

# Research on Character Recognition of Steel Embossing Based on YOLOv2

Huining Huang<sup>1</sup>, Xuejun Zhang<sup>1,2\*</sup>, Ju Huang<sup>1</sup>, Chan Liang<sup>1</sup>, Yinghua Sun<sup>3</sup>

<sup>1</sup>School of Computer, Electronics and Information, Guangxi University, Nanning Guangxi

<sup>2</sup>Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning Guangxi

<sup>3</sup>Guangxi Big White and Little Black Robots Co., Ltd., Nanning Guangxi

Email: \*xjzhang@gxu.edu.cn

Received: Jan. 1<sup>st</sup>, 2020; accepted: Jan. 12<sup>th</sup>, 2020; published: Jan. 19<sup>th</sup>, 2020

---

## Abstract

Aiming at the influence of the same color as the background area of industrial parts and uneven illumination of the steel embossing characters, there is a problem of poor efficiency and precision of traditional computer vision algorithms to identify steel embossing characters. This research proposes a steel embossing character recognition method based on YOLOv2. Through some basic image preprocessing methods, the steel embossing character data set is expanded, and the fast and reliable deep learning algorithm YOLOv2 is used to automatically extract the features of the image to realize the recognition of the steel embossing characters (including numbers and letters). Compared with other traditional image recognition algorithms, the experimental results show that the accuracy of the network model for the identification of steel embossing characters is 98.6%, and the average processing time of the algorithm is 0.3 s, which meets the accuracy and efficiency requirements of engineering applications. In addition, the output of the model is improved by using the character position information, and the correct production label can be output directly. It has good stability and real-time performance in industrial production environment and has certain application significance.

## Keywords

Deep Learning, Character Recognition, YOLOv2, Target Detection, Image Processing

---

# 基于深度学习YOLOv2算法的 钢材压印字符识别研究

黄慧宁<sup>1</sup>, 张学军<sup>1,2\*</sup>, 黄菊<sup>1</sup>, 梁婵<sup>1</sup>, 孙映华<sup>3</sup>

<sup>1</sup>广西大学计算机与电子信息学院, 广西 南宁

\*通讯作者。

<sup>2</sup>广西多媒体通信与网络技术重点实验室, 广西 南宁

<sup>3</sup>广西大白小黑智能机器人有限公司, 广西 南宁

Email: \*xjzhang@gxu.edu.cn

收稿日期: 2020年1月1日; 录用日期: 2020年1月12日; 发布日期: 2020年1月19日

## 摘要

针对工业生产钢材部件上压印字符与背景区域同色和光照不均影响, 传统计算机视觉算法识别钢印字符存在效率与精度不佳的问题, 本研究提出一种基于YOLOv2的钢材压印字符识别方法。通过一些基本的图像预处理方式扩充钢印字符数据集, 采用快速可靠的深度学习算法YOLOv2自动提取图像的特征, 实现对钢印字符(包括数字和字母)的识别。相较于其他传统的图像识别算法, 实验结果表明, 该网络模型对钢印字符识别的准确率达98.6%, 算法平均处理时间为0.3 s, 达到了工程应用的精度和效率要求。此外, 利用字符位置信息对模型的输出进行改进, 实现直接输出正确的生产标号。在工业生产环境下具有较好的稳定性和实时性, 有一定的应用意义。

## 关键词

深度学习, 字符识别, YOLOv2, 目标检测, 图像处理

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

在钢材器件生产过程中, 往往会在每个部件上标注不同的生产编号, 这样不仅为了便于对部件的质量监控和信息管理追踪, 还为了方便对部件与相应器件间的匹配问题进行把控。压印字符是利用字模对物体表面施压形成的, 呈凹凸状, 因其不能随意更改而可以永久保存的优点, 在工业生产中得以广泛应用。钢材器件通常是大批量的生产, 传统的人工手动记录生产编号的方法需消耗大量的劳动力, 同时大大延长了生产时间, 还容易出错。计算机视觉技术不断发展, 应用于工业生产过程当中, 极大地减少了工业生产对人工的依赖。现如今, 随着工业智能化生产的发展要求不断提高, 在工业生产中迫切需要对钢材器件上的压印字符标号进行视觉实时识别。

