License Plate Recognition

CHOI JUNHO 22172113

ANDRE IVAN 22172304

YUN JUNGUN 22181250

인하대학교 영상 및 시각컴퓨팅 연구실
Image and Vision Computing Laboratory

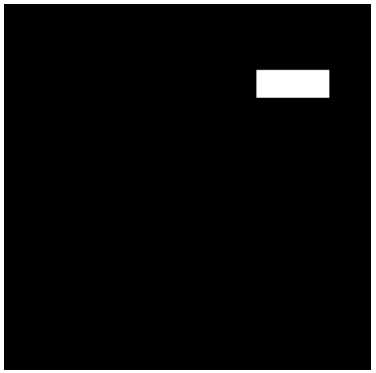
Outline

1. Framework
   - Dataset preparation
   - License plate detection networks
   - License plate recognition network
2. Evaluation Result

Framework

- **Data preparation (Plate detection)**
  - Binary class label for segmentation
  - [X, Y, W, H] plate position
  - Plate characters
  - Manually remove missing or wrongly annotated data

< annotated data >

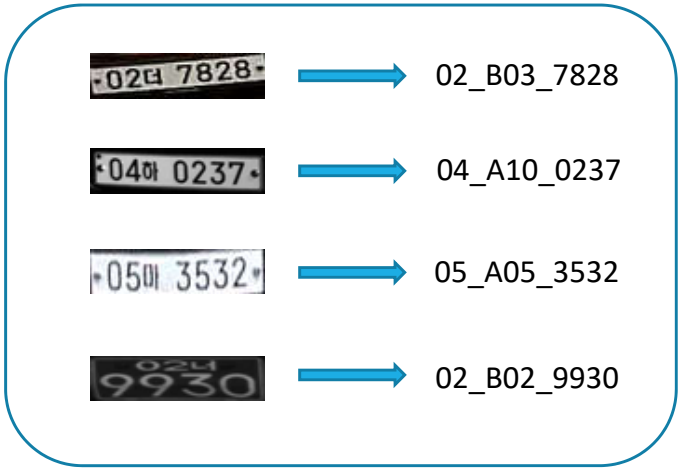


< label data for training >

Framework

- **Data preparation (Plate recognition)**
  - Encode Korean characters
  - Create dictionary

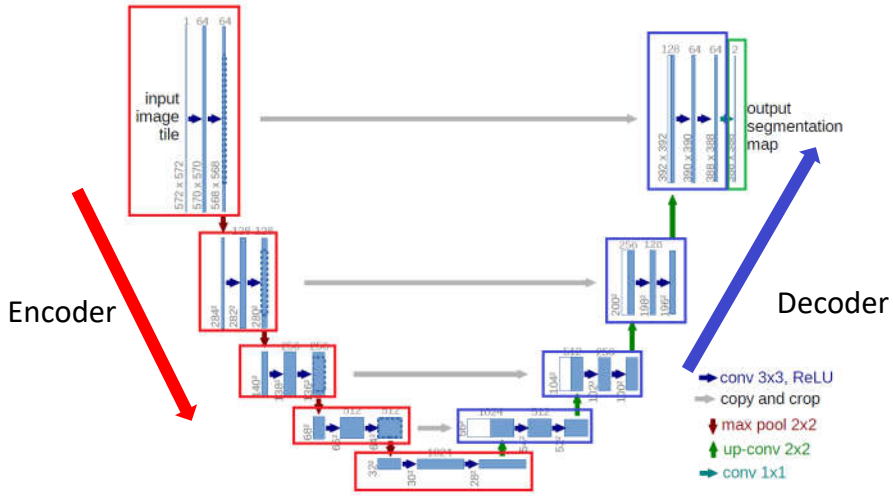
```
#Possible characters in Korean plate
CHARS = ['가', '나', '다', '라', '마','아', '바', '사', '자', '하',
         '거', '너', '더', '러', '머', '버', '서', '어', '저', '허',
         '고', '노', '도', '로', '모', '보', '소', '오', '조', '호',
         '구', '누', '두', '루', '무', '부', '수', '우', '주',
         '배',
         '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '_'
        ]

#Character dictionary
dict = {'A01':'가', 'A02':'나', 'A03':'다', 'A04':'라', 'A05':'마',
        'A06':'바', 'A07':'사', 'A08':'아', 'A09':'자', 'A10':'하',
        'B01':'거', 'B02':'너', 'B03':'더', 'B04':'러', 'B05':'머',
        'B06':'버', 'B07':'서', 'B08':'어', 'B09':'저', 'B10':'허',
        'C01':'고', 'C02':'노', 'C03':'도', 'C04':'로', 'C05':'모',
        'C06':'보', 'C07':'소', 'C08':'오', 'C09':'조', 'C10':'호',
        'D01':'구', 'D02':'누', 'D03':'두', 'D04':'루', 'D05':'무',
        'D06':'부', 'D07':'수', 'D08':'우', 'D09':'주', 'D10':'배',
       }
```



