

1.使用fastdeploy进行trt推理

正常情况会出现如下日志：

```
[1, 3, 224, 224]
[WARNING] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(330)::Infer      TensorRT engine will be rebuilt once shape range information changed, this may take
lots of time, you can set a proper shape range before loading model to avoid rebuilding process. refer https://github.com/PaddlePaddle/FastDeploy/blob/develop/
docs/en/faq/tensorrt_tricks.md for more details.
[INFO] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(556)::BuildTrtEngine Start to building TensorRT Engine...
[INFO] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(643)::BuildTrtEngine TensorRT Engine is built successfully.
[INFO] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(645)::BuildTrtEngine Serialize TensorRT Engine to local file /home/txy/code/FastDeploy/model/v3/ch_PP-C
Rv3_det_infer/det_trt_cache.trt.
[INFO] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(656)::BuildTrtEngine TensorRT Engine is serialized to local file /home/txy/code/FastDeploy/model/v3/ch
P-OCRv3_det_infer/det_trt_cache.trt, we can load this model from the serialized engine directly next time.
[ERROR] fastdeploy/runtime/backends/tensorrt/trt_backend.cc(239)::log      3: [runtime.cpp::~Runtime::346] Error Code 3: API Usage Error (Parameter check failed at:
```

大部分模型会存在动态Shape，例如分类的输入为`[-1, 3, 224, 224]`，表示其第一维（batch维）是动态的；检测的输入`[-1, 3, -1, -1]`，表示其batch维，以及高和宽是动态的。而TensorRT在构建引擎时，需要知道这些动态维度的范围。因此，FastDeploy通过以下两种方式来解决

1. 自动设置动态Shape；在加载模型时，如若遇到模型包含动态Shape，则不会立刻创建TensorRT引擎，而是在实际输入数据预测时，获取到数据的Shape，再进行构建。
 - 1.1 由于大部分模型在推理时，Shape都不会变，因此相当于只是将构建的过程推迟到预测阶段，整体没太大影响；
 - 1.2 如若预测过程中，Shape在变化，FastDeploy会不断收集新的Shape，扩大动态维度的变化范围。每次遇到新的Shape且超出范围的，则更新范围，并重新构建TensorRT引擎。因此这样在遇到超过范围的Shape时，会重新花一定时间构建引擎，例如OCR模型存在这种现象，但随着不断预测，数据的Shape范围最终稳定后，便不会再重新构建。

通常来说可以自动进行动态shape的获取，也可以通过手动，先手动设置好其动态范围，这样可以避免预测时重新构建

2.加载模型耗时长

TensorRT每次构建模型的过程较长，FastDeploy提供了Cache机制帮助开发者将构建好的模型缓存在本地，这样在重新运行代码时，可以通过加载Cache，快速完成模型的加载初始化。

接口传入文件路径字符串，当在执行代码时，

- 如若发现传入的文件路径不存在，则会构建TensorRT引擎，在构建完成后，将引擎转换为二进制流存储到此文件路径
- 如若发现传入的文件路径存在，则会跳过构建TensorRT引擎，直接加载此文件并还原成TensorRT引擎

因此，如有修改模型，推理配置（例如Float32改成Float16），需先删除本地的cache文件，避免出错。

3.预测结果转为numpy

```

1 import fastdeploy as fd
2 import cv2
3 import numpy as np
4
5 model = fd.vision.ocr.PPOCRv4(
6     det_model="",rec_model="",cls_model=None)
7 im = cv2.imread(image)
8 result = model.predict(im)
9 # convert label_map and score_map to numpy format
10 numpy_label_map = np.array(result.label_map)
11 numpy_score_map = np.array(result.score_map)
12 result = fd.vision.OCRResult()
13 result.label_map = numpy_label_map.tolist()
14 result.score_map = numpy_score_map.tolist()

```

4.使用CV-CUDA/CUDA加速GPU端到端推理性能

在编译时开启：export ENABLE_CVCUDA=ON

model.preprocessor.use_cuda(True, 0)

示例：

- 如果预处理第一个算子是resize，则要根据实际情况决定resize是否跑在GPU。因为当resize跑在GPU，且图片解码在CPU时，需要把原图copy到GPU内存，开销较大，而resize之后再copy到GPU内存，则往往只需要copy较少的数据

5.切换后端

```

1 import fastdeploy as fd
2 option = fd.RuntimeOption()
3
4 # 切换使用CPU/GPU
5 option.use_cpu()
6 option.use_gpu()
7
8 # 切换不同后端
9 option.use_paddle_backend() # Paddle Inference
10 option.use_trt_backend() # TensorRT
11 option.use_openvino_backend() # OpenVINO
12 option.use_ort_backend() # ONNX Runtime
13 ##在我自己写的推理脚本中，删除了OpenVINO，但是在预编译时也将这几项进行了open


```

6.创建初始化

```
1 ocr = fd.vision.ocr.PPOCRv4(  
2 det_model=det_model,  
3 cls_model=None,  
4 rec_model=rec_model)  
5 ##可添加自己转化后的推理模型
```

