



# (12) 发明专利

(10) 授权公告号 CN 110533030 B

(45) 授权公告日 2023. 07. 14

(21) 申请号 201910765276.1

(22) 申请日 2019.08.19

(65) 同一申请的已公布的文献号  
申请公布号 CN 110533030 A

(43) 申请公布日 2019.12.03

(73) 专利权人 三峡大学  
地址 443002 湖北省宜昌市西陵区大学路8号

(72) 发明人 曾曙光 左肖雄 郑胜 张佳锋  
曾祥云

(74) 专利代理机构 宜昌市三峡专利事务所  
42103  
专利代理师 吴思高

(51) Int. Cl.  
G06V 30/19 (2022.01)  
G06V 30/148 (2022.01)  
G06V 10/26 (2022.01)  
G06N 3/0464 (2023.01)  
G06N 3/08 (2023.01)  
G06N 3/048 (2023.01)  
G06N 3/084 (2023.01)

(56) 对比文件

CN 101751568 A, 2010.06.23  
CN 105528606 A, 2016.04.27  
CN 105528600 A, 2016.04.27  
WO 2017020723 A1, 2017.02.09  
CN 106611174 A, 2017.05.03  
CN 108921163 A, 2018.11.30  
US 6026177 A, 2000.02.15  
US 2015278626 A1, 2015.10.01  
CN 109784342 A, 2019.05.21  
CN 109359695 A, 2019.02.19  
曾祥云, 郑胜等. 基于SVM手绘太阳黑子图像背景提取方法.《微型机与应用》.2016, 全文.  
A.Krizhevsky, I.Sutskever, G.E. Hinton. Imagenet classification with deep convolutional neural networks.《Proceedings of the 25th International Conference on Neural Information Processing Systems》.2012, 全文.  
朱道远; 郑胜; 曾祥云; 徐高贵. 手绘太阳黑子图手写字符分割方法研究. 微型机与应用. 2015, (20), 全文. (续)

审查员 刘霞

权利要求书3页 说明书9页 附图8页

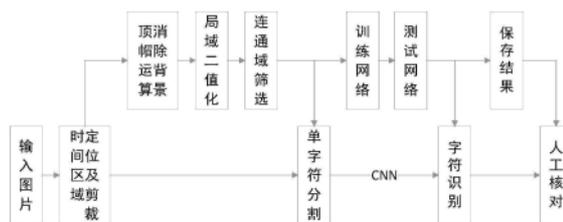
## (54) 发明名称

基于深度学习的太阳胶片图像时间戳信息提取方法

## (57) 摘要

基于深度学习的太阳胶片图像时间戳信息提取方法, 包括步骤1: 对太阳色球胶片图像中的时间戳信息区域定位、并剪裁; 时间戳信息为太阳色球胶片图像中用于表示拍摄时间的年、月、日、时、分信息; 步骤2: 单字符分割, 即将时间戳信息区域的字符进一步分割, 得到单个字符; 步骤3: 字符识别, 即先采用大量的样本训练网络, 然后将步骤2分割得到的单个字符使用训练好的网络对其进行识别, 并将其识别结果整合并保存。该方法对太阳观测图像胶片中的数字时间戳

进行机器自动识别, 并将识别出的时间信息输出。以减少人工识别和编写时间信息的工作量, 从而可加快这批胶片数据的数字化进程, 使这些珍贵的历史资料可以更便捷地用于太阳物理研究。



CN 110533030 B

[接上页]

(56) 对比文件

唐铭豆;陶青川;冯谦.基于神经网络的芯片

表面字符检测识别系统.现代计算机(专业版)

.2018, (09), 全文.

1. 基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于包括以下步骤:

步骤1:对太阳色球胶片图像中的时间戳信息区域定位、并剪裁;

步骤2:单字符分割,即将时间戳信息区域的字符进一步分割,得到单个字符;

步骤3:字符识别,即先采用大量的样本训练网络,然后将步骤2分割得到的单个字符使用训练好的网络对其进行识别,并将其识别结果整合并保存;

所述步骤1包含以下步骤:

步骤1.1,基于垂直投影的太阳球面切除步骤:

对图像在垂直分量上进行累加操作得到一个 $1 \times n$ 的向量,假设图片的大小是 $m \times n$ ,像素在 $i$ 行 $j$ 列的像素值为 $f_{ij}(x, y)$ ,则在垂直方向投影为:

$$S_{1j} = \sum_{i=1}^m f_{ij}(x, y)$$

其中, $S_{1j}$ 表示图像第 $i$ 列像素点求和的结果, $S_{1j}$ 的大小为 $1 \times n$ ;通过计算图片在垂直方向的投影可以进一步判断太阳球面的位置;在 $S_{1j}$ 向量 $[400, 1800]$ 之间为太阳色球部分投影的结果;又因为太阳是对称的,所以只需要得知在向量 $S_{1j}$ 中的最大值,即可定位太阳圆心的位置,然后根据太阳球面在垂直方向所占的像素长度,将包含太阳球面部分的图片去除;

步骤1.2,基于方差判断时间戳位置及翻转校正:

包含时间戳的图片的方差,远远大于不包含时间戳的图片的方差;从而判断出时间戳所在的图片,得知时间戳所在图片后,需要对图片进行反转校正,若为左侧图片则需顺时针翻转90度,反之逆时针旋转90度;

步骤1.3,基于投影法的时间戳字符区域精分割:

对于大小为 $m \times n$ 的图片,像素在 $i$ 行 $j$ 列的像素值为 $x_{ij}$ ,则在水平方向和垂直方向投影分别为:

$$S_{1j} = \sum_{i=1}^m x_{ij}$$

$$S_{i1} = \sum_{j=1}^n x_{ij}$$

其中, $S_{1j}$ 表示图像第 $i$ 列像素点求和的结果, $S_{1j}$ 的大小为 $1 \times n$ ;  $S_{i1}$ 表示图像第 $j$ 行像素点求和的结果, $S_{i1}$ 的大小为 $m \times 1$ ;通过计算图片在水平方向和垂直方向的投影,能够进一步时间戳区域的具体位置从而实现图片的精确分割;所述步骤3中,字符识别的过程为:

采用深度学习中的卷积神经网络算法来进行字符识别,搭建的字符识别的卷积神经网络包含两个卷积层,两个池化层和一个全连接层,第一个卷积层通过6个尺寸为 $5 \times 5$ 的不同的卷积核对输入尺寸为 $28 \times 28$ 的字符图片进行卷积,经过第一层卷积后,原字符图片变为 $24 \times 24 \times 6$ 的特征图;第一个池化层采用滑动窗口为 $2 \times 2$ 的池化函数对第一个卷积层的结果进行特征再提取,经历该层池化后,变为 $12 \times 12 \times 6$ 的特征图;第二个卷积层采用12个尺寸为 $5 \times 5$ 的不同卷积核对池化层的特征图进行特征再提取,经提取后的特征图尺寸为 $8 \times 8 \times 12$ ;第二个池化层对第二个卷积层卷积后的特征图进行池化,池化后特征图尺寸变为 $4 \times 4 \times 12$ ;将第二次池化操作后的特征图输入全连接层,得到该字符的特征向量;最后将字符的特征向量进行分类并与实际数字相对应便完成了时间戳中时间字符识别;将训练得到的卷积神经网络,对太阳色球胶片图中的时间信息中的单个字符进行识别,将识别出来的字

符按顺序组合起来与原图的文件名相对应并填入Excel表中,用于后期人工核对以及建立数据库。

2. 根据权利要求1所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:基于步骤1.3得到的图片在水平和垂直方向的投影结果,对图片进行切割:为了保证图片的连续性不被破坏,以第一个大于均值的点作为起点,最后一个大于均值的点作为终点,保留起点到终点区域的所有图像;假设原图S的大小为 $m \times n$ ,切割后图片P的大小为 $m' \times n'$ ,其切割公式为:

$$P=S(a:b,c:d), (a,c>1,b<m,d<n)$$

其中:

$$\begin{cases} a = \min(x > \bar{x}) \\ b = \max(x > \bar{x}) \\ c = \min(y > \bar{y}) \\ d = \max(y > \bar{y}) \end{cases}$$

