及时、认真协助学生解决论文撰写中的问题，对论文论文可存在的问题及时复修。
- 四、导师应严格按照学生论文的书写格式（另附格式文件通知），指导學生搜集对论文工作有指导作用的参考文献，督促學生按计划完成论文工作进程。
- 五、學生应与导师充分沟通，在导师指导下，确定研究工作计划，论文撰写工作计划和进度计划，撰写好正式的开题报告，经导师审查，同意签字后，提交学院备案。
- 六、學生在导师的指导下认真完成论文撰写，管理工作，在撰写论文的过程中，學生应及时向导师汇报工作进程，遇到问题及时向导师请教。
- 七、學生在开题2-3个月内进入论文中期检查阶段并提交中期检查报告，导师应对學生论文工作中存在的不足给予意见，學生应在导师的指导下，协助导师进行中期检查，在意见完成论文撰写。
- 八、學生在导师的指导下，独立完成开题论文初稿、终稿，论文要求论点正确，论据充分，具有逻辑性或创造性，经导师审查《中山大学本科生毕业论文格式规定（试行）》的要求及学院提供的《本科毕业论文》模板，做好论文的格式审查，给出指导意见和评分成绩，导师应对论文初稿格式审查和答辩工作。
- 九、导师应督促學生做好答辩准备，对學生在答辩过程中存在的问题给出指导意见，协助學生顺利完成论文答辩。
- 十、导师在指导、协助學生完成论文的过程中，撰写及答辩过程中，因學生自身问题或突发事件不能按期完成论文或未能通过论文答辩或答辩，导师应及时做好完成对學生论文的指导工作。
- 十一、导师因工作或其他原因不能指导學生的课题研究或论文撰写工作时，应及时向院系领导提出中请调整的请求，并积极配合学院调配，推荐合格人员替代导师，并做好导师工作的交接，不得私自、随意中断协议；原则上學生不得提出更换导师人选。
- 十二、本协议一式二份，导师、学院、學生各執一份，其他未尽事宜由双方协商解决。

指导教师签名:

李坤

学生本人签名:

陈伟

导师所在单位:

学生学号、所在方队:

软件学院

签名日期:

签名日期:

中山大学软件学院本科毕业论文

指导协议

一、 导师负责全面指导学生本科毕业论文（以下简称论文）的撰写过程。除此之外，导师还应负责指导学生确定论文的研究方向、指导选修专业课程，并为学生解答专业方向发展等相关问题。

二、 导师应及时与学生沟通，帮助学生确定论文的选题，指导学生做好开题报告，选题要求在理论上或应用上能产生一定意义或价值。

三、 导师应及时、认真协助学生解决论文撰写中的问题，对学生论文可能存在的问题持预见性，有效控制学生课题研究和论文撰写的顺利进行。

四、 导师应严格规范学生论文的书写格式（具体格式另行通知），指导学生搜集对论文工作有指导作用的参考文献，督促学生按计划完成论文工作进程。

五、 学生须在与导师充分交流、指导的基础上，确定研究工作计划、论文撰写工作计划和初步提纲，撰写好正式的开题报告，经师审查、同意签字后，提交学院存档。

六、 学生在导师的指导下认真完成论文撰写、答辩工作；在撰写论文的过程中，学生应及时向导师汇报工作进程，遇到问题及时向导师请教。

七、 学生在开题2-3个月内进入论文中期检查阶段并提交中期检查报告，导师应对学生论文工作中存在的不足给予意见，学生应在导师的督导、协助下对论文进行积极修改，直至完成论文撰写。

八、 学生应按照学院要求按时、独立完成并提交论文初稿、终稿。论文要求论点正确、论据充分、具有前瞻性或应用性。导师应参照《中山大学本科生毕业论文的有关规定（试行）》的要求及学院提供的《本科毕业论文》模版，做好论文的形式审查，给出评阅意见和初评成绩，学院组织对论文的形式审查和答辩工作。

