# BRCars: a Dataset for Fine-Grained Classification of Car Images

Daniel M. Kuhn
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
daniel.kuhn@inf.ufrgs.br

Viviane P. Moreira
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
viviane@inf.ufrgs.br

***Abstract*—Fine-grained computer vision tasks refer to the ability of distinguishing objects that belong to the same parent class, differentiating themselves by subtle visual elements. Image classification in car models is considered a fine-grained classification task. In this work, we introduce BRCars, a dataset that seeks to replicate the main challenges inherent to the task of classifying car images in many practical applications. BRCars contains around 300K images collected from a Brazilian car advertising website. The images correspond to 52K car instances and are distributed among 427 different models. The images are both from the exterior and the interior of the cars and present an unbalanced distribution across the different models. In addition, they are characterized by a lack of standardization in terms of perspective. We adopted a semi-automated annotation pipeline with the help of the new CLIP neural network, which enabled distinguishing thousands of images among different perspectives using textual queries. Experiments with standard deep learning classifiers were performed to serve as baseline results for future work on this topic. BRCars dataset is available at https://github.com/danimtk/brcars-dataset.**

## I. Introduction

Supervised classification approaches typically require a large set of annotated data. This is especially true for Deep Learning (DL) techniques. In the context of computer vision, a convolutional neural network (CNN) is a DL algorithm widely used for image classification tasks. CNNs have constantly advanced the state-of-the-art in image classification tasks in datasets such as ImageNet [1] and PASCAL VOC [2].

Naturally, the success obtained in these general tasks led researchers to evaluate these image classification approaches in domain-specific classification tasks, such as classifying bird species [3], [4], plant diseases [5], aircraft models [3], [4], among others. These domain-specific tasks are called *fine-grained visual classification tasks* (FGVC).

Several datasets were built for the most diverse FGVC tasks. The Caltech-UCSD Birds dataset (CUB-200-2011) with 11,788 images from 200 wild bird species [6] and the FGVC-Aircraft dataset, with 10,000 images of aircrafts belonging to 100 models [7] are examples of datasets designed for FGVC tasks. In the domain of vehicles, the first dataset for FGVC was Cars by Krause *et al.* [8]. More recently, Yang *et al.* [9] created the CompCars dataset, which has more images and classes. Both these datasets have been designed for FGVC tasks. They are properly labeled and have undergone careful procedures to ensure data diversity. Their images were prepared in a way that a single car is shown in the center of the image. In the case of CompCars, which also has internal images among the images of specific parts of the vehicle, all images were obtained from the same perspective. Clearly, a controlled environment is a fundamental part of the evaluation of an experiment. However, real-world car images do not follow that pattern. It is common to find partial images, images of the interior of the vehicle, images of specific parts, such as the dashboard, seats, engine, wheels, *etc.* In addition, real images are taken from several different perspectives. In order to account for a more realistic environment, we created BRCars, a dataset with car images collected from a car advertising website, characterized by the lack of standardization with regards to the perspectives.

There are several applications that can benefit from the automatic classification of car models, including intelligent transport monitoring, surveillance, self-inspection for car insurance, and automatic ad verification for advertising websites. In all these applications, the images are not likely to be standardized. As a consequence, there is a need for datasets that contain this type of image.

Another important issue is that car models vary across countries. In some cases, the same model is identified by different names (*e.g.,* the Brazilian Kia Mohave is known as Borrego in the US) and, in other cases, the same name refers to different cars (*e.g.,* the Brazilian Fiat Fiorino is derived from the Uno model, while the Italian version is a minivan). These differences illustrate the need for datasets that are specific for each country.

The main contributions of this paper are: ($i$) BRCars, a *real world* dataset containing images of cars collected from a Brazilian car advertisement website; and ($ii$) experimental results of classification algorithms that can be used as a reference for future work on this topic.