式中,S(a:b,c:d)表示图片S中,a到b行,c到d列;x表示在水平投影中某点的值, $\bar{x}$ 表示水平投影中各点均值;y表示在垂直投影中某点的值, $\bar{y}$ 表示垂直投影中各点均值; $\min(x > \bar{x})$ 表示在投影向量中,x大于 $\bar{x}$ 的位置的最小值; $\max(x > \bar{x})$ 表示在投影向量中,x大于 $\bar{x}$ 的位置的最大值。

3. 根据权利要求1所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:所述步骤2中,单字符分割的过程为:

首先采用顶帽运算对图片的背景进行去除,然后采用局部二值化算法去除噪声,最后采用连通域算法提取字符区域;该算法默认字符颜色为白色,若字符颜色为黑色则经过连通域提取后将提取不到任何有效区域,所以若不存在有效区域则返回到局部二值化部分,对局部二值化后的图片进行颜色翻转即可。

4. 根据权利要求3所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:

采用顶帽运算算法进行背景去除,其算法原理是原图像与原图像开运算结果做差;图片经过顶帽运算后能够消除一部分背景噪声,并且能够突出图像中的字符。

5. 根据权利要求3所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:

采用Sauvola局部二值化算法进行噪声去除;只需要将符合字符大小的连通域提取出来就可以完成字符的切割。

6. 根据权利要求3所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:

采用连通域算法进行字符区域提取,通过判断每个连通域是否符合标准字符的长度,宽度以及长宽比来进一步删除无效区域;字符的高度在[90,110]之间,字符宽度在[10,60]之间,字符的长宽比不小于1;

根据二值图中每个连通域的位置对应原图中的位置,能够将原图中的单个字符区域分

别切割出来;为了保证得到的图片尺寸一致,该算法分别对每张图片进行填充,并将其大小变换为 $28 \times 28$ 的标准图片。

7.根据权利要求1所述基于深度学习的太阳胶片图像时间戳信息提取方法,其特征在于:还包括步骤4,人工核对日期:

对于一段时间内的色球图像,只需要输入第一张图像的年、月、日信息,能够自动推算得出之后的每张图片的拍摄日期;中间偶尔会存在一些未进行太阳观测的日期,采用人工核对的方式,将自动生成的有误的日期信息进行修改。

## 基于深度学习的太阳胶片图像时间戳信息提取方法

### 技术领域

[0001] 本发明涉及太阳观测图像处理技术领域,具体涉及一种基于深度学习的太阳胶片图像时间戳信息提取方法。

### 背景技术

[0002] 太阳色球层是光球层之上的一层大气,作为光球层到日冕层的过渡区,磁场不太稳定,经常产生剧烈的耀斑爆发现象。色球中太阳耀斑的辐射常常以细长的带状出现在磁极反转线(PIL)的两侧,这被认为是磁重联的典型形态的证据。为了研究耀斑爆发现象,需相关人员长期对太阳色球进行拍摄记录。因历史资料数量巨大,很大一批色球图像的时间信息仍是以图像形式呈现在数据中,并未形成可供计算机直接读取的数字信息。这对利用这些资料的展开科学研究带来很大的不便。

[0003] 图像的拍摄时间数字化一方面加深了对历史资料有效信息的挖掘,另一方面可以极大的减轻科研人员的检索工作量,以便于他们从这些数据中得到更多有价值的结果,对科研工作的进展有很大的帮助。

[0004] 历史上的太阳观测图像多采用胶片保存,且其拍摄时间印在胶片上。为了便于科研人员有效的使用这些图像资料,有必要将胶片中的时间戳信息提取出来。图片数量十分庞大,人工识别和提取费时费力。因此,利用计算机自动识别图像中的时间戳信息,是能否高效利用这些资料的关键所在。

### 发明内容

[0005] 为解决上述技术问题,本发明提供一种基于深度学习的太阳胶片图像时间戳信息提取方法,该方法对太阳观测图像胶片中的数字时间戳进行机器自动识别,并将识别出的时间信息输出,以减少人工识别和录入时间信息的工作量。从而可加快这批胶片数据的数字化进程,使这些珍贵的历史资料可以更便捷地用于太阳物理研究。

[0006] 本发明采取的技术方案为:

[0007] 基于深度学习的太阳胶片图像时间戳信息提取方法,包括以下步骤:

[0008] 步骤1:对太阳色球胶片图像中的时间戳信息区域定位、并剪裁;

[0009] 时间戳信息为太阳色球胶片图像中用于表示拍摄时间的年、月、日、时、分信息;

[0010] 步骤2:单字符分割,即将时间戳信息区域的字符进一步分割,得到单个字符;

[0011] 步骤3:字符识别,即先采用大量的样本训练网络,然后将步骤2分割得到的单个字符使用训练好的网络对其进行识别,并将其识别结果整合并保存。

[0012] 所述步骤1包含以下步骤:

[0013] 步骤1.1,基于垂直投影的太阳球面切除步骤:

[0014] 对图像在垂直分量上进行累加操作得到一个 $1 \times n$ 的向量,假设图片的大小是 $m \times n$ ,像素在 $i$ 行 $j$ 列的像素值为 $f_{i,j}(x,y)$ ,则在垂直方向投影为:

$$[0015] \quad S_{1j} = \sum_{i=1}^m f_{ij}(x, y)$$

[0016] 其中,  $S_{1j}$  表示图像第  $i$  列像素点求和的结果,  $S_{1j}$  的大小为  $1 \times n$ 。通过计算图片在垂直方向的投影可以进一步判断太阳球面的位置。在  $S_{1j}$  向量  $[400, 1800]$  之间为太阳色球部分投影的结果。又因为太阳是对称的, 所以只需要得知在向量  $S_{1j}$  中的最大值, 即可定位太阳圆心的位置, 然后根据太阳球面在垂直方向所占的像素长度, 将包含太阳球面部分的图片去除。

[0017] 步骤1.2, 基于方差判断时间戳位置及翻转校正:

[0018] 包含时间戳的图片的方差, 远远大于不包含时间戳的图片的方差。从而判断出时间戳所在的图片, 得知时间戳所在图片后, 需要对图片进行反转校正, 若为左侧图片则需顺时针翻转90度, 反之逆时针旋转90度。

[0019] 步骤1.3, 基于投影法的时间戳字符区域精分割:

[0020] 对于大小为  $m \times n$  的图片, 像素在  $i$  行  $j$  列的像素值为  $x_{ij}$ , 则在水平方向和垂直方向投影分别为:

$$[0021] \quad S_{1j} = \sum_{i=1}^m x_{ij}$$

$$[0022] \quad S_{i1} = \sum_{j=1}^n x_{ij}$$

[0023] 其中,  $S_{1j}$  表示图像第  $i$  列像素点求和的结果,  $S_{1j}$  的大小为  $1 \times n$ 。 $S_{i1}$  表示图像第  $j$  行像素点求和的结果,  $S_{i1}$  的大小为  $m \times 1$ 。通过计算图片在水平方向和垂直方向的投影, 能够进一步时间戳区域的具体位置从而实现图片的精确分割。

[0024] 步骤1.4, 对时间戳区域进行切割:

[0025] 基于图片在水平和垂直方向的投影结果, 对图片进行切割。为了保证图片的连续性不被破坏, 所以以第一个大于均值的点作为起点, 最后一个大于均值的点作为终点, 保留起点到终点区域的所有图像。假设原图  $S$  的大小为  $m \times n$ , 切割后图片  $P$  的大小为  $m' \times n'$ , 其切割公式为:

$$[0026] \quad P = S(a:b, c:d), (a, c > 1, b < m, d < n)$$

[0027] 其中:

$$[0028] \quad \begin{cases} a = \min(x > \bar{x}) \\ b = \max(x > \bar{x}) \\ c = \min(y > \bar{y}) \\ d = \max(y > \bar{y}) \end{cases}$$

[0029] 式中,  $S(a:b, c:d)$  表示图片  $S$  中,  $a$  到  $b$  行,  $c$  到  $d$  列。 $x$  表示在水平投影中某点的值,  $\bar{x}$  表示水平投影中各点均值。 $y$  表示在垂直投影中某点的值,  $\bar{y}$  表示垂直投影中各点均值。 $\min(x > \bar{x})$  表示在投影向量中,  $x$  大于  $\bar{x}$  的位置的最小值。 $\max(x > \bar{x})$  表示在投影向量中,  $x$  大于  $\bar{x}$  的位置的最大值, 其他同理。

