Optimized Neural Network ALPR System for Microprocessors

Hamza Khan (UID 205223240), Omead Pooladzandi (UID 105035701),
Zach Harris (UID 105221897)

Outline

1. Background
2. LP Localization
3. LP Segmentation
4. Data Augmentation
5. End-to-end system
6. Lossless network compression
7. Future work
    a. Lossy compression
    b. Low rank approximation

Background: Problem description

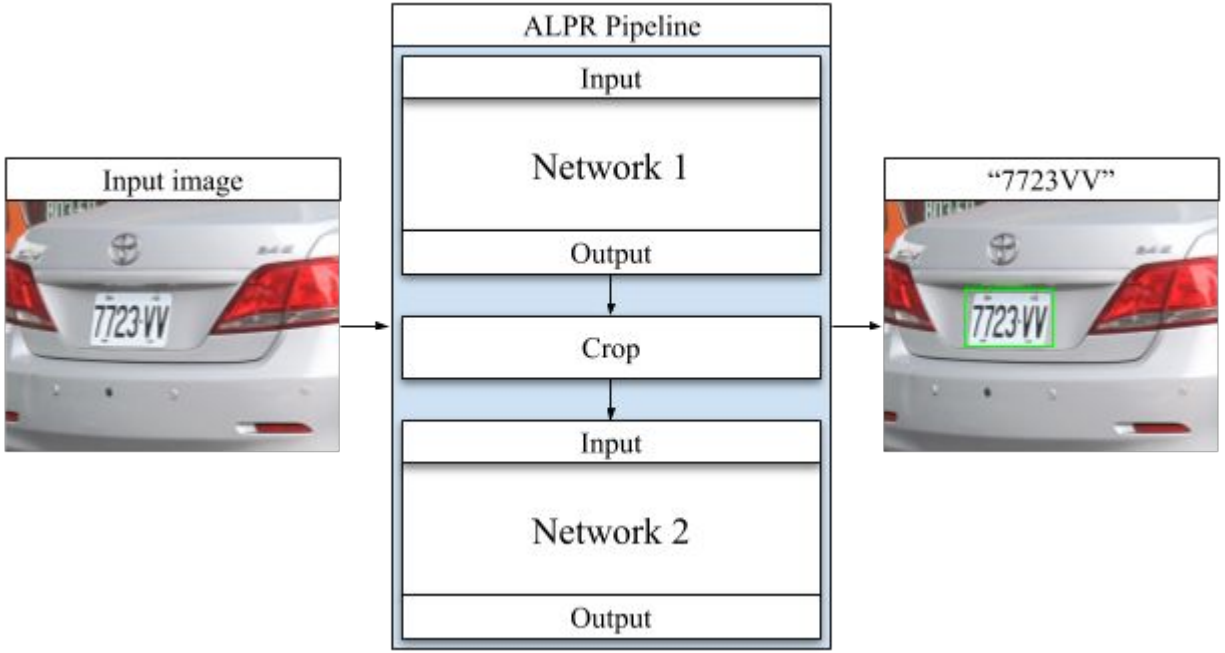
The performance of Automated License Plate Recognition (ALPR) can be greatly improved by the implementation of neural networks.

Neural networks are designed to perform large amounts of computations and store large amounts of memory in order to improve ALPR.

How can these networks be reduced, in order for low-power microprocessors to use them at high speeds?

