Robustness in Computer Vision

Sebastian Cygert

Gdansk University of Technology
sebcyg@multimed.org

MI^2*DataLab*, 31.05.2021

Bio

- 2008-2012, Warsaw Military University of Technology - Cryptography
- 2012-2013, Warsaw University of Technology, CS **@MINI** faculty (the simulation of relativistic heavy ion collisions with Physics' faculty)
- 2013-2017 - Moody's Analytics, London, financial modeling
- 2016 - now - **PhD studies** @Gdansk University of Technology
- 2019 - Amazon Scout project, Tubingen (Germany)
- 2021 - **GUMED**, Centrum Analiz Biostatystycznych i Bioinformatycznych.

Overview

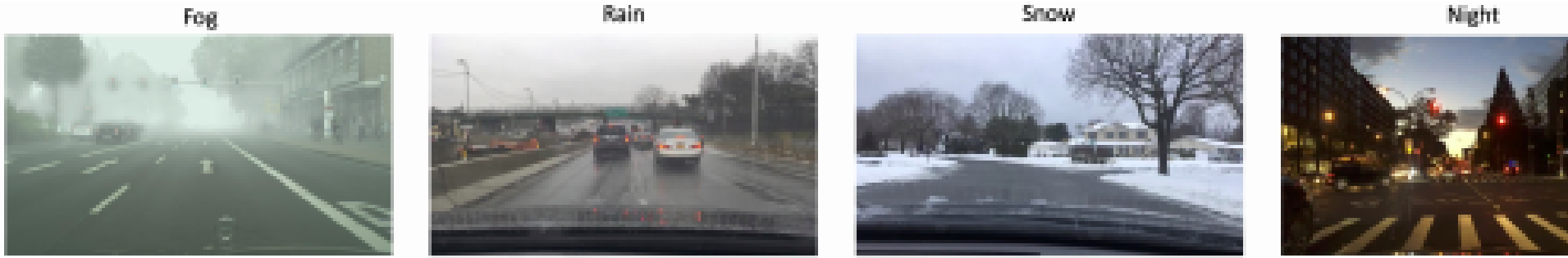
1. Motivation
2. On importance of o.o.d. testing
3. Semantic segmentation uncertainty under distributional shift
4. Efficient ensembling for Object Detection

Motivation

- Dramatic performance reduction in novel / rare conditions
- Current architectures don't need to be more precise
- We need stability / robustness in novel / rare conditions

Safety-critical AI

- Real world is very complicated and full of rare events (and combinations of rare events:) **Impressive benchmark results do not transfer well to the real-world.**
- Accuracy (**and uncertainty calibration**) in out of distribution setting (o.o.d.) is crucial for for applications deployed to the **open-world** (autonomous driving, medicine)



Michaelis et. al, Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming

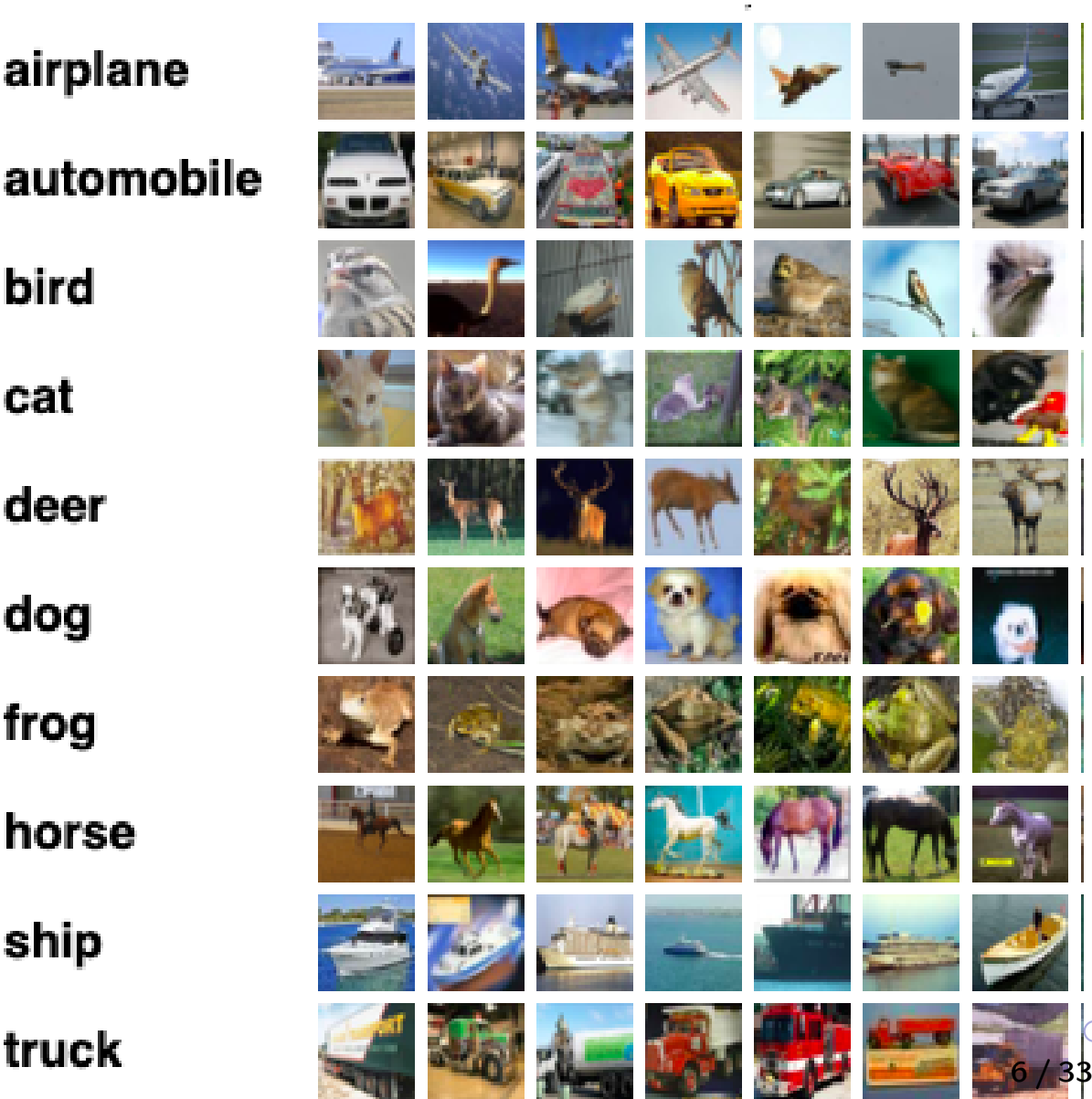
Memorization in deep learning, [Zhang 2017]