Framework

- **License plate detection (Parking)**
  - Dark images
  - Plate size are similar
  - Unbalanced plate type distribution

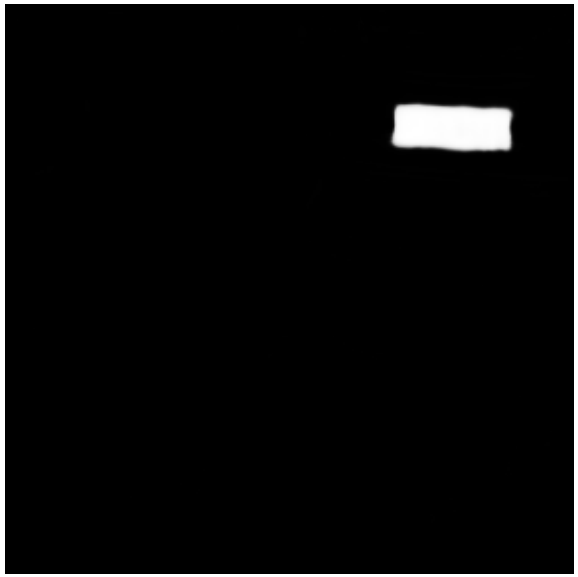
- **Network**
  - Semantic segmentation approach using U-Net[(1)]



Framework

- **License plate detection (Parking)**

< normalized input >



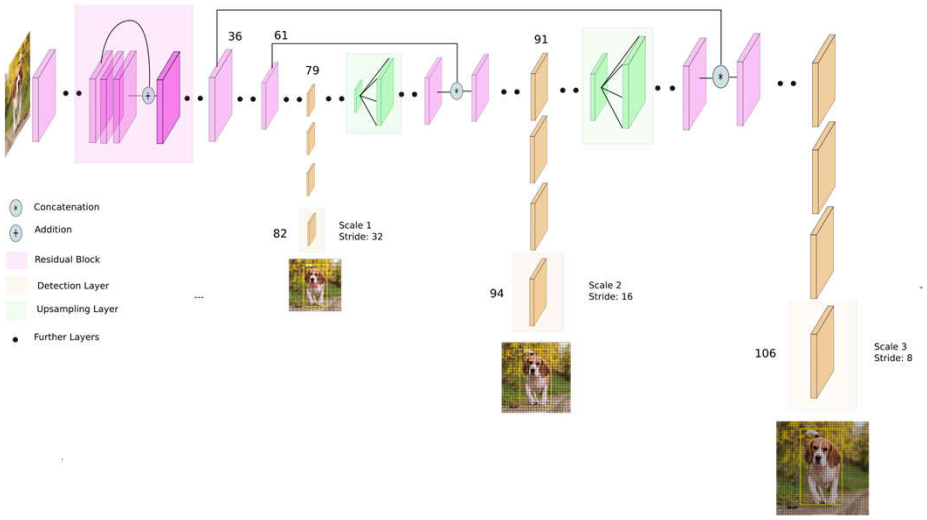
< segmented output >

< detection result >

Framework

- **License plate detection (CCTV)**
  - Variative license plate size
  - Using U-Net alone leads to poor result
  - Adopt **coarse-to-fine** approach (car detection → plate detection)

- **Network**
  - Car detection using YOLO v3 [(2)]
    - Pre-trained model (COCO dataset)
  - Semantic segmentation approach using U-Net