Background: ALPR Pipeline

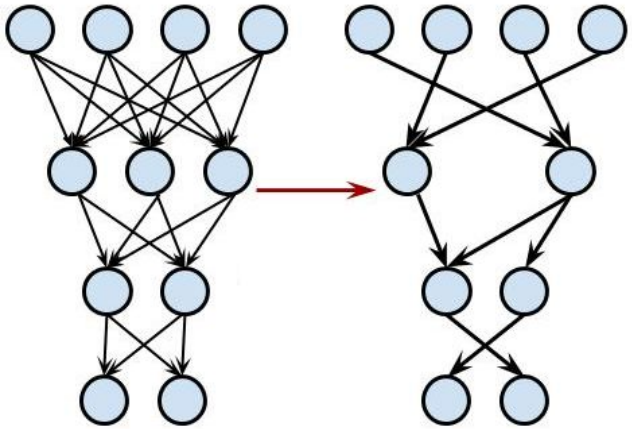
1. Detect the license plate bounding box, using the first dataset that was provided.
2. Convert the bounding boxes to fixed-size images through cropping and transformations
3. Detect each letter in the cropped license plate
4. Put together each detected letter for a final license plate as string



Background: Pruning methods

Some of the parameters in trained neural networks are often either redundant, or don’t contribute much to the output. Removing these parameters can reduce both model size and complexity.

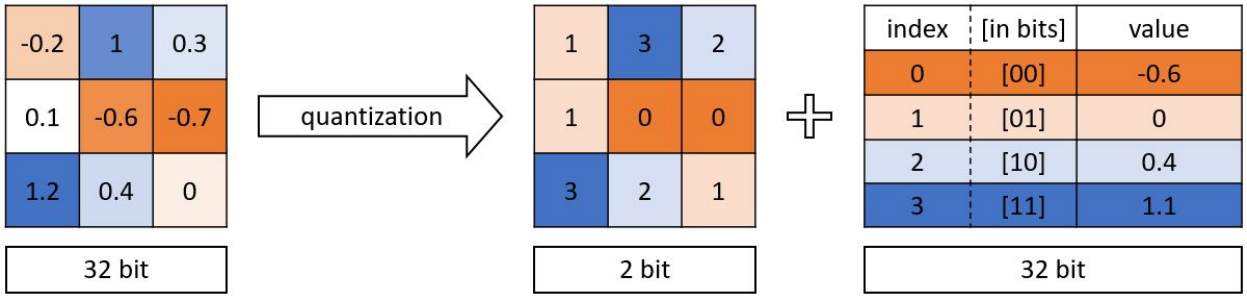
To this end, we implemented iterative weight pruning on both our networks. We repeatedly pruned a few percent, then trained the remaining parameters.



Background: Quantization methods

Typical TensorFlow weights are stored in 32-bit floating point variables. To reduce computational intensity and model size, we implemented quantization using a codebook.

We quantized layer weights to 256 values by splitting our total range of the layer's values into 255 evenly-spaced regions, and including 0 as a value as well. This way, each 32-bit number can be represented with only 8 bits (4x compression).



LP Localization: Network performance survey [2]

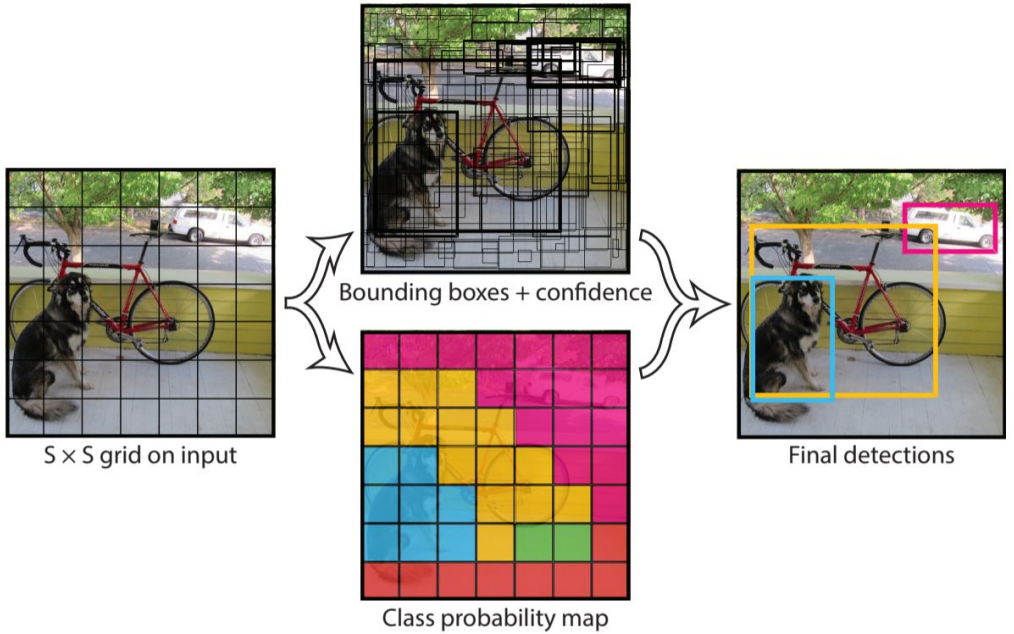
Architecture	**mAP**	**FPS**
R-CNN	66.0	0.05
Fast R-CNN	70.0	0.5
Faster R-CNN with VggNet	73.2	7
Faster R-CNN with ResNet	76.4	5
Mask R-CNN	73.2	5
YOLO	63.4	45
YOLOv2 288 × 288	69.0	91
YOLOv2 352 × 352	73.7	81
YOLOv2 416 × 416	76.8	67
YOLOv2 480 × 480	**77.8**	**59**