## II. Related work

In this section, we discuss existing datasets of car images. Krause *et al.* [8], created the Cars dataset, which contains 16,185 images belonging to 196 classes, which refer to car models. Their images were collected from Flickr, Google, and Bing. With the intention of saving costs in the annotation process, as well as ensuring data diversity, the authors applied a deduplication procedure. The remaining candidate images were submitted to *Amazon Mechanical Turk* to determine

whether the images belong to the target class. The images were divided into training and test sets, with about 8K images each. The Cars dataset is characterized by having images from different external perspectives of the cars. The images have different dimensions and resolutions. Predominantly, the images contain a single car positioned at the center, with no other cars in the background.

Yang *et al.* [9] introduced the Comprehensive Cars (CompCars) dataset, which contains data from two scenarios: *web-nature* and *surveillance-nature*. The web-nature scenario has 136K images capturing the entire car and 27K images capturing specific parts of the cars. These images refer to 1,716 models from 163 makes. The images of the entire car are labeled with bounding boxes and viewpoints. The surveillance-nature group contains 50K images of cars captured in the frontal view. As reported by the authors, the CompCars dataset is well-prepared for the following computer vision tasks: ($i$) FGVC task: the dataset allows classification tasks to be performed at three different levels of granularity: make, model, and year. ($ii$) attribute prediction: From the images, predict five attributes: ($i$) maximum speed; ($ii$) displacement; ($iii$) number of doors; ($iv$) number of seats; and ($v$) type of car. ($iii$) model verification: Given two car images, verify whether the images belong to the same model. Similar to the Cars dataset, in CompCars the images of the cars from external perspectives are characterized by presenting the car without occlusions and arranged in the central region of the image (except the set of images of specific parts).

Still on the domain of cars but focusing on Automatic License Plate Recognition (ALPR), we can cite UFPR-ALPR [10] and Vehicle-Rear [11]. Both these datasets contain Brazilian cars but were not designed for FGVC.

Both Cars and CompCars were created with the aim of providing a controlled environment for carrying out experiments. In fact, these datasets were designed for the FGVC task and have become widely used benchmarks in the literature [3], [12], [13]. However, although a controlled environment is a fundamental part of verifying an experiment, in practice, car images do not follow that pattern. Therefore, in this work, we assembled a dataset with the goal of reflecting the challenges inherent to FGVC tasks in practical applications. Furthermore, as discussed in the Introduction, car models vary across countries and, to the best of our knowledge, there is no such dataset for Brazilian cars.

## III. The BRCars Dataset

In this section, we present the data collection and the construction process that were followed when assembling the BRCars dataset.

Throughout this paper, we use the term *instance* to refer to a specific car advertisement, which is composed of $M$ images. For the construction of the dataset, we collected images of car advertisements from the *webmotors.com.br* website[1], one of the largest car advertisement websites in Brazil. We chose *webmotors* due to its large number of vehicle advertisements. It is important to mention that according to the current privacy terms of the website, commercial use of the data is prohibited without the prior and express consent of *webmotors*. In total, a set of 2,808,846 images was collected, which are distributed among 1,005 different car models. The images belong to a set of 336,660 car advertisements. The ground truth classes come from the website itself since the advertisements contain the make and the model of the vehicles. Our inspection found that the class assignments were trustworthy. Thus, no further annotation effort was required.

Due to the high class imbalance (*i.e.,* some car models have many more images than others), we created two sets to evaluate the FGVC task; they are: BRCars-196 and BRCars-427. Those are described in the next subsections. BRCars datasets are available at https://github.com/danimtk/brcars-dataset.

### A. BRCars-196 set

BRCars-196 has 196 classes, with each class referring to a car model. For the assembly of this set, we first selected only the models that have at least 200 instances of vehicles. Then, for each car model, 200 instances were randomly selected. Finally, each of the 200 sets of images from the 200 random selections were grouped to compose the images of each model. As a result of this selection process, this set is composed of the most common car models. Since each car instance has a variable number of images, the final number of images belonging to each model has slight variations. After removing noisy images using the procedure described in Section III-C, BRCars-196 has 212,609 images, from which 170,151 are intended for training, while 42,458 are intended for testing. Table I shows more statistics. The average images by model is 1,084.74 with a standard deviation of 81.92. Of the total images, 40,665 images are internal images (with an average of images by model of 207.47 and a standard deviation of 37.00).

