基于MXNET的图像检测开发案例

目录

- 个人简介
- 业务背景
- 业务流程
- 开发案例

个人简介

- 王威
- 平安科技智能引擎部深度学习与机器人领域深度学习应用组
- 平安科技联邦学习团队 资深算法研究员
- 中国科学院数学与系统科学研究院数学研究所博士
- 人工智能算法，计算机视觉算法
- 主要业务：各类发票和证件的检测、识别以及信息提取
- 主要开发语言：python
- Email：wangweih45@pingan.com.cn

业务背景

- 存量文档电子化
  - 图书，发票等
- 快速理赔
  - 各类证件的在线识别
- 智能录入
  - 房产证、身份证等
- 文字翻译
  - 手机拍照图片文字的检测识别

业务流程

- 通用模型
  - 文字方向检测
  - 文字位置检测
  - 通用文字识别
  - 关键信息提取
- 专用模型
  - 特定票据检测
  - 特定信息位置检测
  - 特定信息文字识别

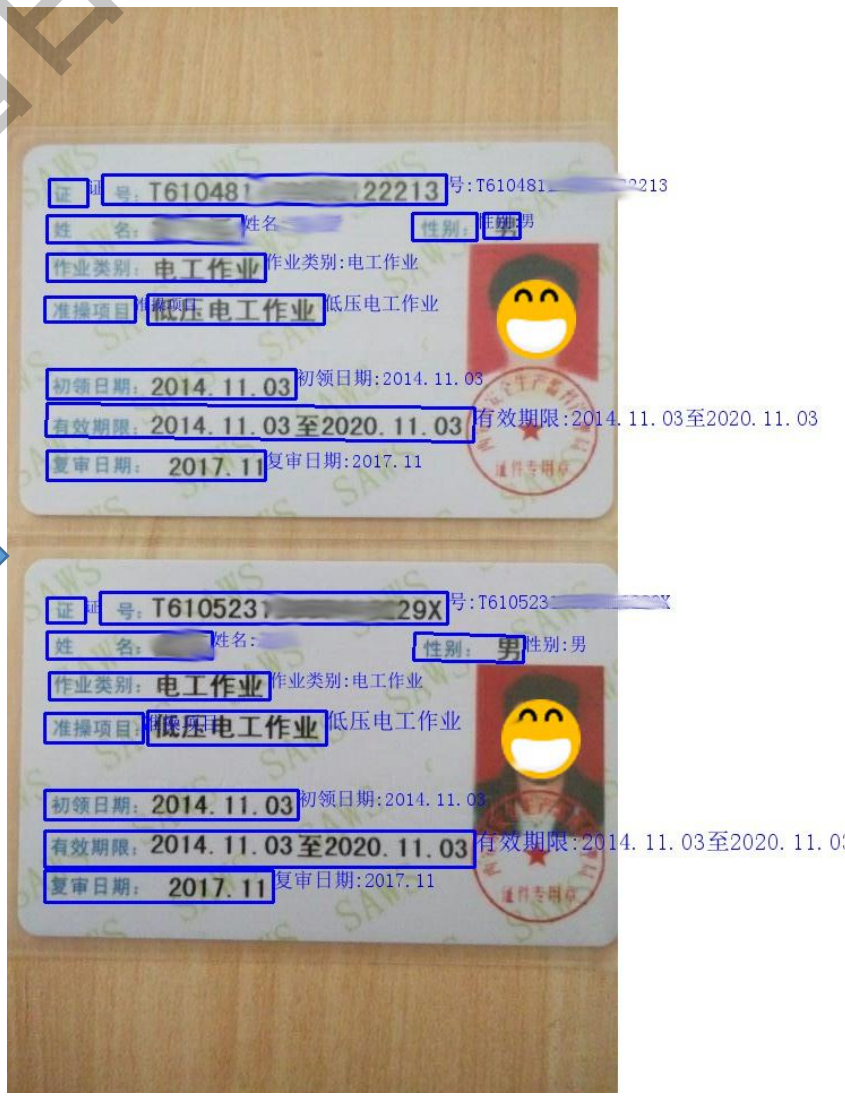
业务流程：通用模型

原始图片

方向检测

方向校正

文字检测 识别



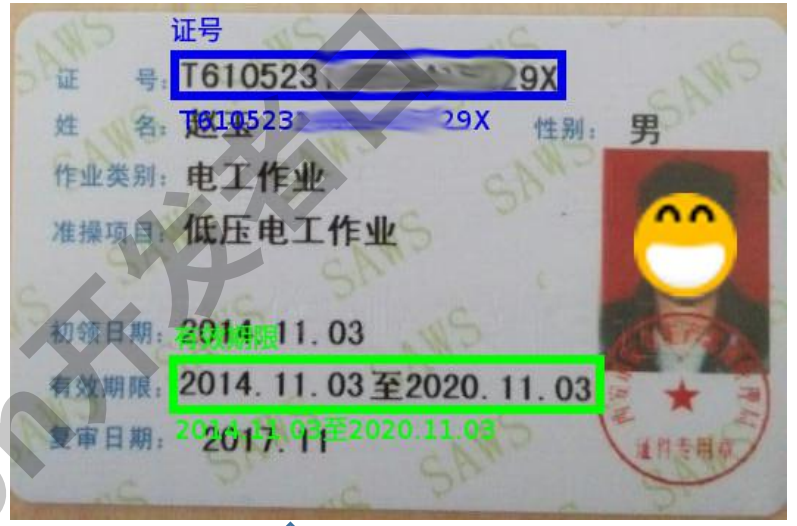
业务流程：专用模型

原始图片

目标检测

裁切校正

特定字段检测识别



开发案例

- 选取案例：特定种类的票据位置检测
  - 原因：目标简单
- 选取框架：MXNET中的Gluon
  - Gluon接口简单易上手
  - 与计算机视觉配套的gluonCV库包含常用的检测网络