压印字符与手写字符、印刷字符等一般的光学字符有所不同, 压印字符的颜色与背景色是一样的, 因而这两者的对比度不鲜明, 而且在复杂多变的工业生产环境下, 会出现噪声、光照不均衡、抖动模糊、位置不确定等问题, 这些问题又进一步增加了压印字符识别的难度。因此, 传统的字符识别技术(如模板匹配[1])无法直接应用于钢材压印字符的识别中。可将钢材压印字符识别看成是具有一般性的目标检测任务。目标检测是计算机视觉[2]研究的热点问题之一, 它学习每个对象的可视模型, 并找到合适的边界框区域和对象类别。早期较常用的目标检测算法是DPM (Deformable Parts Model) [3], 其通过HOG (Histograms of Oriented Gradients) [4]特征提取, 将提取到的图像特征作为输入, 利用SVM (Support Vector Machine) [5]进行分类。DPM在行人检测[6]等目标检测任务上有较好的效果。但实质上, 这样的方法对于特征设计十分依赖于人工, 过程也比较复杂繁琐。

近年来,随着深度卷积神经网络的兴起,目标检测的性能与以往基于手工的方法(如文献[7] [8])相比有了显著提高。最初利用神经网络进行目标检测的方法是基于候选区域的方法。Girshick 等[9]提出的区域卷积神经网络(R-CNN)先从图像中采用选择性搜索算法[10]提取候选区域,再通过标准的卷积神经网络去调整以及分类这些区域。但是,对于每个候选区域都需要对整个网络进行重新计算,存在大量重复的计算,进一步增加了计算复杂度,从而降低了目标检测网络的速度。随后,Girshick [11]提出了快速区域卷积神经网络(Fast R-CNN),直接从 CNN 的编码特征图中提取候选区域,使得网络可以共享计算结果,从而让模型提速,帧速率达到 5 frame/s。然而, Fast R-CNN 仍采用选择性搜索算法来提取候选区域,计算还是相当复杂。为解决这一问题,2015 年 Ren 和 Girshick [12]提出超快区域卷积神经网络(Faster R-CNN),采用基于深度学习的区域建议网络(RPN)来提取目标候选区域。在 RPN 中,预定义的锚点(Anchor)和框(Anchor Box)预测目标感兴趣区域(RoI)。通过引入 RPN,检测精度和速率都有所提高。尽管 Faster R-CNN 的计算速度有了很大提升,但是仍然无法满足实时检测的要求,因此在 2016 年 Redmon 等人[13]提出了 YOLO (You Only Look Once)网络,将整幅待检测图像作为输入,而卷积神经网络作为回归器,回归目标在待检测图像中的位置信息,实现端到端的目标检测和识别。相比于基于候选区域的方法,YOLO 算法不需要生成目标建议区域,而是将输入图像看成一个候选区域,简单地划分为几个小区域,然后通过 CNN 来判断每一个区域是否存在目标,并预测目标的种类和边界框,这大大节省了图像处理的时间,帧速率达到了 45 frame/s。当然,YOLO 在提升检测速度的同时精度有所降低。随后,2017 年 Redmon 等人[14]基于 YOLO 网络进行一系列的改进提出了 YOLOv2 网络,其准确率和检测速度均得到了显著提高。从 VOC 2007 数据集的模型检测测试结果来看,YOLO v2 的准确率为 76.8%,检测速度可达到 67 frame/s,满足了实时性,在目标检测领域表现出色。

本研究提出了一种基于 YOLOv2 网络的钢材压印字符识别方法,结合图像灰度化、滤波去噪、边缘检测、霍夫变换算法对工业生产线上钢材部件的生产标号图像进行预处理,处理之后的数据集通过 YOLOv2 网络训练,学习各类字符的特征,对字符进行分类的同时得到字符的位置信息,利用图像中各个字符所得的位置信息进行排序操作,即可得到钢材部件的生产标号,实现实时地、快速地、准确地一步步式识别钢材部件的生产标号。

