Postprint of: Cygert S., Czyżewski A., Robustness in Compressed Neural Networks for Object Detection (2021), International Joint Conference on Neural Networks (IJCNN), 18-22.07.2021, DOI: 10.1109/IJCNN52387.2021.9533773

Robustness in Compressed Neural Networks for Object Detection

Sebastian Cygert
Multimedia Systems Department
Gdańsk, Poland
sebcyg@multimed.org

Andrzej Czyżewski
Multimedia Systems Department
Gdańsk University of Technology

***Abstract*—Model compression techniques allow to significantly reduce the computational cost associated with data processing by deep neural networks with only a minor decrease in average accuracy. Simultaneously, reducing the model size may have a large effect on noisy cases or objects belonging to less frequent classes. It is a crucial problem from the perspective of the models' safety, especially for object detection in the autonomous driving setting, which is considered in this work.**

It was shown in the paper that the sensitivity of compressed models to different distortion types is nuanced, and some of the corruptions are heavily impacted by the compression methods (i.e., additive noise), while others (blur effect) are only slightly affected. A common way to improve the robustness of models is to use data augmentation, which was confirmed to positively affect models' robustness, also for highly compressed models. It was further shown that while data imbalance methods brought only a slight increase in accuracy for the baseline model (without compression), the impact was more striking at higher compression rates for the structured pruning. Finally, methods for handling data imbalance brought a significant improvement of the pruned models' worst-detected class accuracy.

***Index Terms*—pruning, robustness, object detection, CNN, class imbalance**

I. Introduction

Optimization of the size of visual recognition models is of great importance, for example, for autonomous driving, because of energy consumption, hardware cost, and size. Typical methods to reduce the computation cost include deploying specialized architectures [1], model compression techniques such as reducing model precision (quantisation) [2] and/or setting the number of weight or filters to zero (pruning) [3]. For example Han et. al showed [4], that it is possible to reduce the size of the VGG network by a factor of 13 when benchmarking on the ImageNet dataset [5] with no loss in accuracy.

Another essential aspect for real-world deployment is the models' robustness. Many works have shown that current machine learning models for visual recognition from RGB images are vulnerable to tiny changes in the input image, such as adversarial examples [6], noisy input [7], [8] small transformations of the input image [9], [10] or varying background [11]. Yet, most of the works in model compression focus on clean test-set accuracy ignoring model robustness, such as out-of-distribution (o.o.d.) accuracy, which is crucial for systems operating in the real world.

Model compression is also a very interesting problem from a research perspective. It is well-known that current machine learning models are heavily over-parameterized, which allows them to easily fit random labels [12]. This over-parameterization is exploited by compression techniques which greatly reduce the model size with only a small decrease in accuracy. But, investigating only the mean accuracy might not give the full picture of compression methods' impact on model predictions. Highly accurate models (in terms of average precision) can still fail in rare and atypical cases [13], [14].

It was only recently shown that pruning significantly affects robustness in the image classification task and might disproportionately impact different object classes [15], [16]. In our work, we start with those observations and apply them to the task of object detection from RGB images. Further, we test the effect of naturalistic data augmentation on compressed models. We focus on autonomous driving datasets, as both model robustness and computational efficiency are of great importance for such an application. It was demonstrated that using test accuracy alone might not give the complete picture of the model compression impact. Measuring out-of-distribution performance or per-class accuracy is crucial in safety-critical applications. The contribution of this paper are as follows:

- First, both structured and unstructured compression techniques are evaluated on object detection tasks. The further effect of adding texture invariant data augmentation was measured. It was shown that such an intervention has a positive effect on model robustness, showing that highly compressed models, in spite of their limited capacity, are able to build more texture-invariant object representation.
- When evaluating model robustness on synthetic distributional shift (adding different types of distortions to the images), it was shown that compressed models' sensitivity is remarkably varied between different distortion types and some of them are only slightly affected by the

This work has been partially supported by Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdańsk University of Technology. This work was supported in part by the Polish National Centre for Research and Development (NCBR) through the European Regional Development Fund entitled: INFOLIGHT Cloud-Based Lighting System for Smart Cities under Grant POIR.04.01.04/2019.

compression.

- It was shown that compression techniques have a disproportionate impact on different classes. To reduce that effect, several class-balancing techniques are evaluated which significantly improve accuracy for many classes, and also improve mean average precision. Noticeably, for structured pruning, the positive effect of using methods for handling data imbalance is the most striking at higher compression rates.