  - 文档详细，方便对照相似案例实现定制化目标

开发案例：票据检测

- 相关参考
- https://mxnet.incubator.apache.org/
- https://gluon-cv.mxnet.io/#

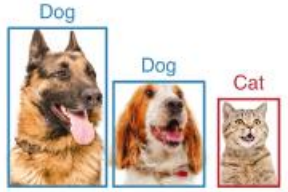
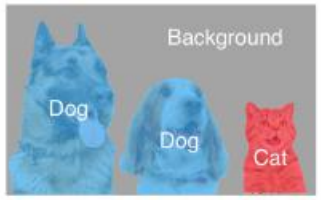
Secure | https://gluon-cv.mxnet.io/index.html

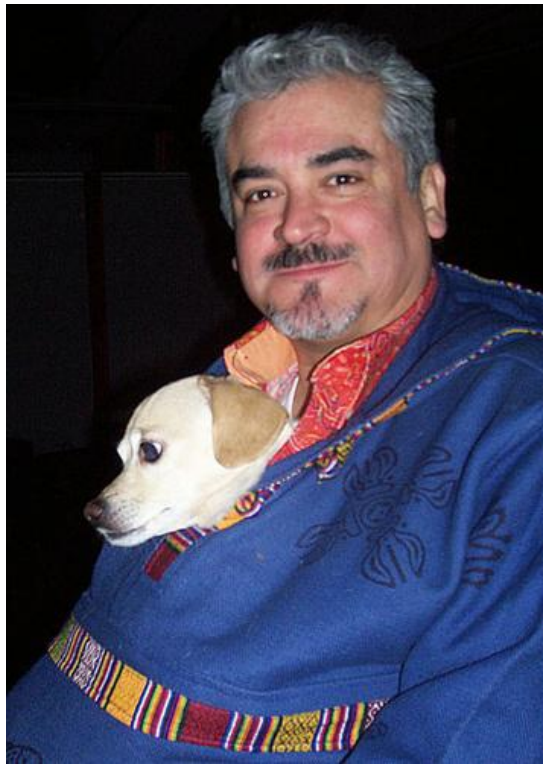
GLUON

- Model Zoo
- Tutorials
  - Image Classification
  - Object Detection
    - 01. Predict with pre-trained SSD models
    - 02. Predict with pre-trained Faster RCNN models
    - 03. Predict with pre-trained YOLO models
    - 04. Train SSD on Pascal VOC dataset
    - 05. Deep dive into SSD training: 3 tips to boost performance
    - 06. Train Faster-RCNN end-to-end on PASCAL VOC
    - 07. Train YOLOv3 on PASCAL VOC