a

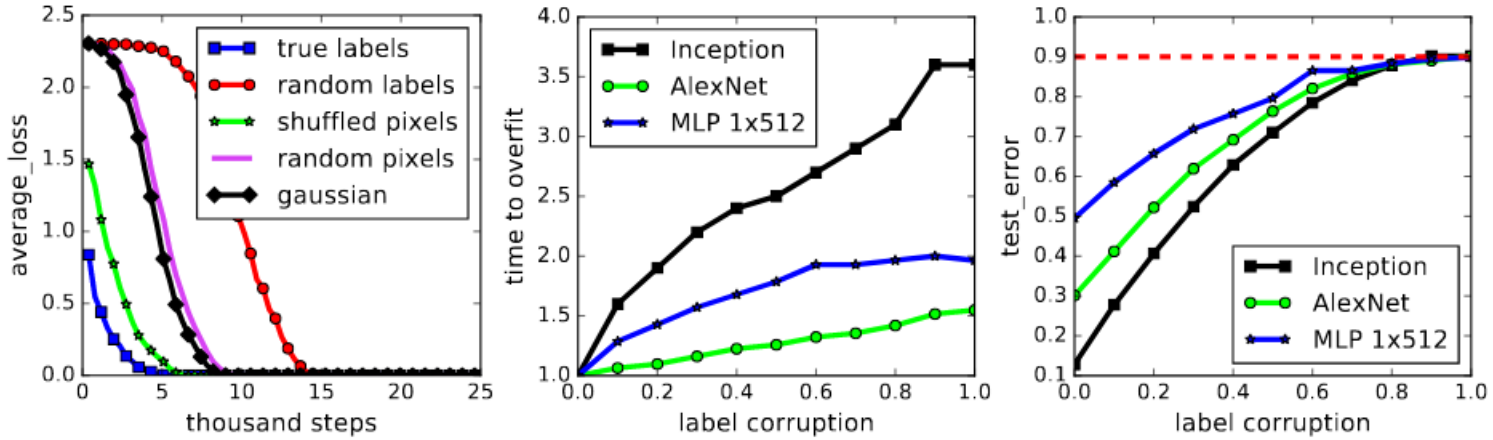
Baselines:

- Standard training
- Shuffled pixels
- **Random labels**

How hard those tasks would be for neural networks?



Memorization in deep learning, [Zhang 2017]



(a) learning curves (b) convergence slowdown (c) generalization error growth

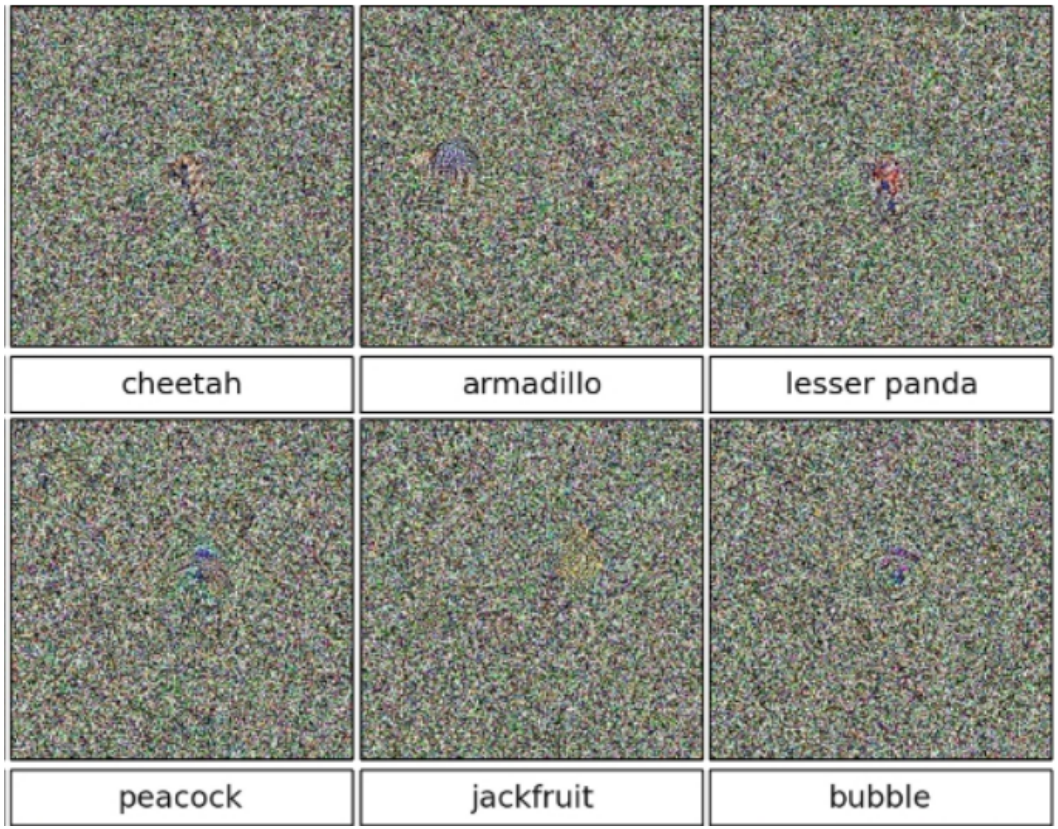
Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

- Neural nets have capacity to memorize all (random) training data (**0 training loss)** - over-parametrization
- Fitting true labels is only **slightly easier** than fitting random data
- Compression methods could be a way to "undestand" deep learning

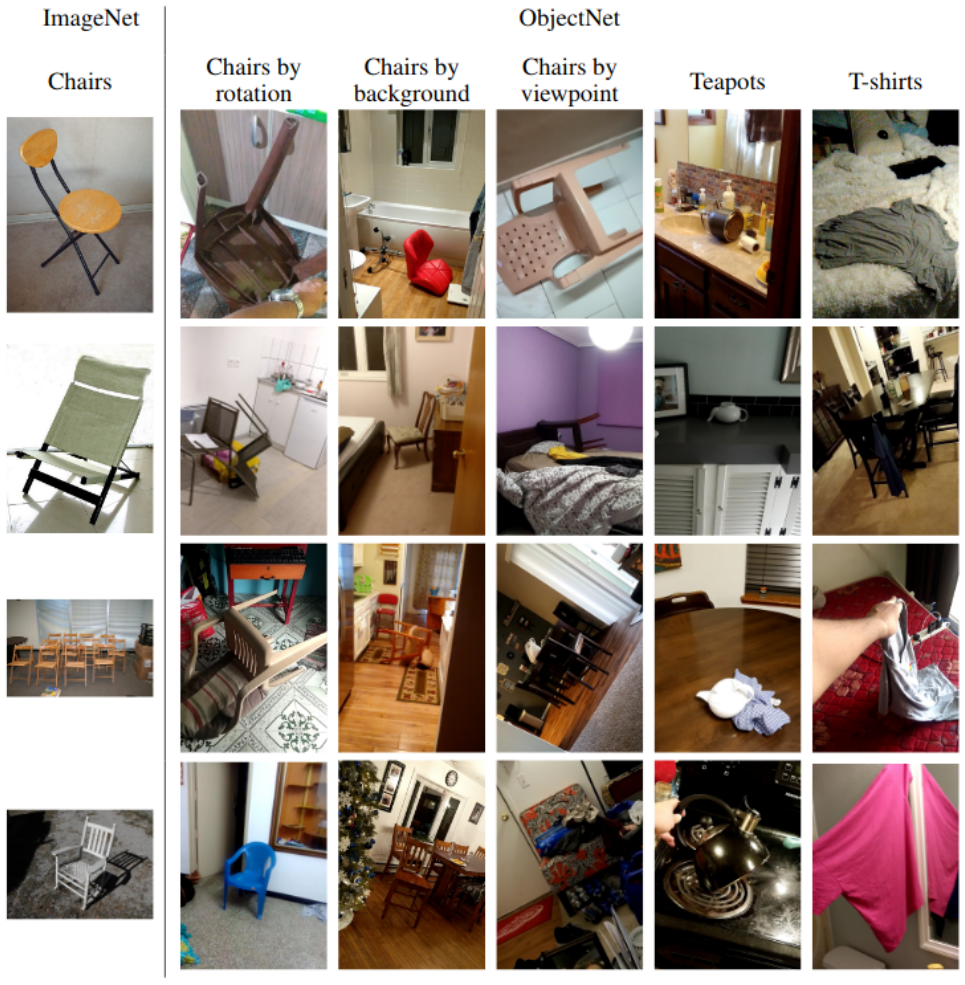
Overview

1. Motivation
2. On importance of o.o.d. testing
3. Semantic segmentation uncertainty under distributional shift
4. Efficient ensembling for Object Detection