II. Related work

Model compression. A significant number of methods have been proposed for reducing the computational footprint of neural networks by reducing the model size. The most popular approach is magnitude pruning which removes a number of small magnitude weights resulting in only a small decrease in accuracy [4], [17]. However, in order to actually reduce the computational cost of such pruned models, specialized hardware is required which optimizes sparse operations. As a result, structured pruning was proposed where entire filters and/or layers are removed [18]–[20]. Standard approaches to model compression assume training the base model, pruning and then a fine-tuning stage [4] or gradually pruning the model during training [17], [20]. For the task of visual recognition, model compression has mostly been applied to the image classification, with very few works on the task of object detection [21]. As object detection is a more complex task than image classification, and also has significant application potential, it is important to evaluate model compression methods in the object detection task.

Robustness. Current CNN-based models show impressive performance when the test data comes from a similar data distribution to the training data, but fails to generalise when this assumption does not hold [22], which is a significant challenge when deploying machine learning models to the real world. To improve model robustness, several methods have been proposed. They include using data augmentation techniques such as style-transfer [23], [24], noise injection [25], naturalistic augmentation (color distortion, noise, and blur) [26], and interpolating between different images [27]. It has also been shown that large models improve robustness [7], [28], however our goal is orthogonal, we studied whether small models can also be robust.

On the evaluation side, evaluating the models by adding synthetic distortions during test time (e.g., noise, blur, changes in contrast) [7], using test data coming from a different distribution, and by testing models on natural transitions (for example from day to nighttime images) [29] have been proposed. Still, measuring the effect of compression on model robustness remains an unstudied problem. It was shown that it is possible to optimise for both adversarial robustness and model size at the same time [30], [31]. A parallel work to ours also investigates effects of model compression in out-of-distribution setting and confirms that such testing is critical in the context of safety-critical systems [32]. In our work, we focus on the object detection task and measure its impact on both synthetic and real distributional shifts, and additionally on per-class accuracy.

Class imbalance. Real-world datasets often follow a long-tail distribution: a few dominant classes are represented by a great number of examples, significantly higher than of other less represented classes. Models trained on such datasets provide poor accuracy on the underrepresented classes [33]. Significant research exists on dealing with such data imbalance which can be categorised into two groups: re-sampling and cost-sensitive learning. In re-sampling strategies, some of the training examples for the minority classes are repeated [34] or examples from dominating classes are undersampled). Cost-sensitive learning deals with the problem by assigning a relatively higher cost to the minority classes, e.g., computing the loss using the inverse of the class frequencies [35] or the inverse of the effective number of samples [36].

At the same time, the effect of model compression techniques on certain classes remains largely unexplored. Indeed, a recent work suggests that some classes may be more impacted by compression techniques than others [15]. As such, we decided to evaluate the effect of model compression techniques on different classes in the safety-sensitive domain of autonomous driving.

III. Methodology

A. Object detection

A goal of object detection is to find where object are located in the image (object localization) and to which class they belong (object classification). Faster R-CNN [37] is a popular algorithm in object detection that works in two stages: regions of interest selection with Region Proposal Network (RPN) is followed by a regions classification into one of the classes $c \in \{1, ..., C\}$. Both stages share a common set of convolutional layers, a so-called a backbone network. RPN outputs a list of anchors (bounding boxes) which are likely to contain an object, and each region proposal is processed by the classification layer, which computes a logit vector $z \in R^c$ for each region. Finally, a sigmoid function is applied $p = sigmoid(z)$, to obtain a list of predicted class probabilities and class with the highest probability is used as predicted class for given region proposal. Whole model is trained by optimizing multi-task loss function which consists of cross-entropy loss for classification task and L1 smooth loss for bounding boxes localizations regression.

B. Model compression

In our work, standard magnitude pruning approaches were utilized. While, more advanced approaches exist, magnitude pruning has been shown to consistently achieve very good results across a number of datasets and tasks [38]. Another advantage is that magnitude pruning is a very general method than can be applied to a wide range of tasks and architectures. During training, the automatic gradual pruning technique is used which progressively increases the sparsity in the network over the course of the training up to the desired compression

Fig. 1: Examples of augmented images: color drop (top left image), color distortion (top right), overexposed image (bottom left), gaussian noise (bottom right).

rate. Specifically, the sparsity s_t at epoch t is computed as: [17]

$$s_t = s_f + (s_i - sf) * (1 - \frac{t - t_0}{n\Delta t})^3 \text{ for } t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\} \quad (1)$$

where n is the number of pruning steps, Δt is the pruning frequency, s_f is a final sparsity value, s_i an initial sparsity value (usually 0) and t_0 is an epoch at which pruning starts. At each iteration L_1-norm is computed for each tensor and tensors with the lowest norm are zeroized, such that the desired level of sparsity s_t at given epoch is achieved. Similarly, for structured pruning the L_1-norm is computed at the filter level, which weights are set to 0.