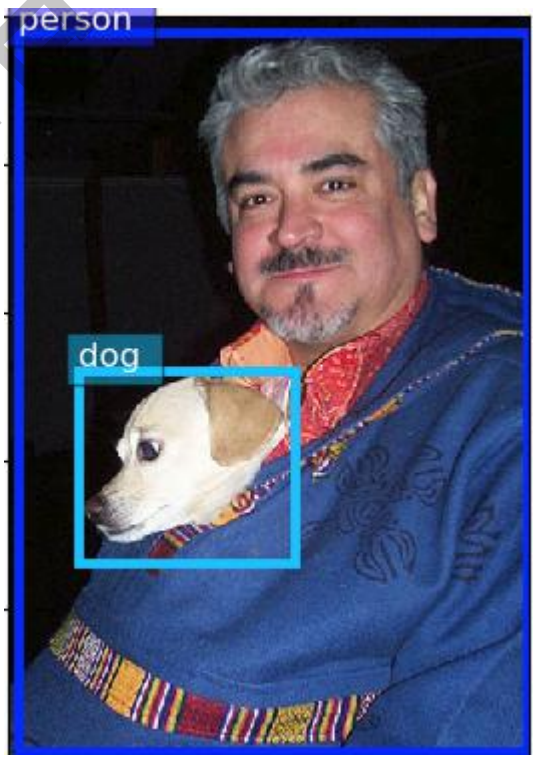
GluonCV: a Deep Learning Toolkit for Computer Vision

Supported Applications

Application	Illustration	Available Models
Image Classification: recognize an object in an image.		50+ models, including ResNet, MobileNet, DenseNet, VGG, ...
Object Detection: detect multiple objects with their bounding boxes in an image.		Faster RCNN, SSD, Yolo-v3
Semantic Segmentation: associate each pixel of an image with a categorical label.		FCN, PSP, DeepLab v3
Instance Segmentation: associate each pixel of an image with an instance label.		Mask RCNN



```
...
<object>
    <name>dog</name>
    <pose>Left</pose>
    <truncated>1</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>48</xmin>
        <ymin>240</ymin>
        <xmax>195</xmax>
        <ymax>371</ymax>
    </bndbox>
</object>
...
```



原图像

标注片段

画图

数据集封装

```
from gluoncv.data import VOCDetection
train_dataset = VOCDetection(splits=[(2007, 'trainval'), (2012, 'trainval')])
val_dataset = VOCDetection(splits=[(2007, 'test')])

class VOCDetection(VisionDataset):
    def __init__(self, ...):
        ...

    ...

    def __len__(self):
        return len(self._items)

    def __getitem__(self, idx):
        img_id = self._items[idx]
        img_path = self._image_path.format(*img_id)
        label = self._label_cache[idx] if self._label_cache else self._load_label(idx)
        img = mx.image.imread(img_path, 1)
        if self._transform is not None:
            return self._transform(img, label)
        return img, label
```

数据集加载以及模型加载

```
from gluoncv.data.transforms import presets
from gluoncv.data.batchify import Tuple
from mxnet.gluon.data import DataLoader

train_transform = presets.ssd.SSDDefaultTrainTransform(width, height)
train_transform = presets.ssd.SSDDefaultTrainTransform(width, height, anchors)
batchify_fn = Tuple(Stack(), Stack(), Stack())
train_loader = DataLoader(
    train_dataset.transform(train_transform),
    batch_size,
    shuffle=True,
    batchify_fn=batchify_fn,
    last_batch='rollover',
    num_workers=num_workers)
```

```
from gluoncv import model_zoo
net = model_zoo.get_model('ssd_300_vgg16_atrous_voc', pretrained_base=False)
```

损失函数及训练

```
from gluoncv.loss import SSDMultiBoxLoss
mbox_loss = SSDMultiBoxLoss()
trainer = gluon.Trainer(
    net.collect_params(), 'sgd',
    {'learning_rate': 0.001, 'wd': 0.0005, 'momentum': 0.9})

for ib, batch in enumerate(train_loader):
    if ib > 0:
        break
    print('data:', batch[0].shape)
    print('class targets:', batch[1].shape)
    print('box targets:', batch[2].shape)
    with autograd.record():
        cls_pred, box_pred, anchors = net(batch[0])
        sum_loss, cls_loss, box_loss = mbox_loss(
            cls_pred, box_pred, batch[1], batch[2])
        # some standard gluon training steps:
        # autograd.backward(sum_loss)
        # trainer.step(1)
```

