Car Design Transfer with Generative Adversarial Networks

Nicolas Schäfer Anurag Das Samim Zahoor

Abstract

We perform image to image translation using Generative Adversarial Networks to translate an image of a car of one make e.g. *BMW to another make* e.g. *Mercedes. Our goal is to change design related features of a car (*e.g. *grill, logo) in an image, such that after applying the learned transformation, the car is recognizable as having the target make. At the same time we want to preserve aspects that are not related to the make of a car like body color, the license plate, background, reflections, people in the car etc. We first consider the baseline model which is the unmodified StarGAN architecture. We build on the baseline model by introducing data augmentations. We next consider how the number of domains and the resolution of the training images affects the learning task and the quality of the generated images. We report improvements and argue about the reasons for improvements. Finally we perform an evaluation of the generated images from our best model and report our findings.*

1. Introduction

Image to image translation refers to the task of changing a particular aspect of a given image. Significant improvements have been achieved in the image to image translation task. Recent unsupervised methods like [1] [2] have shown good performance for tasks like style transfer (*e.g*. image to painting, sketch to image), facial features and expression transfer. We further explore and study the performance of state of the art generative adversarial networks on a more challenging task. Our goal is to translate images of cars of one make *e.g*. BMW to a different make *e.g*. Mercedes. (simpler and reads better). We want to change design related features of a car (*e.g*. grill, logo) in an image, such that after translation, the car is recognizable as having the target make. At the same time we want to preserve aspects that are not related to the make of a car like color, the license plate, background, reflections, people in the car etc. This task has applications in photo editing and car design inspiration. We start with the unmodified StarGAN architecture [3] as the baseline (section 5.2), considering only images showing the front views of the cars and improve the results by introducing data augmentations (section 5.3). We next consider how the number of domains (section 5.4) and the resolution of the training images (section 5.5) affect the learning task and the quality of the generated images. We can efficiently reduce perturbations in color and structure and improve the sharpness of the generated images with these approaches. Our evaluation based on a pre-trained make classifier (section 6) reveals weaknesses in representing the target make in the output image which we address with future work mentioned in section 8.

2. Related Work

Image to image translation has received significant research attention and rapid advances have been made for this task. Most recent methods are based on Conditional Generative Adversarial Networks[4]. Pix2Pix [5] considers a supervised approach and requires pairwise data (x,y). It combines the Conditional GAN loss with an L_1 loss over the generated output and the target image y such that the generator learns to produce realistic images which are pixel wise similar to y. CycleGAN [2] introduces the Cycle Consistency Loss and overcomes the need for paired images. StarGAN [1] conditions the Generator on the input image and a target domain label by spatially replicating and appending the target label to the input image. This allows a single discriminator and generator to perform translations among multiple domains and thus provides a more scalable framework than CycleGAN.

3. Method

Our method is based on the StarGAN [1] architecture. Following the StarGAN terminology, we define a domain as the set of images of cars that have the same make. So, all images of BMWs is one domain, all images of Audis is another domain and so on. We train a single generator G that learns mappings among multiple domains. To achieve this, G is trained to translate an input image x to an output image y by being conditioned on the input image x and the target label c, $G(x, c) \rightarrow y$. Target labels are randomly generated at training time so that G learns to flexibly translate among all the domains. The conditioning on x and c is achieved by onehot encoding the target label c and then

spatially replicating and appending it to the input image x. We add an auxiliary classifier to the discriminator D, which produces probability distributions over domains. Thus, D produces probability distributions over sources (real/fake) and domains $D : x \rightarrow \{D_{src}(x), D_{cls}(x)\}$. The generated image $G(x, c)$ is translated back to its original domain c' giving $G(G(x, c), c')$ and this generated image is forced to be similar to the original image x using the Reconstruction Loss which is the L_1 norm of reconstructed image minus the original image *i.e.* $\mathcal{L}_{rec} = \|G(G(x, c), c') - x\|_1$. The reconstruction loss is added to the overall objective of the G and is weighted by a factor λ_{rec}. The weight factor governs how similar the generated image should be to the input image. The losses and the full objectives that we optimize are the same as original StarGAN [1] and their details are in the paper.