## 2. 识别方法

### 2.1. 图像预处理

#### 2.1.1. 倾斜矫正

从工业相机获取的钢材压印字符原始图像如图 1(a)所示。由于在压制压印字符过程中会出现偏差,或者在摄取压印字符图像过程中没有完全对准,难免会出现字符倾斜的问题,这就需要对字符进行倾斜校正。霍夫变换(Hough Transform)是图像处理中的一种检测特定形状的技术。它实质上是一种线-点变换,把图像空间转换到一个参数空间中,通过统计寻找累加的局部最大值得到一个符合该特定形状的集合作为霍夫变换结果。霍夫变换常用来检测图像中的直线,此外霍夫变换也可用于对曲线的识别,例如检测圆和椭圆。霍夫变换检测直线具有受直线中的间隙影响小和抗干扰性强的优点。其原理是:将图像中直线上的点 $(x, y)$ 进行变换,利用过该点的直线在直角坐标中的表达式 $y = kx + b$ (其中 $k$ 为斜率, $b$ 为截距)可变换成 $b = -kx + y$ ,即把点 $(x, y)$ 转换到 $k-b$ 参数空间的一条直线,图像空间的直线上多个点的累加变为在参数空间上的多条直线相交于一点,在参数空间寻找这个最大点从而可以返回找到原始图像上的直线。根据钢材部件的矩形条状结构特点,可利用霍夫变换检测直线的算法获取图像中最长的线,然后计算该长线的倾斜角度 $\gamma$ ,最后将图像旋转 $-\gamma$ 即可。结果如图 1(b)所示。

另外,图像在实际成像、获取的过程中会出现字符倒置的情况,如图 1(c)所示。为了统一数据集样本的基本形式,就需要对这些倒置的图像进行 $180^\circ$ 的旋转处理,结果如图 1(d)所示。

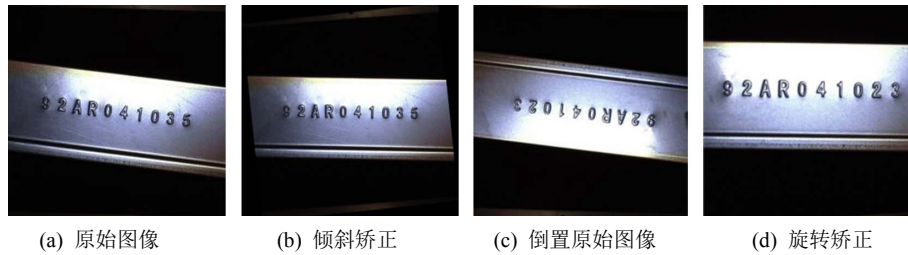


Figure 1. Tilt correction picture  
图 1. 倾斜矫正图片

### 2.1.2. 图像增强

为了加强和突出钢材部件图像中压印字符特征,减少字符图像的噪声干扰,本研究首先对图像进行灰度化处理,然后利用中值滤波抑制噪声,同时字符的边缘细节还能保留下来。中值滤波[15]是一种非线性图像平滑方法,其输出像素由邻域像素的中间值决定。首先要选定一个奇数像素的窗口  $W$ ,将窗口内各像素灰度值按从小到大进行排列  $(x_1, x_2, \dots, x_n)$ ,选取排在灰度序列中间的灰度值  $y$  作为对应窗口中心位置的像素灰度值。具体的计算方式为

$$y = \text{med}(x_1, x_2, \dots, x_n) = \begin{cases} x_{\frac{n+1}{2}}, & n \text{ 为奇数} \\ \frac{1}{2} \left( x_{\frac{n}{2}} + x_{\frac{n}{2}+1} \right), & n \text{ 为偶数} \end{cases} \quad (1)$$

其中,  $n$  为像素值的数量,通常是奇数。但当  $n$  为偶数时,计算中间两个像素的平均值即为窗口中心位置的灰度值。通过中值滤波后图像的背景更加平滑,字符更突出,有助于在字符识别的过程中更好地提取出字符特征,从而达到较好的识别效果。图像增强的效果如图 2 所示。

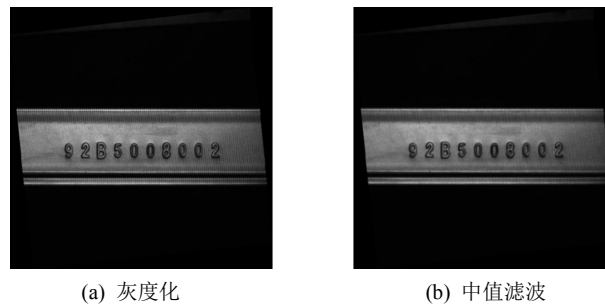


Figure 2. Image enhancement effect map  
图 2. 图像增强效果图

### 2.1.3. 数据集扩充

由于没有公开的钢材压印字符数据集,本研究从实际的工厂生产线上通过工业相机获取了 1500 幅钢材压印字符原始图像作为数据集样本。每幅钢材生产标号图像均包含了 10 个字符,由数字和大写字母组成,但经过统计发现字母的数量远远少于数字字符的数量。为了平衡网络训练的各类样本数量,对现有样本进行扩充,本研究采用泊松融合的方法对部分字母样本进行重新自由组合,产生新的样本。