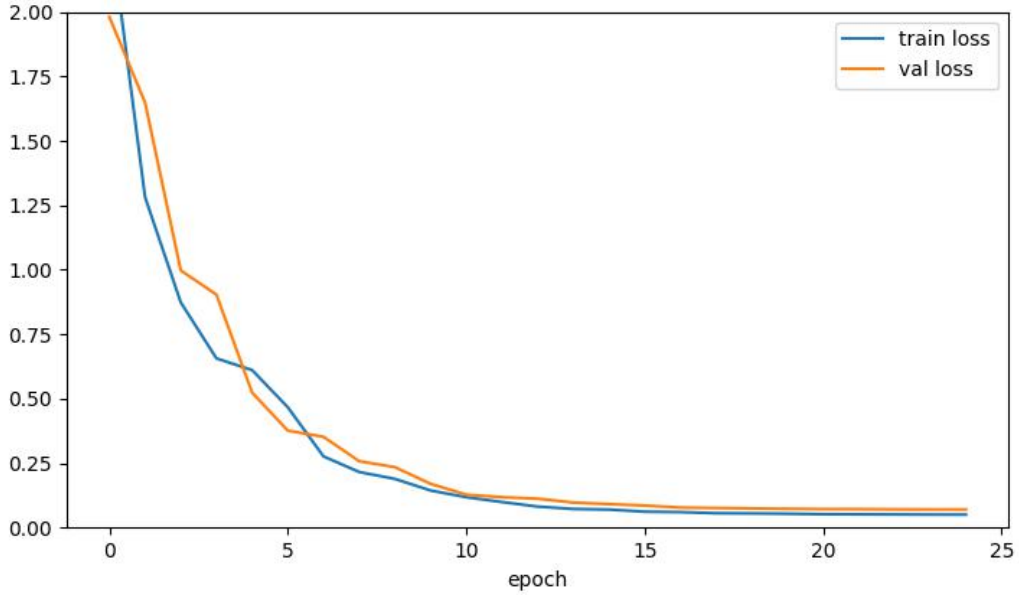
需要注意的问题

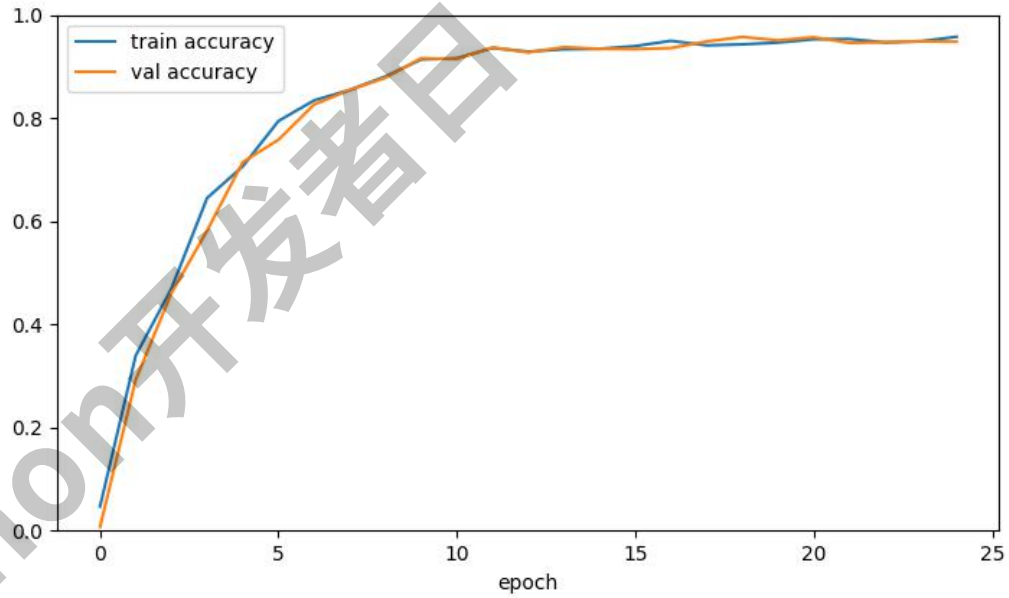
- 损失函数的权重
- 学习率策略
- 多卡训练
- 状态监控
- 模型保存
- 终止条件
- 数据清洗