LP Localization: YOLOv2 network [1]

Single neural network

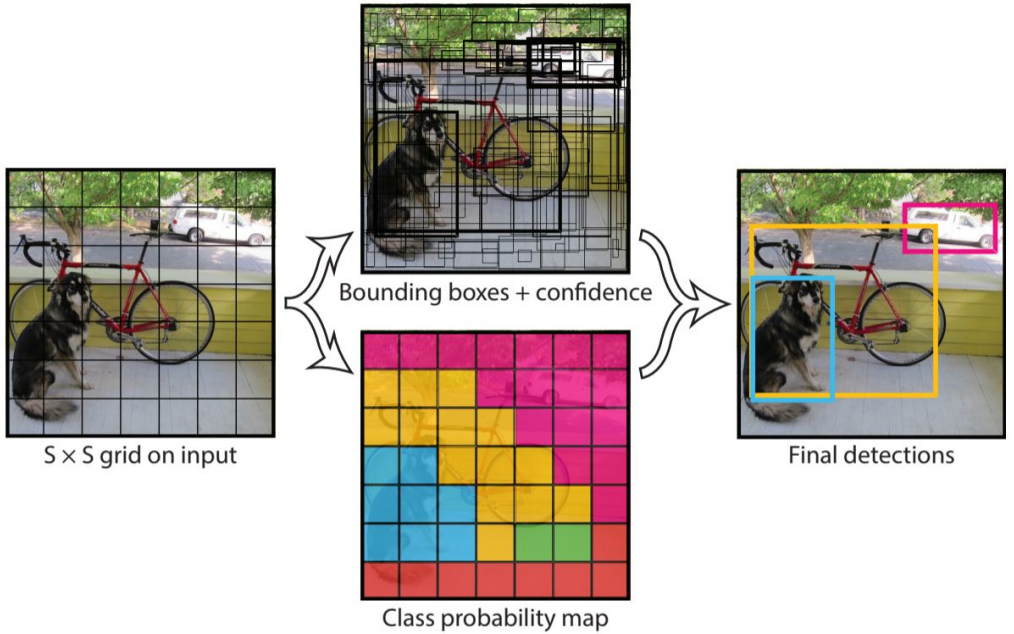
Predicts bounding boxes and class probabilities directly from full images in one evaluation

Learns very general representations of objects



LP Localization: YOLOv2 network [1]

The input image is divided into an S × S grid and each grid cell predicts B bounding boxes, confidence for those boxes and C class probabilities



LP Localization: YOLOv2 architecture [5]

The MobileNet framework is used as the base image classifier for the YOLO algorithm

Depthwise separable convolutions keep the architecture light-weight

Pretrained on ImageNet 2016 dataset, fine-tuned using the LP dataset

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

LP Localization: Accuracy measures

Intersection over Union (IoU):

- Measures the amount of overlap between the truth and predicted bounding boxes

Mean Average Precision (mAP):

- The average of the maximum precisions at different IoU thresholds

Accuracy:

- For testing, images not overlapping any of the truth bounding boxes (IoU == 0.0) are thrown out and classified as an error.
- The accuracy of the overlapping boxes will be determined by the CRNN.