7.脚本文件



 trt_infer_fp16_cache.py

单独测试识别部分：rec_bs:测速分析：16：fp32: 1.97ms fp32:1.82

8: fp32: 2.02ms fp32 : 1.85

8.安装+编译（官方）

编译过程需要满足

- gcc/g++ >= 5.4(推荐8.2)
- cmake >= 3.18.0
- python >= 3.6
- CUDA >= 11.2

- cudnn >= 8.2

Python打包依赖，编译前请先执行 `wheel pip install wheel`

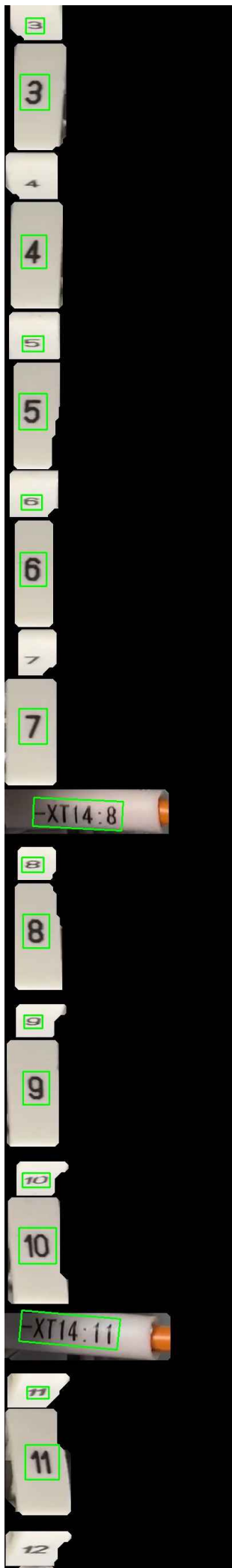
所有编译选项通过环境变量导入（可对文档中的第4点进行添加）

```
1 git clone https://github.com/PaddlePaddle/FastDeploy.git
2 cd FastDeploy/python
3 export ENABLE_ORT_BACKEND=ON
4 export ENABLE_PADDLE_BACKEND=ON
5 export ENABLE_OPENVINO_BACKEND=ON
6 export ENABLE_VISION=ON
7 export ENABLE_TEXT=ON
8 export ENABLE_TRT_BACKEND=ON
9 export WITH_GPU=ON
10 export TRT_DIRECTORY=/Paddle/TensorRT-8.4.1.5
11 export CUDA_DIRECTORY=/usr/local/cuda
12 # OPENCV_DIRECTORY可选，不指定会在编译过程下载FastDeploy预编译的OpenCV库
13 export OPENCV_DIRECTORY=/usr/lib/x86_64-linux-gnu/cmake/opencv4 \
14
15 python setup.py build
16 python setup.py bdist_wheel
17 ###可对文档中的第4点进行添加
```

9.添加数据结果对比

现在模型效果







识别结果:

- 1 cost : 0.03218507766723633 s
- 2 det boxes: [[47,153],[78,153],[78,201],[47,201]]rec text: 1 rec score:0.589659
- 3 det boxes: [[74,303],[333,303],[333,444],[74,444]]rec text: XT14 rec score:0.999860
- 4 det boxes: [[98,569],[257,556],[261,607],[102,619]]rec text: -XT14:3 rec score:0.920201
- 5 det boxes: [[31,703],[82,703],[82,767],[31,767]]rec text: 2 rec score:0.997361
- 6 det boxes: [[37,885],[69,885],[69,912],[37,912]]rec text: 3 rec score:0.992192
- 7 det boxes: [[27,984],[78,984],[78,1046],[27,1046]]rec text: 3 rec score:0.998466
- 8 det boxes: [[29,1266],[73,1266],[73,1324],[29,1324]]rec text: 4 rec score:0.998958
- 9 det boxes: [[31,1443],[68,1443],[68,1470],[31,1470]]rec text: 5 rec score:0.995508
- 10 det boxes: [[25,1544],[74,1544],[74,1607],[25,1607]]rec text: 5 rec score:0.999206
- 11 det boxes: [[29,1722],[65,1722],[65,1748],[29,1748]]rec text: 6 rec score:0.972368
- 12 det boxes: [[27,1823],[73,1823],[73,1882],[27,1882]]rec text: 6 rec score:0.988535
- 13 det boxes: [[25,2097],[74,2097],[74,2159],[25,2159]]rec text: 7 rec score:0.995513
- 14 det boxes: [[51,2253],[207,2260],[204,2311],[49,2303]]rec text: -XT14:8 rec score:0.886726

15 det boxes: `[[30,2358],[68,2358],[68,2384],[30,2384]]`rec text: 8 rec
score:0.985233

16 det boxes: `[[32,2458],[78,2458],[78,2517],[32,2517]]`rec text: 8 rec
score:0.992275

17 det boxes: `[[34,2635],[66,2635],[66,2658],[34,2658]]`rec text: 9 rec
score:0.957484

