Real time detection of traffic signs on mobile device

Nicolas Six

Georgia Institute of Technology
School of Computer Science

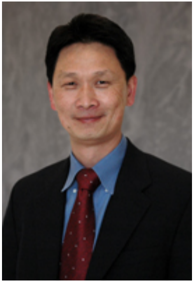
December 3^{rd}, 2019

Acknowledgments

Traffic sign detection on smartphone

Nicolas SIX

Dr. Zsolt KIRA



Professor Yi-Chang James TSAI

Dr. Zhaohua WANG

Overview

Traffic sign detection

Figure: Example of detection, from Yolo v3, classification with MobileNet.

Current procedure

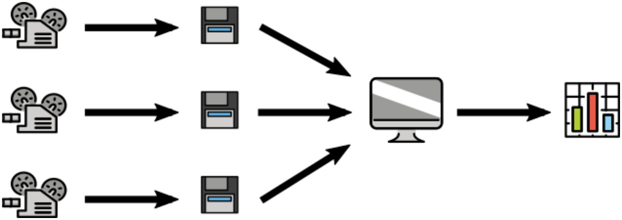


Figure: Current procedure for automatic inventory of traffic signs

Proposed procedure

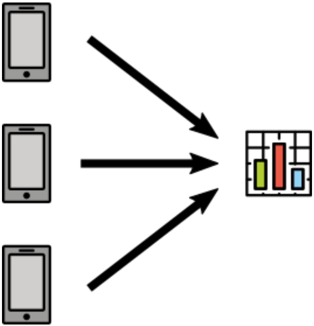


Figure: Target procedure for automatic inventory of traffic signs

On device processing

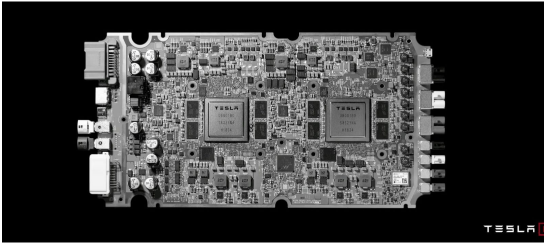
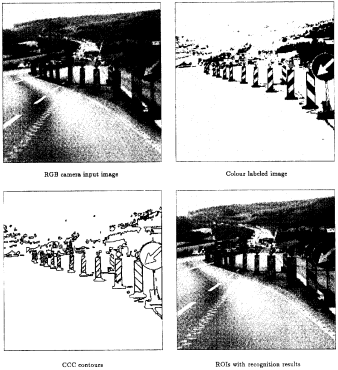
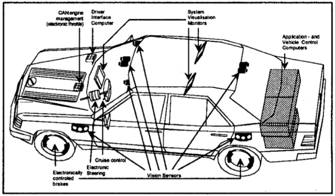


Figure: Tesla new chip for autonomous driving, consuming 72W and providing 144 TOPS

Historical traffic sign detection



(a) Hybrid processing to traffic sign detection proposed by Ruud Janssen, W. Ritter in 1993 [1]



(b) Vita-II autonomous vehicle in 1994 [7]

Generic object detection

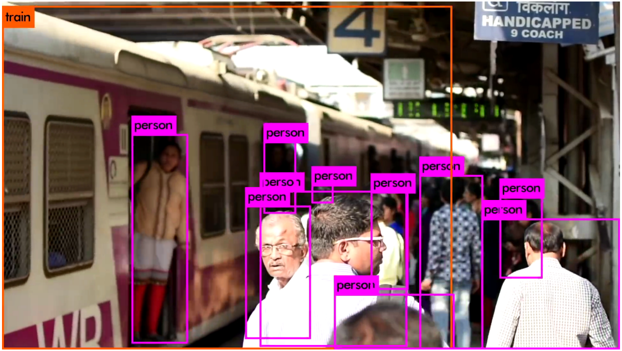


Figure: Example of detection done by Yolo v3 [5] after training on MS COCO [2]