泊松融合是由 Perez 等[16]提出的基于泊松方程的图像编辑算法,通过求解公式(2)极值问题,构造泊松方程从而求解像素最优值。

$$\min_f \iint_{\Omega} |\nabla f - V|^2, f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (2)$$

式中,  $\Omega$  代表前景图像中需要融合的字区域,  $\partial\Omega$  则为该区域的边界,  $V$  是在  $\Omega$  上的梯度向量场。  $f$  代表字符融合后的图像, 其梯度为  $\nabla f$ , 而  $f^*$  代表融合的字区域以外的背景图像。在满足融合图像和背景图像的边界值相等的同时, 求解融合图像的梯度与  $V$  引导下的梯度最小差值, 从而达到比较自然的无缝融合效果。该问题可以转化成利用狄利克雷边界条件求解泊松方程, 如式(3)所示。

$$\Delta f = \text{div}V, f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (3)$$

式中,  $\Delta$  为拉普拉斯算子,  $\text{div}$  为散度算子。本研究利用泊松融合方法最终得到 2000 幅字符图像, 其中 1600 幅作为训练集, 400 幅作为测试集。

## 2.2. YOLOv2 原理

### 2.2.1. YOLOv2 网络结构

大部分目标检测框架常用准确率高的 VGG-16 [17] 作为特征提取网络, 但 VGG-16 网络结构复杂且计算量庞大。而 YOLO 采用类似 GoogleNet [18] 的网络结构, 相较于 VGG-16 而言, 计算量减小, 但准确率略低于 VGG-16。为了兼顾复杂度与准确率以提升网络的检测性能, YOLOv2 对 YOLOv1 进行了改进, 采用了新的分类网络 Darknet-19 作为网络骨架来进行特征提取。Darknet-19 网络包含了 19 个卷积层和 5 个最大值池化层, 去除了 YOLOv1 中的全连接层, 得到了更多的空间信息。网络在所有的卷积层后增加 Batch Normalization (批量归一化) 并移除网络的 dropout, 这可以显著改善网络的收敛性, 并且有利于正则化模型, 防止过拟合。

基于 YOLOv2 的钢材压印字符识别网络结构如图 3 所示。将归一化为  $608 \times 608$  大小、通道数为 3 的图像作为输入, 采用多个  $3 \times 3$  卷积核, 每经过一次池化操作都会将通道数翻倍。借鉴了 network in network [19] 的思想, 使用全局平均池化进行预测, 将  $1 \times 1$  卷积核置于  $3 \times 3$  卷积核之间进行交替操作以压缩卷积间的特征表示。经过 19 个卷积层和 5 次池化后, 图像转化为  $19 \times 19 \times 1024$  的特征图, 其中第 13 个卷积层输出的  $38 \times 38 \times 512$  特征图经过一层卷积和重组之后得到  $19 \times 19 \times 256$  的特征图, 将其与  $19 \times 19 \times 1024$  的特征图进行融合, 最后再经过两层卷积得到最终的特征图。通过添加这样一个转移层 (passthrough layer) 把浅层特征图和深层特征图相连接, 使得最终获得的特征图可以拥有更好的细粒度特征。