18 det boxes: `[[32,2734],[78,2734],[78,2792],[32,2792]]`rec text: 9 rec
score:0.950624

19 det boxes: `[[31,2912],[79,2912],[79,2937],[31,2937]]`rec text: 10 rec
score:0.993363

20 det boxes: `[[25,3008],[90,3008],[90,3071],[25,3071]]`rec text: 10 rec
score:0.995965

21 det boxes: `[[30,3153],[202,3168],[196,3222],[25,3207]]`rec text: -XT14:11 rec
score:0.941879

22 det boxes: `[[39,3287],[77,3287],[77,3308],[39,3308]]`rec text: 11 rec
score:0.557845

23 det boxes: `[[36,3389],[95,3389],[95,3450],[36,3450]]`rec text: 11 rec
score:0.991938

24 det boxes: `[[31,3665],[97,3665],[97,3729],[31,3729]]`rec text: 12 rec
score:0.998867

25 det boxes: `[[42,3847],[73,3847],[73,3859],[42,3859]]`rec text: rec
score:0.000000

26 det boxes: `[[30,3942],[98,3942],[98,4006],[30,4006]]`rec text: 13 rec
score:0.998505

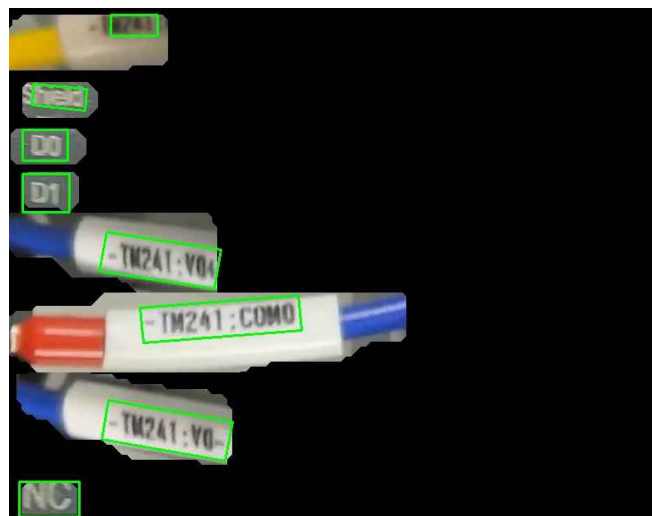
27 det boxes: `[[40,4118],[82,4118],[82,4139],[40,4139]]`rec text: 14 rec
score:0.470080

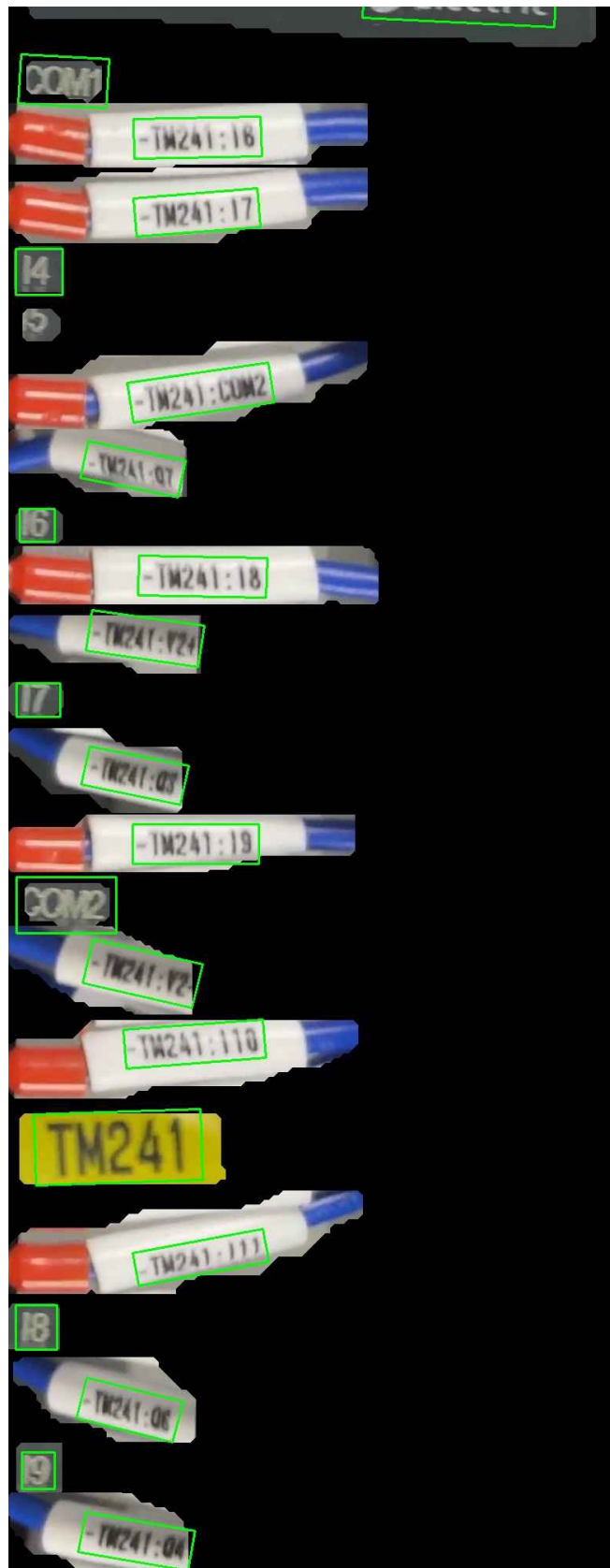
28 det boxes: `[[30,4219],[101,4219],[101,4281],[30,4281]]`rec text: 14 rec
score:0.998462