Example: SSD

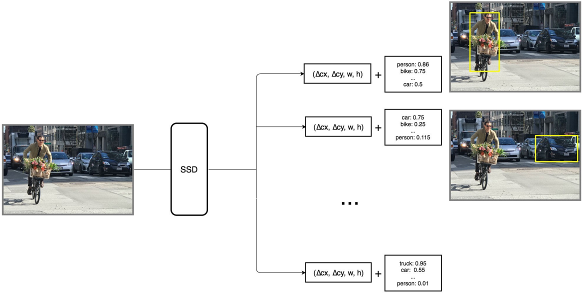


Figure: Schematic explanation of how the SSD head work [4]

Anchors

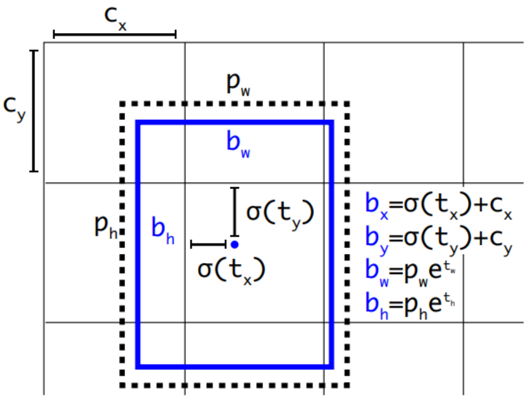


Figure: Visual description of an anchor

Real data

Figure: Example of a clear image from the dataset

Real data

Figure: Example of a smartphone collected image from the dataset

Artificial data

Figure: Example of artificially generated data

Aspect ratio

Figure: Example of sign randomly sampled from our dataset, one example per class.

Position and size accuracy

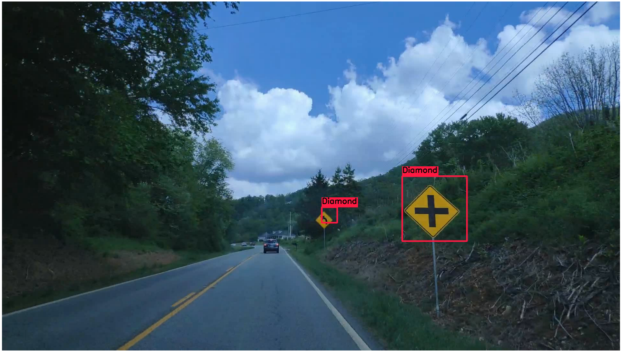
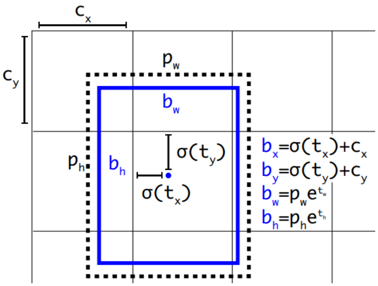


Figure: Example of not perfectly accurate detection boxes

Task complexity

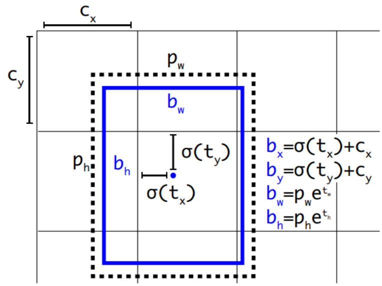
Figure: Example of image with an owl, from MS COCO dataset [2]

Assumption implementation



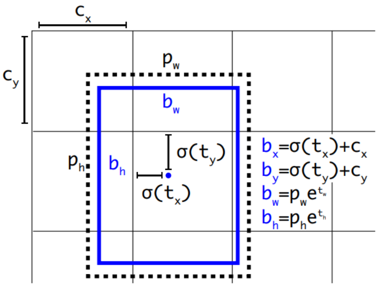
- Set $t_w = t_h$

Assumption implementation



- Set $t_w = t_h$
- Set $t_x = t_y = 0 \Leftrightarrow b_x = c_x + 0.5$ and $b_y = c_y + 0.5$

Assumption implementation



- Set $t_w = t_h = 0$
- Set $t_x = t_y = 0 \Leftrightarrow b_x = c_x + 0.5$ and $b_y = c_y + 0.5$