C. Data augmentation

Several data augmentation techniques have been proposed to improve model robustness, in particular style-transfer data augmentation is quite often used [23], [24], [29]. However, the computation itself is quite costly and one has to decide which data to use as the source of the style. On the contrary, recent work has shown that adding simple data augmentation such as color distortion, noise, and blur can also be a very efficient strategy to improve model robustness [26]. As such a procedure is very simple and very efficient, it was used in our work. Namely, during training, the following augmentation is used in the pipeline:

- Color distortion with a probability of 50%. This includes changes in the brightness, contrast, saturation and hue of the image as specified in [39].
- Color drop (grayscale image) with a probability of 20%.
- Gaussian blur with a probability of 50%.
- Gaussian noise with a probability of 50%.

D. Imbalanced data

In this subsection, a few techniques for handling data-imbalance are described, the first technique being based on sampling and the others based on cost-sensitive learning.

The **repeat factor sampling (RFS)** strategy was recently shown to yield competitive results on class imbalance problems [40]. For each category c, let's define f_c as a portion of images that contain at least one instance of object category c. The category-level repeat factor is defined as:

$$r_c = max(1, \sqrt{(t/f_c)}) \quad (2)$$

where t is a hyperparameter. Intuitively, this means that categories which frequency f_c is below threshold t, will be over-sampled. Then the image-level repeat factor is computed as the maximum value over the categories in the image i:

$$r_i = \max_{c \in i} r_c \quad (3)$$

Cost sensitive learning, on the other hand applies class-specific weights w_c to the cross entropy loss in the classification task. For a given observation, the weighted cross entropy can be computed as:

$$L_{wCE} = -\sum_{c=1}^{C} w_c * y_c log(\hat{y_c}) \quad (4)$$

where $y_c \in \{0, 1\}$ indicates whether class c is the correct class for given observation and $\hat{y_c}$ is a predicted probability for class c, and w_c is a weighting factor for every class. If $w_c = 1$ for all classes then the above formulation relates to the standard cross entropy loss. Below, different approaches to computing w_c for data imbalance problems are briefly described.

Inverse square root of class frequency computes the w_c in exactly the same way as the repeat factor r_c was computed for the RFS algorithm. For our experiments, another variant was also tested where the weights were computed as $w_c = \sqrt{(t/f_c)}$ (so removing the max function), which allowed the weights of some frequent classes to be smaller than 1.

Computing class weights by means of a category-level repeat factor, as defined above, may yield suboptimal results, since some of the images may contain just one instance of a given category, while others may contain dozens of them. As such, it was proposed in [36] to compute the weighting factors using the number of instances. Our implementation follows the details provided in [41]. First, the number of instances for each category N_c is computed. Then, the **effective number of samples** E_n for each category c can be computed as:

$$E_n = \frac{1 - \beta^{Nc}}{1 - \beta} \quad (5)$$

The final class weights are obtained by taking inverse of the E_n and applying a normalisation term.

However, note that the above methods have mostly been tested on the image classification task, and object detection brings further challenges. First, object detection has a multi-task objective and scaling classification loss may introduce side effects to the overall performance (for example by changing the accuracy of the regional proposal network). Second, the above calculation does not take into account the background class (because it is hard to estimate the "frequency" of the background class, a class-weight of 1 is

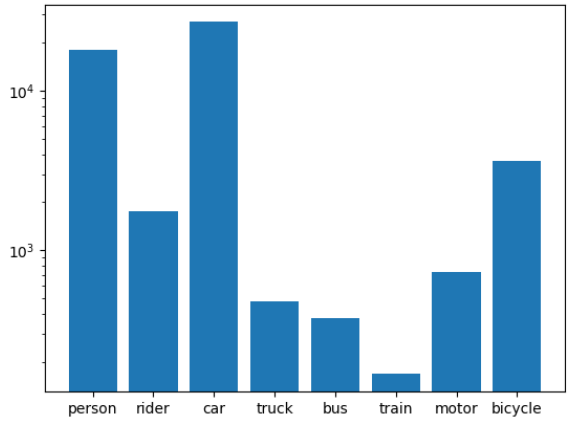

Fig. 2: Cityscapes dataset class histogram (logarithmic scale).

applied to the background class in all cases as in [41]). Since foreground/background separation is also a very important part of the object detection, one has to be very careful when applying different class balancing methods. As such, in the experiments section, experiments are also conducted with linearly scaled variants of the above methods.