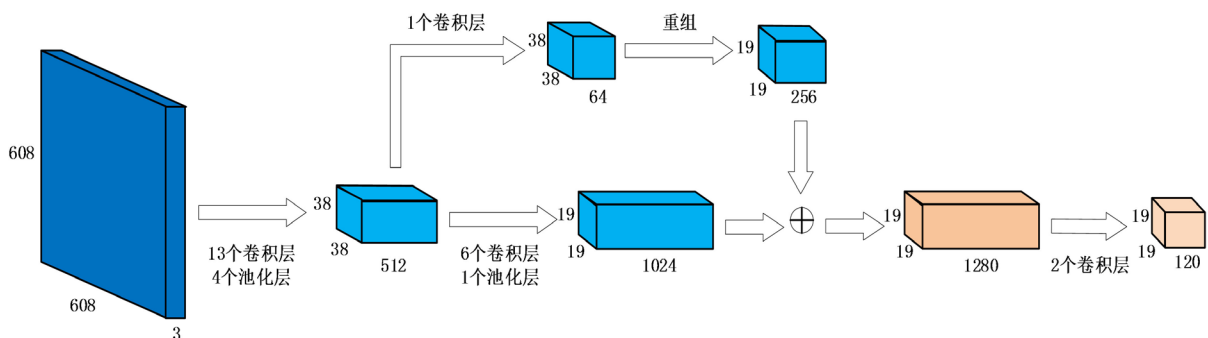


Figure 3. The network structure of YOLOv2

图 3. YOLOv2 的网络结构

### 2.2.2. 检测过程

YOLOv2 在对钢材压印字符检测识别的过程中, 会将整个输入图像划分成  $S \times S$  的网格, 每个网格负责检测中心点落在网格中的对象。同时, YOLOv2 网络借鉴 Faster R-CNN 的思想, 引入 Anchor Boxes 先验框, 不同于 Fast R-CNN 需要人工定义 Anchor Boxes, YOLOv2 采用 K-means 聚类方法对数据集中的人工标记框进行聚类分析, 确定 Anchor boxes 的数量和大小, 最后在网格的周围生成几个一定比例的边



框。那么，每个网格单元预测  $B$  个边界框和这些框的置信度分数，每个边界框包含了该区域中心点的位置  $(x, y)$ ，高度  $h$ ，宽度  $w$  和置信度  $confidence$  这 5 个信息。

置信度表征边界框中是否含有检测对象以及边框预测的准确程度，其计算公式为

$$confidence = P(object) \times IOU_{pred}^{truth} \quad (4)$$

式中， $P(object)$  表示边界框中是否有包含检测对象，若有取值为 1，则置信度  $confidence$  等于  $IOU_{pred}^{truth}$  的值，若边界框中没有包含检测对象， $P(object)$  取值为 0，则置信度  $confidence$  的值也为 0。而  $IOU_{pred}^{truth}$  表示预测边界框与标记的真实边界框的重合程度，其计算方式为

$$IOU_{pred}^{truth} = \frac{Area(pred) \cap Area(truth)}{Area(pred) \cup Area(truth)} \quad (5)$$

即预测边界框与真实边界框所占面积的交集比上两者的并集(Intersection over Union)，该值越接近于 1，表明两者的重合度越高。那么，预测边框某一类别的置信分数  $Conf_i$  就可以通过式(6)进行计算。

$$\begin{aligned} Conf_i &= P(class_i | object) \times P(object) \times IOU_{pred}^{truth} \\ &= P(class_i) \times IOU_{pred}^{truth} \end{aligned} \quad (6)$$

置信分数表征预测边界框中含有某一类别的概率及边框坐标的准确度。综上，当存在  $C$  类别时，网络的输出尺寸为  $S \times S \times B \times (5 + C)$  的张量。根据置信分数的大小，采用非极大值抑制(NMS)筛选获得最终的检测结果。

### 2.2.3. 输出改进

在检测结果输出时，为了能使检测到的字符按生产标号顺序输出，本研究利用检测到的字符位置信息通过冒泡排序按从小到大进行排序，相应的字符从左到右排列，从而得到准确的生产标号。利用字符位置信息进行冒泡排序的方式比较简单，依次比较存放字符位置信息的数组中相邻两个数的大小，若排在第  $i$  位的数比排在第  $i + 1$  位的数要大，则两者进行交换，否则不需要交换。同时位置信息对应的字符数组也进行相同的操作，最终得到位置信息从小到大的序列和按生产标号从左到右的字符序列。

## 3. 实验结果及分析

### 3.1. 实验平台

本实验是在 PC 端上完成，PC 配置为 CPU Intel Core i7-6700，显卡/GPU 为 NVIDIA GeForce GTX 960M，8G 运行内存，训练和测试框架均为 Darknet。

关于网络参数的设置方面，将初始学习率(learning\_rate)设为 0.001，采用 steps 学习率调整策略(policy)；每迭代一次训练的样本数(batch)为 64，最大迭代次数(max\_batches)为 10000 次；学习率在迭代次数为 4000, 6000 时，在原来的基础上再分别乘以 0.1, 0.1；动量(momentum)为 0.9，衰减系数(decay)为 0.0005。

### 3.2. 训练

对 YOLOv2 网络的训练是钢材压印字符识别的关键一环，训练所得模型的好坏直接影响字符识别的效果。本实验利用 LabelImg 软件对训练集中所有图片上的字符打上各自相应的标签，生成每一幅图片对应的.xml 文件，然后将每一个.xml 文件生成 txt 文件，存储标签信息。最后利用设置好参数的 YOLOv2 网络对训练集样本进行训练。