Structure chosen

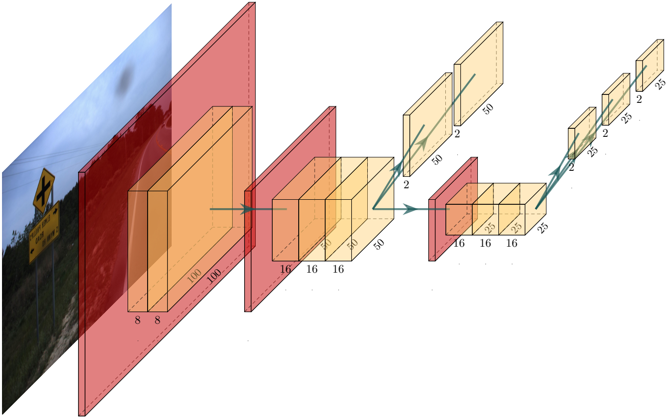


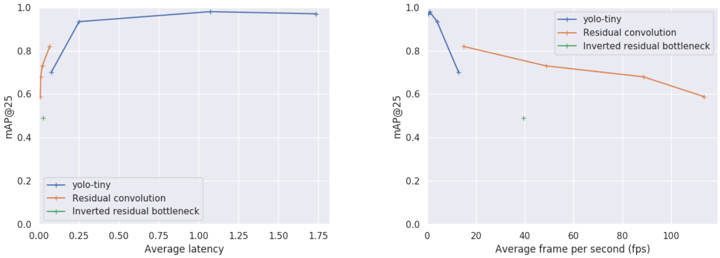
Figure: Final architecture chosen, based on MobileNet v2 first block

Using the artificial data

Block type	Filters	mAP				Improvement	
		Direct Training		Fine tuning			
		@25	@50	@25	@50	@25	@50
Residual	(8, 16)	0.58	0.45	0.68	0.55	+17.2%	+22.2%
	(16, 24)	0.00	0.00	0.73	0.61	$+\infty$	$+\infty$
	(32, 64)	0.62	0.52	0.82	0.67	+32.3%	+28.8%
Inverted Residual	(8, 16)	0.49	0.38	0.43	0.30	−12.24%	−21.05%

Table: Summary of the evolution of the mAP with and without fine turning

Latency study



(a) Evolution of the mAP@25 with the latency

(b) Evolution of the mAP@25 with the frame per second

Figure: Plots of the ratio computation speed vs accuracy for different architectures

West Westley

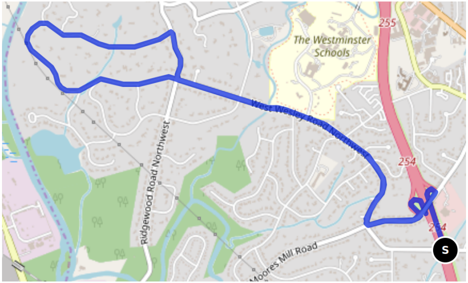


Figure: Map of the West Westley test road

Frame count	Sign count	TP	FP	FN	Precision	Recall
22, 198	34	31	10	3	76%	91%

West Wesley False Positive cases

(a) Small yellow flag

(b) School Sign (MUTCD: S1-1)

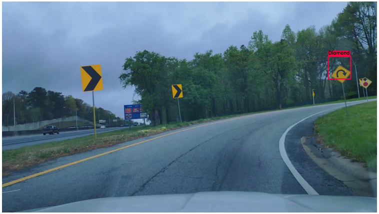
(c) Chevron sign

(d) Back of traffic light

West Wesley False Negative cases

(a) Whitened sign

(b) Dirty, faded sign in the shadow



(c) Small often obstructed sign

West Wesley True Positive example

(a) Obstructed sign

(b) Obstructed sign

(c) Obstructed sign

Wrap up

Assumptions:

- Low position accuracy ✓
- Small backbone ✓

Future work

- Implement and deploy android / iOS app

Future work

- Implement and deploy android / iOS app
- Experiment with higher position accuracy

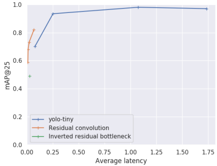
Future work

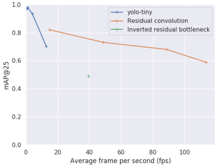
- Implement and deploy android / iOS app
- Experiment with higher position accuracy
- Try full model quantification

Future work

- Implement and deploy android / iOS app
- Experiment with higher position accuracy
- Try full model quantification
- Try on other signs

Contributions





- Accurate detection of traffic sign on mobile device at above real time speed
- New artificial data generation method for pretraining

Questions?