High Confidence Predictions for Unrecognizable Images, [Nguyen 2015]



Typical objects in untypical poses, [Barbu 2019]



Typical objects in untypical poses, [Barbu 2019]

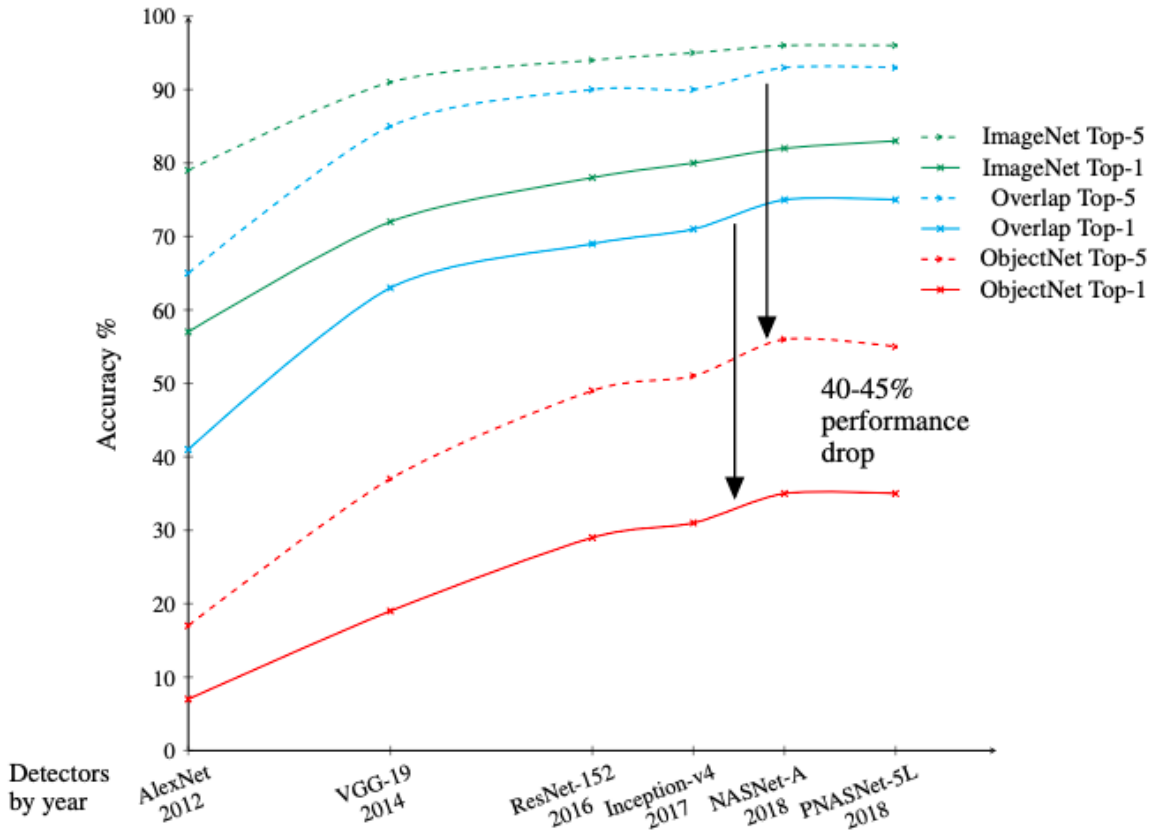
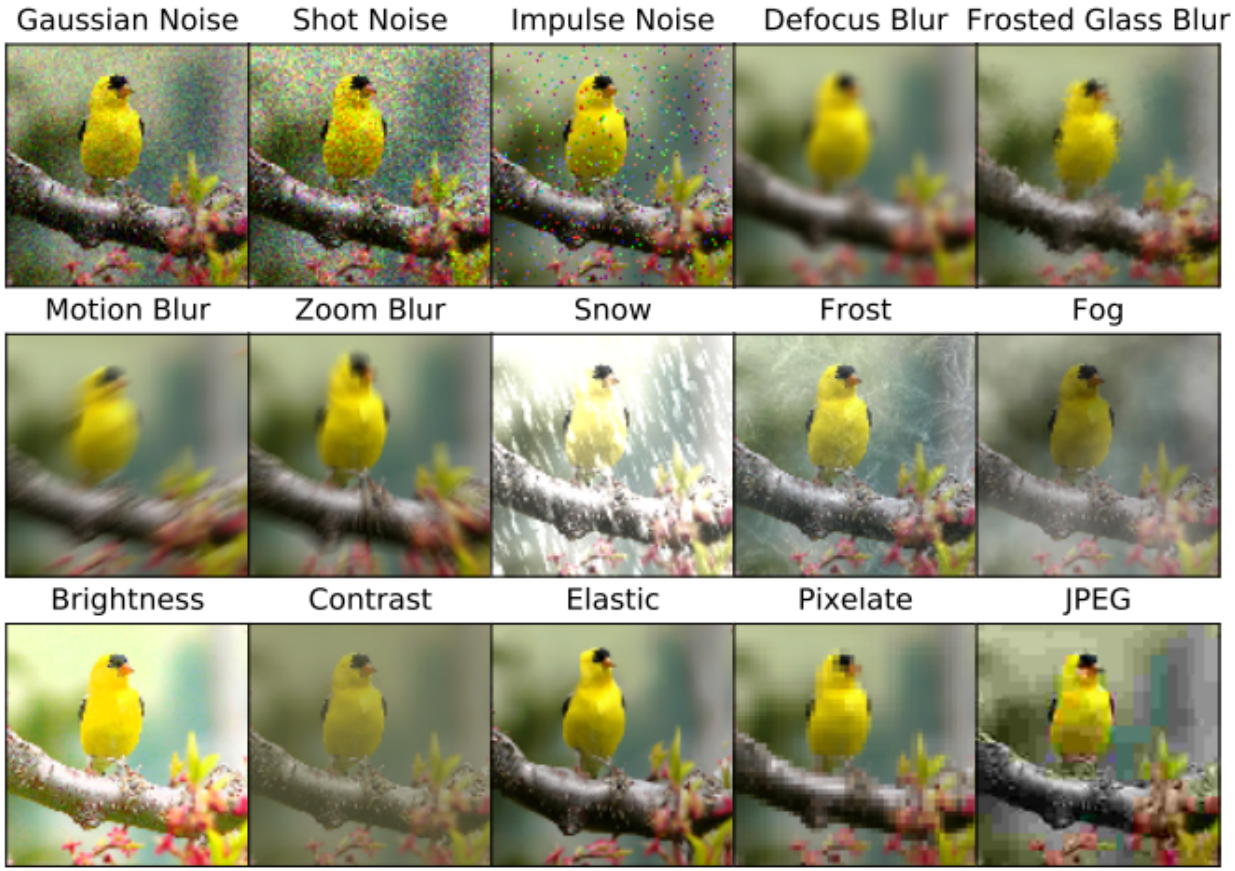