The StarGAN architecture scales well with the number of domains (only one discriminator and 1 generator is trained irrespective of the number of domains) as opposed to CycleGAN which requires 2 Generators for each possible combination of domain pairs. So StarGAN provides a practical approach for multi-domain image to image translation. We find that this is useful to easily observe the effect of number of domains on the quality of generated images. We discuss this further in the experiments section. Besides this, StarGAN also utilizes PatchGAN [5] which is shown to produce sharper and less noisy images. Being able to train the generator and discriminator on images of different domains simultaneously, extends the training set for real/fake classification for all domains. Furthermore, learned features can be shared across different classification tasks. We assume that in our case, where images of all domains share basic structures (*e.g.* all images show cars of the same view) this cross learning scenario comes with a beneficial generalization effect.

4. Datasets

The primary source of our data is the Comprehensive Cars Dataset [6]. The data set contains around 130,000 images of full cars. The images are labelled with the following information:

- Make, model and year of manufacture.
- Bounding boxes around the cars
- The viewpoint from which the image was captured.

We also include images from the Stanford Cars Dataset [7]. The dataset contains around 16,000 images and the labels provided are make, model, year of manufacture and bounding boxes around the cars. The viewpoint information is not provided so we manually filter images according to the viewpoints we need. We also collect some images publicly available on the internet and add them to our dataset.

5. Experiments

5.1. Data Preparation

- To simplify the learning task we only consider car images showing the front. Including more perspectives like the back view without additional constraints can theoretically lead to a transformation from a front view to a back view what is undesired.
- We crop the images to extended bounding boxes. This means, if necessary, we extend the given bounding boxes to match a 1:1 ratio and then crop to this extended bounding box. We do this to prevent distortion of the car when resizing.
  We crop the images to reduce scale variance and make the images structurally more similar, to simplify the learning task.
- We resize the images to 128x128 to reach feasible training time but still having sufficient quality. To minimize aliasing effects, we used Lanczos resampling for resizing.
- We remove cars with uncharacteristic design (*e.g.* concept cars), because we want to focus on characteristic design patterns. Having a concept car in the target domain means that having an output similar to this concept car is a valid output. That is not desired.

5.2. Baseline (Model 0)

For the baseline we choose the following domains (sample counts in braces):

- Mercedes (615)
- BMW (568)
- Audi (528)
- Skoda (215)
- Citroen (302)

We choose these because they belong to the domains with the highest sample counts and have comparable car shapes and not too dissimilar designs. Some SUVs for example differ fundamentally in design and shape from ordinary limousines. Experiments showed that transformations to such targets do not lead to convincing results without further adjustments (*e.g.* reducing reconstruction loss weight). We use the StarGAN architecture as it is with preset hyperparameters [3]

Results for this model are shown in Figure 1 and Figure 2 in the *first* row of each figure. We see that some basic design structures as logo and grill shapes are learned but