29 det boxes: `[[37,4375],[189,4392],[183,4448],[31,4431]]`rec text: -XT14:15 rec
score:0.952122

30 det boxes: `[[47,4508],[82,4508],[82,4527],[47,4527]]`rec text: 15 rec
score:0.806172

31 det boxes: `[[39,4608],[106,4608],[106,4673],[39,4673]]`rec text: 15 rec
score:0.997307





1 cost : 0.16536998748779297 s

2 det boxes: [[122,8],[178,8],[178,33],[122,33]]rec text: 741 rec score:0.680486

3 det boxes: [[29,91],[93,97],[91,123],[27,116]]rec text: Ted rec score:0.796900

4 det boxes: [[16,146],[70,146],[70,183],[16,183]]rec text: D0 rec score:0.859039

5 det boxes: [[16,199],[72,199],[72,245],[16,245]]rec text: D1 rec score:0.923178

6 det boxes: **[[117,269],[254,290],[246,337],[108,316]]**rec text: -TM241:V0 rec score:**0.761900**

7 det boxes: **[[157,360],[347,345],[351,387],[161,402]]**rec text: -TM241:COMO rec score:**0.852212**

8 det boxes: **[[119,471],[268,497],[259,544],[111,518]]**rec text: -TM241:V0- rec score:**0.773561**

9 det boxes: **[[12,569],[84,569],[84,611],[12,611]]**rec text: NC rec score:**0.960080**

10 det boxes: **[[103,644],[152,644],[152,683],[103,683]]**rec text: -T rec score:**0.539720**

11 det boxes: **[[2,702],[85,702],[85,758],[2,758]]**rec text: NC rec score:**0.986101**

12 det boxes: **[[115,784],[168,784],[168,821],[115,821]]**rec text: rec score:**0.000000**

13 det boxes: **[[12,824],[140,831],[137,893],[9,886]]**rec text: COMO rec score:**0.875429**

14 det boxes: **[[174,913],[337,903],[340,953],[176,962]]**rec text: -TM241:**11** rec score:**0.841814**

15 det boxes: **[[160,996],[325,984],[329,1029],[164,1041]]**rec text: -TM241:**12** rec score:**0.888785**

16 det boxes: **[[6,1071],[63,1071],[63,1120],[6,1120]]**rec text: **10** rec score:**0.938446**

17 det boxes: **[[187,1130],[337,1141],[333,1190],[183,1179]]**rec text: TM241:**13** rec score:**0.920443**

18 det boxes: **[[16,1224],[68,1224],[68,1284],[16,1284]]**rec text: **1** rec score:**0.962844**

19 det boxes: **[[12,1299],[63,1299],[63,1344],[12,1344]]**rec text: **12** rec score:**0.413053**

20 det boxes: **[[106,1355],[250,1381],[242,1429],[98,1403]]**rec text: -TM241:V14 rec score:**0.795149**

21 det boxes: **[[167,1453],[343,1449],[344,1496],[168,1500]]**rec text: -TM241:COM1 rec score:**0.878773**

22 det boxes: **[[156,1614],[310,1590],[318,1641],[164,1665]]**rec text: TM241:**14** rec score:**0.795654**

23 det boxes: **[[115,1721],[253,1745],[245,1792],[107,1768]]**rec text: -TM241:01 rec score:**0.815473**

24 det boxes: **[[110,1812],[236,1834],[228,1879],[102,1858]]**rec text: -TM241:Q2 rec score:**0.717482**

25 det boxes: **[[122,1913],[250,1945],[238,1991],[110,1959]]**rec text: -TMZA1:00 rec score:**0.746011**

26 det boxes: **[[100,2011],[239,2039],[230,2089],[89,2060]]**rec text: **241:1-** rec score:**0.686419**

27 det boxes: **[[166,2146],[321,2121],[329,2168],[174,2193]]**rec text: -T4241:**15** rec score:**0.825224**

28 det boxes: **[[559,2268],[754,2276],[751,2349],[556,2341]]**rec text: M241 rec score:**0.999164**

29 det boxes: **[[363,2366],[736,2382],[733,2440],[360,2424]]**rec text: TM241CE40T rec score:**0.996103**