Figure 1: Performance on ObjectNet for high-performing detectors trained on ImageNet in recent years: AlexNet [4], VGG-19 [5], ResNet-152 [6], Inception-v4 [7], NASNET-A [8], and PNASNet-5 Large [9]. Solid lines show top-1 performance, dashed lines show top-5 performance. ImageNet performance on all 1000 classes is shown in green. ImageNet performance on classes that overlap with ObjectNet is shown in blue; the two overlap in 113 classes out of 313 ObjectNet classes, which are only slightly more difficult than the average ImageNet class. Performance on ObjectNet for those overlapping classes. We see a 40-45% drop in performance. Object detectors have improved substantially. Performance on ObjectNet tracks performance on ImageNet but the gap between the two remains large.

Common Corruptions



- Large decrease in image recognition models
- Can be used to approximate robustness (test-time only)

D. Hendrycks, T. Dietterich: Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. ICLR 2019

Correlation

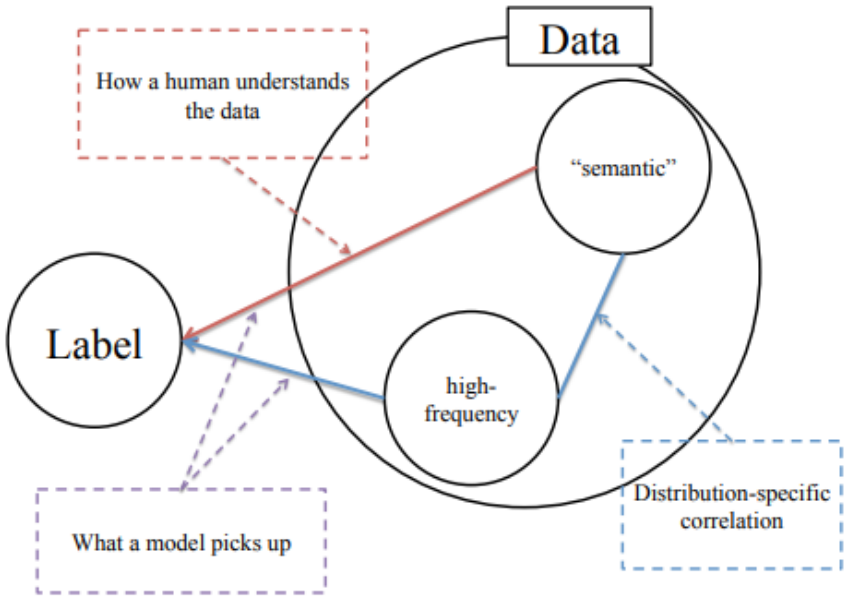


Figure 1. The central hypothesis of our paper: within a data collection, there are correlations between the high-frequency components and the "semantic" component of the images. As a result, the model will perceive both high-frequency components as well as the "semantic" ones, leading to generalization behaviors counter-intuitive to human (*e.g.*, adversarial examples).

- Models learn the simplest feature that does the job on the data.
- Explains effectiveness of data augmentation methods

H. Wang et al.,: High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. CVPR 2020

Other domains

- NLP - sentiment analysis (1-word attacks) [Ren, Shuhuai, et al. "Generating natural language adversarial examples through probability weighted word saliency", 2019]
- Text-to-speech [N. Carlini et al., "Audio adversarial examples: Targeted attacks on speech-to-text", 2018]
- Reinforcement learning Gleave, A., et al. "Adversarial policies: Attacking deep reinforcement learning", ICLR 2020.

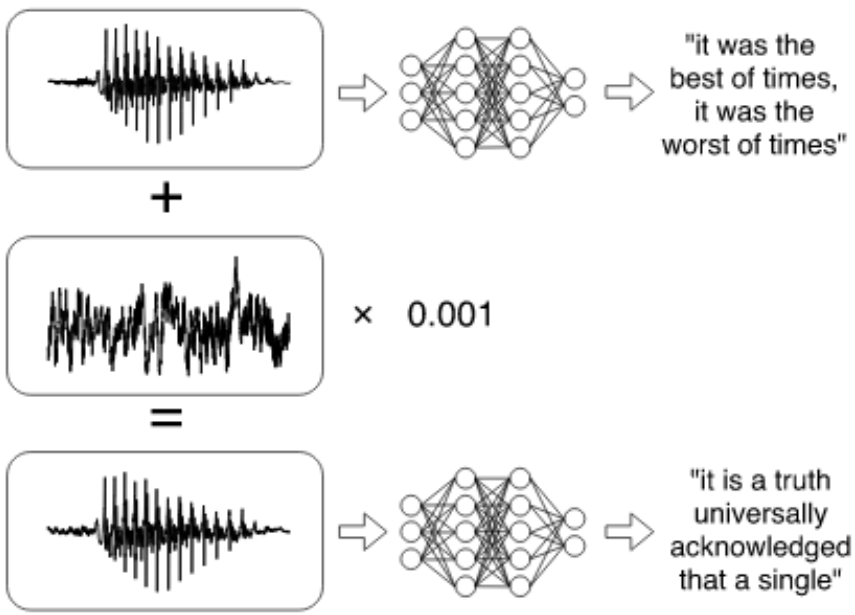


Figure 1. Illustration of our attack: given any waveform, adding a small perturbation makes the result transcribe as any desired target phrase.

Summary

- Current DNNs are very brittle (**low-level features, context**) in their predictions and may fail in o.o.d. setting (with **high confidence**)
- Current models are heavily overparametrized and can be efficiently compressed
- Literature on robustness suggests: bigger models, more data, data augmentation, sampling-based methods (**deep ensembling**)

Summary

- Current DNNs are very brittle (**low-level features, context**) in their predictions and may fail in o.o.d. setting (with **high confidence**)
- Current models are heavily overparametrized and can be efficiently compressed
- Literature on robustness suggests: bigger models, more data, data augmentation, sampling-based methods (**deep ensembling**)
- **Can we combine model ensembling and model compression?**

Overview

1. Motivation
2. On importance of o.o.d. testing
3. Semantic segmentation uncertainty under distributional shift
4. Efficient ensembling for Object Detection

Scope of the study

- Study accuracy and uncertainty estimation under **varying level of distributional shift** for the high-level task of semantic segmentation
- Study how state-of-the art method (model ensembling) works in that setting
- **Model ensembling** by means of different augmentation methods (color jittering and style-transfer) and backbones (**to improve ensemble diversity**)
- Make use of improved model accuracy and uncertainty estimation (model ensembling) to collect pseudo-labels on unseen data, and finetune the model in the target domain (**domain adaptation**)

Model ensemble

Table: Ensemble of models performance (ResNet-101, Xception41, Xception65). Also mean performance of all models is reported.