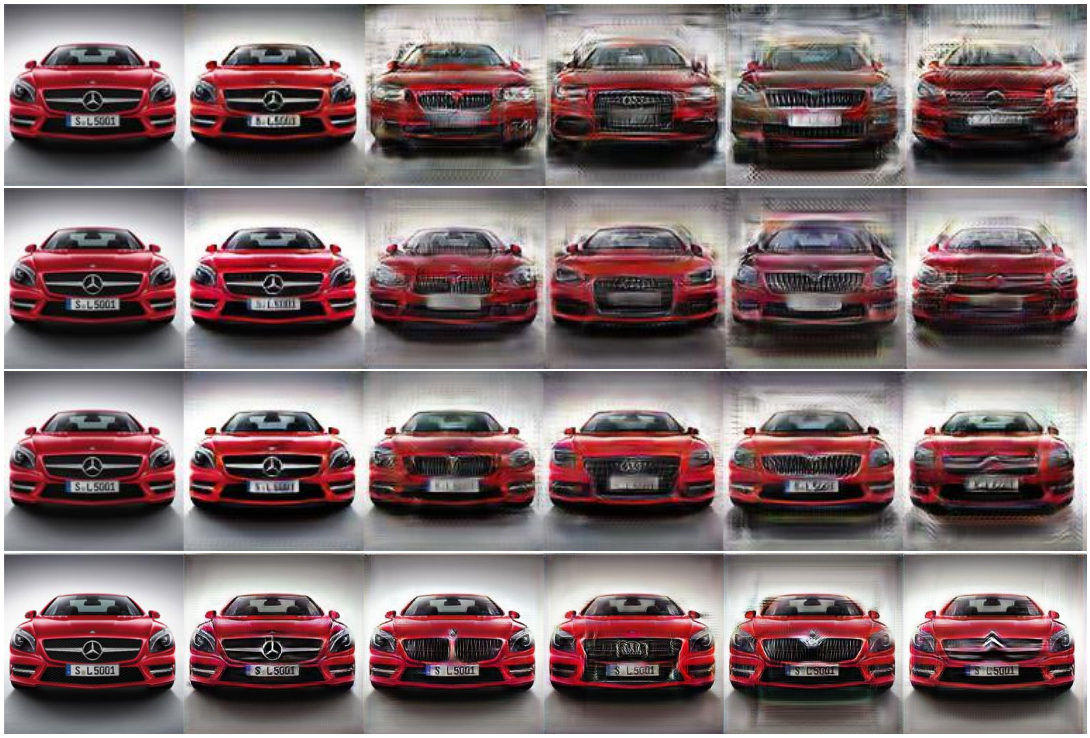

Figure 1. Showing outputs for a Mercedes test image for models 0 to 3. First image in each row is input.

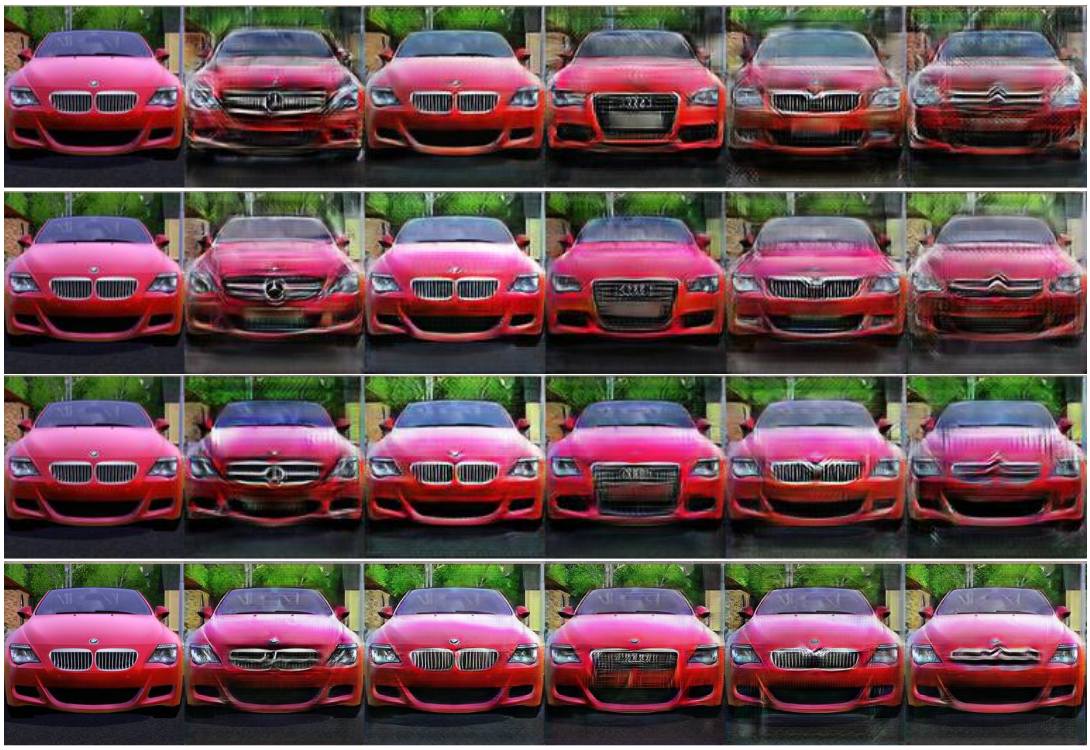

Figure 2. Showing outputs for a BMW test image for models 0 to 3. First image in each row is input.

the output images are dominated by perturbations in color and structure. We interpret these as overfitting effects (*e.g.* most cars have a darker background, so we see this as part of the output)

5.3. Data Augmentation (Model 1)

To address overfitting on color, background and reflections we use data augmentation.