References

R Janssen, W Ritter, F Stein, and S Ott.
Hybrid approach for traffic sign recognition.
In *Proceedings of the Intelligent Vehicles' 93 Symposium*, pages 390–395. IEEE, 1993.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick.
Microsoft COCO: common objects in context.
CoRR, abs/1405.0312, 2014.

Hanxiao Liu, Karen Simonyan, and Yiming Yang.
Darts: Differentiable architecture search.
arXiv preprint arXiv:1806.09055, 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg.
Ssd: Single shot multibox detector.
In *European conference on computer vision*, pages 21–37. Springer, 2016.

Joseph Redmon and Ali Farhadi.
Yolov3: An incremental improvement.
arXiv, 2018.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen.
Mobilenetv2: Inverted residuals and linear bottlenecks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

Berthold Ulmer.
Vita ii-active collision avoidance in real traffic.
In *Proceedings of the Intelligent Vehicles' 94 Symposium*, pages 1–6. IEEE, 1994.

mAP vs speed numbers

Architecture	Input size	Filters	mAP@50	mAP@25	Latency (s)	FPS
Yolov3-tiny	704x416	-	0.92	0.97	1.738	0.6
	576x320	-	0.96	0.98	1.075	0.9
	160x288	-	0.77	0.93	0.252	4.0
	96x160	-	0.39	0.70	0.078	12.8
Residual convolution	110x200	(6,10)	0.48	0.59	0.009	113.7
	110x200	(8,16)	0.55	0.68	0.011	88.7
	110x200	(16,24)	0.61	0.73	0.020	48.9
	110x200	(32,64)	0.67	0.82	0.068	14.8
Inverted residual bottleneck	110x200	(8,16)	0.38	0.49	0.025	39.4

Table: Value of the mAP, latency and frame per second for different configuration and different models

A feature extractor

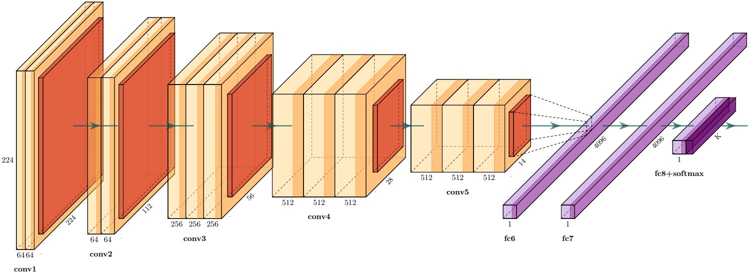


Figure: Example of the VGG16 architecture, on a classification task

Doing it manually

Figure: Original Image

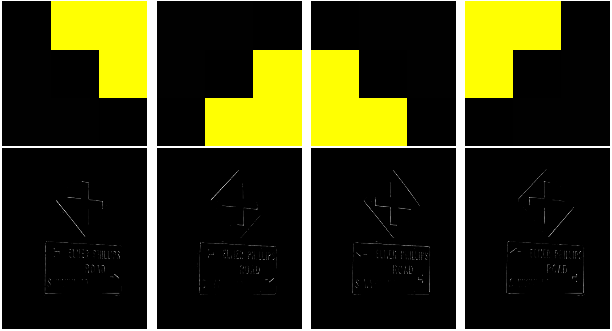
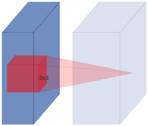


Figure: Edge detector convolution filters and results.