Model name	mIoU	pix. acc	ECE	mIoU	pix. acc	ECE
	GTA-to-GTA			**GTA-to-Cityscapes**		
M=3	**81.9**	**96.8**	0.81	43.2	84.7	2.45
M=5	81.4	96.7	1.02	**44.5**	**86.3**	**1.1**
Models mean	79.0	96.3	**0.21**	41.4	83.7	6.08
	Cityscapes-to-Cityscapes			**Cityscapes-to-BDD**		
M=3	**77.2**	**96.0**	0.36	55.7	91.3	1.99
M=5	77.0	**96.0**	**0.29**	**56.2**	**91.7**	**1.09**
Models mean	73.8	95.4	1.16	49.3	89.8	4.56

Uncertainty estimation

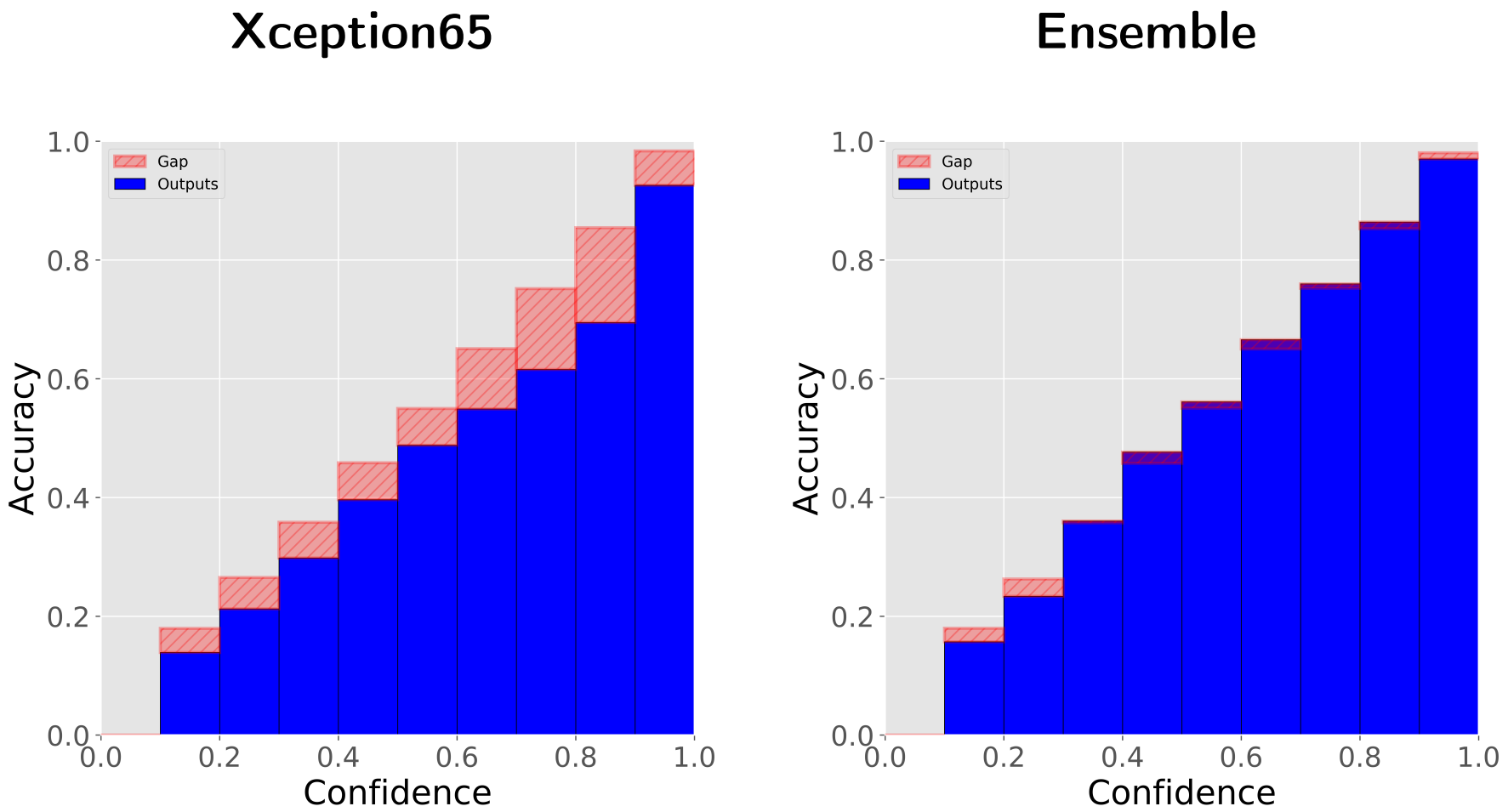
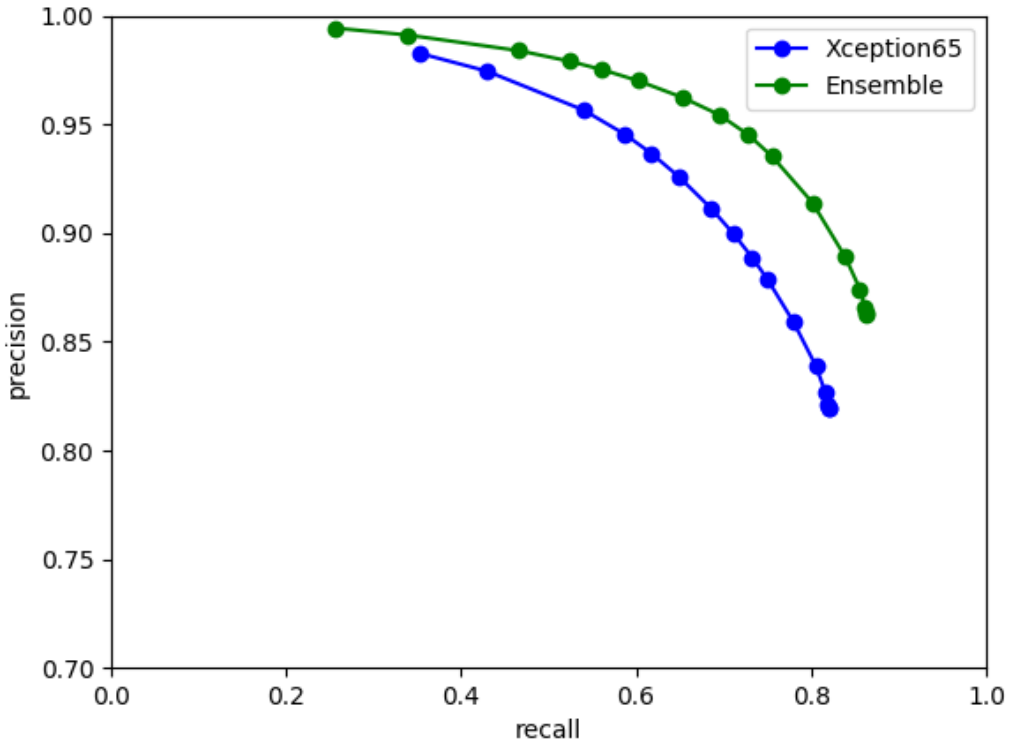


Figure: Calibration plots for Xception65 model and model ensemble (M=5) evaluated on the GTA-to-Cityscapes adaptation.