30 det boxes: [[27,2439],[261,2463],[253,2539],[19,2515]]rec text: atibility rec
score:0.966913

31 det boxes: [[658,2463],[740,2463],[740,2518],[658,2518]]rec text: ce rec
score:0.991547

32 det boxes: [[23,2542],[145,2549],[141,2610],[19,2603]]rec text: ce: rec
score:0.965030

33 det boxes: [[658,2542],[735,2542],[735,2593],[658,2593]]rec text: er rec
score:0.997671

34 det boxes: [[39,2637],[270,2649],[266,2727],[35,2715]]rec text: rec
score:0.000000

35 det boxes: [[348,2644],[379,2644],[379,2676],[348,2676]]rec text: 9 rec
score:0.492025

36 det boxes: [[351,2697],[375,2697],[375,2716],[351,2716]]rec text: - rec
score:0.113345

37 det boxes: [[409,2764],[589,2780],[585,2826],[405,2810]]rec text: Scan me rec
score:0.993952

38 det boxes: [[382,3038],[714,3058],[710,3128],[378,3108]]rec text: Schneider
rec score:0.998015

39 det boxes: [[457,3096],[703,3108],[700,3163],[454,3151]]rec text: SElectric
rec score:0.956866

40 det boxes: [[16,3198],[129,3204],[125,3267],[12,3261]]rec text: COMI rec
score:0.816508

41 det boxes: [[160,3281],[323,3278],[325,3329],[161,3332]]rec text: -TM241:16
rec score:0.828331

42 det boxes: [[160,3383],[319,3370],[323,3420],[164,3432]]rec text: -TM241:17
rec score:0.891662

43 det boxes: [[9,3448],[69,3448],[69,3507],[9,3507]]rec text: 14 rec
score:0.972269

44 det boxes: [[152,3620],[333,3594],[341,3645],[160,3671]]rec text: -M241:CUM2
rec score:0.814113

45 det boxes: [[102,3695],[227,3720],[217,3767],[92,3742]]rec text: -M241-07 rec
score:0.731979

46 det boxes: [[14,3782],[59,3782],[59,3824],[14,3824]]rec text: 16 rec
score:0.952959

47 det boxes: [[167,3842],[332,3845],[330,3896],[166,3893]]rec text: -TM241:18
rec score:0.892993

48 det boxes: [[107,3912],[251,3932],[243,3985],[99,3964]]rec text: -N241:V24 rec
score:0.699081

49 det boxes: [[10,4006],[66,4006],[66,4049],[10,4049]]rec text: 17 rec
score:0.779501

50 det boxes: [[104,4087],[231,4112],[220,4162],[93,4136]]rec text: -24103 rec
score:0.709776

51 det boxes: [[159,4187],[321,4187],[321,4238],[159,4238]]rec text: -1M241:19
rec score:0.857441

52 det boxes: [[10,4255],[138,4255],[138,4327],[10,4327]]rec text: COME rec
score:0.793805

53 det boxes: `[[108,4335],[249,4367],[235,4423],[95,4391]]`rec text: `-241` rec
score:`0.780834`

54 det boxes: `[[147,4451],[326,4438],[330,4487],[151,4501]]`rec text: `-TH241:110`
rec score:`0.894988`

55 det boxes: `[[32,4560],[246,4553],[249,4644],[35,4651]]`rec text: `TM241` rec
score:`0.998224`

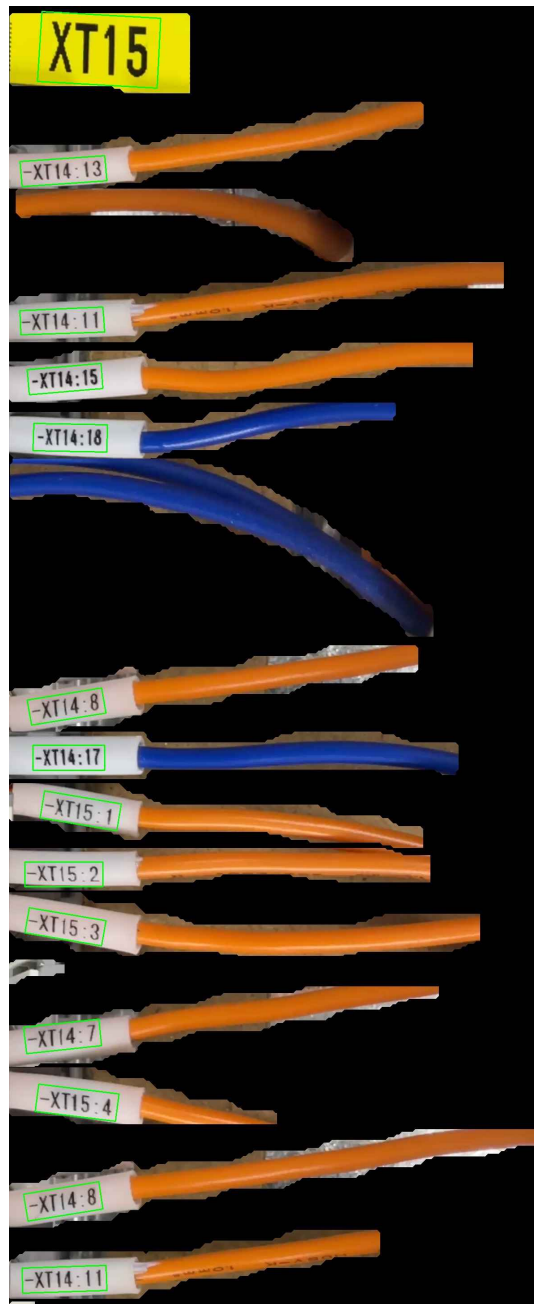
56 det boxes: `[[159,4738],[326,4707],[334,4749],[167,4780]]`rec text: `-T241:111`
rec score:`0.700617`