- Color Jitter with changes in the interval of $[-20\%, +20\%]$ for brightness and saturation and $[-10\%, +10\%]$ for contrast and hue to reduce the color and brightness bias in the data set. Experiments showed that using heavier changes in color lead to undesired color shifts in the output.
- Horizontal Flip which flips the background and reflections but preserves the design of the symmetric car.

Results for this model are shown in Figure 1 and Figure 2 in the *second* row of each figure. We see less perturbations in color and structures, but they are still dominating the images.

5.4. Domain Set Extension (Model 2)

Since the real/fake classification task is learned across all domains, extending the domain set gives more samples to this task. This can have a generalization effect and reduce perturbations. Furthermore, the domain set extension can have a generalization effect on the domain classification task which can lead to both an improvement if the discriminator overfits on certain domains or domain samples or a deterioration if the discriminator already underfits because of lack of capacity. In our situation it is hard to asses over- or under-fitting for the domain classification task based on the output on model 1 (section 5.3). Images are dominated by perturbations which can be caused by overfitting of the real/fake classification and/or domain classification.
We add 1483 new samples (+83%) by adding three new domains:

- VW(754)
- Chevrolet(446)
- Volvo(283)

Using more domains did not give better results concerning color perturbations but lead to a worse representation of the target domain in the output, because introducing more domains extends the domain classification task. The discriminator with unchanged capacity tends to underfit.

Results for this model are shown in Figure 1 and Figure 2 in the *third* row of each figure. The additional domains are not shown but in the appendix more examples can be found containing all 8 domains. The results show significantly less perturbations in color and structure but in Figure 1 in the fourth image (Audi) the shape and some details in the spoiler are now nearer to the input image than before, which can be considered as a deterioration. This could be caused by generalization of the discriminator in terms of the the domain classification task.

5.5. Higher Resolution Input Images (Model 3)

To improve sharpness and high frequency details we increase the resolution of the input images from 128x128 to 256x256. This is motivated by two reasons.

- Down-sampling has a smoothing effect. So high frequencies are dumped. Going to a higher resolution reduces this effect. So higher frequencies are better preserved in the input image. The model has the chance to learn something about higher frequency structures.
- A higher resolution input image leads to a higher resolution patch grid for real/fake classification. StarGAN has a fully convolutional discriminator which last layer for real/fake classification has not one activation but several. One for a patch in the input (sub image). The amount of patches and the patch size is implicitly defined by the depth of the network. Because of a stride of 2 for each layer, deeper layers in the network have activation maps of smaller resolution. Because of a fully convolutional network, activations in the last layer implicitly correspond to a certain patch in the input image based on the receptive field. So higher resolution input images lead to a higher resolution activation map for the last layer if we do not change the architecture of the discriminator and thereby implicitly to more patches in the input image. Focusing on smaller local patches gives a more fine-granular fake/real prediction, better preserving high frequencies.

Results for this model are shown in Figure 1 and Figure 2 in the *fourth* row of each figure. We see a significant sharpness improvement and new high frequency details in the output (*e.g.* in Figure 1, third image, the BMW logo) which do not exist in detail in the 128x128 input image. The license plate is now significantly better readable in all outputs. But we see a worse representation of the target domain for Audi and Mercedes in Figure 2. The Audi grill is incomplete. This can be explained by a focus shift. The discriminator is now especially more demanding in terms of the fake/real loss. This implicitly dumps the importance of the other losses in the sum of losses for the generator. Another problem is that any influence on the discriminator can have an effect on both the domain classification task and the real/fake classification task and so on their performance, since both tasks are realized in one network and the

capacity is shared. We show how we want to address these problems in future work (section 8).

6. Evaluation

For now there does not exist a universal metric to evaluate generative models as shown in [8] and finding appropriate metrics is still an open issue. For evaluation we use a pre-trained classifier to measure quality properties of generated images as for example, in Zhang *et al*. [9] where a classifier was used to evaluate the performance of their style transfer approach using conditional GANs. Similarly, Radford *et al*. [10] used a classifier trained on the ImageNet dataset to evaluate their approach on representation learning via DCGAN.