[0030] 所述步骤2中, 单字符分割的过程为:

[0031] 首先采用顶帽运算对图片的背景进行去除,然后采用局部二值化算法去除噪声,最后采用连通域算法提取字符区域;该算法默认字符颜色为白色,若字符颜色为黑色则经过连通域提取后将提取不到任何有效区域,所以若不存在有效区域则返回到局部二值化部分,对局部二值化后的图片进行颜色翻转即可。

[0032] 采用顶帽运算算法进行背景去除,其算法原理是原图像与原图像开运算结果做差。图片经过顶帽运算后可以消除一部分背景噪声,并且可以突出图像中的字符。

[0033] 采用Sauvola局部二值化算法进行噪声去除。只需要将符合字符大小的连通域提取出来就可以完成字符的切割。并且采用连通域提取的方法速度快,可以节约很多程序运行时间。

[0034] 采用连通域算法进行字符区域提取,先对图片进行局部二值化,然后去掉过大或过小的连通域,以消除部分干扰。局部二值化阈值为 $16 \times 16$ 时效果较好,字符的连通域面积均处于 $[500, 5000]$ 之间。经过上述处理之后仍然有一些无效区域没有消除,因此本发明通过判断每个连通域是否符合标准字符的长度,宽度以及长宽比来进一步删除无效区域。由统计信息,字符的高度在 $[90, 110]$ 之间,字符宽度在 $[10, 60]$ 之间,字符的长宽比不小于1。

[0035] 根据二值图中每个连通域的位置对应原图中的位置,便可以将原图中的单个字符区域分别切割出来。为了保证得到的图片尺寸一致,本发明分别对每张图片进行填充并将其大小变换为 $28 \times 28$ 的标准图片。

[0036] 所述步骤3中,字符识别的过程为:

[0037] 采用深度学习中的卷积神经网络算法来进行字符识别。本发明搭建的字符识别的卷积神经网络包含两个卷积层,两个池化层和一个全连接层。第一个卷积层通过6个尺寸为 $5 \times 5$ 的不同的卷积核对输入尺寸为 $28 \times 28$ 的字符图片进行卷积,经过第一层卷积后,原字符图片变为 $24 \times 24 \times 6$ 的特征图。第一个池化层采用滑动窗口为 $2 \times 2$ 的池化函数对第一个卷积层的结果进行特征再提取,经历该层池化后,变为 $12 \times 12 \times 6$ 的特征图。第二个卷积层采用12个尺寸为 $5 \times 5$ 的不同卷积核对池化层的特征图进行特征再提取,经提取后的特征图尺寸为 $8 \times 8 \times 12$ 。第二个池化层对第二个卷积层卷积后的特征图进行池化,池化后特征图尺寸变为 $4 \times 4 \times 12$ 。将第二次池化操作后的特征图输入全连接层,得到该字符的特征向量。最后将字符的特征向量进行分类并与实际数字相对应便完成了时间戳中时间字符识别。

[0038] 将训练得到的卷积神经网络对太阳色球胶片图中的时间信息中的单个字符进行识别,将识别出来的字符按顺序组合起来与原图的文件名相对应并填入Excel表中,用于后期人工核对以及建立数据库。

[0039] 还包括步骤4,人工核对日期:

[0040] 对于一段时间内的色球图像,只需要输入第一张图像的年月日信息,便可以自动推算得出之后的每张图片的拍摄日期。中间偶尔会存在一些未进行太阳观测的日期,这时需要采用人工核对的方式,将自动生成的有误的日期信息进行修改。

[0041] 本发明一种基于深度学习的太阳胶片图像时间戳信息提取方法,技术效果如下:

[0042] 1)、本发明提出了一种基于深度学习的时间戳信息提取方法,用来系统地识别和整理美国国立太阳天文台1956-2003年间经过扫描数字化的700多万张太阳色球胶片图像资料的时间信息。首先,定位图像中时间戳信息区域并将其分割;其次,采用顶帽运算、局域二值化、连通域筛选等方法消除噪声的干扰,对时间戳信息区域图像进行字符分割;然后,

选取10000张已经分类的字符图片训练卷积神经网络,并测试得到的网络的识别效果;最后,使用训练得到的网络对10000张色球图像中的时间戳信息进行批量识别,之后对识别结果进行定量分析。结果表明该方法可以自动、准确、快速地实现扫描太阳胶片图像中时间戳信息的定位与识别。

[0043] 2)、采用基于深度学习中卷积神经网络的方法,对美国国立天文台拍摄的近50年太阳色球胶片图片中时间信息的识别问题进行了研究。结果表明:该方法对图片中的字符识别具有很强的适用性,识别正确率可以达到98%以上,平均处理一张图片不超过0.1秒,可以满足本发明在实际应用中对识别速度、识别质量的需要,并且具有很强的可移植性,对解决后期同类型问题具有很高的参考价值。

## 附图说明

[0044] 下面结合附图和实施例对本发明作进一步说明:

[0045] 图1为通用卷积神经网络结构图。

[0046] 图2为时间戳字符识别流程图。

[0047] 图3(a)为带有时间戳信息的太阳色球胶片图像一;

[0048] 图3(b)为带有时间戳信息的太阳色球胶片图像二;

[0049] 图3(c)为带有时间戳信息的太阳色球胶片图像三。

[0050] 图4为图像中的时间戳信息区域示意图。

[0051] 图5为 $S_{1j}$ 向量在坐标轴的显示图。

[0052] 图6为去除太阳球面后得到的图。

[0053] 图7为包含时间戳的图片。

[0054] 图8(a)为投影向量图一;

[0055] 图8(b)为投影向量图二。

[0056] 图9为时间戳区域切割结果图。

[0057] 图10为单字符提取算法流程图。

[0058] 图11(a)为字符的强度分布图(顶帽运算前);

[0059] 图11(b)为字符的强度分布图(顶帽运算后)。

[0060] 图12为二值化图片。

[0061] 图13为去除噪声后的二值图。

[0062] 图14为消除无效区域后的二值图。

[0063] 图15为字符切割结果图。

[0064] 图16为字符识别卷积神经网络结构图。

[0065] 图17为日期核对图形界面图。

## 具体实施方式

[0066] 以下通过特定的具体实例说明本发明的实施方式:

[0067] 一个通用的卷积神经网络的架构,包括输入层、卷积层、池化层、全连接层和输出层等,其结构如图1所示。通过softmax逻辑回归用输出层输出的特征向量来对输入层输入的数据进行分类,当输入层为字符图像数据时,可以通过输出层分类的结果对字符图像进

行分类,进而实现字符识别。一个卷积神经网络中可以根据需要有多层卷积层、池化层和全连接层,图1仅仅表示其一般形式。

[0068] 通过卷积神经网络(CNN)对于扫描太阳色球胶片图像中的时间戳信息进行提取,主要分为三部分,如图2所示:

[0069] 步骤1,对图片时间戳信息区域定位并剪裁;

[0070] 步骤2,将时间戳信息区域的字符进一步分割得到单个字符;

[0071] 步骤3,先采用大量的样本训练网络,然后将分割得到的字符使用训练好的网络对其进行识别并将其识别结果整合并保存。

[0072] 步骤1:对图片时间戳信息区域定位并剪裁。

[0073] 原始太阳色球胶片图片如图3(a)、图3(b)、图3(c)所示,每张图片的分辨率为 $1600 \times 2048$ ,时间戳信息通常放在图片的左侧或右侧,位置不固定。时间信息的字符格式大致分为两类,如图3(a),图3(b),且每一类的字符样式不一,因此需要分类对字符进行识别。又由于每张图片的清晰度不一,大部分图片比较昏暗,字符难以辨识,如图3(c)所示。因此需要对每张图片进行预处理。整张图片只有时间戳字符是本发明需要得到的,如图4红框中的部分。时间戳信息中分别记录有拍摄该照片的年、月、日以及时、分、秒的信息,如图4中黄框中的部分。根据这些资料的观测时间精度,只需要得到年、月、日以及时、分的信息即可。由于时间信息的位置不固定,所以本发明首先应该将时间戳区域进行定位并切割。时间戳的位置在左侧或者右侧,并不会在太阳的球面上,所以本发明需要先将包含太阳的部分切除,然后再寻找时间戳所在的位置。