LP Localization: Training Results

	YOLOv2 Original	YOLOv2 Pruned	YOLOv2 8-bit Quantized
mAP	0.9930	0.9848	0.9897
Train accuracy	96.55% (1792/1856)	97.31% (1806/1856)	98.01% (1819/1856)
Test accuracy	99.35% (306/308)	98.70% (304/308)	99.35% (306/308)
Model SIze	100%	12.87%	3.3%

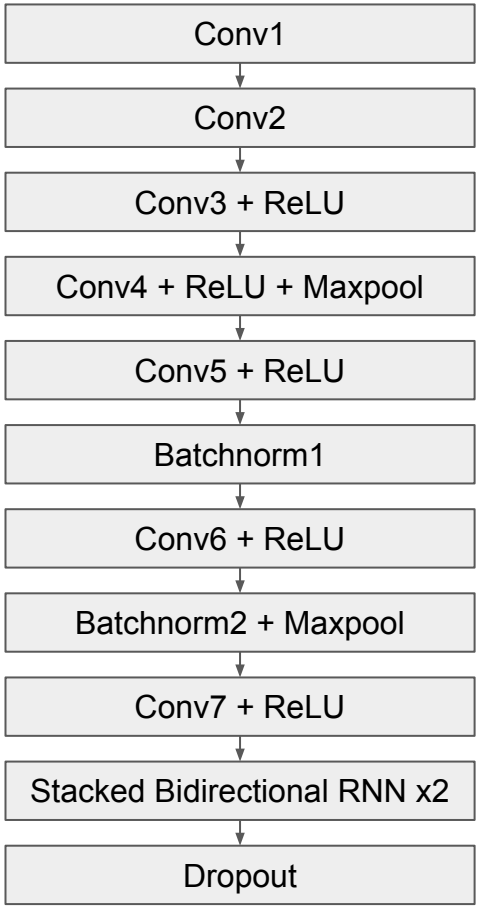
Minimal loss in mAP. This loss will easily be made up by the spatial invariance of the CRNN

After further training, the pruned and quantized models achieve a higher test and train accuracy than original

LP Segmentation: CRNN

To go from the cropped license plate to letters, we used a combination of a convolutional and recurrent neural network.

When pruning this network, we ended up pruning only kernels/weights for best results.



LP Segmentation: Results

The accuracy of the CRNN was evaluated after training, as well as with pruning and quantization with two different pruning amounts:

	Model Size (relative)	**0 errors**	**1 or fewer**	**2 or fewer**	**3 or fewer**
Original	100%	90.79%	98.09%	99.37%	100%
Pruning (68.74%) and 8-bit Quantization	17.18%	91.53%	98.79%	100%	100%
Pruning (51.55%) and 8-bit Quantization	12.88%	59.80%	82.35%	90.20%	95.09%

Dataset Augmentation

- The given Train set consisted of 1748 images of license plates.
- The augmented Train set consists of 31751 images.
- Train set was probabilistically augmented via transformations:
  - Rotation
  - Shearing
  - Skew Corner
  - Skew Left Right
  - Gaussian Distortion
  - Random Distortion

Original vs Aug

End-to-end system: Results

A combination of original, pruned, and quantized models of both networks are tested in the pipeline.

The number of inaccurately bounded license plates from YOLO is combined with the accuracy measures of CRNN to produce the following table.