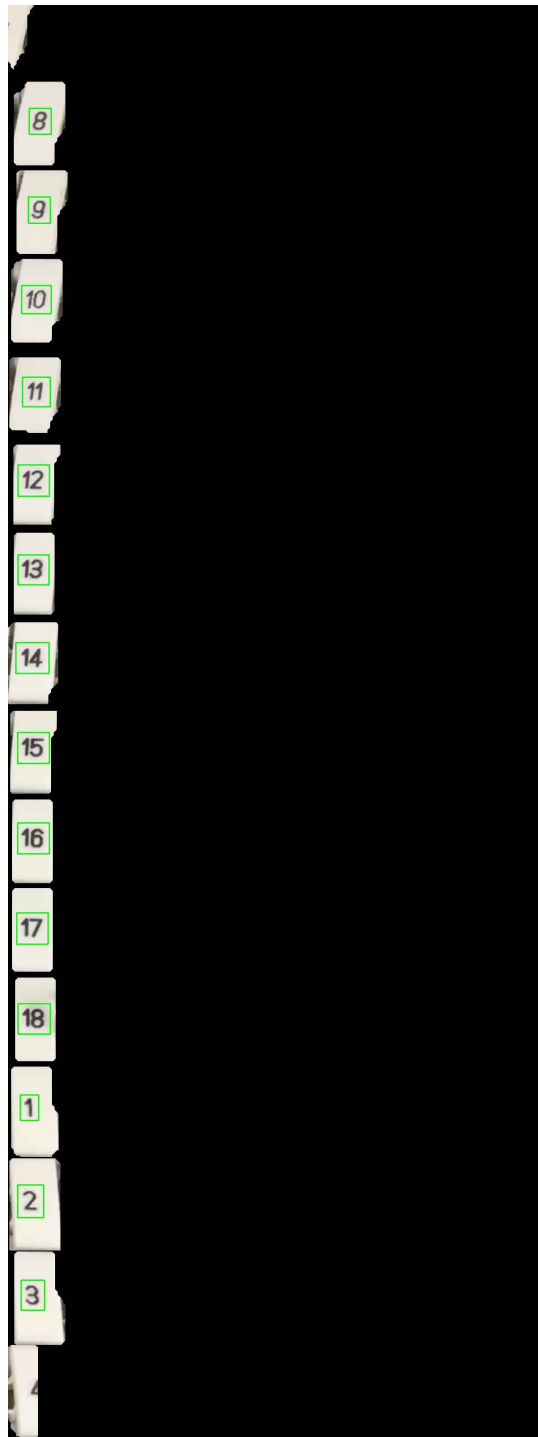
57 det boxes: `[[9,4805],[62,4805],[62,4862],[9,4862]]`rec text: `18` rec
score:`0.946872`

58 det boxes: `[[100,4898],[224,4929],[210,4981],[87,4950]]`rec text: `-T241:06` rec
score:`0.702381`

59 det boxes: `[[17,4994],[59,4994],[59,5041],[17,5041]]`rec text: `19` rec
score:`0.979047`

60 det boxes: `[[99,5071],[239,5095],[230,5146],[89,5122]]`rec text: `-TM241:04` rec
score:`0.805946`





识别结果

```
1 cost : 0.16294026374816895 s
2 det boxes: [[73,12],[338,27],[330,183],[63,168]]rec text: XT15 rec
  score:0.999683
3 det boxes: [[24,347],[213,334],[218,390],[29,403]]rec text: -XT14:13 rec
  score:0.926431
4 det boxes: [[24,680],[209,672],[213,728],[26,734]]rec text: -XT14:11 rec
  score:0.922627
5 det boxes: [[40,812],[205,798],[209,854],[45,868]]rec text: -XT14:15 rec
  score:0.924615
6 det boxes: [[55,932],[219,935],[218,996],[53,993]]rec text: -XT14:18 rec
  score:0.934322
```

7 det boxes: `[[40,1550],[206,1523],[216,1579],[50,1606]]`rec text: -XT14:8 rec score:0.904030

8 det boxes: `[[53,1651],[213,1651],[213,1707],[53,1707]]`rec text: -XT14:17 rec score:0.947486