Pseudo-labelling



Threshold = 0.95 was used in the domain adaptation (70.1% of the pixels to be annotated with 92.6% accuracy)

Final accuracy

TABLE IV: Domain adaptation results for our models with per-class evaluation.

Name	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
Gta to Cityscapes																				
Baseline	29.9	17.4	62.8	13.2	14.7	15.5	26.8	10.7	79.0	8.4	47.4	53.5	10.3	48.2	25.7	3.04	0.	11.4	4.5	25.4
CJ	80.3	28.9	80.9	30.9	22.5	25.8	37.0	17.5	83.8	31.0	76.6	**58.4**	**19.6**	83.0	28.7	24.7	0.	**27.4**	**11.0**	40.4
CJ + fine	86.1	36.4	83.1	24.9	28.7	27.8	**39.6**	19.4	85.7	38.4	79.5	56.9	13.0	86.5	31.0	23.6	0.	22.6	0.	41.2
CJ + ens	**88.6**	**43.2**	**85.0**	**36.3**	**33.8**	**30.7**	37.4	**21.9**	**86.8**	**44.9**	**83.9**	57.5	14.5	**87.3**	**37.2**	**32.2**	0.	15.0	0.	**44.0**
Cityscapes to BDD																				
Baseline	88.9	52.4	65.2	18.5	18.7	35.2	35.7	31.9	78.2	36.1	75.8	47.3	22.3	78.5	23.4	32.7	0.	41.2	32.0	42.8
CJ	91.8	54.5	79.9	**19.8**	27.1	**41.9**	**43.3**	43.8	82.5	39.1	91.4	58.2	29.7	85.2	27.7	25.5	0.	**49.1**	42.6	49.1
CJ + fine	93.2	60.4	**81.4**	18.7	36.6	37.4	40.5	44.2	83.0	42.0	91.7	62.2	43.7	85.1	36.4	23.6	0.	47.6	48.7	51.4
CJ + ens	**94.4**	**62.5**	81.0	17.5	**37.7**	38.6	38.6	**45.5**	**85.0**	**43.2**	**92.2**	**63.2**	**46.8**	**87.1**	**42.6**	**54.7**	0.	44.9	**53.4**	**54.2**

Qualitative results

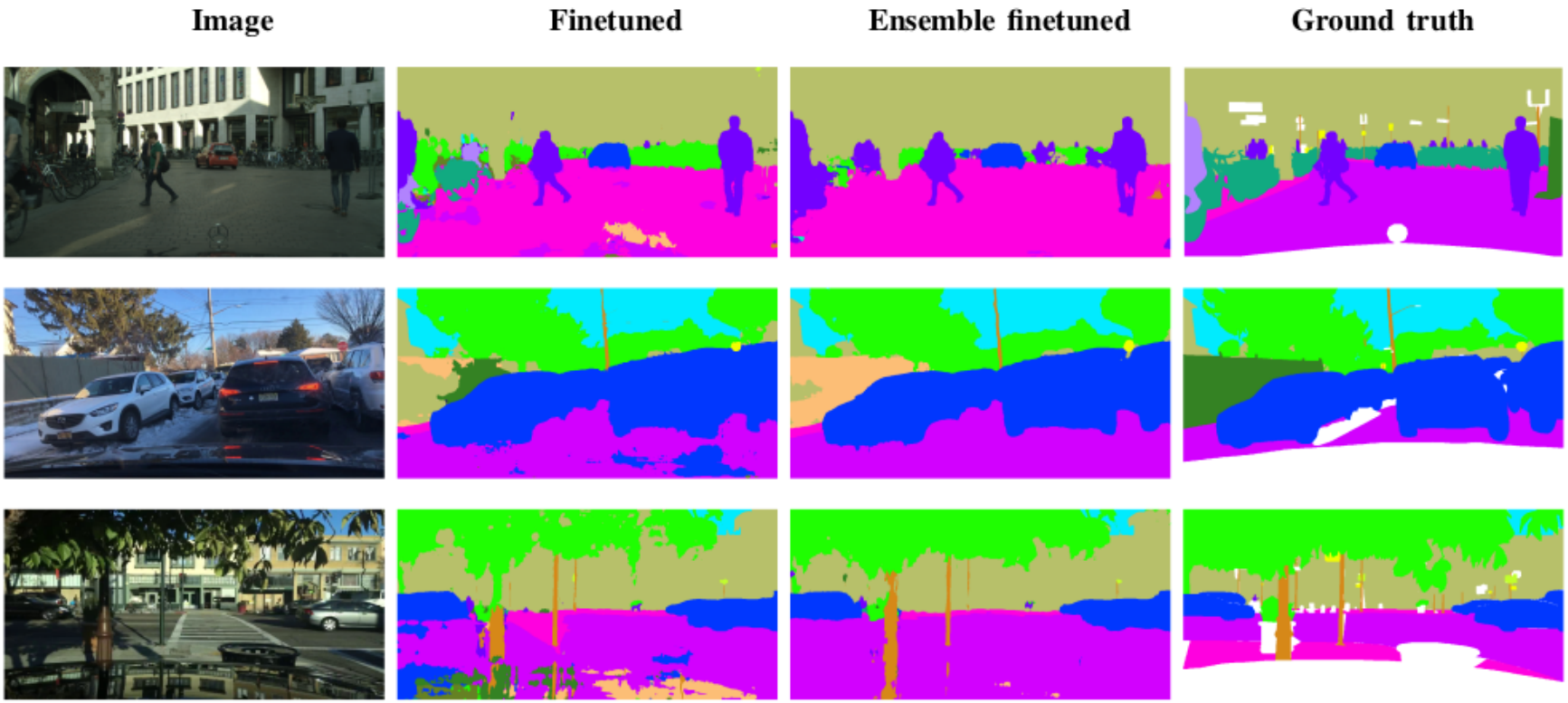


Fig. 5: Qualitative results of trained models on GTA-to-Cityscapes transfer (first row) and Cityscapes-to-BDD transfer (consecutive rows). White color corresponds to the ignore label.