IV. Experiments

A. Datasets

Cityscapes [42] is a large-scale autonomous driving dataset for semantic segmentation and object detection (Fig. 2 shows class histogram). It contains 5000 images of street scenes recorded in 27 cities, mostly in Germany. However, a potential limitation is the fact that the Cityscapes datasets were mostly recorded during the daytime in good weather conditions. As such, more challenging datasets are being developed. EuroCity Persons (ECP) [43] contains 47,300 images recorded in 31 cities in 12 European countries. Additionally, data were recorded during all seasons in different weather conditions. A significant subset of images was recorded during the nighttime. This allows us to evaluate model robustness on a day to night transition (when the model was trained using daytime images and evaluated at nighttime). Finally, Berkeley Deep Drive (BDD) dataset [44] was used as it is one of the most diverse datasets for object detection in autonomous driving.

However, even the biggest datasets cannot account for all different conditions that may occur in the real world, e.g., bad illumination conditions, adverse weather conditions, sensor noise, or a mixture of these. As such, using simulated distortions is often used as an additional proxy to evaluate model robustness. The common corruptions benchmark [7] is a great example of such an approach, which contains procedures to generate synthetic distortions, which are applied during model evaluation. In total, 15 distortion types can be generated, which are grouped into 4 categories: noise (Gaussian noise, shot noise, impulse noise, salt-and-pepper noise), blur (defocus blur, frosted glass blur, motion blur, zoom blur), weather corruptions (snow, fog, brightness, contrast) and digital noise (elastic transformations, pixelation, JPEG lossy compression). Each corruption has 5 levels of intensity. For simplicity, in our evaluation, distortions are applied at the medium intensity level.

B. Implementation details

The models were trained using the Faster R-CNN general purpose object detector. The Distiller package [45] was used for pruning, using both structured and unstructured methods. Similar to [23], [29], the Cityscapes model was trained for 64 epochs, with a learning rate step reduction by factor of 10 at epoch 48. Initial learning rate was 0.01 and the batch size was 6 as this is the maximum that the GPU used is able to concurrently process. The pruning used the automated gradual pruning scheme [17] starting from the first epoch until epoch 56.

For ECP and BDD datasets the model was trained for 11 epochs, with a learning rate step reduction by factor of 10 at epoch 7. Initial learning rate was 0.01 and the batch size was also 6. The pruning was gradual starting from the first epoch until epoch 8.

The models were pruned at 30%, 50%, and 70% compression rates for the structured pruning and at 50%, 80% and 95% compression rates for the unstructured pruning. For each method, all of the compression rates can be considered to be a reasonable setup, with the first compression rates being more conservative and the last being more aggressive. Note, that for the unstructured pruning, higher compression rates can be achieved, which is why the compression rates were higher in that setting. Above models were trained 5 times, and the mean accuracy is reported.

C. Measuring impact of model compression on the robustness

Table I presents the results obtained for the models trained on the Cityscapes dataset using different compression strategies and evaluated on the clean Cityscapes dataset and its corrupted versions. The results from the second to the last column measure the robustness of the models (o.o.d. test). The first thing that can be noticed is that the models clearly lack robustness and is very vulnerable to different kinds of distortions, as the mAP metric is very low for all distortion types. Further, for the structured pruning, it was possible to prune 30% of the filters and still achieve the same accuracy on the clean dataset (first column), however models' sensitivity to different distortion types was already negatively affected.

While the previous experiment measured robustness to some synthetically generated distortions, using the ECP dataset, one can measure the robustness to natural distortion such as the transition from day to night (Table II). Specifically, a model trained on daytime images is evaluated on daytime images (first column) and also on nighttime images (second column, o.o.d. test). The decrease in mAP metric is comparable for both tests, when compared to the baseline model, for both pruning methods and for both evaluation (ECP-day and ECP-night).

TABLE I: Accuracy comparison for models trained with different pruning strategies tested on the Cityscapes dataset (first column) and different corruption types from the Common Corruptions benchmark (the remaining columns).

Model	Clean	Noise	Blur	Weather	Digital
Baseline	0.352	0.0	0.049	0.152	0.146
Unstructured pruning (compression rate)					
50%	0.351	0.0	0.047	0.151	0.14
80%	0.338	0.0	0.041	0.138	0.135
95%	0.323	0.0	0.029	0.115	0.118
Structured pruning (compression rate)					
30%	0.352	0.0	0.037	0.134	0.135
50%	0.337	0.0	0.027	0.105	0.131
70%	0.33	0.0	0.023	0.088	0.125

TABLE II: Accuracy comparison for models trained using daytime and tested on daytime images (first column) and nighttime images (second column).