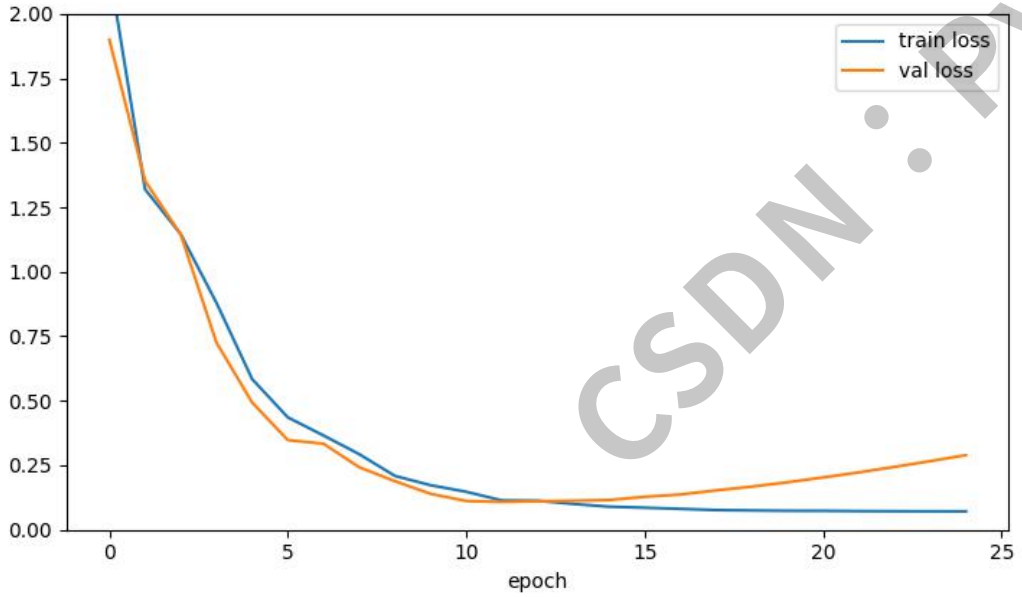
训练过程

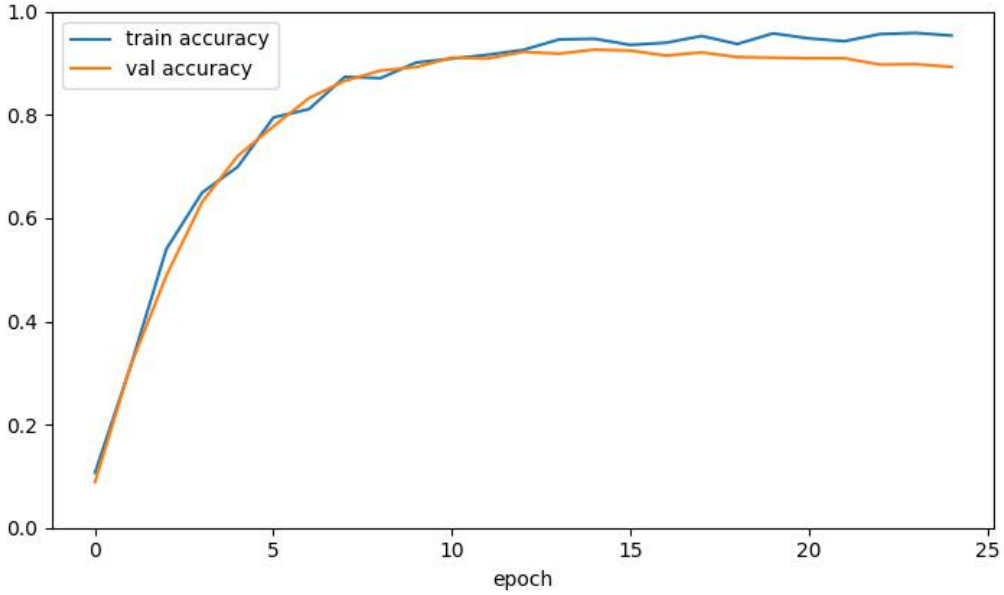
07:39:06 - Epoch 20 train, Batches 001815/001845: Total Loss: 0.596, Cls Loss: 0.326, Box Loss: 0.108, time: 30.43s, epoch left: 0.05h
07:39:18 - Epoch 20 train, Batches 001820/001845: Total Loss: 0.517, Cls Loss: 0.332, Box Loss: 0.074, time: 11.76s, epoch left: 0.02h
07:39:25 - Epoch 20 train, Batches 001825/001845: Total Loss: 0.512, Cls Loss: 0.312, Box Loss: 0.080, time: 7.19s, epoch left: 0.01h
07:39:34 - Epoch 20 train, Batches 001830/001845: Total Loss: 0.583, Cls Loss: 0.388, Box Loss: 0.078, time: 8.84s, epoch left: 0.01h
07:39:41 - Update[35037]: Change learning rate to 1.88677e-04
07:39:45 - Epoch 20 train, Batches 001835/001845: Total Loss: 0.529, Cls Loss: 0.324, Box Loss: 0.082, time: 10.69s, epoch left: 0.01h
07:40:10 - Epoch 20 train, Batches 001840/001845: Total Loss: 0.579, Cls Loss: 0.328, Box Loss: 0.100, time: 25.50s, epoch left: 0.00h