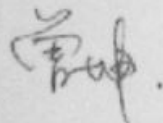
九、 导师应督促学生做好答辩准备，对学生在答辩过程中可能存在的问题给出指导意见，协助学生顺利完成论文答辩。

十、 导师在指导、协助学生完成论文的选题、撰写及答辩过程中，因学生自身问题或突发事件不能按期完成论文或未能通过论文评阅或答辩，导师仍应继续完成对学生论文的指导工作。

十一、 导师因工作或身体等原因不能指导学生的课题研究和论文撰写工作时，应及时向院领导提出中断协议的请求，并积极配合学院物色、推荐合格人选替代导师，作好导师工作的交接，不得私自、随意中断协议；原则上学生不得提出更改导师人选。

十二、 本协议一式三份，导师、学院、学生各执一份。其他未尽事宜由双方协商解决。

指导老师签名：



导师所在单位：

学生本人签名：



学生学号、所在方向：软件学院

对比第二次边缘检测的效果图：

中山大学教育科学学院本科毕业设计

指导意见

一、选题与开题报告。选题是毕业设计的关键，应结合专业培养目标、社会需求和自身兴趣，选择具有理论意义或应用价值的课题。开题报告应明确研究目的、意义、内容、方法和进度。

二、文献综述。通过查阅国内外相关文献，了解研究现状，明确研究的创新点和贡献。文献综述应条理清晰，重点突出。

三、研究方法与数据收集。根据研究内容，选择合适的研究方法，确保数据的真实性和可靠性。数据收集过程应详细记录。

四、数据分析与结果讨论。运用统计软件对收集到的数据进行分析，得出研究结论。讨论部分应结合理论和实际，对结果进行深入探讨。

五、结论与展望。总结研究成果，指出研究的不足之处，并对未来研究提出建议。结论应简洁明了，具有说服力。

六、论文撰写与答辩准备。按照学术规范撰写论文，确保格式正确、内容完整。答辩前应做好充分准备，能够清晰阐述研究过程和结论。

七、学术诚信与规范。严格遵守学术道德规范，杜绝抄袭和剽窃行为。引用他人成果时应注明出处，确保论文的原创性和学术价值。

八、其他事项。包括论文的查重、格式调整、打印装订等。答辩时应着装得体，举止大方，回答问题时思路清晰、逻辑严密。

九、导师职责。导师应全程指导学生的毕业设计工作，定期检查进度，提供必要的学术支持和资源。学生应主动与导师沟通，及时反馈研究进展。

十、总结与反思。毕业设计结束后，学生应进行自我总结和反思，总结经验教训，提高学术水平和实践能力。导师也应对学生的表现进行评价和反馈。

十一、致谢。对指导老师和提供帮助的同事、同学表示衷心的感谢。致谢部分应真诚、简洁，体现学生的感恩之心。

十二、参考文献。列出论文中引用的主要文献，确保格式规范、信息准确。参考文献应涵盖国内外相关领域的研究成果。

指导教师姓名:



学号/姓名:

学号/姓名:



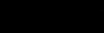
指导教师:

学号/姓名:



指导教师:

学号/姓名:



中山大学软件学院本科毕业论文

指导协议

- 一、导师负责全面指导学生本科毕业论文（以下简称论文）的撰写过程，除此之外，导师还应负责指导学生确定论文的研究方向、指导选修专业课程，并为学生解答专业方向发展等相关问题。
- 二、导师应及时与学生沟通，帮助学生确定论文的选题，指导学生做好开题报告。选题要求在理论上或应用上能产生一定意义或价值。
- 三、导师应及时、认真协助学生解决论文撰写中的问题，对学生论文可能存在的问题持预见性，有效控制学生课题研究 and 论文撰写的顺利进行。
- 四、导师应严格规范学生论文的书写格式（具体格式另行通知），指导学生搜集对论文工作有指导作用的参考文献，督促学生按计划完成论文工作进程。
- 五、学生应在与导师充分交流、指导的基础上，确定研究工作计划、论文撰写工作计划和初步提纲，撰写好正式的开题报告，经师审查、同意签字后，提交学院存档。
- 六、学生在导师的指导下认真完成论文撰写、答辩工作。在撰写论文的过程中，学生应及时向导师汇报工作进程，遇到问题及时向导师请教。
- 七、学生在开题2-3个月内进入论文中期检查阶段并提交中期检查报告，导师应对学生论文工作中存在的不足给予意见，学生应在导师的督导、协助下对论文进行积极修改，直至完成论文撰写。
- 八、学生应按学院要求按时、独立完成并提交论文初稿、终稿。论文要求论点正确、论据充分、具有前瞻性或应用性。导师应参照《中山大学本科生毕业论文的有关规定（试行）》的要求及学院提供的《本科毕业论文》模版，做好论文的形式审查，给出评阅意见和初评成绩。学院组织对论文的形式审查和答辩工作。
- 九、导师应督促学生做好答辩准备，对学生在答辩过程中可能存在的问题给出指导意见，协助学生顺利完成论文答辩。
- 十、导师在指导、协助学生完成论文的选题、撰写及答辩过程中，因学生自身问题或突发事件不能按期完成论文或未能通过论文评阅或答辩，导师仍应继续完成对学生论文的指导工作。
- 十一、导师因工作或身体等原因不能指导学生的课题研究 and 论文撰写工作时，应及时向院领导提出中断协议的请求，并积极配合学院物色、推荐合格人选替代导师，做好导师工作的交接，不得擅自、随意中断协议；原则上学生不得提出更改导师人选。
- 十二、本协议一式三份，导师、学院、学生各执一份。其他未尽事宜由双方协商解决。

指导老师签名：

导师所在单位：

签名日期：

学生本人签名：

学生学号、所在方向：软件学院

签名日期：

申请人填写说明。

(1) 选择“理论研究”类别的申请人应提交具有较高学术造诣论文；选择“应用-技术”与“应用研究”类别的申请人应提交具有较高应用研究及其应用成果(专利、技术开发与工程应用成果(理、工、医类)、人才培养与应用研究成果(公共外语与体育类))为证。

(2) 代表性学术成果应填写下列相应表格填写：

- ◆ 论文：按全部作者排序(注明学生、通讯作者、共同第一作者、共同通讯作者)、论文题目、刊物名称(注明ISSN号)、刊物类别、出版年月、卷号、期号、起止页码、第一作者单位列出。
- ◆ 著作：按全部作者排序、著作名称、出版年份、出版单位、总字数、本人撰写字数列出。
- ◆ 研究报告：按全部作者排序、报告题目、报告时间、报告对象、总字数、本人撰写字数列出。
- ◆ 决策咨询成果/研究成果：按全部作者排序、报告内容、报告时间、报告对象、总字数、本人撰写字数列出。
- ◆ 应用性科技成果：按全部作者排序、成果名称、发表日期、奖项名称(注明奖项级别)、获奖部门、奖项等级、本人承担的工作列出。
- ◆ 国家标准/行业标准/行业标准/技术标准：按全部作者排序、标准名称、标准号、发布日期、起草部门列出。
- ◆ 国家专利：按全部作者排序、专利名称、专利号、专利申报日、专利类型、授权时间、专利权人列出。
- ◆ 软件著作权：按全部作者排序、软件名称、登记号、发证日期列出。
- ◆ 国家新药证书：按全部作者排序、药品名称、新药编号、持有人、注册日期列出。
- ◆ 药物临床试验材料：按申请人排序、药物名称、批件号、核准日期列出。
- ◆ 发明专利使用证书：按全部作者排序、产品名称、证书编号、持有人、核准日期、有效期列出。
- ◆ 《公共外语与体育类》类学生参加竞赛获奖：获奖时间、竞赛等级(专业、全国、外省)、奖项名称(注明奖项级别)、获奖等级、本人承担的工作列出。
- ◆ 专利：按全部作者排序、成果名称、发表日期、内容、专利号。