[0074] 步骤1.1:首先是基于垂直投影的太阳球面切除步骤。

[0075] 垂直投影方法是根据图像中像素点的信息在垂直方向对像素点的分布特点进行检查的一种方法。其计算方法是,对图像在垂直分量上进行累加操作得到一个 $1 \times n$ 的向量。假设图片的大小是 $m \times n$ ,像素在 $i$ 行 $j$ 列的像素值为 $f_{ij}(x, y)$ ,则在垂直方向投影为:

$$[0076] \quad S_{1j} = \sum_{i=1}^m f_{ij}(x, y) \quad (1)$$

[0077] 其中, $S_{1j}$ 表示图像第 $j$ 列像素点求和的结果, $S_{1j}$ 的大小为 $1 \times n$ 。通过计算图片在垂直方向的投影可以进一步判断太阳球面的位置。对图3(a)进行垂直方向投影计算后,得到的 $S_{1j}$ 向量如图5所示。由图5可以看出,在 $S_{1j}$ 向量[400, 1800]之间为太阳色球部分投影的结果。又因为太阳是对称的,所以本发明只需要得知在向量 $S_{1j}$ 中的最大值,即可定位太阳圆心的位置,然后根据太阳球面在垂直方向所占的像素长度将包含太阳球面部分的图片去除。图6为除太阳球面后得到的两张小图。

[0078] 步骤1.2:基于方差判断时间戳位置及翻转校正。

[0079] 图片在本质上是包含有多个像素点的矩阵,像素点值的大小便反映在图片中的颜色上。例如:二值图像中“0”表示黑色,“1”表示白色。图6中并不是二值图像而是256级灰度图像。每个像素点的大小表示该点的亮度等级,共有256级,级数越高在该像素点位置的亮度越大。由图6可知,包含时间戳的图片中,时间戳以亮度高的点进行表示,而不包含时间戳的图片中,绝大多数像素点都是亮度较暗的。本步骤采用基于方差的时间戳位置判断方法,即根据图像矩阵中像素值的方差大小来判断时间戳在哪张图片上。对于大小为 $m \times n$ 的图片,其方差计算公式表达如下。

$$[0080] \quad STD = \frac{\sum_{i=1}^m \sum_{j=1}^n (x_{ij} - \bar{x})}{n * m} \quad (2)$$

[0081] 其中,  $x_{ij}$  表示像素点在  $[m, n]$  处的像素值,  $\bar{x}$  表示总体的像素值均值,  $n * m$  表示总的像素个数。

[0082] 显然, 包含时间戳的图片的方差远远大于不包含时间戳的图片的方差。从而判断出时间戳所在的图片。得知时间戳所在图片后, 需要对图片进行反转校正。若为左侧图片则需顺时针翻转90度, 反之逆时针旋转90度。图像旋转公式如下:

$$[0083] \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

[0084] 其中,  $x, y$  表示原始像素位置,  $x', y'$  表示旋转变换后的像素位置,  $\beta$  表示逆时针旋转的角度。以图6为例, 经过方差判断和旋转后的图片如图7所示。

[0085] 从图7中可知, 包含时间戳的图片中仍然存在大量无用信息。这在一定程度上会对计算速度产生影响。考虑到巨大的图片量, 这些无用区域会造成不必要的存储消耗。

[0086] 步骤1.3: 本发明采用水平垂直投影法实现时间戳区域的精分割。

[0087] 水平垂直投影方法是根据图像中像素点的信息在水平方向和垂直方向分别对像素点的分布特点进行检查的一种方法。经常用于对目标区域进行精确投影, 用于后期的分割操作。其计算方法是, 对图像在水平和垂直分量上分别进行累加操作并得到两个的向量。对于大小为  $m \times n$  的图片, 像素在  $i$  行  $j$  列的像素值为  $x_{ij}$ , 则在水平方向和垂直方向投影分别为:

$$[0088] \quad \begin{aligned} S_{1j} &= \sum_{i=1}^m x_{ij} \\ S_{i1} &= \sum_{j=1}^n x_{ij} \end{aligned} \quad (4)$$

[0089] 其中,  $S_{1j}$  表示图像第  $i$  列像素点求和的结果,  $S_{1j}$  的大小为  $1 \times n$ 。  $S_{i1}$  表示图像第  $j$  行像素点求和的结果,  $S_{i1}$  的大小为  $m \times 1$ 。通过计算图片在水平方向和垂直方向的投影可以进一步时间戳区域的具体位置从而实现图片的精确分割。以图7为例, 其水平方向和垂直方向投影的向量如图8(a), 图8(b) 所示。

[0090] 步骤1.4: 对时间戳区域进行切割。

[0091] 基于图片在水平和垂直方向的投影结果, 对图片进行切割。从图8(a), 8(b) 中可以得知, 图片中时间戳区域对应的水平和垂直方向的投影结果较高。根据这一特点, 限定只保留投影结果大于平均值的部分即可完成时间戳区域分割。为了保证图片的连续性不被破坏, 所以以第一个大于均值的点作为起点, 最后一个大于均值的点作为终点, 保留起点到终点区域的所有图像。假设原图  $S$  的大小为  $m \times n$ , 切割后图片  $P$  的大小为  $m' \times n'$ , 其切割公式为:

$$[0092] \quad P = S(a:b, c:d), (a, c > 1, b < m, d < n) \quad (5)$$

[0093] 其中:

$$[0094] \quad \begin{cases} a = \min(x > \bar{x}) \\ b = \max(x > \bar{x}) \\ c = \min(y > \bar{y}) \\ d = \max(y > \bar{y}) \end{cases}$$

[0095] 式中,  $S(a:b, c:d)$  表示图片  $S$  中,  $a$  到  $b$  行,  $c$  到  $d$  列。 $x$  表示在水平投影中某点的值,  $\bar{x}$  表示水平投影中各点均值。 $y$  表示在垂直投影中某点的值,  $\bar{y}$  表示垂直投影中各点均值。 $\min(x > \bar{x})$  表示在投影向量中,  $x$  大于  $\bar{x}$  的位置的最小值。 $\max(x > \bar{x})$  表示在投影向量中,  $x$  大于  $\bar{x}$  的位置的最大值, 其他同理。以图7为例, 经切割后如图9所示。

[0096] 步骤2: 单字符分割。

[0097] 由图9可知, 日期字符较小容易且部分模糊不清, 在后期处理过程中容易被当作噪声干扰而被删除, 从而对结果的影响较为剧烈。因此, 在识别过程中只考虑识别时分信息, 对于日期采用人工填写的方式且工作量不大。

[0098] 在实际操作过程中, 单个字符的切割是最为困难的。图9是从质量比较好的图片中切割得到的结果。然而实际情况下, 大部分图片类似图3(c), 字符部分模糊不清, 有的甚至人眼无法识别。因此单个字符的获取要比字符区域的获取困难的多, 同时也最容易影响最终的识别结果。另外, 如图3(a), 3(b)、3(c)所示, 时间戳信息区域字符格式不一, 故需处理过程中更要考虑算法的通用性。对于单个字符切割流程如图10所示, 首先采用顶帽运算对图片的背景进行去除, 然后采用局部二值化算法去除噪声, 最后采用连通域算法提取字符区域。因为字符类型分为两类, 如图3(a), 3(b), 分别以白色和黑色表示字符。该算法默认字符颜色为白色, 若字符颜色为黑色则经过连通域提取后将提取不到任何有效区域, 所以若不存在有效区域则返回到局部二值化部分, 对局部二值化后的图片进行颜色翻转即可。

[0099] 步骤2.1: 背景去除基于顶帽运算。其算法原理是原图像与原图像开运算结果做差。其算法可以描述为:

$$[0100] \quad \text{topimg} = \text{tophat}(\text{img}, \text{element}) = \text{img} - \text{open}(\text{img}, \text{element}) \quad (6)$$

[0101] 其中,  $\text{topimg}$  表示经过顶帽运算后的图片,  $\text{img}$  表示原图,  $\text{element}$  表示用于顶帽运算和开运算的核。

