

Report of LPR

@Haoyu, Zhang

Date: 2023/05/24

描述: 针对 LPR 的改进

摘要

- 1. 使用多任务学习的方法，同时进行车牌字符分割和车牌识别，以进一步提高车牌识别的准确率，针对双行车牌的识别准确率有显著提升。并且改用 Focal CTC Loss 以减少韩文字符和数字分布不均对模型训练的影响。
- 2. 对于新的硬件平台 coral 进行了针对性的优化了车牌识别算法，使得车牌识别的速度得到了提升。
- 3. 利用合成数据，针对车牌检测和车牌识别缺少的双行车牌的数据进行了补充。并对于一行车牌的数据进行扩展，尤其是在电动车牌和新式车牌部分进行了补足。

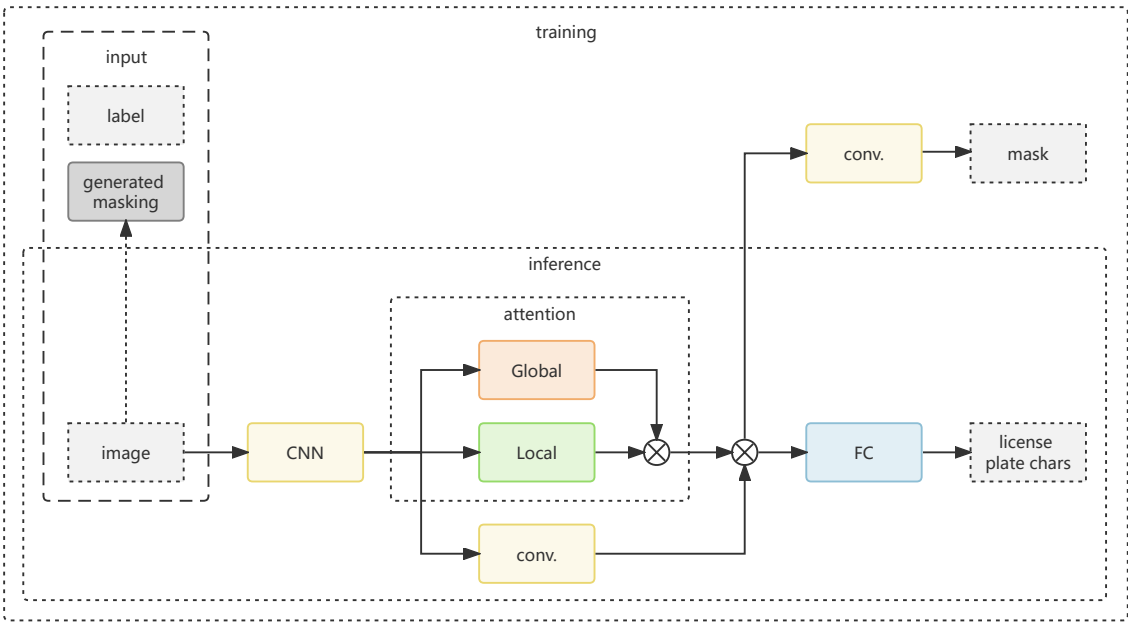
内容

算法改进

近些年来随着多任务学习的发展，越来越多的实验证明多任务学习能够进一步提高准确度。与单任务学习相比，主要有以下几个方面的优势。

- 1. 多任务学习通过挖掘任务之间的关系，能够得到额外的有用信息，大部分情况下都要比单任务学习的效果要好。在有标签样本比较少的情况下，单任务学习模型往往不能够学习得到足够的信息，表现较差，多任务学习能克服当前任务样本较少的缺点，从其他任务里获取有用信息，学习得到效果更好、更鲁棒的机器学习模型
- 2. 多任务学习有更好的模型泛化能力，通过同时学习多个相关的任务，得到的共享模型能够直接应用到将来的某个相关联的任务上。

受此启发，针对 LPR 的任务设计了如下的网络。



针对同一张输入的车牌，进行图像分割和车牌识别两个任务的多任务学习。模型在训练时，同时生成分割的结果以及车牌结果，但在实际部署的推理模型中，我们去除了负责分割任务的head以进一步精简模型和提高推理速度。

Workflow

首先，对输入的车牌图像进行简单的图像处理，得到大致的字符掩码。由于数据量较大，且图像分割的准确度并不是我们优先考虑的指标，所以仅仅使用简单的 OpenCV 的图像处理操作来生成对应的掩码。

这里补一个 opencv 的图像处理的 pipeline

Consider the white and black color chars in the license plate, we use different pipeline to generate mask.