### B. BRCars-427 set

BRCars-427 includes BRCars-196 and additional 231 classes referring to models that have fewer than 200 car instances. In order to remove classes that are extremely underrepresented in terms of instances, we discarded models with fewer than 20 instances. These additional 231 classes have a varied number of images. The goal of adding the classes with fewer instances is to replicate the challenge of dealing with rarer models. After the noise removal procedure (see Section III-C), BRCars-427 is composed of 300,325 images, from which 239,668 are intended for training, while 60,657 are intended for testing. Table I shows that the average number of images by model is 703.34 with a standard deviation of 403.78. Of the total images, 58K are internal images (with an average of images of 135.76 by model and a standard deviation of 81.26). The bigger standard deviation, when compared with BRCars-196, gives highlights the imbalance from the 231 classes with fewer images.

[1] http://www.webmotors.com.br

TABLE I
STATISTICS OF THE DATASETS: NUMBER OF IMAGES PER MODEL

| Dataset | Perspective | Images | Avg images by model (SD) |
|---|---|---|---|
| BRCars-196 | all | 212,609 | 1,084.74 (81.92) |
| | external | 171,944 | 877.27 (60.09) |
| | internal | 40,665 | 207.47 (37.00) |
| BRCars-427 | all | 300,325 | 703.34 (403.78) |
| | external | 242,354 | 567.57 (325.95) |
| | internal | 57,971 | 135.76 (81.26) |

### C. Removing Noise

Due to the nature of the source of the data, a considerable number of images can be characterized as noise, among them: images of keys and car documents, images that have only textual elements, images of specific parts that are not representative (such as wheels, parts of vehicle, doors, *etc.*). Noise images do not present visual information that can contribute to the recognition of the car model, and thus, they should be discarded. In order to do that, we adopted a hybrid annotation procedure for visual aspects that mixes manual and automatic procedures. We consider that relevant images should include an external perspective of the car (in whole or in part), or of the cockpit.

Our procedure for removing noise images relies on the new pre-trained CLIP (Contrastive Language – Image Pre-training) architecture [14]. CLIP consists of a text encoding architecture combined with an image encoding architecture. It is trained with the contrastive function loss in pairs of images and texts collected from the Web. This way, it is able to learn visual concepts from natural language supervised training. After being trained, given a pair containing a textual query and an image, it allows calculating the similarity between their respective vectors.

We use CLIP as an annotator for the perspectives of the cars in order to assist in the selection of the relevant images. More specifically, we use the version that relies on Vision Transformers ViT-B/32 [15] with the encoding. We created two sets of textual queries. The first set contains four text queries: ($i$) engine; ($ii$) wheel; ($iii$) external body view; and ($iv$) interior view. The second set has seven text queries: ($i$) seats; ($ii$) cockpit; ($iii$) instrument panel or dashboard; ($iv$) gearshift; ($v$) sunroof; ($vi$) radio; and ($vii$) door panel. These queries define the classes that will be automatically assigned to the images.

All images were submitted to CLIP and then, for each image vector, the cosine similarity with the query vectors was calculated. These similarities were grouped according to the query sets and submitted to a softmax function to obtain the probabilities of each query in relation to the image. The most likely textual query for each query set was assigned as the image class, so that each image receives the two most likely classes, one from the first query set and the other from the second query set.

Then, two samples were generated, each containing around 5K images. The first sample has random images among those that were labeled by CLIP as *external body view* and the second sample has random images that were labeled both as *internal view* and *cockpit* (*internal view & cockpit*).