	CRNN Original	CRNN Quantize, Light Prune	CRNN Quantize, Heavy Prune
YOLOv2 Original	< 4 mistakes 99.35% < 3 mistakes 99.03% < 2 mistakes 97.43% **No mistakes 88.13%**	< 4 mistakes 99.35% < 3 mistakes 98.58% < 2 mistakes 95.88% No mistakes 86.99%	< 4 mistakes 97.46% < 3 mistakes 94.63% < 2 mistakes 88.97% No mistakes 53.12%
YOLOv2 Pruned	< 4 mistakes 98.70% < 3 mistakes 97.72% < 2 mistakes 96.42% No mistakes 84.69%	< 4 mistakes 98.70% < 3 mistakes 97.55% < 2 mistakes 95.23% No mistakes 82.55%	< 4 mistakes 97.67% < 3 mistakes 93.55% < 2 mistakes 81.17% No mistakes 57.46%
YOLOv2 Quantized	< 4 mistakes 99.03% < 3 mistakes 98.38% < 2 mistakes 95.79% **No mistakes 86.41%**	< 4 mistakes 98.67% < 3 mistakes 98.59% < 2 mistakes 96.70% **No mistakes 87.99%**	< 4 mistakes 98.37% < 3 mistakes 97.39% < 2 mistakes 86.60% No mistakes 60.13%

Results without YOLOv2 errors

Lossless network compression

- Implemented two lossless compression algorithms:
  - Huffman coding
    - Generates dictionary based on most frequently used symbols
    - Optimal prefix code - achieves lowest possible expected codeword length
  - Lempel Ziv Welsh
    - Generates a dictionary of substrings which are recursively referenced to exploit patterns in data
    - Lempel Ziv gives asymptotically optimal compression ratios for sequences

Compression performance for YOLOv2

Encoding combinations	**Compression ratio**
Huffman	4.6559x
Lempel Ziv	4.8015x
Lempel Ziv then Huffman	3.0485x
Huffman then Lempel Ziv	**5.3210x**
Huffman then Huffman	4.8356x
Lempel Ziv then Lempel Ziv	3.1657x

By encoding the quantized network with Huffman, a codebook and encoded representation are generated. By encoding this encoded representation with Lempel Ziv, the maximum compression ratio is achieved.

Compression performance for CRNN

Encoding combinations	Compression ratio of lightly pruned	Compression ratio of heavily pruned
Huffman	1.4295x	1.7576x
Lempel Ziv	1.0827x	1.3796x
Lempel Ziv then Huffman	0.9868x	1.1123x
Huffman then Lempel Ziv	**1.6226x**	**2.0259x**
Huffman then Huffman	1.4584x	1.8002x
Lempel Ziv then Lempel Ziv	1.0385x	1.1686x

This same combination of encoding provides the highest compression ratio for the CRNN network.

Final Results

The final output size of each network (after pruning, quantization, and lossless network compression) is shown in the table below.

Network	Pruning Size	Quantization Size	Compression	Final Size
YOLOv2	12.87%	25%	5.32x	0.6%
CRNN	51.55%	25%	2.03x	6.3%

Future work

- Lossy Compression Algorithms
  - Just as we lose information in quantization and pruning, lossy compression algorithms trade information for space.
  - Run lossy compression and test those weights in a network to check the loss in accuracy
- Low Rank Approximation of channels.
  - In linear algebra we know that we can have a rank 1 approximation of a matrix that is near to the original matrix. It would be good to see if we can use this to retain the purpose of the network channel while having the benefit of being able to decompose the channel into two vectors, hence reducing the weights needed in storage drastically. .

Bibliography

[1] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” CoRR, vol. abs/1506.02640, 2015.

[2] Hogne, Jørgensen, Automatic License Plate Recognition using Deep Learning Techniques. Norwegian University of Science and Technology, Department of Computer Science. Web access Feb. 10, 2019.