We use a pre-trained classifier to evaluate if the generated image can be recognized as belonging to the target make. If the generated images capture the target make, as real images of the target make do, classifiers trained on real images should be able to classify the synthesized images correctly. We use the pre-trained classifier (10,000 training iteration snapshot) from github [11] which was originally used for fine grained car model and make classification on the CompCars dataset [6]. It was initially trained on the ImageNet dataset [12] and further fine tuned on the CompCars dataset. It obtained an accuracy of 82.9 % on the CompCars test data for all views. The pre-trained model[11] gives 431 posteriors for the existing 431 CompCars models which further can be used to get the corresponding make posterior by summing up the model posteriors corresponding to a given make. We use a publicly available classifier to give the metric more meaning by making our results comparable to potentially other work using the same metric.

What we do not evaluate are basic quality properties of the generated images (*e.g*. sharpness) and how well features that we consider as design unrelated are preserved in the output (*e.g*. background, color, reflections, basic shape of the car). Such aspects could be evaluated with a user study.

For building the test set we collected samples from the Stanford Car Dataset [13] and online sources. The test set size per make is shown in Table 1.

Make	Size
Mercedes	77
Audi	71
BMW	60
Citroen	55
Skoda	65

Table 1. Test set sample count per domain

Since the model gives posteriors corresponding to specific models, we sum up all posteriors corresponding to one make what gives the posteriors for makes. Table 2 shows accuracy for the prediction of make on real images and on generated images for model 3 (section 5.5). In the most cases the results show significantly worse accuracy for generated images. One reason for this could be the following. Model 3 (see section 5.5) especially focuses on preserving sharpness and reducing perturbations by the cost of worse representing fundamental design features of the target domain in comparison to model 2 (see section 5.5). Furthermore, we aim on transferring fundamental design features by preserving the basic shape of the car but not on fully embedding a car of the target domain into the input image (*e.g*. a car contained in an image of the target domain appears as being copied into the background of the input image is not the goal). The classifier was trained to predict *specific* models of makes. However, our generated images are supposed to be not similar to images of specific existing models. A possibility to address this is to train a classifier for the more general task of predicting the make, not specific models. In that way the classifier generalizes to the fundamental design aspects of makes rather than fitting to special properties of specific models. Furthermore, the result shows only 11.72% accuracy for BMW and 24.61 % accuracy for Skoda. Most of the generated and wrongly classified BMW cars are classified as Skoda and vice versa. This can be explained by the similarity of grill and the missing of a precisely represented logo in the generated image (see fourth row in Figure 1).

Make	Acc. (real data)	Acc. (generated data)
Mercedes	83.22 %	29.87 %
Audi	76.99 %	49.29 %
BMW	75.59 %	11.72 %
Citroen	87.03 %	38.18 %
Skoda	92.57 %	24.61 %

Table 2. Classification Accuracy per domain

7. Conclusion

Our aim was to transfer the car design from one domain to another by preserving design unrelated features (*e.g*. background, color) of the input image. Initially the output images were dominated by perturbations in shape and color. We could efficiently reduce these perturbation and furthermore significantly improve the sharpness with different approaches we showed. We have succeeded in transferring design related features (*e.g*. grill, logo) to a certain degree from one domain to another, but face weaknesses concerning the representation of the target domain in the output. We see barely changes to other structures then logo and grill (*e.g*. front spoiler). We mention how to address this in future work.

8. Future Work

To better control the performance of the domain classification, we plan to separate the current discriminator network into two separate discriminator networks, such that real/fake classification and domain classification are performed separately. In that way the capacity of the discriminator networks are not longer shared and taking influence on one task does not unintentionally influence the other. Together with adjustments of the loss weight for the classification loss for the generator network (λ_{cls}), we want to improve the representations of the target domain in the output. Currently we see for example incomplete grills for images with Audi as target domain.

Additionally, we want to adjust classification (λ_{cls}) and reconstruction loss (λ_{rec}) further to control the degree of the transformation. This means by how far the car in the input is adopted to the target domain. Currently we see mostly changes of logo and grill, but these could be extended to changes of front spoiler, lights etc.

Furthermore, our model is restricted to use front views only. We also plan to extend our approach to include multiple views (*e.g*. back view, side view).

To improve the evaluation of our approach we want to address several weaknesses. The classifier we use for predicting make does not fully capture the requirements of predicting make in a more generalized why as described in section 6. We want to address this by training an appropriate classifier focusing on make rather than on specific models. We do not evaluate basic quality properties of the generated images (*e.g*. sharpness) and how well features that we consider as design unrelated are preserved in the output (*e.g*. background, color, reflections, basic shape of the car). We want to evaluate these properties with a user study.