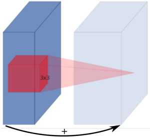
Doing it manually

Figure: Example of detection of a manually made model after training

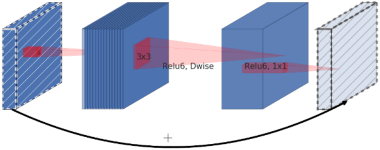
Different blocks



(a) 2D Convolution block



(b) Residual convolution block



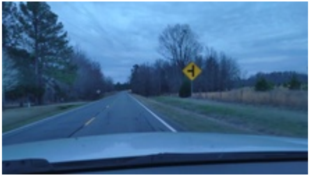
(c) Inverted residual block [6]

Figure: Graphical representation of the different block used in this study, Illustrations from MobileNet V2 paper [6]

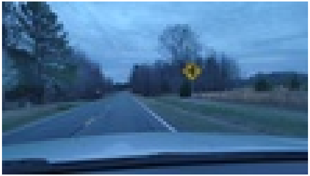
About input size

(a) 1080 × 1920

(b) 440 × 800



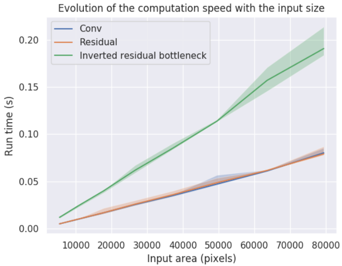
(c) 220 × 400



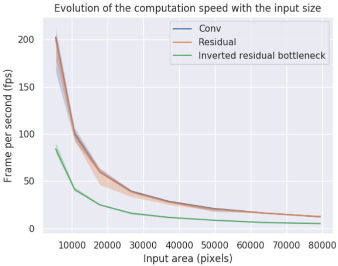
(d) 110 × 200

Figure: Example of an image resized at different size and displayed at same size

Input size and speed



(a) Latency evolution with input size



(b) FPS evolution with input size

Figure: Evolution of the prediction speed for different block type, using Tensorflow Lite version of the model on Samsung S6

Neural network structure

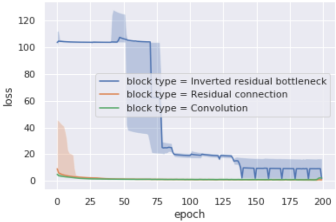
Name	Filter	Kernel	Stride	Activation	BN	Res	Input size	Input
Conv1	8	(3, 3)	(2, 2)	ReLu6	✓		(111, 201, 3)	
block_0	8	(3, 3)	(1, 1)	ReLu6	✓	✓	(55, 100, 8)	Conv1
block_1	16	(3, 3)	(2, 2)	ReLu6	✓		(55, 100, 8)	block_0
block_2	16	(3, 3)	(1, 1)	ReLu6	✓	✓	(28, 50, 16)	block_1
block_3	16	(3, 3)	(1, 1)	ReLu6	✓	✓	(28, 50, 16)	block_2
block_4	16	(3, 3)	(2, 2)	ReLu6	✓		(28, 50, 16)	block_3
block_5	16	(3, 3)	(1, 1)	ReLu6	✓	✓	(14, 25, 16)	block_4
block_6	16	(3, 3)	(1, 1)	ReLu6	✓	✓	(14, 25, 16)	block_5
output_1	2	(3, 3)	(1, 1)	Linear			(28, 50, 16)	block_3
output_2	2	(3, 3)	(1, 1)	Linear			(28, 50, 16)	block_3
output_3	2	(3, 3)	(1, 1)	Linear			(14, 25, 16)	block_6
output_4	2	(3, 3)	(1, 1)	Linear			(14, 25, 16)	block_6
output_5	2	(3, 3)	(1, 1)	Softmax			(14, 25, 16)	block_6

Table: Description of the neural network structure, block by block, giving the link between the blocks and some information such as the use Batch Normalization (BN) or residual connections (Res).