(3)“说明”栏填写成果特点和影响力情况，例如基金类别、成果部门及经费情况、奖励情况等可注明成果影响力的内容，不超过20字。所发表论文的刊物类别，文科参照学校社科类论文的《中山大学人文社会科学重要期刊目录》(执行《2014年版目录》)按A、B、C类重要期刊排序，理工医和农林工 Thompson Reuters JCR-SCI一区、二区、三区、四区期刊收录，或中国科学院文献情报中心JCR-SCI一区、二区、三区、四区期刊收录，或ESI高引区的文献源点论文，或EI收录，或ISTP收录。

Point (923,200) = [187 187 187]

可以看出还是边缘检测图的效果要好一些，主要是霍夫变换检测出的直线更加贴合实际的边，所以进行矫正的效果也更好。

2. 手写字体的检测

(1) Adaboost进行手写字体的检测

导入mnist数据集

```
import tensorflow.examples.tutorials.mnist.input_data as input_data
data_dir = '../MNIST_data/'
mnist = input_data.read_data_sets(data_dir, one_hot=False)
batch_size = 50000
test_x = mnist.test.images[:10000]
test_y = mnist.test.labels[:10000]
```

一共60000个数据集，取50000用于训练，10000用于测试训练出的模型。

调用Adaboost分类器进行训练：

```
batch_x, batch_y = mnist.train.next_batch(batch_size)
clf_rf = AdaBoostClassifier(n_estimators = 60)
clf_rf.fit(batch_x, batch_y)
```

评估预测的效果：

```
y_pred_rf = clf_rf.predict(test_x)
acc_rf = accuracy_score(test_y, y_pred_rf)
print("%s n_estimators = 60, accuracy:%f" % (datetime.now(), acc_rf))
```

选取较好的参数(弱分类器数量)：

先通过调节弱分类器数量来获得一个训练效果比较不错的数量参数(虽然之后发现预测率好像是改变的)

```
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
start Gradient Boosting
2018-12-19 11:09:24.558654 n_estimators = 10, random forest accuracy:0.595900
2018-12-19 11:10:07.601420 n_estimators = 20, random forest accuracy:0.682300
2018-12-19 11:11:09.929975 n_estimators = 30, random forest accuracy:0.686700
2018-12-19 11:12:33.281707 n_estimators = 40, random forest accuracy:0.710200
2018-12-19 11:14:17.427947 n_estimators = 50, random forest accuracy:0.724100
2018-12-19 11:16:22.440998 n_estimators = 60, random forest accuracy:0.726700
2018-12-19 11:18:46.291313 n_estimators = 70, random forest accuracy:0.721500
2018-12-19 11:21:29.213744 n_estimators = 80, random forest accuracy:0.720000
2018-12-19 11:24:33.746690 n_estimators = 90, random forest accuracy:0.726500
2018-12-19 11:28:07.282343 n_estimators = 100, random forest accuracy:0.721600
2018-12-19 11:31:48.173228 n_estimators = 110, random forest accuracy:0.703500
2018-12-19 11:35:36.821454 n_estimators = 120, random forest accuracy:0.710300
2018-12-19 11:39:57.086900 n_estimators = 130, random forest accuracy:0.723000
2018-12-19 11:44:48.909089 n_estimators = 140, random forest accuracy:0.720800
2018-12-19 11:49:55.488914 n_estimators = 150, random forest accuracy:0.723200
2018-12-19 13:06:17.475205 n_estimators = 160, random forest accuracy:0.716200
2018-12-19 13:09:35.562056 n_estimators = 170, random forest accuracy:0.715200
2018-12-19 13:13:09.355173 n_estimators = 180, random forest accuracy:0.702300
2018-12-19 13:17:41.520215 n_estimators = 190, random forest accuracy:0.705800
Total time 7718.69 s
```