Model name	ECP-day	ECP-night
Baseline	0.468	0.392
Unstructured pruning (compr. rate)		
50%	0.462	0.396
80%	0.45	0.383
95%	0.414	0.331
Structured pruning (compr. rate)		
30%	0.457	0.382
50%	0.444	0.363
70%	0.431	0.34

D. Naturalistic data augmentation

A standard way to improve model robustness is using specialized data augmentation, it is however unclear what the effect of such augmentation will be on compressed models, especially at the highest compression rates. In this section, the models' robustness was again evaluated, but this time a naturalistic data augmentation' [26] was used during training. The results are presented in Table III and Table IV. Overall, one can see that, as expected, the out-of-distribution detection accuracy has greatly increased for both datasets for all models. For example, looking at the structurally pruned model at the 50% compression rate, one can see that the accuracy on the distortions unseen during training has greatly increased (0.243 and 0.251 mAP for weather and digital distortions compared to 0.105 and 0.131 mAP, respectively). Also, the accuracy on the clean dataset (first column) has significantly increased for all models, with the only exception of structurally pruned model at the highest compression rate, where the accuracy has slightly decreased. Interestingly, the effect of using naturalistic data augmentation is smaller at the highest compression rates.

On the ECP dataset, the loss in accuracy on daytime images was significant (0.15 and 0.13 at the highest compression rates for unstructured and structured pruning, however this might be because a similar decrease can be noticed for the uncompressed model (decrease in mAP from 0.468 to 0.456). This shows that one has to be careful when setting the data augmentation parameters, probably using less aggressive augmentation on the ECP dataset would improve the results for daytime images. Nevertheless, the results for the nighttime images greatly improved at all compression rates. For example, for the model structurally pruned at the 50% compression rate, after using naturalistic data augmentation, the mAP on the nighttime images increased from 0.363 to 0.393. This shows that, in spite of limited capacity, the compressed models were still able to learn more texture-invariant representation of the objects.

It is also worth looking at how the dynamics of change in accuracy for specific corruptions are affected, as the compression rate is increased (Fig. 3). A few, very interesting observations can be made. First, the accuracy for each corruption type was differently impacted by the pruning. The models' sensitivity to noise was the most heavily impacted by model pruning. While the initial accuracy was fair (0.194 mAP without any compression), the accuracy started to deteriorate very quickly when more than 30% of the filters were pruned. On the other hand, the accuracy for the blur distortions was almost flat, being only slightly reduced at the highest compression rates. Digital and weather distortions were similarly impacted by model compression, comparably to the performance of the original Cityscapes dataset. Relating the results to other work [8], it is worth noting that different distortions had different Fourier statistics. Some of them (i.e., shot and impulse noise) were concentrated in the high-frequency components of the

TABLE III: Accuracy comparison for models trained using naturalistic data augmentation with different pruning strategies tested on the Cityscapes dataset and corruption types from the Common Corruptions benchmark, when using naturalistic data augmentation. Values in brackets show accuracy change due to the added augmentation.

Name	Clean	Noise	Blur	Weather	Digital
Baseline	0.367 (+0.015)	0.194	0.126	0.258	0.271
Unstructured pruning (compr. rate)					
50%	0.364 (+0.013)	0.193	0.127	0.258	0.264
80%	0.359 (+0.021)	0.18	0.125	0.255	0.256
95%	0.326 (+0.003)	0.064	0.112	0.226	0.233
Structured pruning (compr. rate)					
30%	0.36 (+0.008)	0.178	0.122	0.252	0.252
50%	0.352 (+0.015)	0.154	0.122	0.243	0.251
70%	0.324 (-0.006)	0.103	0.113	0.221	0.232

TABLE IV: Accuracy comparison for models trained using naturalistic data augmentation on daytime images and tested on daytime images (first column) and nighttime images (second column), when using naturalistic data augmentation. Values in brackets show accuracy change due to the added augmentation.

Model name	ECP-day	ECP-night
Baseline	0.456 (-0.012)	0.419 (+0.027)
Unstructured pruning (compr. rate)		
50%	0.453 (-0.009)	0.417 (+0.023)
80%	0.444 (-0.006)	0.407 (+0.024)
95%	0.399 (-0.015)	0.363 (+0.032)
Structured pruning (compr. rate)		
30%	0.447 (-0.01)	0.407 (+0.025)
50%	0.433 (-0.011)	0.393 (+0.03)
70%	0.418 (-0.013)	0.381 (+0.041)