[0102] 采用  $29 \times 29$  的核对图9处理为例, 图11(a)、图11(b) 分别为顶帽运算前后的强度分布图对比。从图11(b) 中可以看出, 图片经过顶帽运算后可以消除一部分背景噪声, 并且可以突出图像中的字符。虽然不能完全去除噪声, 但是可以减少后期运算的工作量, 避免过多噪声影响局部二值化的结果。

[0103] 步骤2.2: 噪声去除则采用基于Sauvola局部二值化算法。

[0104] Sauvola算法解释如下:

[0105] Step1: 计算像素点  $f(x, y)$  在  $n \times n$  范围内的均值MEAN和方差STD;

$$[0106] \quad \text{MEAN}(x, y) = \frac{1}{n^2} \sum_{i=x-\frac{n}{2}}^{x+\frac{n}{2}} \sum_{j=y-\frac{n}{2}}^{y+\frac{n}{2}} f(x, y) \quad (7)$$

$$[0107] \quad STD(x, y) = \sqrt{\frac{1}{n^2} \sum_{i=x-\frac{n}{2}}^{x+\frac{n}{2}} \sum_{j=y-\frac{n}{2}}^{y+\frac{n}{2}} (f(x, y) - MEAN(x, y))^2} \quad (8)$$

[0108] Step 2:根据公式计算像素点 $f(x, y)$ 的阈值 $T(x, y)$ ;

$$[0109] \quad T(x, y) = MEAN(x, y) \left[ 1 + k \left( \frac{STD(x, y)}{N} - 1 \right) \right] \quad (9)$$

[0110] 其中, $k$ 为自定义参数且 $0 < k < 1$ 。 $N$ 为标准方差的动态范围;

[0111] 采用 $k=35, N=0.08$ 处理后的图片如图12所示。

[0112] 从二值化图片中,可以看出字符已经可以和背景分开,背景或过大过小,也有很多噪点。只需要将符合字符大小的连通域提取出来就可以完成字符的切割。并且采用连通域提取的方法速度快,可以节约很多程序运行时间。

[0113] 步骤2.3:字符区域提取基于连通域算法。

[0114] 为了尽可能多的去除噪声并保留有效信息,若为第1、2类图片则需先对图片进行局部二值化,然后去掉过大或过小的连通域,以消除部分干扰。若为第2类图片则对图片进行阈值为0.25的二值化操作并进行反转得到以白色表示的字符,之后处理同前类相同。经实验表明,局部二值化阈值为 $16 \times 16$ 时效果较好,字符的连通域面积均处于 $[500, 5000]$ 之间。处理结果如图13所示。

[0115] 由图13可以看出,经过上述处理之后仍然有一些无效区域没有消除,因此本发明通过判断每个连通域是否符合标准字符的长度,宽度以及长宽比来进一步删除无效区域。由统计信息,字符的高度在 $[90, 110]$ 之间,字符宽度在 $[10, 60]$ 之间,字符的长宽比不小于1。得到的结果如图14所示。

[0116] 从图14可得,图片的无效区域被全部消除,这样根据二值图中每个连通域的位置对应原图中的位置便可以将原图中的单个字符区域分别切割出来。为了保证得到的图片尺寸一致,本发明分别对每张图片进行填充并将其大小变换为 $28 \times 28$ 的标准图片,单个字符切割最终结果如图15所示。

[0117] 步骤3:字符识别。

[0118] 采用深度学习中的卷积神经网络算法来进行字符识别。本发明搭建的字符识别的卷积神经网络包含两个卷积层,两个池化层和一个全连接层,如图16所示。第一个卷积层通过6个尺寸为 $5 \times 5$ 的不同的卷积核对输入尺寸为 $28 \times 28$ 的字符图片进行卷积,经过第一层卷积后,原字符图片变为 $24 \times 24 \times 6$ 的特征图。第一个池化层采用滑动窗口为 $2 \times 2$ 的池化函数对第一个卷积层的结果进行特征再提取,经历该层池化后,变为 $12 \times 12 \times 6$ 的特征图。第二个卷积层采用12个尺寸为 $5 \times 5$ 的不同卷积核对池化层的特征图进行特征再提取,经提取后的特征图尺寸为 $8 \times 8 \times 12$ 。第二个池化层对第二个卷积层卷积后的特征图进行池化,池化后特征图尺寸变为 $4 \times 4 \times 12$ 。将第二次池化操作后的特征图输入全连接层,得到该字符的特征向量。最后将字符的特征向量进行分类并与实际数字相对应便完成了时间戳中时间字符识别。

[0119] 本发明中,时间字符识别的卷积神经网络的训练步骤分为如下3步:

[0120] 步骤3.1:对从色球图像中得到的单个字符图片并添加标签作为训练网络所需的

数据样本。

[0121] 步骤3.2:将样本数据整合为一个 $28 \times 28 \times N$ 的矩阵作为输入层的X向量,其中N为字符样本的个数。将X向量中每一维矩阵对应的数字标签作为输入层的Y向量。

[0122] 步骤3.3:通过前向传播和反向传播(BP)来训练该网络,循环迭代更新其系数,采用ReLU激活函数和最大池化函数,最终得到一个识别准确度较高的网络结构。其中输入向量X和Y被送入迭代100次。

[0123] 将训练得到的卷积神经网络对太阳色球胶片图中的时间信息中的单个字符进行识别,将识别出来的字符按顺序组合起来与原图的文件名相对应并自动填入Excel表中,用于后期人工核对以及建立数据库。

[0124] 步骤4:人工核对日期,即采用人工的方式核对自动生成的图片日期信息是否有误。

[0125] 当时间戳中“时、分”等时间信息自动识别完成后,还需进行的重要一步就是人工核对日期信息(年、月、日),参照图17。由于拍摄的时间大多是连续的,且采用24小时制,因此很容易便能判断拍摄日期是否变化。例如第一张识别结果为“2359”,第二张识别结果为“000”,则第二张的拍摄日期在第一张拍摄日期基础上加一天即可。所以对于一段时间内的色球图像,只需要知道拍摄的起始日期便可以推算得出之后的每张图片的拍摄日期。采用图17的界面,对每天的前几张数据进行校验,若日期错误便只需修改每天第一张图片的拍摄日期,之后通过递推算法自动更新后面所以图片的日期。

[0126] 用户进行日期核对时,打开此界面填入原始图片路径和相应Excel表路径。点击“Open”按钮,程序依据Excel表中每一个日期第一天所对应编号依次打开图片,并将日期填入右侧文本框中。若Excel表中记录的日期正确且上一张图片为上一个日期,则直接点击“Next Day”按钮进行下一个日期的核对。若错误则需通过“Last”、“Next”按钮找到日期跳转的图片并填入右侧文本框,点击更新按钮程序自动对后面所有的日期进行更新。依次核对,直到程序运行到最后一个日期即为完成。经测试,1个人完成一万张图片的日期核对所需时间约为10分钟。本发明中,时间戳中“时、分”等时间信息是采用步骤1-3来自动提取的,因此时间主要耗费在“步骤4.人工核对日期”这一步上面。若采用传统的方式,逐一人工录入“年、月、日、时、分”信息,则1个人至少需要两天才能完成1万张图片的时间戳信息录入。因此,本发明在提高时间戳信息录入效率以及节省人力等方面的效益是显著的。