9. Reports to indicate assignments of each group member

There is no clear separation. We discussed most problems and approaches in the team. All team members contributed equally.

References

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[3] Yunjey Choi. Official pytorch implementation of stargan. `https://github.com/yunjey/stargan`, 2018.

[4] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[6] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. *CoRR*, abs/1506.08959, 2015.

[7] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[8] Ali Borji. Pros and cons of GAN evaluation measures. *CoRR*, abs/1802.03446, 2018.

[9] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[11] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Googlenet model for compcars dataset. `https://gist.github.com/bogger/b90eb88e31cd745525ae`, 2017.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

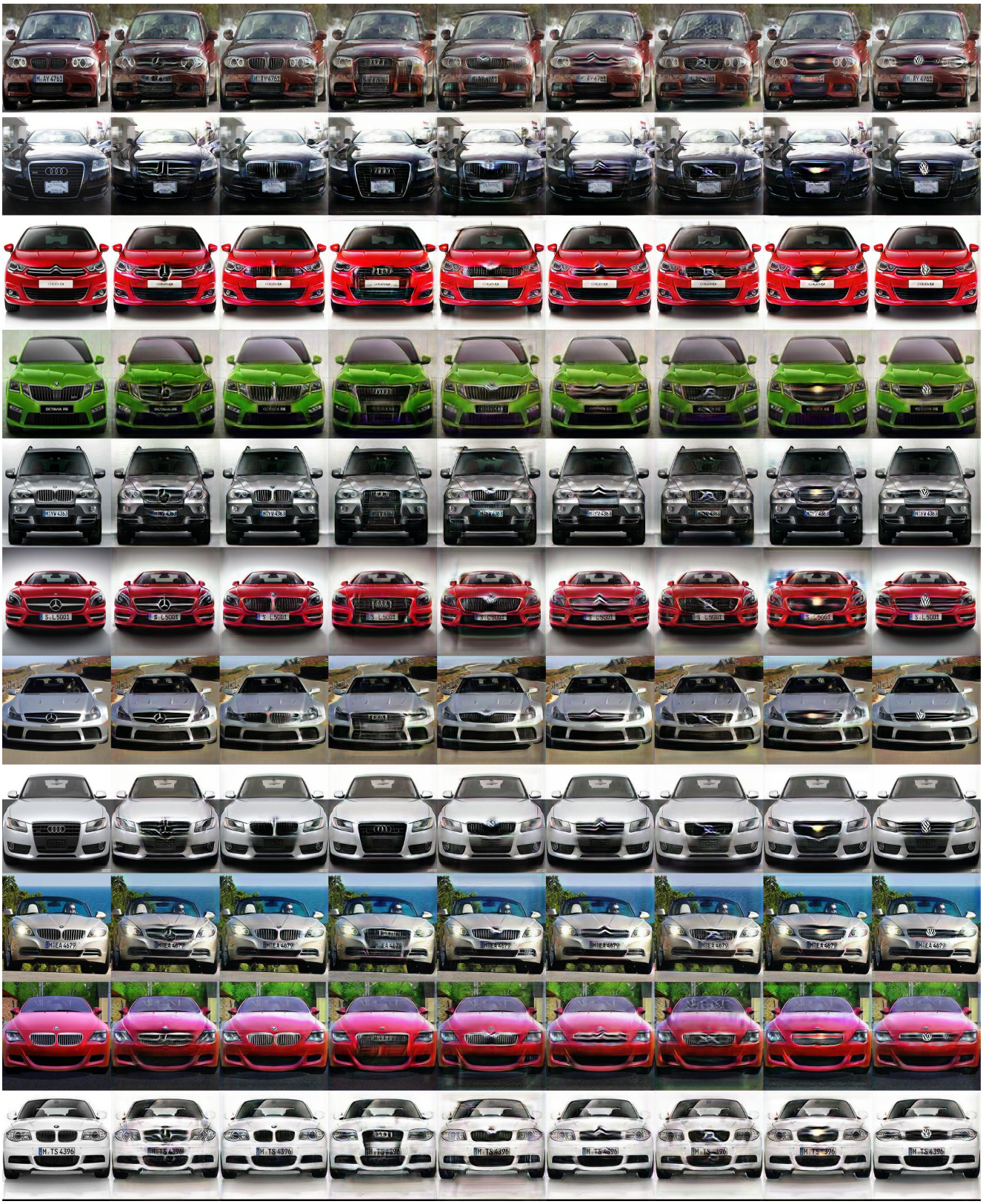

Figure 3. Outputs of model 3 for all 8 domains on test images.