YOLOv2 网络在训练时可以通过计算损失函数  $Loss$  来判断训练的效果，计算方法在等式(7)中示出。

$$\begin{aligned}
 Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{coobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} I_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned} \tag{7}$$

其中,  $I_i^{obj}$  表示目标对象是否出现在网格单元  $i$  中,  $I_{i,j}^{obj}$  表示网格单元  $i$  中的第  $j$  个目标边界框。相反地,  $I_{i,j}^{noobj}$  表示网格单元  $i$  中的第  $j$  个边界框, 其不包含目标对象的任何部分;  $\lambda_{coord}$  为目标坐标预测的损失系数,  $\lambda_{noobj}$  则为不包含有目标的边框预测的损失系数;  $x_i$ 、 $y_i$ 、 $w_i$ 、 $h_i$ 、 $C_i$ 、 $p_i(c)$  为网格  $i$  中预测目标的中心点坐标、宽高、类别及概率,  $\hat{x}_i$ 、 $\hat{y}_i$ 、 $\hat{w}_i$ 、 $\hat{h}_i$ 、 $\hat{C}_i$ 、 $\hat{p}_i(c)$  则为网格  $i$  中目标的各相应信息的真实值。公式(7)的时间复杂度是  $O((k+c) \times S^2)$ , 其是针对一个图像计算的。要得到较好的训练效果, 则期望  $Loss$  值尽可能小。

YOLOv2 网络在训练过程中还采用了多尺度训练策略, 每隔十轮便改变一次图像尺寸进行输入, 以提高模型对不同分辨率图像的鲁棒性。使用 YOLO v2 网络在钢材压印字符训练集训练中随着迭代次数的增加平均损失率所产生的变化如图 4 所示, 从图中可以看出网络收敛速度较快, 在迭代 2000 次左右时基本收敛, 之后处于基本稳定状态, 并且  $Loss$  值越来越接近于 0, 说明训练的效果逐渐趋于最佳。

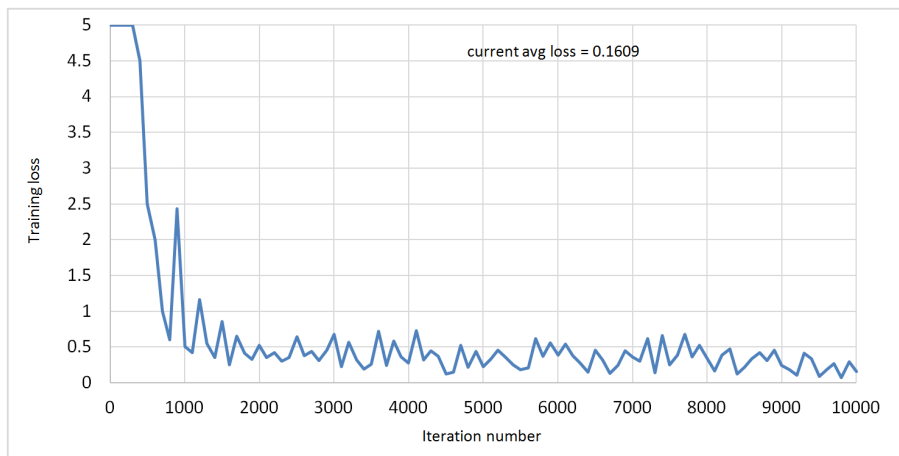


Figure 4. Loss rate curve  
图 4. 损失率曲线图

### 3.3. 测试结果及分析

本研究还将整幅钢材生产标号图像分割成单个字符图像作为样本集, 分别采用 LeNet-5 [20]和 SVM 来对样本集进行训练和分类识别, 对比两者与本研究采用的 YOLOv2 方法的识别效果。采用准确率 (Accuracy, Acc)、假阳性率(False Positive Rate, FPR)、召回率(Recall)、检测单幅图像耗时(Time)等评估参数作为评价性能指标, 如表 1 所示。

Table 1. Performance indicators for recognition assessment

表 1. 识别评估性能指标

方法(Method)	样本(Sample)	测试平均时间(Time)	准确率(Acc)	假阳性率(FPR)	召回率(Recall)
YOLOv2	整幅图像	0.3s	98.6%	0.16	0.97
LeNet-5	单个字符图像	0.6s	96.7%	0.19	0.95
SVM	单个字符图像	0.5s	93%	0.21	0.92