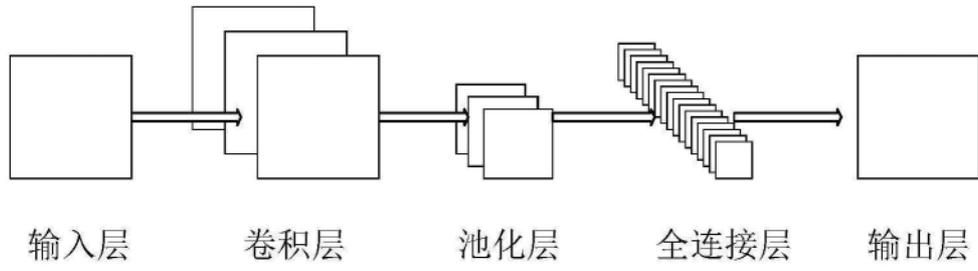


图1

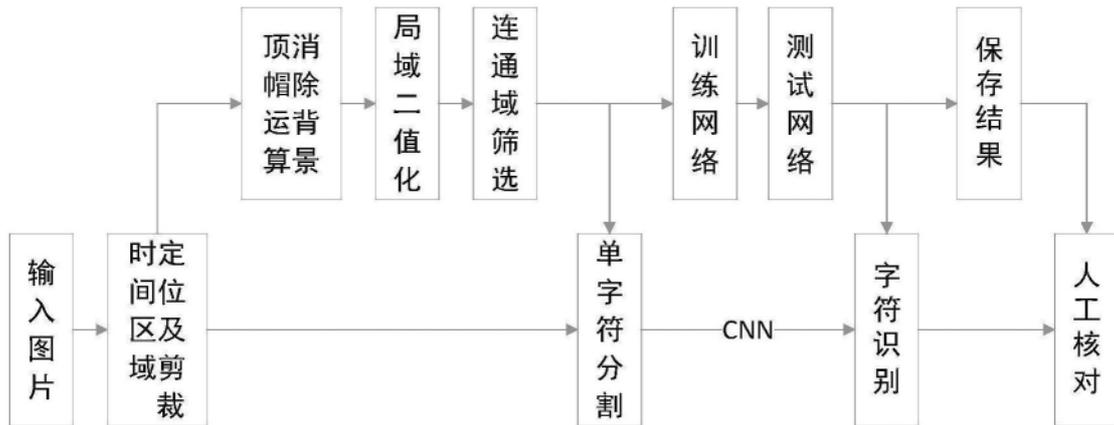


图2

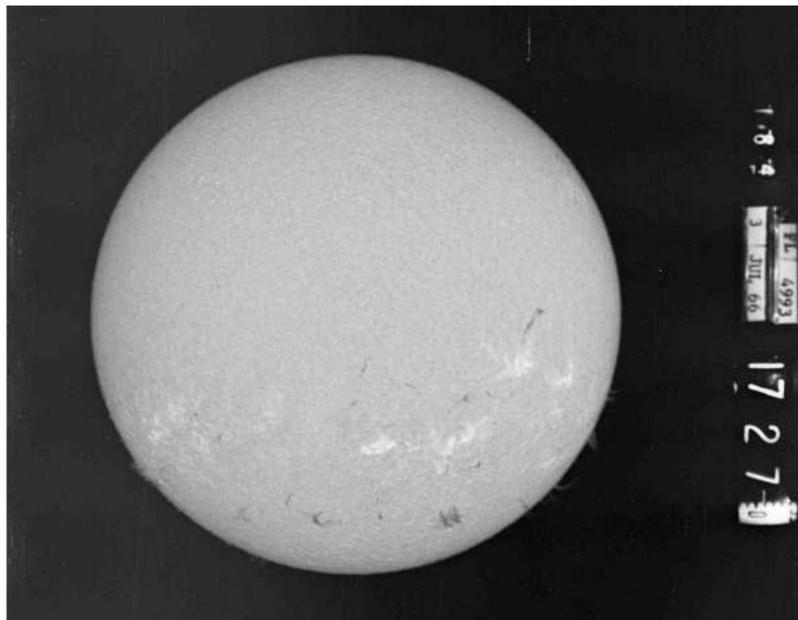


图3(a)

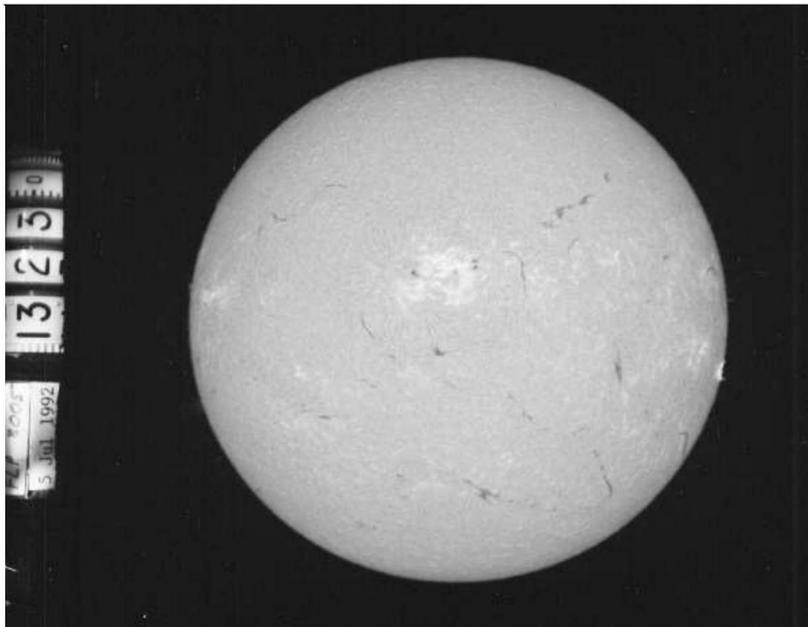


图3(b)

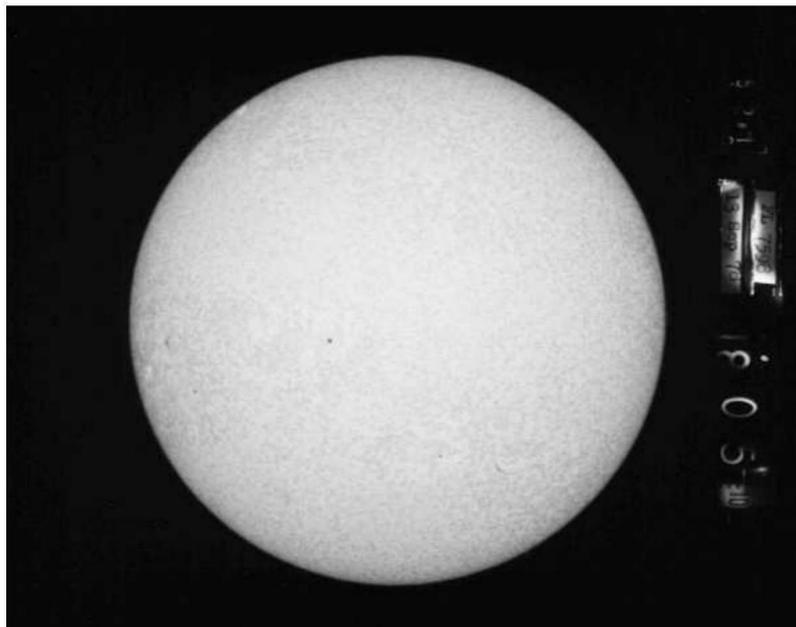


图3(c)