9 det boxes: `[[79,1753],[250,1782],[240,1842],[69,1813]]`rec text: -XT15:1 rec score:0.898311

10 det boxes: `[[35,1912],[209,1912],[209,1967],[35,1967]]`rec text: -XT15:2 rec score:0.911895

11 det boxes: `[[45,2014],[216,2043],[206,2099],[35,2070]]`rec text: -XT15:3 rec score:0.904153

12 det boxes: `[[32,2281],[200,2263],[206,2322],[39,2340]]`rec text: -XT14:7 rec score:0.880238

13 det boxes: `[[66,2408],[245,2430],[237,2491],[58,2469]]`rec text: -XT15:4 rec score:0.887178

14 det boxes: `[[32,2654],[200,2626],[209,2687],[42,2714]]`rec text: -XT14:8 rec score:0.893468

15 det boxes: `[[29,2819],[216,2813],[218,2869],[30,2875]]`rec text: -XT14:11 rec score:0.935541

16 det boxes: `[[47,3149],[96,3149],[96,3207],[47,3207]]`rec text: 8 rec score:0.995606

17 det boxes: `[[45,3347],[94,3347],[94,3406],[45,3406]]`rec text: 9 rec score:0.852599

18 det boxes: `[[30,3545],[96,3545],[96,3608],[30,3608]]`rec text: 10 rec score:0.997616

19 det boxes: `[[32,3750],[94,3750],[94,3816],[32,3816]]`rec text: 11 rec score:0.997870

20 det boxes: `[[22,3946],[92,3946],[92,4016],[22,4016]]`rec text: 12 rec score:0.999527

21 det boxes: `[[22,4147],[91,4147],[91,4214],[22,4214]]`rec text: 13 rec score:0.998755

22 det boxes: `[[17,4343],[91,4343],[91,4412],[17,4412]]`rec text: 14 rec score:0.998401

23 det boxes: `[[21,4544],[92,4544],[92,4613],[21,4613]]`rec text: 15 rec score:0.998512

24 det boxes: `[[21,4746],[92,4746],[92,4815],[21,4815]]`rec text: 16 rec score:0.998548

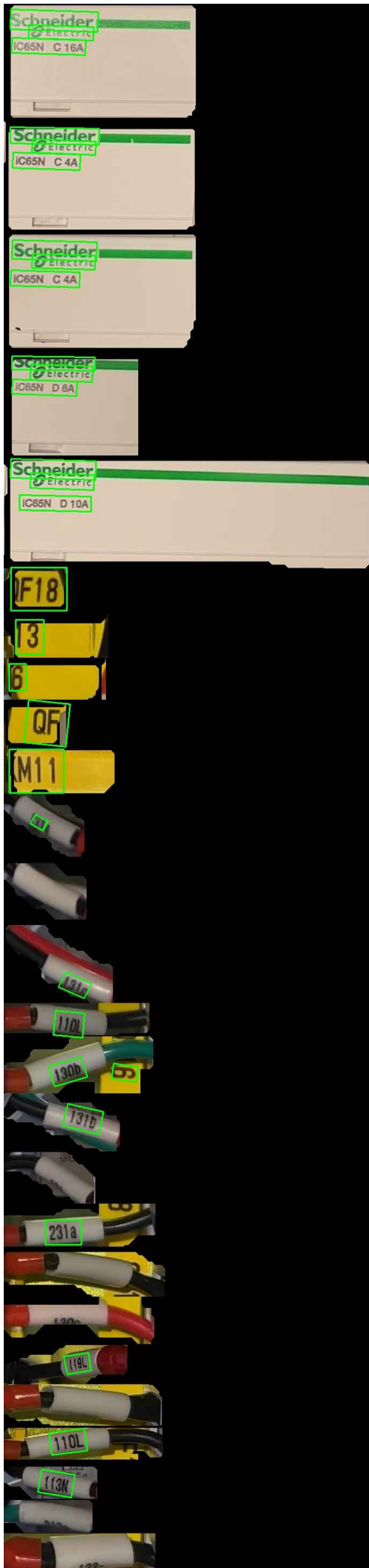
25 det boxes: `[[19,4947],[89,4947],[89,5017],[19,5017]]`rec text: 17 rec score:0.999239

26 det boxes: `[[22,5150],[94,5150],[94,5218],[22,5218]]`rec text: 18 rec score:0.998648

27 det boxes: `[[27,5354],[68,5354],[68,5411],[27,5411]]`rec text: 1 rec score:0.995555

28 det boxes: `[[21,5555],[79,5555],[79,5628],[21,5628]]`rec text: 2 rec score:0.998772

29 det boxes: `[[29,5768],[81,5768],[81,5835],[29,5835]]`rec text: 3 rec score:0.998775