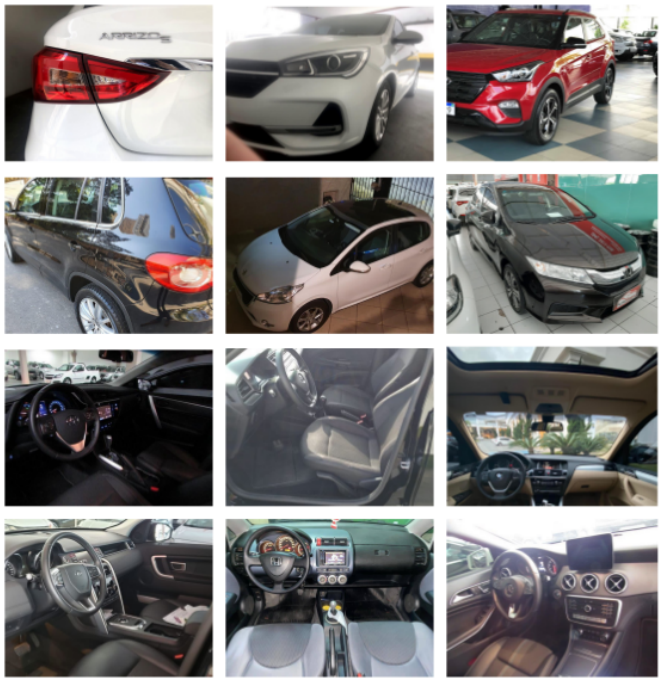


Fig. 1. Samples of the images from the BRCars dataset. Images are taken from different perspectives and may not include the entire vehicle. In addition, there are images from the interior of the car.

The two samples were manually annotated to validate whether the images indeed belonged to the target class. The results of this analysis found accuracies of 96.84% for *external body view* and 81.20% for *internal view & cockpit*. The manual annotations were used to train two binary auxiliary CNNs, one for the *external/not external* classes and another for the *cockpit/not cockpit* classes. Finally, the images previously classified by CLIP as *external body view* and *internal view &cockpit* were submitted to a second annotation step using the respective auxiliary CNNs.

### D. Characteristics of the BRCars Datasets

In this subsection, we describe the properties of the BRCars datasets and their images.

*1) Style of images:* The images are characterized by a lack of standardization with regards to their perspectives. Unlike the Cars and CompCars datasets, which present well-centered images, the images in our three datasets are considerably non-standard. In addition, there are mixed images of internal and external perspectives. This is due to the nature of the images – the advertisers are in diverse environments; some are more and others are less suitable for photography. Also, the advertisers have different levels of photographic skills. Figure 1 shows 12 images from the dataset, highlighting the lack of standardization in terms of perspective.

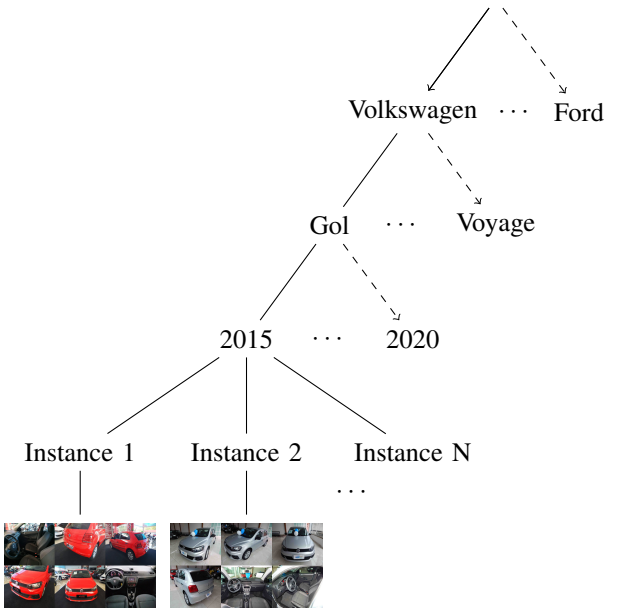



Fig. 2. Hierarchy of the dataset. BRCars can be approached in different granularities: ($i$) make; ($ii$) model; ($iii$) year; and ($iv$) car instances.