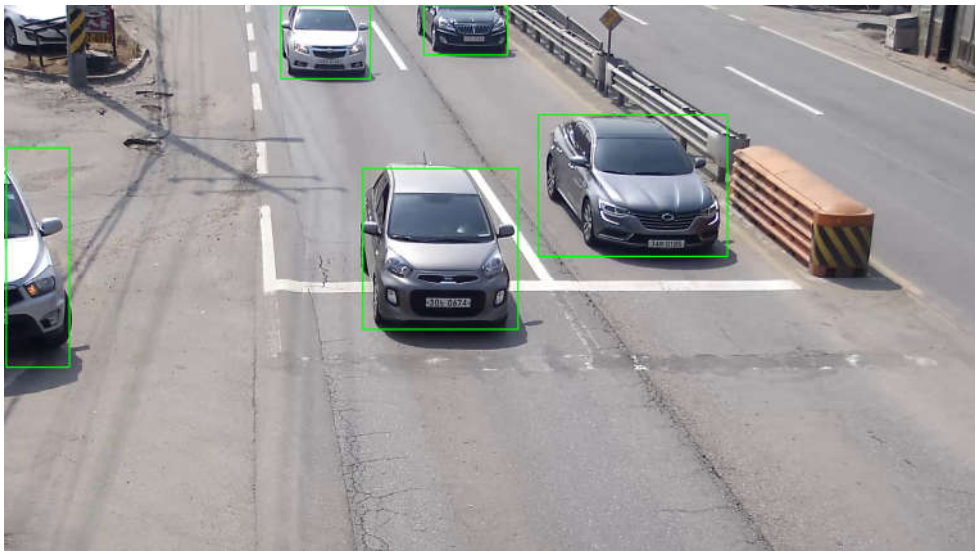
(2) Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement." Technical report, 2018.

Framework

- **License plate detection (CCTV)**

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

< Feature extraction network of YOLO v3 >

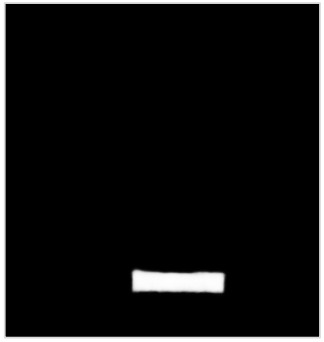


< Car detection result >

Framework

- **License plate detection (CCTV)**
  - Choose 3 largest detected cars
  - Choose 1 largest detected plate

Plate Detection



Final result

Framework

PyTorch

- **License plate detection (Parking)**
  - Training details

Items	Value
Input size	320×320×1
Loss function	Binary cross entropy + weight decay (gradient clipping)
Learning rate	0.01 to 0.0001 (90% decay rate at 20 and 30 epoch)
Optimizer	Stochastic Gradient Descent (SGD)
Batch size	8
Epoch	40

Framework

- **License plate recognition**
  - Various character length
  - Different position of characters
  - Low resolution plate
  - Ambiguous label

- **Network**
  - Automatic character recognition LPR-Net[3]

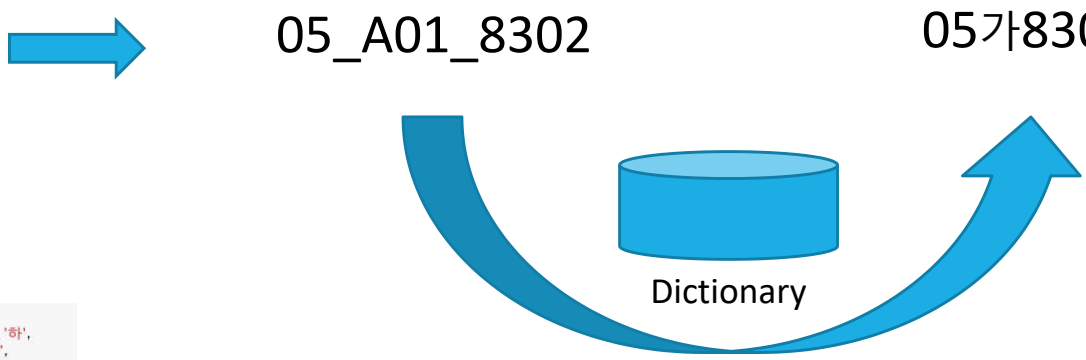
Layer Type	Parameters
Input	94x24 pixels RGB image
Convolution	#64 3x3 stride 1
MaxPooling	#64 3x3 stride 1
Small basic block	#128 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Small basic block	#256 3x3 stride 1
Small basic block	#256 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Dropout	0.5 ratio
Convolution	#256 4x1 stride 1
Dropout	0.5 ratio
Convolution	# class_number 1x13 stride 1