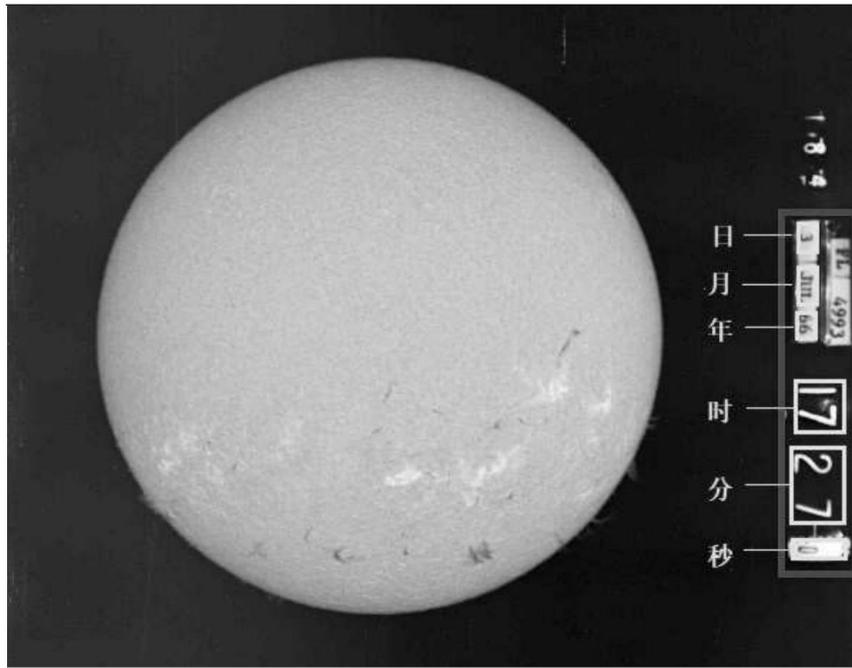


图4

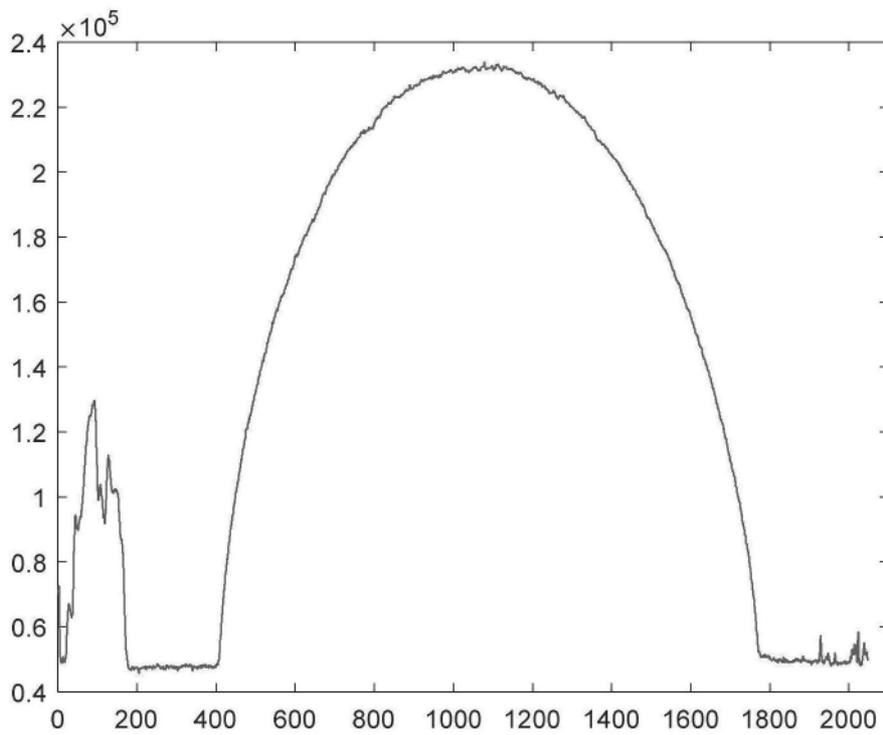


图5

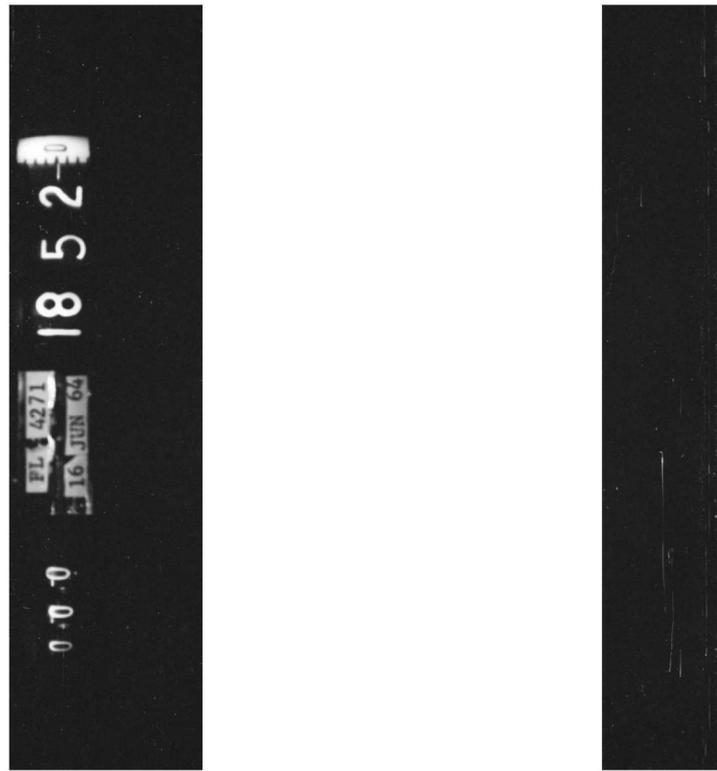


图6

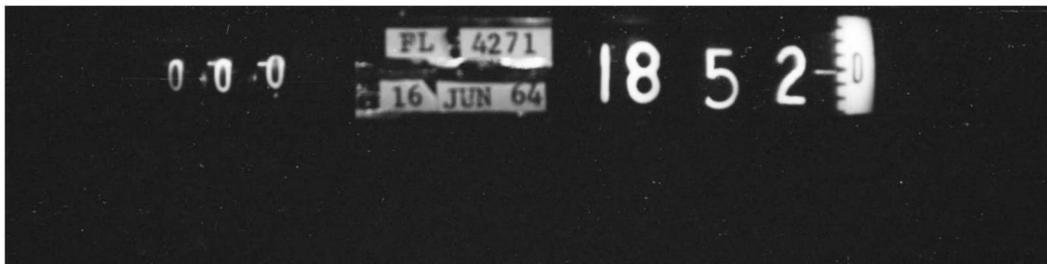


图7

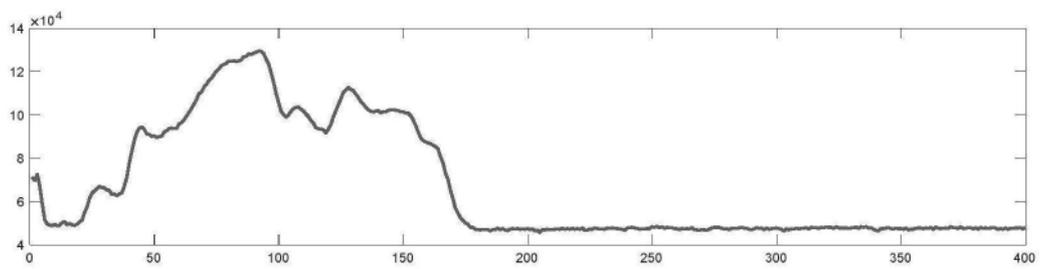


图8(a)

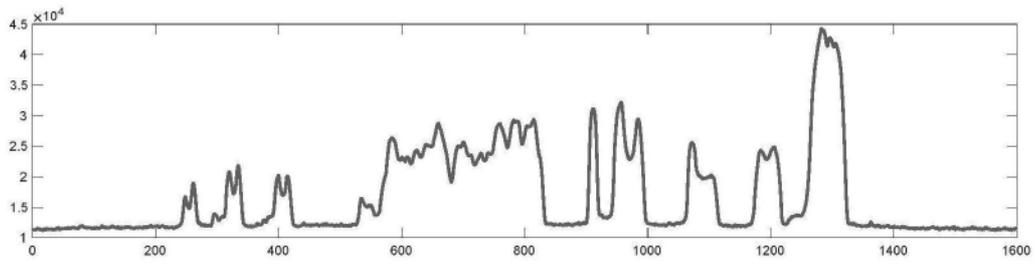


图8(b)