Overview

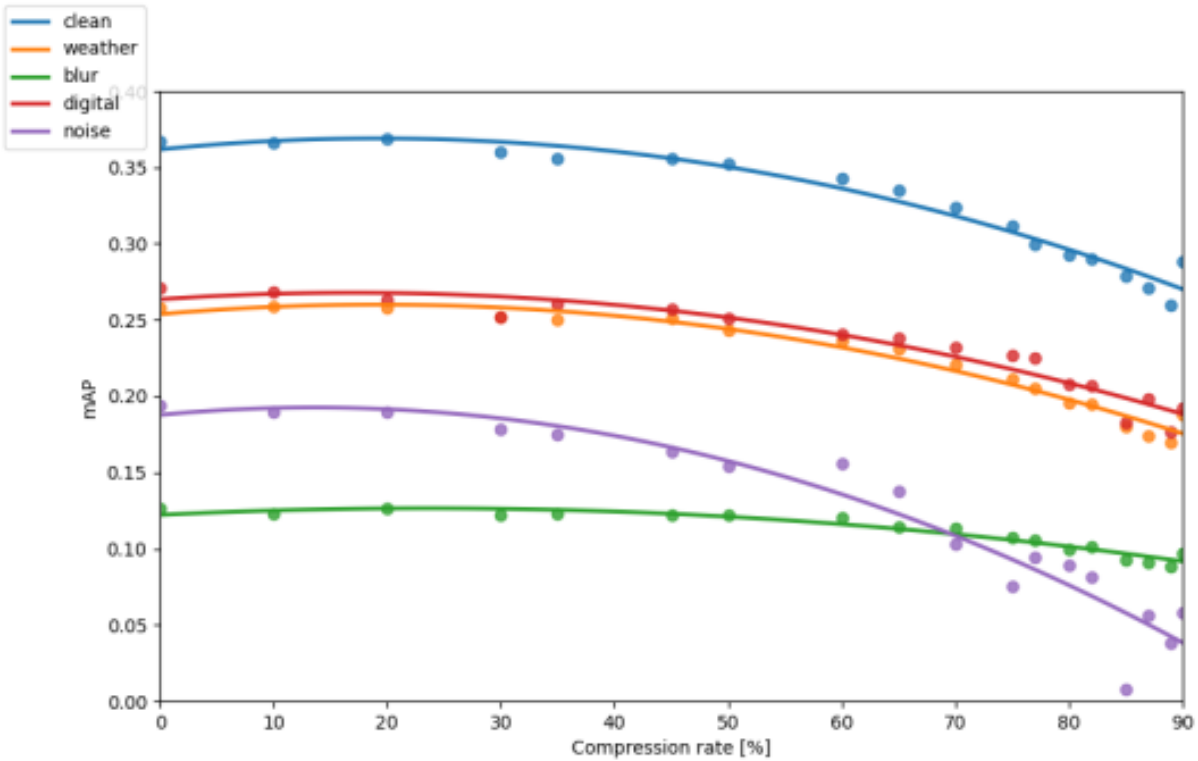
1. Motivation
2. On importance of o.o.d. testing
3. Semantic segmentation uncertainty under distributional shift
4. Efficient ensembling for Object Detection

Can we combine model compression and model ensembling?

- It is possible to prune more than **90% of weights** (unstructured pruning) or **around 50% filters** (structured pruning) without decreasing final accuracy

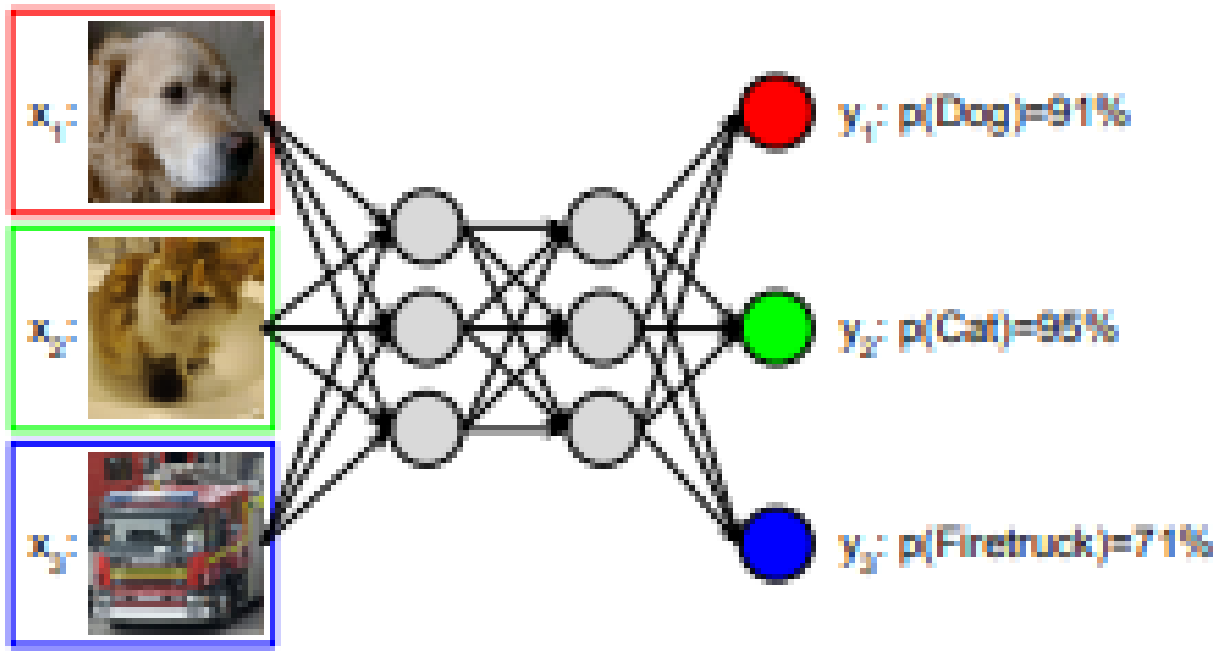
Can we combine model compression and model ensembling?

- It is possible to prune more than **90% of weights** (unstructured pruning) or **around 50% filters** (structured pruning) without decreasing final accuracy