*2) Hierarchy:* The dataset is organized in a tree structure shown in Figure 2. The structure is similar to the one adopted by CompCars [9], where the granularity gets finer as we move down the tree. In the first level, granularity refers to the car make. Then, in the second level, the granularity refers to the car model, followed by the year in the third level. Finally, the actual instances of a specific car are in level four. The fourth level is a novelty in relation to the other datasets since it groups images belonging to a specific car instance. That is, due to the nature of the origin of the images, it is possible to group a set of images belonging to a specific instance.

## IV. Fine-Grained Classification Setup

In this section, we apply classification algorithms to predict the car model of a given input image. The goal is to provide baseline classification results for BRCars.

In our experiments, we compared different CNN architectures, namely *InceptionV3* and *ResNet50*, and Siamese versions of these networks. A Siamese network is an architecture consisting of two or more identical neural networks that aim to generate feature vectors and compare them. Siamese networks have been applied in several fields, such as face-recognition [16], [17], image retrieval [18], and multi-domain tasks [19]. Siamese networks learn a function $f$ able to generate feature vectors with high discriminative capacity among the different classes. Thus, we send three input images to the Siamese architecture: ($i$) an anchor image ($A$); ($ii$) a positive image ($P$) that corresponds to the same class as the anchor and; ($iii$) a negative image ($N$) which belongs to a different class. The goal is that $f$ learns to generate output vectors such that vectors belonging to the $A$ images are close to the $P$ image vectors and away from the $N$ image vectors. To do that, we use the triplet-loss function:

$$L(A, P, N) = max(D(f(A), f(P)) - D(f(A), f(N)) + m, 0) \quad (1)$$

where where $D$ denotes the Euclidean distance and $m$ is the hyperparameter used to set a margin that makes sure that the distance between $A$ and $P$ is smaller than the distance between $A$ and $N$ by at least a margin $m$ [17].

We chose these architectures due to their accuracy being constantly among the highest in image classification tasks. The framework adopted for the execution of the experiments used Keras[2], integrated with Tensorflow 2.3[3].

### A. Pre-processing

Initially, the input images submitted to *InceptionV3* and *ResNet50* were resized to 256×256. We applied the data augmentation procedure so that, in each epoch, for each of the images belonging to the batch, random transformations in series were applied. The transformations were: ($i$) rescaling 1/255; ($ii$) rotations with a range of 30 degrees; ($iii$) zoom transformations with a range of 0.15; ($iv$) width shift with a range of 0.2; ($v$) height shift with a range of 0.2; ($vi$) shear with a range of 0.15; and ($vii$) horizontal flip transformations with nearest fill mode. No data augmentation was used in the Siamese architecture.

### B. Transfer learning

The CNNs were started with the pre-trained weights from the *ImageNet* set [1]. Then, for each architecture, we removed the last layer, keeping only the resource extraction layers. Then, we added a Global Average Pooling layer [20] and two new fully connected layers to each CNN. The first fully connected layer has 1024 units and the second fully connected layer has 512 units. Finally, we added the softmax layer corresponding to the number of desired outputs. This layer had 196, 427, 196, and 431 units for the BRCars-196, BRCars-427, Cars, and CompCars datasets, respectively. Between the second fully connected layer added with 512 units and the softmax layer, we applied a dropout layer [21] of 0.5.

For the Siamese architectures, the CNNs were started with the first-epoch weights of the categorical cross-entropy training. That is, we train the CNN architectures initialized with Imagenet weights for one epoch with categorical cross-entropy. Then, we remove the last two layers and train the networks with the triplet-loss function. The final architecture is: CNN layers → Global Average Pooling layer → 1024 layer Relu → 512 layer. This one-epoch training was essential for the Siamese architectures to converge.