图9

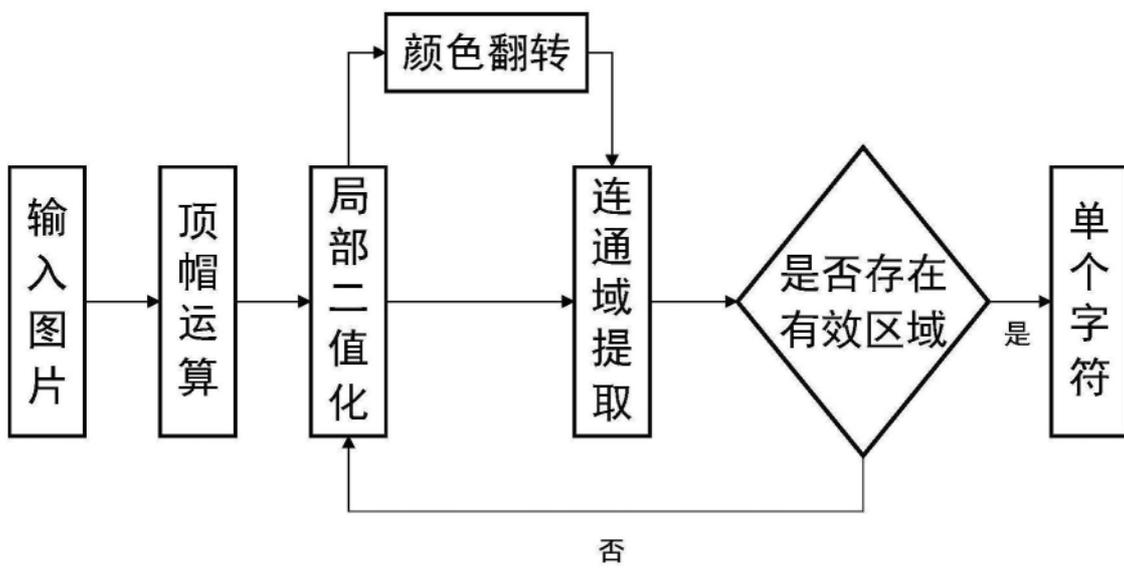


图10

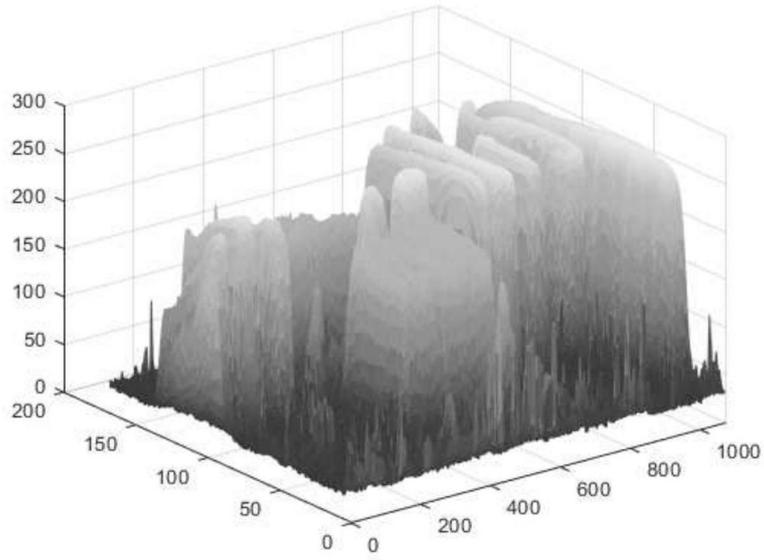


图11(a)

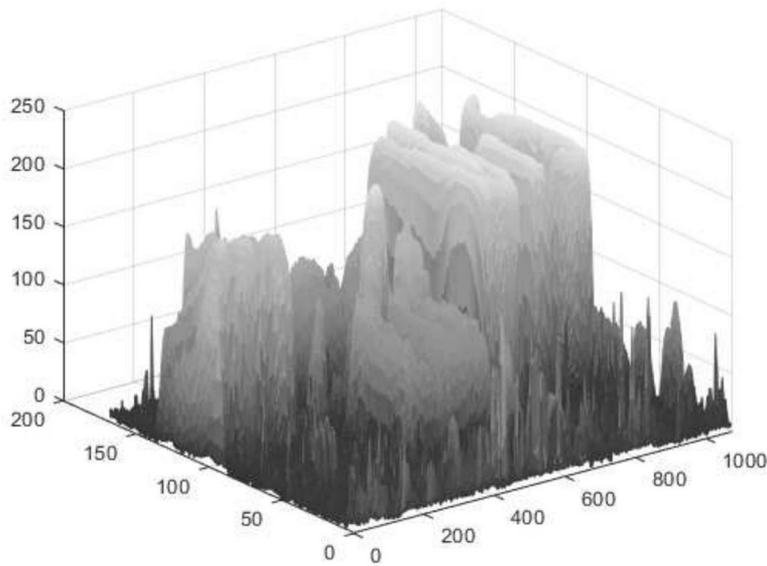


图11(b)

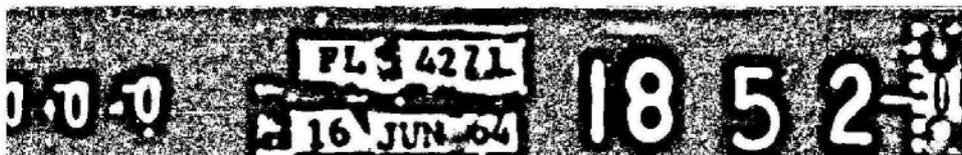


图12



图13



图14

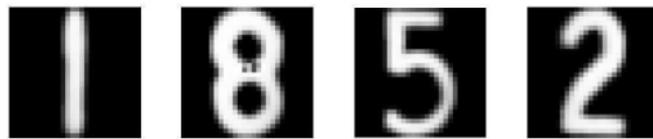


图15

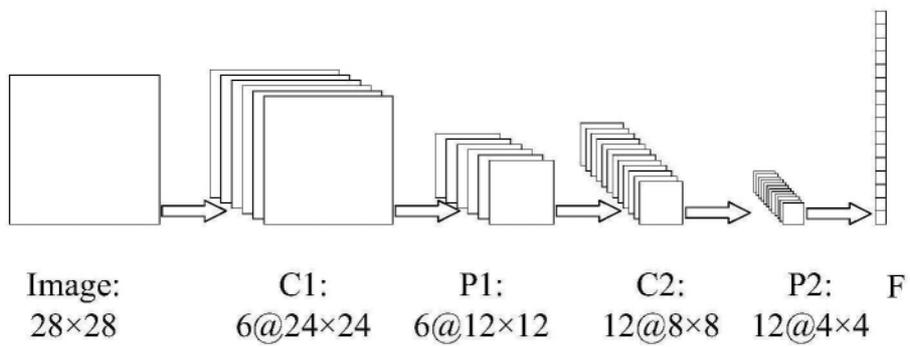


图16

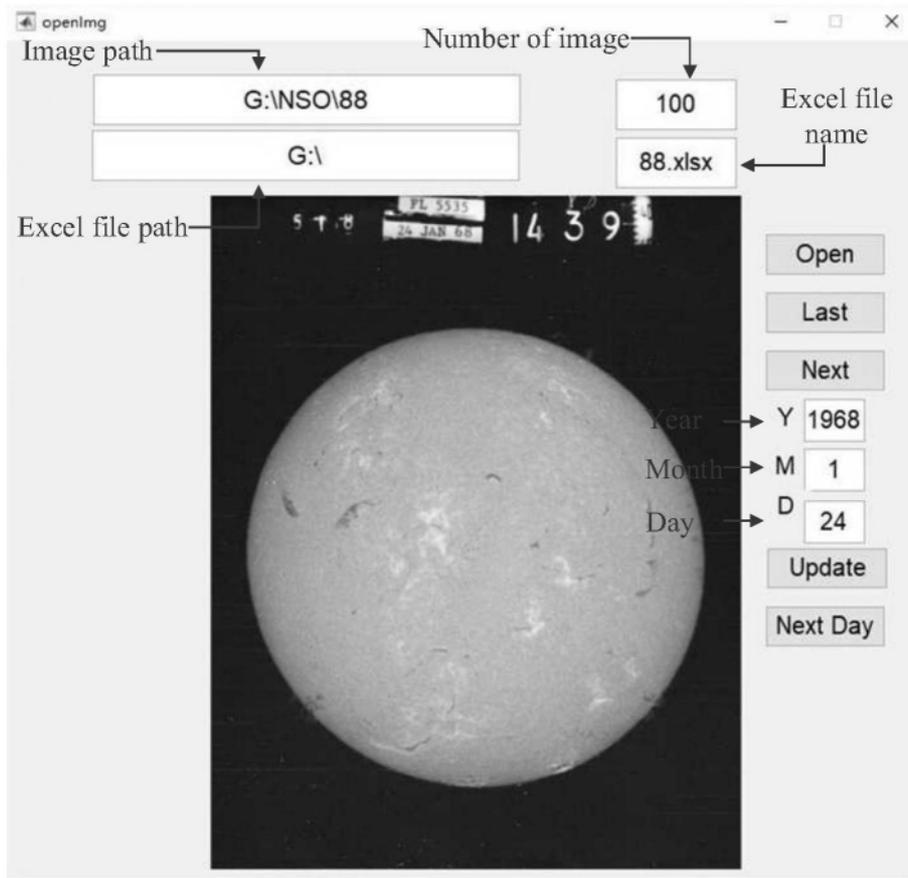


图17