S.Cygert, A. Czyżewski, Robustness in Compressed Neural Networks, IJCNN 2021

Multi-input multi-output networks



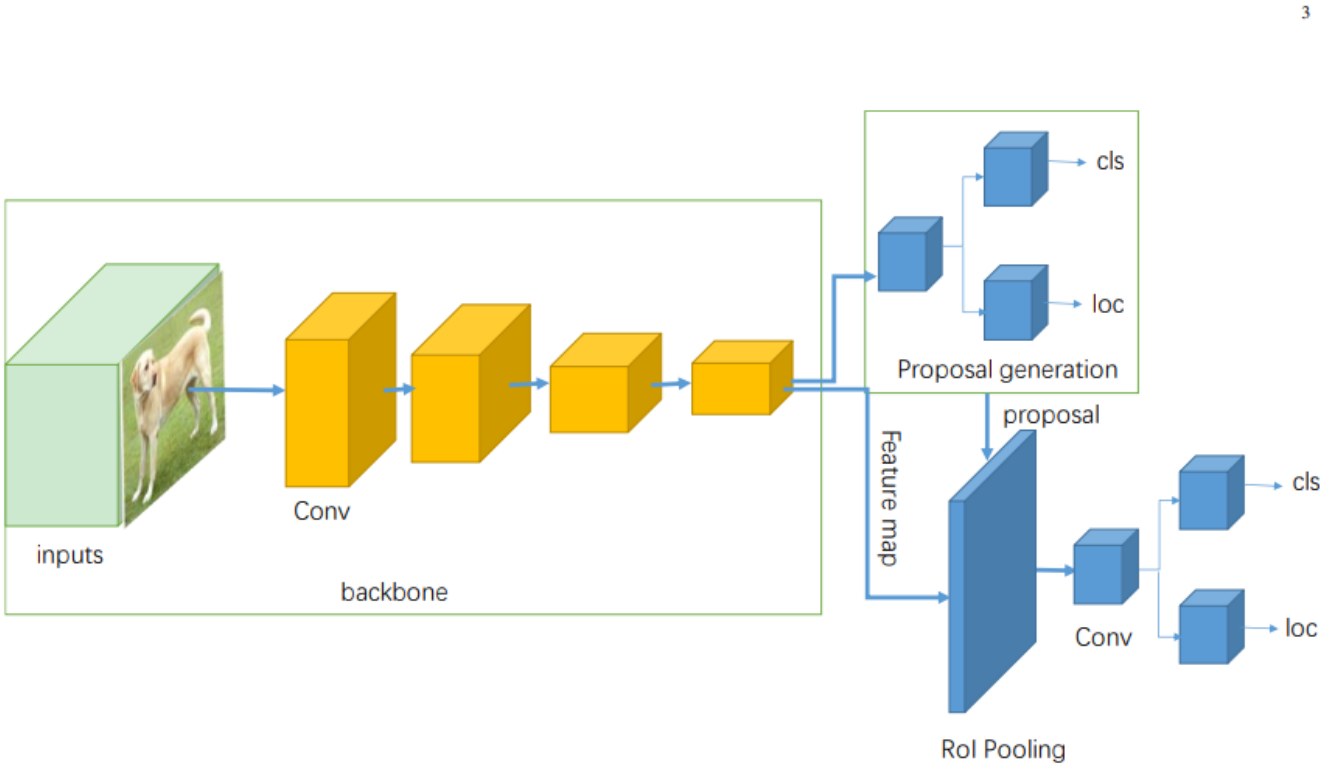
(a) Training

M. Havasi et. al, Training independent subnetworks for robust prediction, ICML 2021

MIMO

- Adds 1% in number of parameters, almost the same inference time.
- CIFAR 10/100. 2/3 subnetworks can be fitted efficiently, **matches Deep Ensemble results** in terms of accuracy and ECE.
- Better utilization of capacity (higher number of active filters)
- ImageNet with 2 subnetworks has decreased accuracy (**not enough capacity)**. Need to relax independence constraint (subnetworks use the same training image with $p = 0.4$ probability

MIMO Object Detection



- Multi-input (backbone with the same capacity computes features for M images at once)
- RPN computes M set proposals at once

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R. (2019). A survey of deep learning-based object detection

Changing p-value

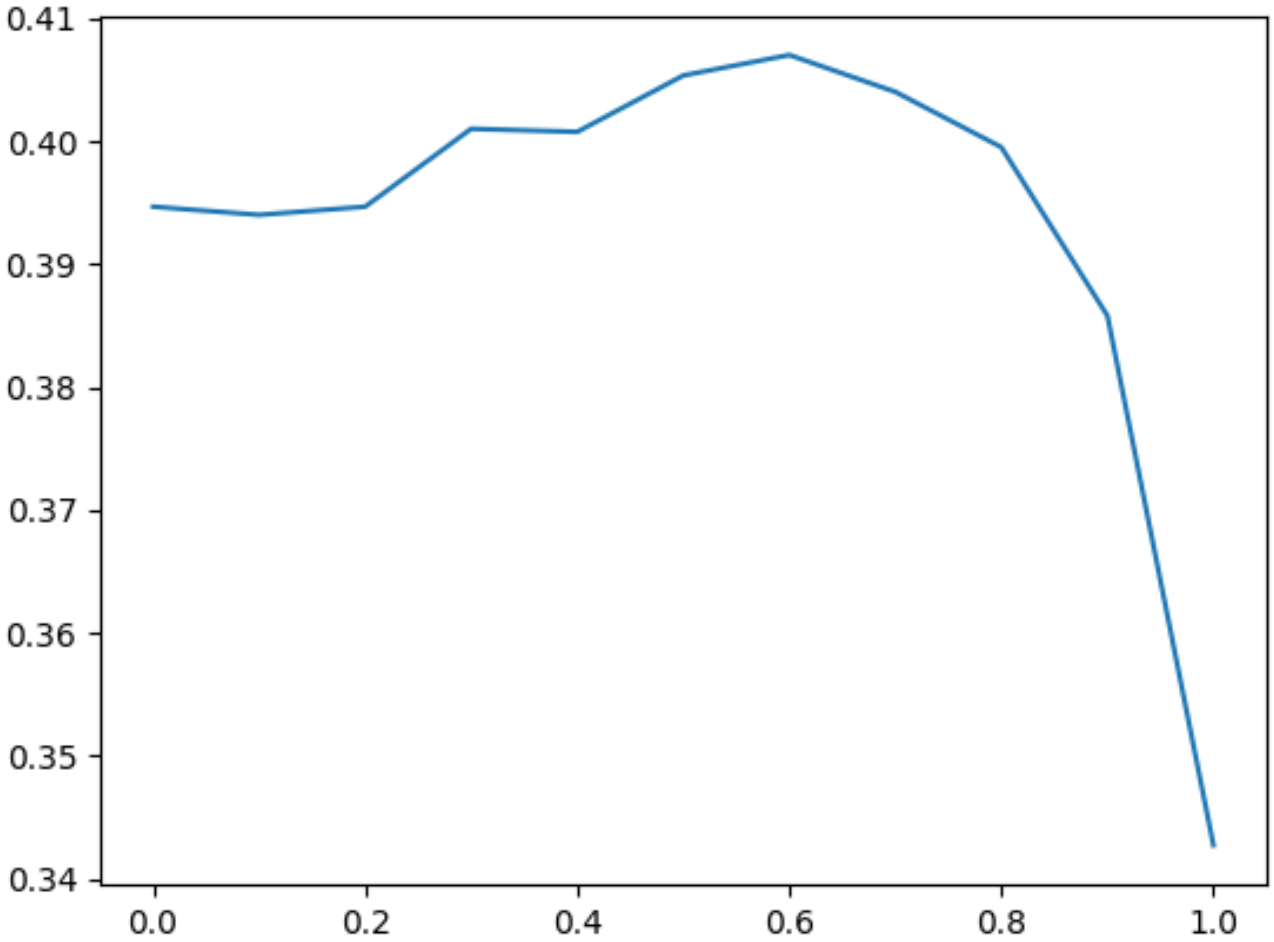


Figure: mAP (y-axis) as the function of subnetworks independence. p = 0 (same data), p = 1 (data sampled independently)

Results

Table: Accuracy and computational cost of different methods.

Model	mAP	c-mAP	parameters	inference time
Baseline	0.386	0.105	41.384M	0.088
MIMO (M=2)	0.409	0.172	41.397M	0.102
Deep Ensemble (M=2)	0.406	0.169	82.768M	0.176
Deep MIMO Ensemble	0.417	0.199	82.794M	0.204

Aggregation method:
https://github.com/ZFTurbo/Weighted-Boxes-Fusion

Conclusions

- O.o.d. accuracy is an important problem for open-world deployment (autonomous driving, medicine)
- Sampling based-methods (i.e. deep ensembling) can improve accuracy and uncertainty calibration in o.o.d. setting
- MIMO approach can be used applied to high-level task for very efficient ensembling

GDAŃSK UNIVERSITY
OF TECHNOLOGY

Thank you for attention

Questions?

sebcyg@multimed.org?