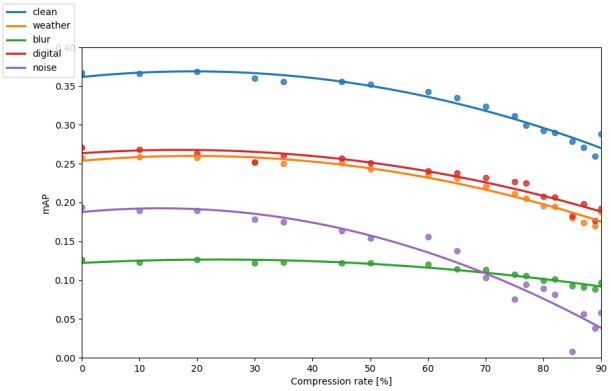

Fig. 3: Effect of structured pruning with different compression rates (x axis) across different distortion types on a mAP metric, for a model using naturalistic data augmentation.

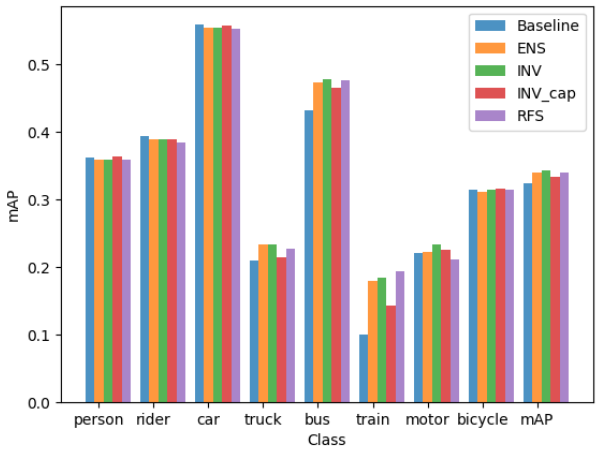

Fig. 4: Per class accuracy for models structurally pruned at the 70% compression rate using different class-balancing strategies.

image, while others (e.g., brightness, contrast) were concentrated in the low-frequency components. This might mean that pruning the visual recognition changes the models' sensitivity to the high- and low-frequency components of the image.

E. Per class evaluation

In this section, instead of observing only the mean accuracy of the model, a per-class accuracy is also examined to see how different classes were impacted by the compression. It is important, since observing effects of compression using mean accuracy alone, may be insufficient [16]. In this section the experiments were conducted on Cityscapes and BDD datasets, as they provide ground-truth for many classes. It is clear that different classes were disproportionately affected by the compression techniques (Table V). Some classes were heavily impacted by the compression (e.g., truck, train, bus) while others were less affected (i.e, car). There are many factors which influence the final impact. One of those is the class imbalance (Fig. 2, i.e., car class is dominant in both datasets), but some classes were also inherently harder than others (because they were similar to other classes, occured with high occlusion rates, or were hard to distinguish from the background).

As some classes seems to be more impacted than others, we have conducted experiments using methods for imbalanced datasets, namely:

- Repeat factor sampling (RFS)
- Inverse squared class frequency re-weighting with (INV_{cap}) and without (INV) setting the minimal weight to be 1.0 (as described in sec. III-D)
- Effective number of samples (ENS)

For the weighting methods, we also experimented with the linear variants of the above methods using scaling factor $\lambda \in \{0.5, 1, 2.\}$ and results are reported for the best performing scale.

Overall, very interesting results were obtained. The best performing method utilized inverse class frequency re-weighting. Interestingly, while the effect of data imbalance was relatively small without any compression (mAP increased from 0.367 to 0.374 on Cityscapes, similarly on BDD, Table V), the effect was much more striking at the highest compression rates, for the structurally pruned models. At the 70% compression rate level, the accuracy significantly increased from 0.324 to 0.343 on Cityscapes and from 0.228 to 0.246 on BDD dataset. As a sanity check, models were also tested at the 75% and 80% compression rates, confirming those results - the overall accuracy increased by around 0.02 mAP in both cases. For the unstructured pruning, the above finding was not observed. It might occur because structured pruning is a harder problem, and in the case of model pruned with unstructured method, it might be easier to accommodate for different classes.

Fig. 4 compares different data balancing methods on Cityscapes dataset. It can be noticed that for some classes (i.e., train, truck), the accuracy greatly increased after data balancing was applied, while on others, the accuracy remained almost the same. In general, all of the methods brought improvement to the compressed model (i.e, for INV method train accuracy increased from 0.1 to 0.184 and bus accuracy increased from 0.432 to 0.478). Recent work studied models performance of the minority groups and show that the overparameterized models seem to learn patterns that generalize well on the majority groups, but do not work well on the underrepresented classes [46]. Our work, on the other hand studies per class accuracy on the real-world dataset in low-capacity models, and showed that different data balancing methods can be very effective (for structurally pruned models).