[3] Han, S. (2016, November 10). Compressing and regularizing deep neural networks. Retrieved from https://www.oreilly.com/ideas/compressing-and-regularizing-deep-neural-networks

[4] Marianne Stecklina, M. S. (2018, March 8). Step by Step to a Quantized Network. Retrieved March 24, 2019, from https://medium.com/@marianne.stecklina/step-by-step-to-a-quantized-network-5d7da6c52af1

[5] Howard, Andrew G.et. al, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. Web access Mar. 24, 2019

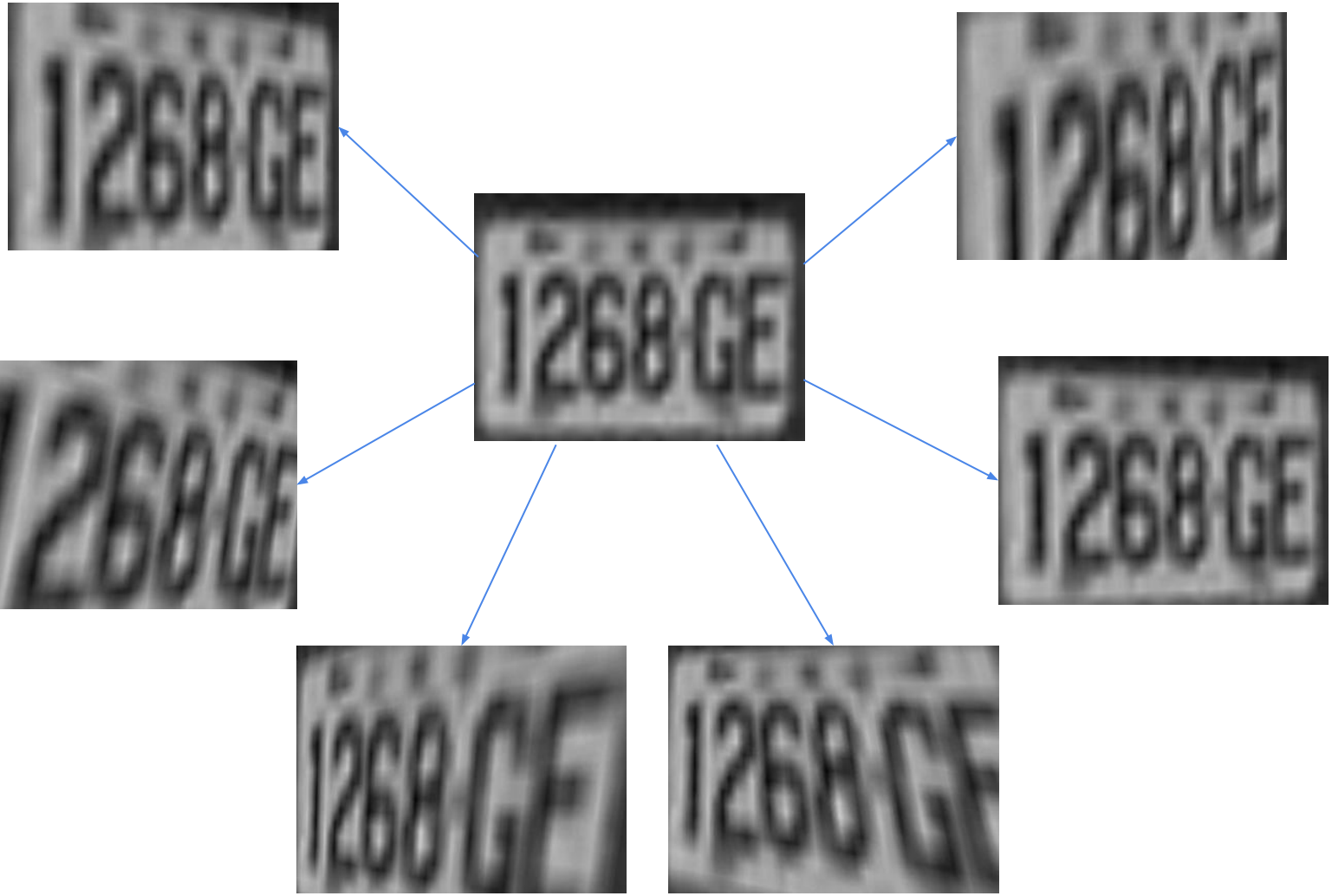
	CRNN Original	CRNN Quantize, Light Prune	CRNN Quantize, Heavy Prune
YOLOv2 Original	< 4 mistakes 100% < 3 mistakes 99.68% < 2 mistakes 98.08% No mistakes 88.78%	< 4 mistakes 100% < 3 mistakes 99.23% < 2 mistakes 96.53% No mistakes 87.64%	< 4 mistakes 98.11% < 3 mistakes 95.28% < 2 mistakes 89.62% No mistakes 53.77%
YOLOv2 Pruned	< 4 mistakes 100% < 3 mistakes 99.02% < 2 mistakes 97.72% No mistakes 85.99%	< 4 mistakes 100% < 3 mistakes 98.85% < 2 mistakes 96.53% No mistakes 83.85%	< 4 mistakes 98.97% < 3 mistakes 94.85% < 2 mistakes 82.47% No mistakes 58.76%
YOLOv2 Quantized	< 4 mistakes 99.68% < 3 mistakes 99.03% < 2 mistakes 96.44% No mistakes 87.06%	< 4 mistakes 99.62% < 3 mistakes 99.24% < 2 mistakes 97.35% No mistakes 88.64%	< 4 mistakes 99.02% < 3 mistakes 98.04% < 2 mistakes 87.25% No mistakes 60.78%