之后选取弱分类器参数为60进行训练

在对自己的手写图片读入前先进行几步处理：

- 将图片转为二值图
- resize为mnist训练集要求的(28*28)尺寸
- 将图像进行膨胀处理

```
img = cv2.resize(img, (28, 28), interpolation=cv2.INTER_CUBIC)
GrayImage = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret,thresh2=cv2.threshold(GrayImage,127,255,cv2.THRESH_BINARY_INV)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(3, 3))
img = cv2.dilate(thresh2,kernel)
```

处理好的图片效果如下：

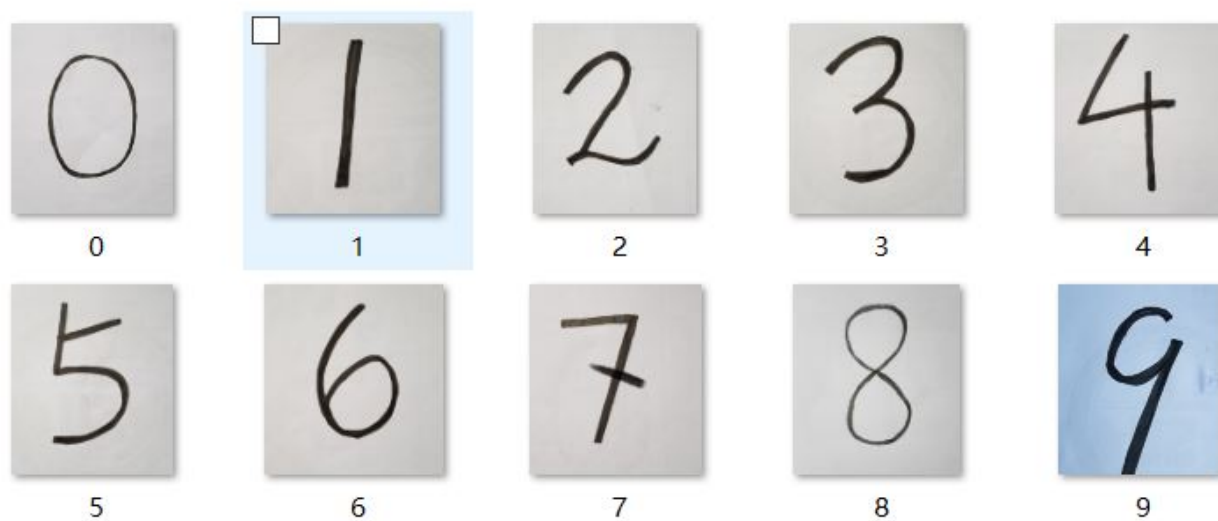


同时有一点很坑的要注意就是mnist要识别的是西方的手写字体，和我们常写的有一点区别。

下面是用于读入的手写数字



进行一步裁剪处理:



下面是预测效果:

```
Instructions for updating.  
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.  
2018-12-20 17:35:06.125152 n_estimators = 60, accuracy:0.752300  
Total time 74.85 s  
the digit is: 0  
the result is: 0  
the digit is: 1  
the result is: 1  
the digit is: 2  
the result is: 2  
the digit is: 3  
the result is: 3  
the digit is: 4  
the result is: 3  
the digit is: 5  
the result is: 5  
the digit is: 6  
the result is: 6  
the digit is: 7  
the result is: 1  
the digit is: 8  
the result is: 6  
the digit is: 9  
the result is: 1
```

test集的准确率为75%左右，自己手写数字的准确率为60%，这是一个比较不期望的结果，于是考虑用一个自己搭建的CNN网络对模型进行测试(参考tensorflow中文社区)

(2) 卷积神经网络(CNN)进行手写字体的检测

1. 权重初始化

```
def weight_variable(shape):  
    initial = tf.truncated_normal(shape, stddev=0.1)  
    return tf.Variable(initial)  
  
def bias_variable(shape):  
    initial = tf.constant(0.1, shape=shape)  
    return tf.Variable(initial)
```