识别结果

- 1 cost : 0.11083579063415527 s
- 2 det boxes: [[13,10],[175,15],[174,52],[12,47]]rec text: Schneider rec score:0.996893
- 3 det boxes: [[47,39],[167,43],[166,68],[46,64]]rec text: SElectric rec score:0.867213
- 4 det boxes: [[15,63],[152,67],[151,95],[14,91]]rec text: IC65N C16A rec score:0.920211
- 5 det boxes: [[12,229],[177,232],[177,262],[12,259]]rec text: Schneider rec score:0.997690
- 6 det boxes: [[53,251],[171,256],[170,281],[52,276]]rec text: SElectric rec score:0.940187
- 7 det boxes: [[17,276],[142,280],[141,309],[16,305]]rec text: iC65N C4A rec score:0.935632
- 8 det boxes: [[16,439],[173,443],[172,478],[15,474]]rec text: Schneider rec score:0.998638
- 9 det boxes: [[53,465],[170,468],[169,494],[52,491]]rec text: OElectric rec score:0.847988
- 10 det boxes: [[13,495],[141,499],[140,527],[12,524]]rec text: iC65N C4A rec score:0.934614
- 11 det boxes: [[15,655],[169,657],[169,682],[15,680]]rec text: Scneider rec score:0.937288
- 12 det boxes: [[50,670],[165,674],[164,703],[49,699]]rec text: Delectric rec score:0.865721
- 13 det boxes: [[16,698],[136,702],[135,729],[15,725]]rec text: jC65N D6A rec score:0.861412
- 14 det boxes: [[13,846],[171,850],[170,885],[12,881]]rec text: Schneider rec score:0.998609

15 det boxes: **[[50,872],[168,876],[167,904],[49,900]]**rec text: SElectric rec
score:0.908258

16 det boxes: **[[29,913],[162,916],[162,944],[29,941]]**rec text: iC65ND10A rec
score:0.940731

17 det boxes: **[[13,1048],[117,1048],[117,1128],[13,1128]]**rec text: F18 rec
score:0.994459

18 det boxes: **[[22,1145],[74,1145],[74,1209],[22,1209]]**rec text: 13 rec
score:0.896445

19 det boxes: **[[10,1227],[41,1227],[41,1277],[10,1277]]**rec text: 6 rec
score:0.869894

20 det boxes: **[[46,1294],[123,1302],[115,1381],[37,1373]]**rec text: QF rec
score:0.977132

21 det boxes: **[[10,1386],[111,1386],[111,1467],[10,1467]]**rec text: M11 rec
score:0.995158

22 det boxes: **[[58,1507],[82,1520],[72,1537],[49,1524]]**rec text: R1 rec
score:0.486743

23 det boxes: **[[115,1800],[163,1822],[149,1849],[101,1828]]**rec text: 131c rec
score:0.818394

24 det boxes: **[[96,1872],[150,1877],[146,1922],[92,1917]]**rec text: 110L rec
score:0.922502

25 det boxes: **[[83,1976],[143,1958],[155,1995],[94,2013]]**rec text: 130b rec
score:0.994865

26 det boxes: **[[204,1970],[250,1977],[246,2005],[200,1998]]**rec text: 9 rec
score:0.517990

27 det boxes: **[[119,2045],[185,2060],[176,2102],[109,2086]]**rec text: 131D rec
score:0.886535

28 det boxes: **[[77,2261],[144,2261],[144,2307],[77,2307]]**rec text: 231a rec
score:0.995538

29 det boxes: **[[111,2515],[157,2508],[162,2541],[118,2548]]**rec text: 119L rec
score:0.903514

30 det boxes: **[[84,2657],[149,2649],[155,2690],[89,2698]]**rec text: 110L rec
score:0.968949

31 det boxes: **[[70,2725],[132,2735],[125,2777],[63,2767]]**rec text: 113N rec
score:0.935571

32 det boxes: **[[122,2957],[179,2973],[168,3013],[110,2996]]**rec text: 410L rec
score:0.979038

33 det boxes: **[[135,3127],[203,3127],[203,3172],[135,3172]]**rec text: 133+ rec
score:0.979463

34 det boxes: **[[91,3203],[152,3209],[148,3247],[87,3241]]**rec text: 110N rec
score:0.846091

35 det boxes: **[[85,3264],[140,3273],[134,3312],[79,3303]]**rec text: 134t rec
score:0.853025

36 det boxes: **[[41,3335],[101,3357],[85,3401],[24,3378]]**rec text: 418L rec
score:0.952032

37 det boxes: **[[119,3488],[147,3498],[141,3515],[112,3505]]**rec text: 2 rec
score:0.330990

38 det boxes: `[[69,3560],[123,3560],[123,3587],[69,3587]]`rec text: SLON rec
score:0.434535

39 det boxes: `[[183,3687],[238,3687],[238,3713],[183,3713]]`rec text: 6181 rec
score:0.497699

40 det boxes: `[[100,3763],[140,3761],[141,3777],[101,3780]]`rec text: 30 rec
score:0.760968

41 det boxes: `[[96,3832],[163,3824],[167,3862],[101,3870]]`rec text: 130a rec
score:0.997343