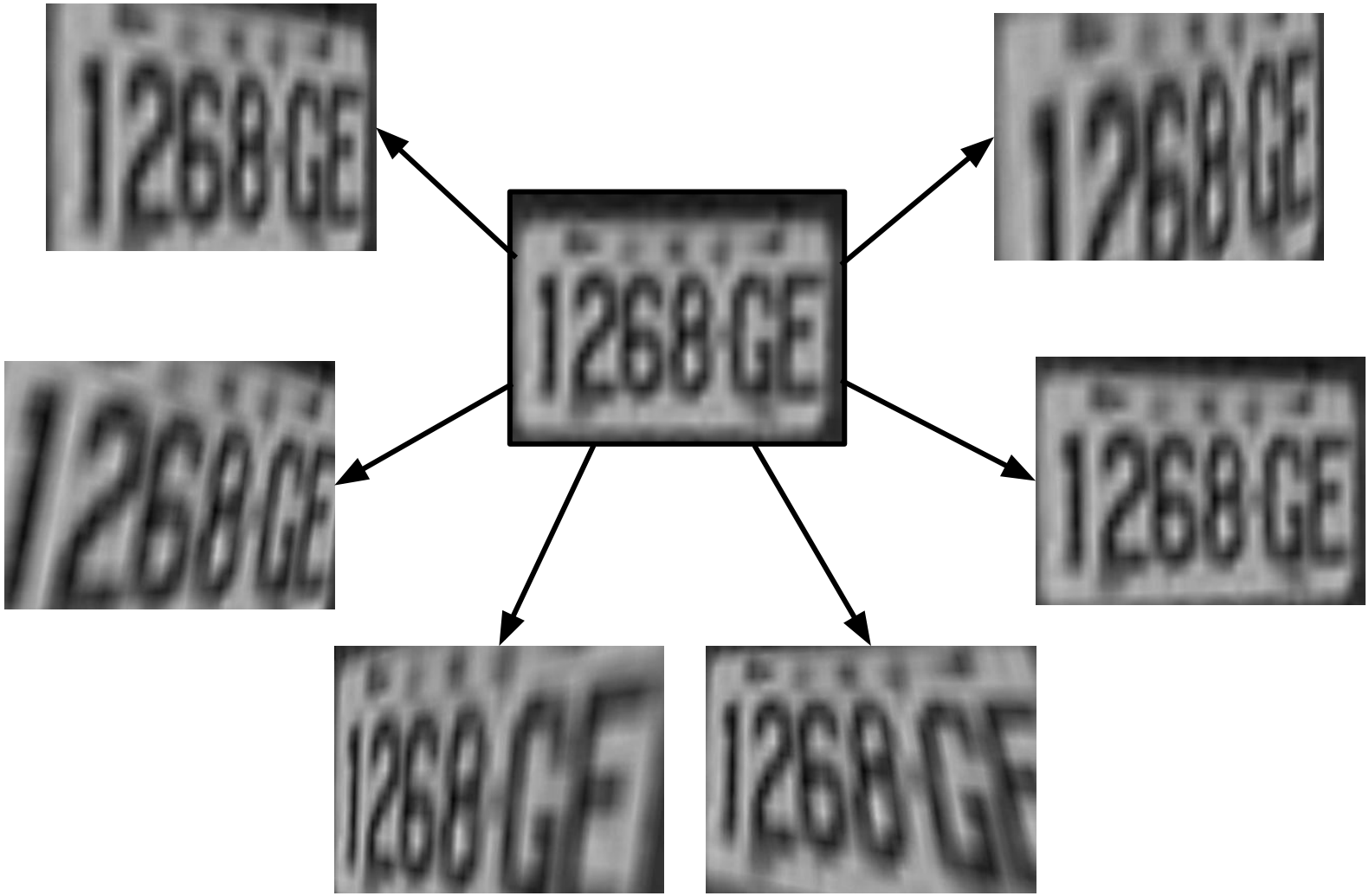
Results without YOLOv2 errors

	CRNN Original			CRNN Quantize, Light Prune			CRNN Quantize, Heavy Prune		
YOLOv2 Original	< 4 mistakes	99.35%	100%	< 4 mistakes	99.35%	100%	< 4 mistakes	97.46%	98.11%
	< 3 mistakes	99.03%	99.68%	< 3 mistakes	98.58%	99.23%	< 3 mistakes	94.63%	95.28%
	< 2 mistakes	97.43%	98.08%	< 2 mistakes	95.88%	96.53%	< 2 mistakes	88.97%	89.62%
	No mistakes	88.13%	88.78%	No mistakes	86.99%	87.64%	No mistakes	53.12%	53.77%
YOLOv2 Pruned	< 4 mistakes	98.70%	100%	< 4 mistakes	98.70%	100%	< 4 mistakes	97.67%	98.97%
	< 3 mistakes	97.72%	99.02%	< 3 mistakes	97.55%	98.85%	< 3 mistakes	93.55%	94.85%
	< 2 mistakes	96.42%	97.72%	< 2 mistakes	95.23%	96.53%	< 2 mistakes	81.17%	82.47%
	No mistakes	84.69%	85.99%	No mistakes	82.55%	83.85%	No mistakes	57.46%	58.76%
YOLOv2 Quantized	< 4 mistakes	99.03%	99.68%	< 4 mistakes	98.67%	99.62%	< 4 mistakes	98.37%	99.02%
	< 3 mistakes	98.38%	99.03%	< 3 mistakes	98.59%	99.24%	< 3 mistakes	97.39%	98.04%
	< 2 mistakes	95.79%	96.44%	< 2 mistakes	96.70%	97.35%	< 2 mistakes	86.60%	87.25%
	No mistakes	86.41%	87.06%	No mistakes	87.99%	88.64%	No mistakes	60.13%	60.78%

Results with YOLOv2 errors


1268 GE


1268 GE
1268 GE
1268 GE
1268 GE
1268 GE
1268 GF
1268 GF


1268 GE
1268 GE
1268 GE
1268 GE
1268 GE
1268 GF
1268 GF