从表中可以看出,采用 YOLOv2 方法进行识别时,每幅图像平均需要花费 0.3 s 的时间,准确率达到 了 98.6%,不仅所用的时间少于其他两种方法,而且识别的准确率还比较高。此外, YOLOv2 的假阳性率比其他两种方法的要低,而召回率高于其他两种方法。实验结果表明,采用 YOLOv2 网络模型对钢材压印字符进行识别具有快速性和高效性,满足工业生产的实时性需求。

采用 YOLOv2 对钢材压印字符进行识别的实际效果如图 5 所示。由图可见, YOLOv2 基本能对数字和字母进行分类,并且能较准确地标识出各字符所在的位置。对于一些特征相近的字符如“8”和“B”,也能被准确地识别出来,而字符“C”和“D”会存在误识成“0”的情况,分析其原因可能跟训练的样本数有关。

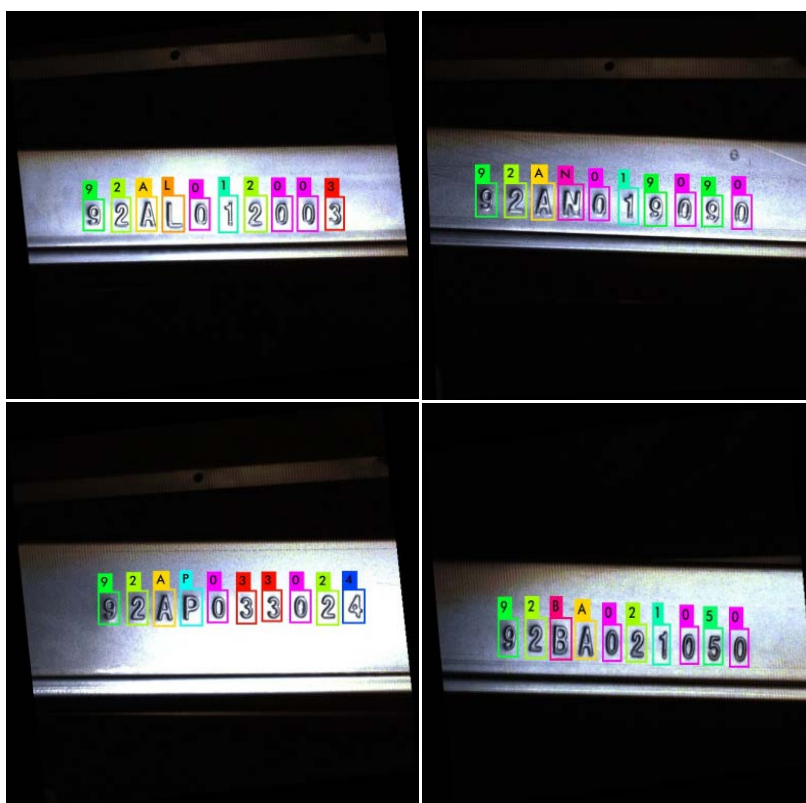


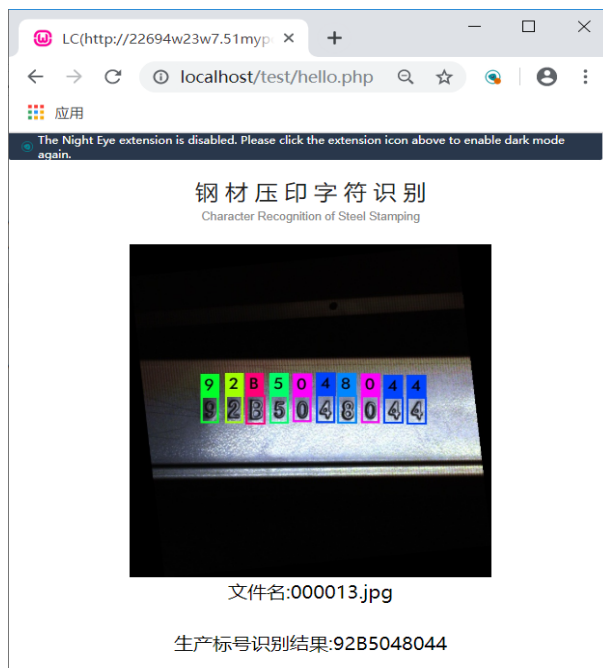
Figure 5. Recognition effect map  
图 5. 识别效果图

同时, YOLOv2 采用整幅生产标号图像作为输入,检测时即能输出一整串生产标号,比先分割字符输入,识别输出单个字符后再进行拼接的方法方便快捷很多。利用 HTML 与 PHP 相结合制作基于 YOLOv2 的钢材压印字符识别平台,可以在网页上远程实现一键得到生产标号,结果显示如图 6 所示。