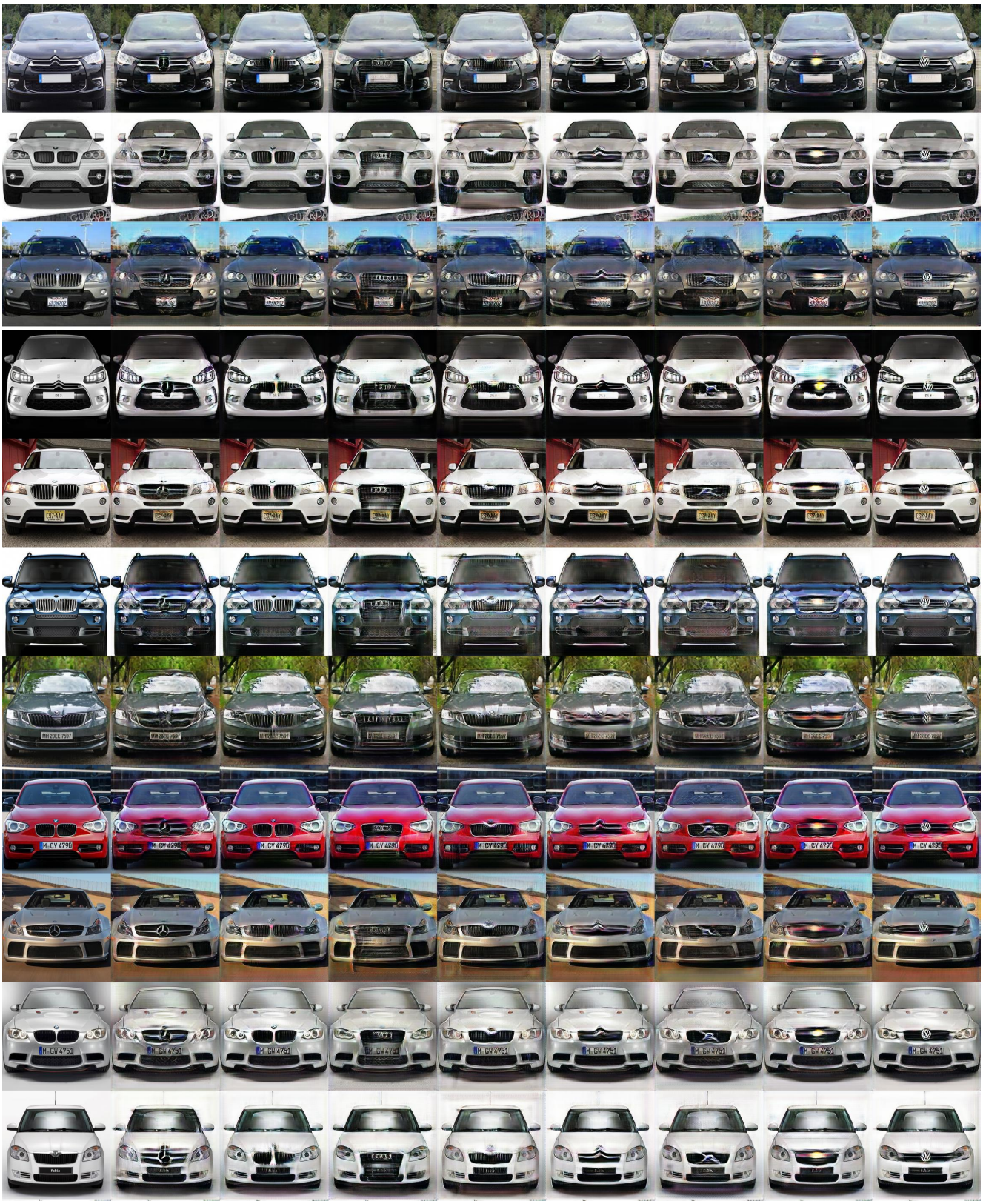

Figure 4. Outputs of model 3 for all 8 domains on test images.