Samples

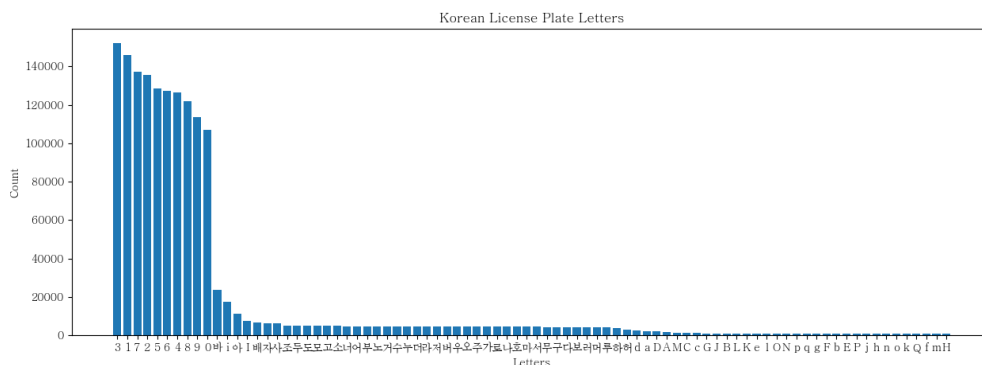


Results

通过将这些字符掩码同样作为输入，不但可以加快注意力网络的学习的收敛速度，从而进一步提升车牌识别的精度。

数据增强

车牌数据存在明显的区域性分布，并且不同字符的分布也存在着较大的差距。



其中，A-Q，a-q分别表示

```
{
    '서울': 'A', '부산': 'B', '대구': 'C', '인천': 'D',
    '광주': 'E', '대전': 'F', '울산': 'G', '세종': 'H',
    '경기': 'I', '강원': 'J', '충북': 'K', '충남': 'L',
    '전북': 'M', '전남': 'N', '경북': 'O', '경남': 'P',
    '제주': 'Q'
}
```

因此，我们采用 Focal CTC Loss 来尽可能的避免字符分布不均存在的问题。

这里补一个focal的介绍，以及focal ctc loss的公式

$$L_{Focal_CTC} = \alpha * (1 - y')^\gamma * \log(y')$$

结合公式和韩国现存车牌的字符分布情况，alpha 取 1.0，gamma 取 5.0。

考虑到 LPR 任务的实时性要求，我们选择了 mobilenet v3 small 作为该网络的 backbone，并且针对 coral 硬件的特性，进行了针对性的优化。

1. 去除了原版 mobilenet v3 small 中包含的所有 SE-Net 模块。SE-Net 虽然可以小幅提高模型的精度，但是SE-Net 将会增加近 40% 的延迟。

Architectural Blocks	Latency (ms)
Baseline	1.53
+ Squeeze-Excite	2.10
+ Skip-Connections	2.62

2. 由于coral 对于 hard-swish 激活函数的支持较差，考虑到后续量化的算子支持。这里我们选择使用 relu6 替代原版 mobilenet v3 small 中的 hard-swish 激活函数。

$$f(x) = x \cdot \text{sigmoid}(\beta x) \quad \leftarrow \quad \frac{\text{ReLU6}(x+3)}{6}$$