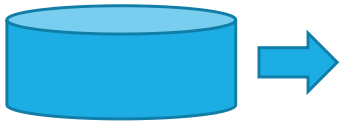
Target platform	1 LP processing time
GPU + cuDNN	3 ms
CPU (using Caffe [22])	11-15 ms
CPU + FPGA (using DLA [23])	4 ms[1]
CPU (using IE from Intel OpenVINO [24])	1.3 ms

(3) Wang, D., Tian, Y., Geng, W., Zhao, L., & Gong, C. LPR-Net: Recognizing Chinese license plate in complex environments. Pattern Recognition Letters (2018).

Framework

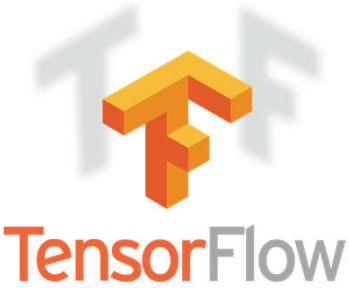
- **License plate recognition**





```
#Possible characters in Korean plate
CHARS = ['가', '나', '다', '라', '마','아', '바', '사', '자', '하',
         '거', '너', '더', '러', '머', '버', '서', '어', '저', '허',
         '고', '노', '도', '로', '모', '보', '소', '오', '조', '호',
         '구', '누', '두', '루', '무', '부', '수', '우', '주',
         '배',
         '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '_'
        ]

#Character dictionary
dict = {'A01':'가', 'A02':'나', 'A03':'다', 'A04':'라', 'A05':'마',
        'A06':'바', 'A07':'사', 'A08':'아', 'A09':'자', 'A10':'하',
        'B01':'거', 'B02':'너', 'B03':'더', 'B04':'러', 'B05':'머',
        'B06':'버', 'B07':'서', 'B08':'어', 'B09':'저', 'B10':'허',
        'C01':'고', 'C02':'노', 'C03':'도', 'C04':'로', 'C05':'모',
        'C06':'보', 'C07':'소', 'C08':'오', 'C09':'조', 'C10':'호',
        'D01':'구', 'D02':'누', 'D03':'두', 'D04':'루', 'D05':'무',
        'D06':'부', 'D07':'수', 'D08':'우', 'D09':'주', 'D10':'배',
       }
```

Framework



- **License plate detection (Parking)**
  - Training details

Items	Value
Input size	94×24×1
Loss function	Connectionist Temporal Classification (CTC) [4] loss
Learning rate	0.001 (90% decay rate every 2000 iteration)
Optimizer	Adam
Batch size	16
Epoch	600

(4) Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. ICML 2006.

Evaluation Result

- **Parking dataset**

Test 1

Item	Value
num_bbox_examples	285
num_bbox_corrects	270
bbox_accuracy	94.74
num_rec_examples	285
num_rec_corrects	183
rec_accuracy	64.21
avg_pt	71.84
score	**161.76**

Test 2

Item	Value
num_bbox_examples	285
num_bbox_corrects	270
bbox_accuracy	94.74
num_rec_examples	285
num_rec_corrects	183
rec_accuracy	64.21
avg_pt	109.6
score	157.99

Evaluation Result

- **CCTV dataset**

Test 1

Item	Value
num_bbox_examples	451
num_bbox_corrects	399
bbox_accuracy	88.47
num_rec_examples	436
num_rec_corrects	269
rec_accuracy	61.7
avg_pt	189.91
score	141.18

Test 2

Item	Value
num_bbox_examples	451
num_bbox_corrects	399
bbox_accuracy	88.47
num_rec_examples	436
num_rec_corrects	269
rec_accuracy	61.7
avg_pt	116.95
score	**148.47**

Conclusion

- **Segmentation approach to detect license plate**
- **Coarse-to-fine approach for small plates (CCTV)**
- **Automatic and robust character recognition with small and fast network**
- **Total score: 310.23**