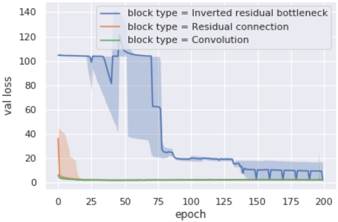
Overfitting study

Traffic sign detection on smartphone

Nicolas Six



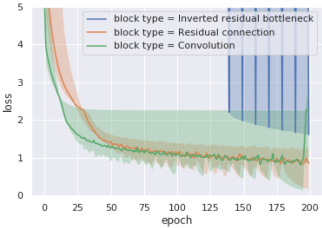
(a) Evolution of the training loss with epoch, for different block type



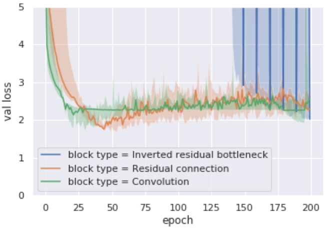
(b) Evolution of the validation loss with epoch, for different block type

Figure: Evolution of the validation and training loss for different block type

Overfitting study



(a) Evolution of the training loss with epoch, for different block type, zoomed on loss between 0 and 5



(b) Evolution of the validation loss with epoch, for different block type, zoomed on loss between 0 and 5

Figure: Evolution of the validation and training loss for different block type

Real data

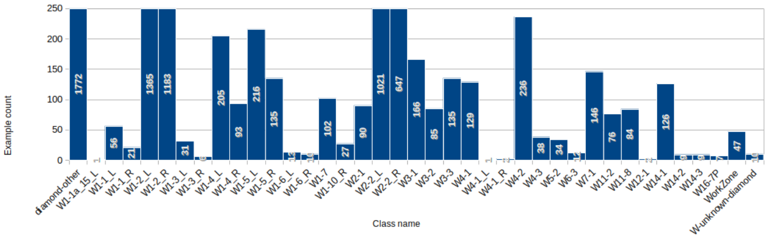


Figure: Count of example for each classes in the dataset, cropped to 250 for readability.

US Diamond Warning signs

(a) W1-1 (b) W1-3 (c) W1-4

Figure: Example of US diamond warning signs, with there MUTCD sign class.

Architecture search

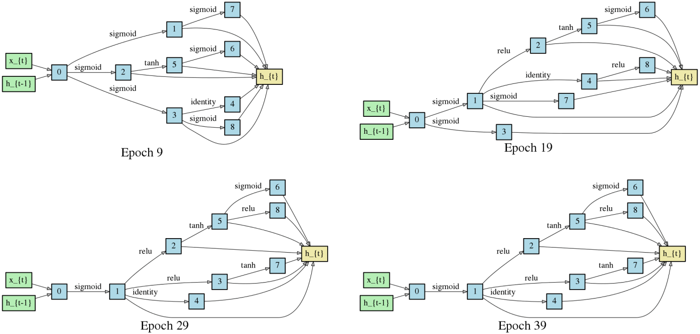


Figure: Demo of the architecture search process as used in Darts [3]

Testing

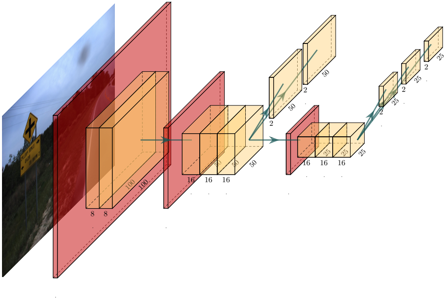


Figure: Architecture used for testing

Filters	Latency (s)	FPS	mAP@25	mAP@50
(8, 16)	0.011	88.7	0.68	0.55

State Road 2

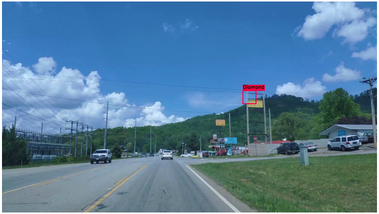
Figure: Map representing the test section we used on State Road 2

SR2 False Positive cases

(a) Chevron Sign

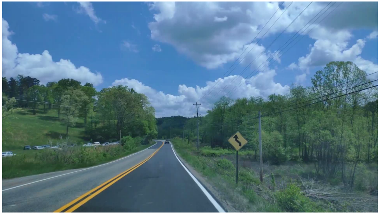
(b) Guard rail signalization

(c) Work zone diamond sign



(d) Advertisement sign

SR2 False Negative cases



(a) Light, faded sign

(b) Sign on the other side of the road

(c) Light, faded sign

(d) Light, faded sign

SR2 True Positive example

(a) Temporary sign

(b) Small sign

(c) Multiple signs

(d) Temporary sign