$$\downarrow$$

$$\text{h-swish}[x] = x \frac{\text{ReLU6}(x+3)}{6}$$

3. 由于深度表征对于 LPR 这个任务的贡献不大，且带来了较大的推理延迟。所以去除了原版 mobilenet v3 small 的最后两个 bneck。

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

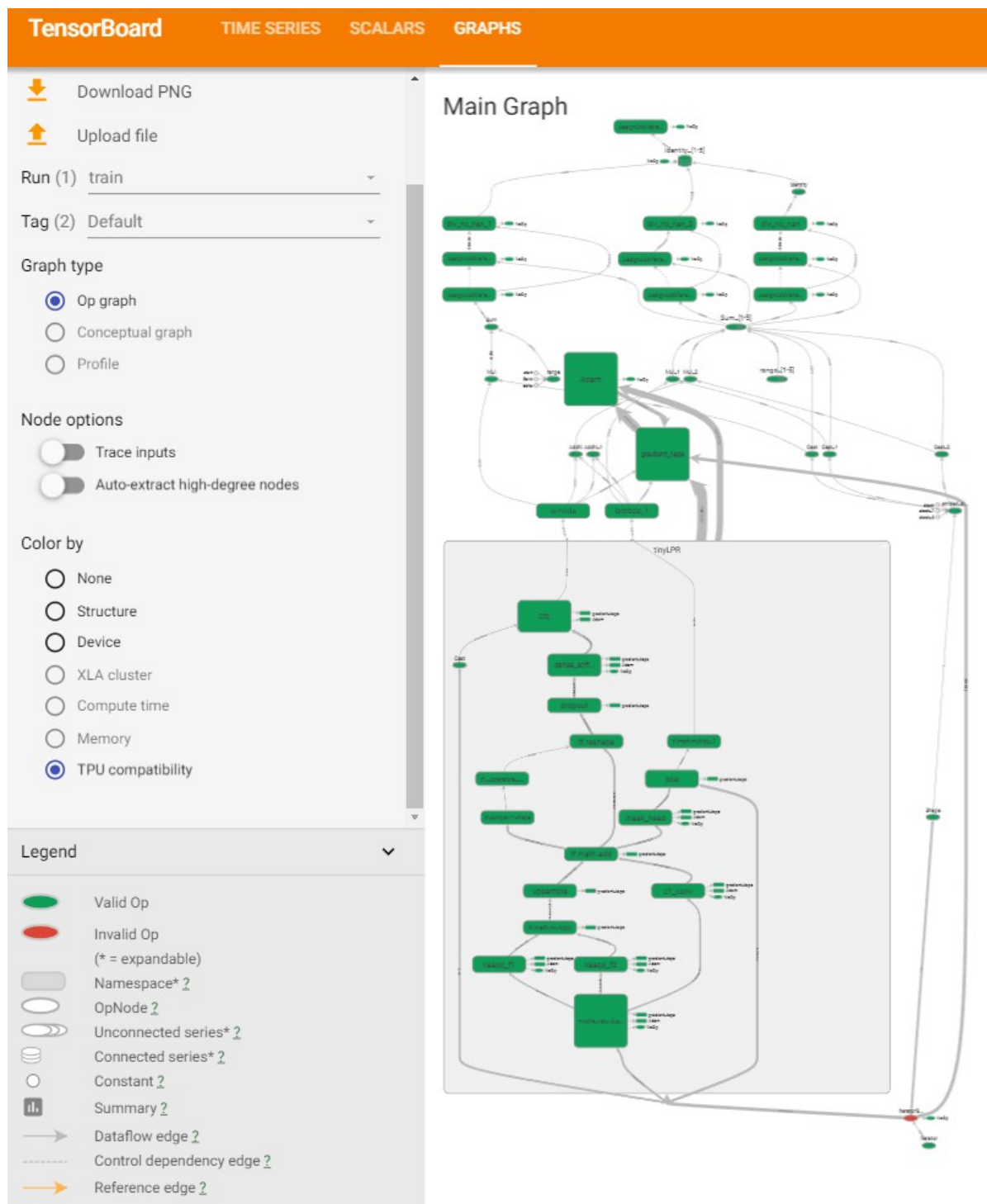
Table 2. Specification for MobileNetV3-Small. See table 1 for notation.

对比上次作业模型，我们的模型处理更大输入分辨率的情况下，在参数和运算量以及推理延迟上都有着显著的降低。

Model	Input Size	Params(M)	FLOPs(M)
LPRNet(our pre version)	94x24	1.93	332.7
tinyLPR	128x64	0.20	47.9

针对硬件优化

针对 EdgeTPU 支持的算子优化神经网络的设计，如下图所示，目标平台（TPU）100%支持该网络的所有算子。图中绿色表示支持的算子，红色为不支持的算子。右下方的红点表示的是 Focal CTC Loss 损失函数，该部分仅在训练时参与运算，部署推理时不参与运算。