Fig. 5 shows some detection examples. In general, the compressed model detects well visible objects in the image, however, the occluded objects might not be detected (the first column, missed motorcycle detection). Additionally, pruned models are much more sensitive to the noise distortion and might include more false-positive detections.

TABLE V: Per class accuracy of trained models. Aug stands for the naturalistic data augmentation and INV for the inverse class frequency re-weighting method. For the unstructured pruning, using data balancing methods bring similar gain across different compression rates, here only the accuracy at the highest compression rate is reported.

Name	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mAP
Cityscapes									
Baseline + aug	0.39	0.404	0.577	0.265	0.495	0.222	0.256	0.329	0.367
Baseline + aug + INV	0.39	0.407	0.576	0.283	0.512	0.229	0.263	0.332	0.374
Unstructured (95%) + aug	0.359	0.38	0.552	0.205	0.454	0.16	0.228	0.312	0.331
Unstructured (95%) + aug + INV	0.354	0.378	0.546	0.221	0.466	0.186	0.233	0.31	0.337
Structured (30%) + aug	0.385	0.409	0.576	0.249	0.497	0.186	0.246	0.333	0.36
Structured (30%) + aug + INV	0.384	0.406	0.572	0.257	0.512	0.222	0.26	0.331	0.368
Structured (50%) + aug	0.378	0.402	0.57	0.251	0.473	0.176	0.236	0.332	0.352
Structured (50%) + aug + INV	0.379	0.399	0.567	0.256	0.502	0.214	0.245	0.331	0.362
Structured (70%) + aug	0.362	0.394	0.559	0.21	0.432	0.1	0.221	0.314	0.324
Structured (70%) + aug + INV	0.36	0.389	0.555	0.233	0.478	0.184	0.233	0.314	0.343
BDD									
Baseline + aug	0.318	0.256	0.408	0.392	0.417	0.0	0.218	0.221	0.279
Baseline + aug + INV	0.317	0.26	0.404	0.396	0.428	0.031	0.232	0.222	0.286
Structured (70%) + aug	0.266	0.193	0.388	0.322	0.339	0.0	0.150	0.163	0.228
Structured (70%) + aug + INV	0.276	0.222	0.389	0.35	0.369	0.0	0.179	0.185	0.246

Fig. 5: Detection samples for base model and the model structurally pruned at 70% compression rate. Detections on the original Cityscapes dataset (first column), ECP dataset (second column) and noise distortion (third column).

V. Conclusions

In this paper, it was shown that, despite limited capacity, compressed models could make effective use of naturalistic data augmentation to learn more texture-invariant representations, which significantly increased model robustness to synthetic distortions and day to night transition. It was found that model compression differently affects models' sensitivity to different distortion types. Some of them, i.e., those concentrated in the high-frequency domain such as Gaussian noise, were heavily affected by pruning techniques, while others (blur distortions), were only slightly affected.

In particular, it was demonstrated that data balancing methods might be especially useful in structurally pruned neural networks. Without any compression applied, using inverse class frequency re-weighting increased the overall mAP by 0.007 (1.9% relative increase). On Cityscapes dataset, at the 70% compression rate, in the case of structured pruning, the mAP increased by 0.019 (5.9% relative increase). Similar results were obtained for the BDD dataset. Both sampling-based methods (repeat factor sampling) and cost-sensitive methods (i.e, inverse squared class frequency re-weighting) turned out to be effective.

Overall, our work explores the relation between models' robustness and the model compression techniques and provides insights on improving both performance and computational cost of deployed models. It was shown that for safety-critical systems, testing compressed models in out-of-distribution setting or measuring per class accuracy, is important to fully understand effects of model pruning. A natural extension of our work would be extending our experiments with quantization techniques, which are also used for reducing computational cost of machine learning models. As a future work, it would be also worth exploring effect of model compression on fine-grained "subclasses", similar as in [14].

References

[1] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018.

[2] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 2704–2713, 2018.

[3] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2, NIPS*, pp. 598–605, 1989.

[4] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, vol. 28, pp. 1135–1143, 2015.

[5] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[7] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *7th International Conference on Learning Representations, ICLR*, 2019.

[8] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, "A fourier perspective on model robustness in computer vision," in *Advances in Neural Information Processing Systems*, pp. 13255–13265, 2019.

[9] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 1802–1811, 2019.

[10] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?," *J. Mach. Learn. Res.*, vol. 20, pp. 184:1–184:25, 2019.

[11] W. Khan, A. Hussain, K. Kuru, and H. Al-Askar, "Pupil localisation and eye centre estimation using machine learning and computer vision," *Sensors*, vol. 20, no. 13, p. 3785, 2020.