2019-03-07 07:40:16,629 - INFO - Epoch 20, model saved ./models/ssd_512-20.params
2019-03-07 07:41:27,491 - INFO - epoch 20 validation result:
2019-03-07 07:41:27,492 - INFO - ID_Kuang_1 0.9926505575019027
2019-03-07 07:41:27,492 - INFO - ID_ZhaoPian 0.9918573615973989
2019-03-07 07:41:27,492 - INFO - ID_Kuang_2 0.983387269507376
2019-03-07 07:41:27,493 - INFO - X_Kuang_1 0.992835376812076
2019-03-07 07:41:27,493 - INFO - X_YinZhang 0.9857278694205598
2019-03-07 07:41:27,493 - INFO - X_Kuang_2 0.9918454945722626
2019-03-07 07:41:27,494 - INFO - X_Kuang_3 0.9725970915795881
2019-03-07 07:41:27,494 - INFO - QiTa_C 0.8413855138689824
2019-03-07 07:41:27,494 - INFO - mAP 0.9690358168575183

训练结果



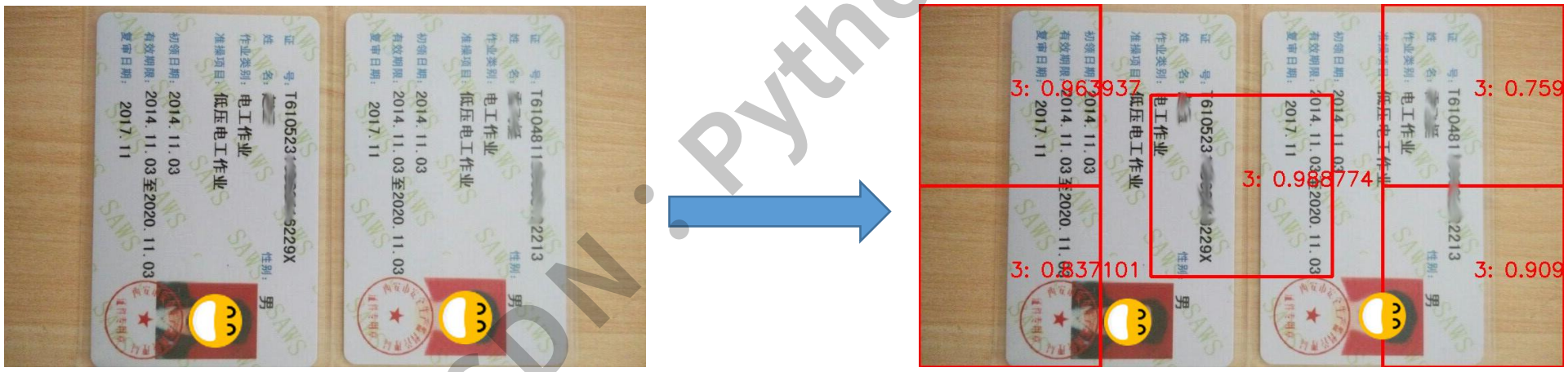






其它简单的例子

- 文字方向检测
- 分类模型
- Mobile net



CSDN

2019
Python开发者日

让开发者紧跟
技术潮流

谢谢