相比较之前版本的，经过 uint8 量化后的模型在下述平台上的运行速度如下：

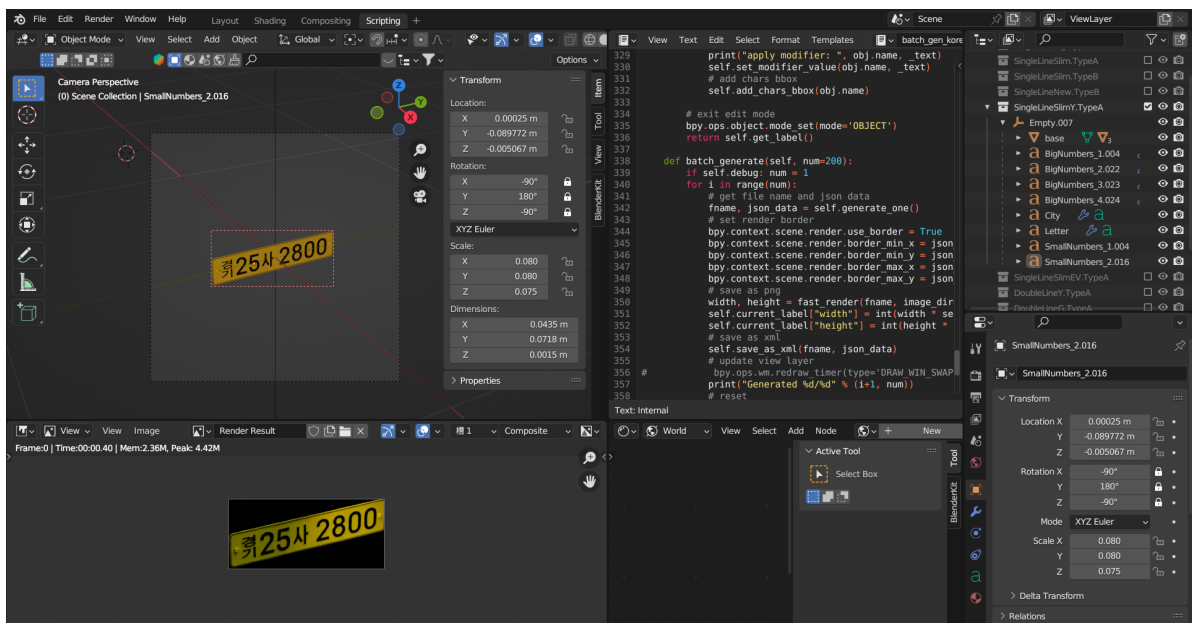
Platform	Description	Inference latency
CPU	rk3399 Cortex-A72@1.8GHz	32 ms
TPU	Coral USB Accelerator	3 ms

合成数据

由于韩国现存车牌种类较多，且年度跨度较大。实际收集到的车牌数据存在明显的地域性，且分布不均。

<div>• 서울 2 •</div> <div>가 1973</div>	1973년 – 1995년 초록색 바탕에 흰색글씨 위에 등록 지역명+숫자 1자리 아래 한글+숫자 4자리
<div>• 부산 33 •</div> <div>나 1996</div>	1996년 – 2003년 초록색 바탕에 흰색글씨 위에 등록 지역명+숫자 2자리 아래 한글+숫자 4자리
<div>• 44 다 •</div> <div>2004</div>	2004년 – 2006년 초록색 바탕에 흰색글씨 지역명이 없어지고 숫자 2자리와 한글 숫자 4자리로 구성됨
<div>• 55라 2006</div> <div>• 55라 2006 •</div>	2006년 – 2019년 8월 흰색 바탕에 검정색글씨 2열로 구성되었던 기존 번호판에서 1열구성으로 변경되었음. 가로가 긴 유럽식 번호판 시작
<div>• 666마 2019 •</div>	2019년 9월 이후 번호판 부족으로 앞자리를 3자리로 변경 번호판 인식을 위해 폰트 변경

为了解决这一问题，我们通过 Blender 这一软件进行车牌建模并生成大量符合实际情况的车牌数据。并且考虑了夜间可能存在的车牌架光照影响，针对性的进行了模拟并最终生成了大量的合成车牌的数据集。



经过实验，我们发现合成数据集对模型的精度提升有着显著的贡献。

// F

特殊字符的识别精度提升最为显著，如下表所示：

Dataset	Count	Accuracy
Origin		85.36
Syn		89.63
Real+Syn		91.25

实验结果

这里补一个可视化的结果

Dataset	Count	Accuracy
Real		94.09
Syn		96.75
Real+Syn		97.38

由于区域限制，我们无法获得韩国各地的车牌数据。因此，我们将实际收集到的车牌数据和合成数据进行了混合，最终得到了包含 210,000 张车牌图片的数据集（以下简称210K），其中 90% 用于训练，10% 用于测试，并且我们将真实数据中选出的 1,000 张图片作为验证集（以下简称 KLP）。

Dataset	Size	Single	Multi	Total	Final*
210K	210,000	91.25	96.75	97.38	97.38
KLP	1,000	85.36	94.09	94.09	94.09

- Final 表示经过正则表达式过滤掉不符合规范的车牌后的数据集。

附录

Final Accuracy 所用到的正则表达式：

```
^([가-힣]{2}\d{2}|\d{2,3})[가-힣]{1}\d{4}$
```