[12] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *5th International Conference on Learning Representations, ICLR*, 2017.

[13] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks," in *8th International Conference on Learning Representations, ICLR*, 2020.

[14] N. S. Sohoni, J. Dunnmon, G. Angus, A. Gu, and C. Ré, "No subclass left behind: Fine-grained robustness in coarse-grained classification problems," in *Advances in Neural Information Processing Systems 33, NeurIPS*, 2020.

[15] S. Hooker, A. Courville, G. Clark, Y. Dauphin, and A. Frome, "What do compressed deep neural networks forget? arxiv e-prints, art," *arXiv preprint arXiv:1911.05248*, 2019.

[16] R. Entezari and O. Saukh, "Class-dependent compression of deep neural networks," *arXiv preprint arXiv:1909.10364*, 2020.

[17] M. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," in *6th International Conference on Learning Representations, ICLR*, 2018.

[18] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems 29*, pp. 2074–2082, 2016.

[19] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *5th International Conference on Learning Representations, ICLR*, 2017.

[20] C. H. Tu, J. H. Lee, Y. M. Chan, and C. S. Chen, "Pruning depthwise separable convolutions for mobilenet compression," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[21] G. Tzelepis, A. Asif, S. Baci, S. Cavdar, and E. E. Aksoy, "Deep neural network compression for image classification and object detection," in *18th IEEE International Conference On Machine Learning And Applications, ICMLA*, pp. 1621–1628, 2019.

[22] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?," in *Proceedings of the 36th International Conference on Machine Learning, ICML*, vol. 97, pp. 5389–5400, 2019.

[23] S. Cygert and A. Czyżewski, "Toward robust pedestrian detection with data augmentation," *IEEE Access*, vol. 8, pp. 136674–136683, 2020.

[24] R. Geirhos *et al.*, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *7th International Conference on Learning Representations, ICLR*, 2019.

[25] J. Gilmer, N. Ford, N. Carlini, and E. Cubuk, "Adversarial examples are a natural consequence of test error in noise," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 2280–2289, 2019.

[26] K. L. Hermann, T. Chen, and S. Kornblith, "The origins and prevalence of texture bias in convolutional neural networks," in *Advances in Neural Information Processing Systems 33, NeurIPS*, 2020.

[27] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *8th International Conference on Learning Representations, ICLR*, 2020.

[28] C. Xie and A. L. Yuille, "Intriguing properties of adversarial training at scale," in *8th International Conference on Learning Representations, ICLR*, 2020.

[29] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," in *Machine Learning for Autonomous Driving Workshop, NeurIPS*, Jul 2019.

[30] V. Sehwag, S. Wang, P. Mittal, and S. Jana, "HYDRA: pruning adversarially robust neural networks," in *Advances in Neural Information Processing Systems 33*, 2020.

[31] S. Kundu, M. Nazemi, P. A. Beerel, and M. Pedram, "DNR: A tunable robust pruning framework through dynamic network rewiring of dnns," in *ASPDAC '21: 26th Asia and South Pacific Design Automation Conference*, pp. 344–350, 2021.

[32] L. Liebenwein, C. Baykal, B. Carter, D. Gifford, and D. Rus, "Lost in pruning: The effects of pruning neural networks beyond test accuracy," in *Proceedings of Machine Learning and Systems 2021, MLSys 2021*.

[33] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, p. 321–357, June 2002.

[35] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, IEEE Computer Society, 2016.

[36] Y. Cui, M. Jia, T. Lin, Y. Song, and S. J. Belongie, "Class-balanced loss based on effective number of samples," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 9268–9277, 2019.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[38] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," *CoRR*, vol. abs/1902.09574, 2019.

[39] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning, ICML*, vol. 119, pp. 1597–1607, 2020.

[40] A. Gupta, P. Dollár, and R. B. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 5356–5364, 2019.

[41] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng, "Overcoming classifier imbalance for long-tail object detection with balanced group softmax," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 10988–10997, 2020.

[42] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

[43] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "Eurocity persons: A novel benchmark for person detection in traffic scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1844–1861, 2019.

[44] F. Yu *et al.*, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2633–2642, IEEE, 2020.

[45] N. Zmora, G. Jacob, L. Zlotnik, B. Elharar, and G. Novik, "Neural network distiller: A python package for dnn compression research," 2019.

[46] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An investigation of why overparameterization exacerbates spurious correlations," in *Proceedings of the 37th International Conference on Machine Learning, ICML*, vol. 119, pp. 8346–8356, 2020.