#### 4. 结论

随着深度学习的深入研究和不断发展,越来越多的针对图像、视频的视觉识别算法被开发出来,这些算法的网络框架性能不断提升并且使用起来十分的便捷。本研究将目前对目标检测比较快速准确的 YOLOv2 算法应用于部件钢印字符识别中,实验结果表明,与其他的经典的字符识别算法如 LeNet-5 和分类器 SVM 相比, YOLOv2 的识别准确率较高,且算法处理时间较少。同时,通过对网络输出结果的改进,可以直接输出图像中正确的生产标号,不需要把图像分割成单个字符再进行识别,更加的方便快捷。





**Figure 6.** Steel stamping character recognition platform based on YOLOv2

**图 6.** 基于 YOLOv2 的钢材压印字符识别平台

这有效提高了工业生产管理的稳定性和实时性。后续的研究工作将不断扩充数据集样本数量，还有对样本图像采用边缘检测等预处理方法。

## 致 谢

本研究室的卢健祥，李壁江等多位同学完成了很多先期准备工作；匿名审稿老师对本文提出了宝贵的修改意见，在此表示感谢！

## 基金项目

本课题得到广西自然科学基金“形状纹理特征量的提取及其优化研究”(No. 2017JJA170765y)；国家自然科学基金“基于 CT 数据的胃癌模型的三维重建和手术仿真”(No. 81760324)资助。

## 参考文献

- [1] 顾晨勤, 葛万成. 基于模板匹配算法的字符识别研究[J]. 通信技术, 2009, 42(3): 220-222.
- [2] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 936-944. <https://doi.org/10.1109/CVPR.2017.106>
- [3] Felzenszwalb, P.F., Girshick, R.B., Mcallester, D., et al. (2009) Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 1627-1645. <https://doi.org/10.1109/TPAMI.2009.167>
- [4] Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, 20-25 June 2005, 886-893. <https://doi.org/10.1109/CVPR.2005.177>
- [5] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. <https://doi.org/10.1007/BF00994018>
- [6] Zhu, Q., Yeh, M.-C., Cheng, K.-T. and Avidan, S. (2006) Fast Human Detection Using a Cascade of Histograms of

- Oriented Gradients. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, 17-22 June 2006, 1491-1498. <https://doi.org/10.1109/CVPR.2006.119>
- [7] Wang, X., Yang, M., Zhu, S. and Lin, Y. (2013) Regionlets for Generic Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, 1-8 December 2013, 17-24. <https://doi.org/10.1109/ICCV.2013.10>
- [8] Azizpour, H. and Laptev, I. (2012) Object Detection Using Strongly-Supervised Deformable Part Models. *Proceedings of the European Conference on Computer Vision*, Florence, 7-13 October 2012, 836-849. [https://doi.org/10.1007/978-3-642-33718-5\\_60](https://doi.org/10.1007/978-3-642-33718-5_60)
- [9] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. <https://doi.org/10.1109/CVPR.2014.81>
- [10] Uijlings, J.R., Van De Sande, K.E., Gevers, T., *et al.* (2013) Selective Search for Object Recognition. *International Journal of Computer Vision*, **104**, 154-171. <https://doi.org/10.1007/s11263-013-0620-5>
- [11] Girshick, R. (2015) Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448. <https://doi.org/10.1109/ICCV.2015.169>
- [12] Ren, S.Q., He, K.M., Girshick, R., *et al.* (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, **28**, 91-99.
- [13] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [14] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 6517-6525. <https://doi.org/10.1109/CVPR.2017.690>
- [15] Nodes, T. and Gallagher, N.C.J. (1982) Median Filters: Some Modifications and Their Properties. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **30**, 739-746. <https://doi.org/10.1109/TASSP.1982.1163951>
- [16] Pérez, P., Gangnet, M. and Blake, A. (2003) Poisson Image Editing. *ACM Transactions on Graphics*, **22**, 313-318. <https://doi.org/10.1145/882262.882269>
- [17] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, San Diego, 7-9 May 2015, 1-14.
- [18] Szegedy, C., Liu, W., Jia, Y., *et al.* (2015) Going Deeper with Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [19] Lin, M., Chen, Q. and Yan, S.C. (2014) Network in Network.
- [20] Lecun, Y., Bottou, L., Bengio, Y., *et al.* (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**, 2278-2324. <https://doi.org/10.1109/5.726791>