2. 卷积和池化

```
def conv2d(x, W):  
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')  
  
def max_pool_2x2(x):  
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')
```

3. 第一层卷积

```

W_conv1 = weight_variable([5, 5, 1, 32])
b_conv1 = bias_variable([32])

x_image = tf.reshape(x, [-1,28,28,1])
h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)

```

4.第二层卷积

```

W_conv2 = weight_variable([5, 5, 32, 64])
b_conv2 = bias_variable([64])

h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
h_pool2 = max_pool_2x2(h_conv2)

```

5. 密集连接层

```

W_fc1 = weight_variable([7 * 7 * 64, 1024])
b_fc1 = bias_variable([1024])

h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1) + b_fc1)

```

6. Dropout

(屏蔽神经元的输出以及自动处理神经元输出值的scale)

```

keep_prob = tf.placeholder(tf.float32)
h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)

```

7. 输出层

```

W_fc2 = weight_variable([1024, 10])
b_fc2 = bias_variable([10])

y_conv=tf.nn.softmax(tf.matmul(h_fc1_drop, W_fc2) + b_fc2)

```

8. 训练和评估模型

```

cross_entropy = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y_conv))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y_conv,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

saver = tf.train.Saver() # defaults to saving all variables

```

```

sess.run(tf.global_variables_initializer())
for i in range(20000):
    batch = mnist.train.next_batch(50)
    if i%100 == 0:
        train_accuracy = accuracy.eval(feed_dict={
            x:batch[0], y_: batch[1], keep_prob: 1.0})
        print("step %d, training accuracy %g"%(i, train_accuracy))

    train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})
saver.save(sess, './model/model.ckpt') #保存模型参数, 注意把这里改为自己的路径





print("test accuracy %g"%accuracy.eval(feed_dict={
    x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))

```

最终跑出的模型准确度接近99.3%

训练出的模型：

电脑 > 桌面 > adaboost > model

<input type="checkbox"/> 名称	修改日期	类型	大小
 checkpoint	2018/12/20 14:34	文件	1 KB
 model.ckpt.data-00000-of-00001	2018/12/20 14:34	DATA-00000-OF...	38,406 KB
 model.ckpt.index	2018/12/20 14:34	INDEX 文件	1 KB
 model.ckpt.meta	2018/12/20 14:34	META 文件	92 KB

9. 用存储的model对自己的手写字体进行测试：

```

init_op = tf.initialize_all_variables()

saver = tf.train.Saver()
with tf.Session() as sess:
    sess.run(init_op)
    saver.restore(sess, "./model/model.ckpt") #这里使用了之前保存的模型参数

    prediction=tf.argmax(y_conv,1)
    predint=prediction.eval(feed_dict={x: [result],keep_prob: 1.0}, session=sess)
    print('the digit is 9: ')
    print('recognize result:')
    print(predint[0])

```

10. 效果：

(最终效果10张图全部都预测准确，再测了几张写得不是特别规范的数字也基本都识别正确)


```
the digit is 0:  
recognize result:  
0
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 1:  
recognize result:  
1
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 2:  
recognize result:  
2
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 3:  
recognize result:  
3
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 4:  
recognize result:  
4
```

```
Use tf.global_variables_initializer instead.  
the digit is 5:  
recognize result:  
5
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 6:  
recognize result:  
6
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 7:  
recognize result:  
7
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 8:  
recognize result:  
8
```

```
(py35) C:\Users\Administrator\Desktop\adaboost  
D:\Anaconda\envs\py35\lib\site-packages\h5py\  
ype from 'float' to 'np.floating' is deprecate  
from ._conv import register_converters as _r  
WARNING:tensorflow:From D:\Anaconda\envs\py35\  
all_variables (from tensorflow.python.ops.vari  
Instructions for updating:  
Use 'tf.global_variables_initializer' instead.  
the digit is 9:  
recognize result:  
9
```