### C. Training

For training *InceptionV3* and *ResNet50*, we used the *Adam optimizer* [22], with a learning rate of 1e-4. The loss function

[2]https://keras.io/
[3]https://www.tensorflow.org/

adopted was the *categorical crossentropy*. After applying the settings, the models were trained for 25 epochs.

For the Siamese architectures, we adopted the *hard-batch* strategy to perform online triplet mining [17]. The batches were generated through a uniform random selection of $c$ classes with uniform random selections of $r$ images for each $c$ selected class. In our experiments, we used $c = 16$ and $r = 4$, that's resulted in batches of 64 images. We used the *Adam optimizer* with a learning rate of 1e-5. The models were trained for 15 epochs. We observed that there is no enhancement in classification accuracy above that number of epochs. After Siamese training, we applied the $k$-nearest neighbors (KNN) algorithm on the image vectors to make predictions for new unseen images. We call these architectures *KNN InceptionV3-S* and *KNN ResNet50-S*. The hyperparameter $k$ was set to 5 as it showed good results in both datasets.

### D. Evaluation Metrics

To evaluate classification quality, we adopted the following standard metrics:

**Precision** measures the ratio of the images classified as belonging to a class $c$ that actually belong to that class.

**Recall** measures the the proportion of instances that belong to a class $c$, which that were classified as such.

**F1** is the harmonic mean between precision and recall. These three metrics were calculated for each class and then macro-averaged. Macro-averaging is especially useful for imbalanced datasets such as BRCars-427 since it gives equal weights to each class (and not each instance). We basically calculate the F1 for each of the $n$ classes and then take the average.

### E. Results

Classification results are shown in Table II. Comparing the results across datasets, we notice that BRCars-427 has lower scores in all metrics. This was expected given that there are over twice as many classes and the dataset is very unbalanced. The F1 achieved by InceptionV3 and ResNet50 were very close, with InceptionV3 being slightly superior in both datasets by a 0.02 margin. On the Siamese runs, both KNN InceptionV3-S and KNN ResNet50-S achieve the same scores for all metrics except for F1 on BRCars-427 (in which KNN ResNet50-S loses by one percentage point). We can also see that the Siamese versions performed worse results than their counterparts. The difference was more noticeable on BRCars-427. We believe this happened due to the training set having both external and internal images, generating confusing image triplets which may have degraded the result of the Siamese networks.

Table III shows the top-3 classes in terms of F1 for all four architectures in both datasets. We can see that some car models are the same for different architectures and datasets. *JEEP RENEGADE*, *FIAT 500*, *CITROEN C4*, and *FIAT MOBI* all appear more than once among the top-ranked models. This suggests that these classes present unique visual information that helps in their accurate identification.

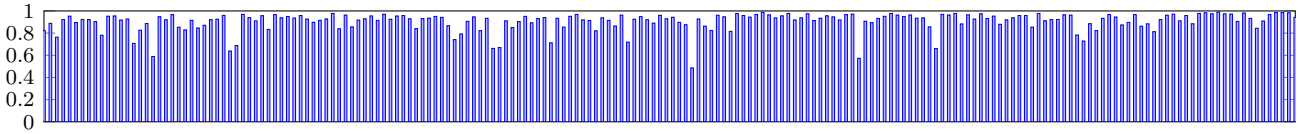
TABLE II
RESULTS OF EACH ARCHITECTURE IN FINE-GRAINED CLASSIFICATION TASK

| Dataset | Architecture | Precision | Recall | F1 |
|---|---|---|---|---|
| BRCars-196 | InceptionV3 | **0.92** | **0.91** | **0.91** |
| | ResNet50 | 0.89 | 0.89 | 0.89 |
| | KNN InceptionV3-S | 0.85 | 0.85 | 0.85 |
| | KNN ResNet50-S | 0.85 | 0.85 | 0.85 |
| BRCars-427 | InceptionV3 | **0.82** | **0.79** | **0.79** |
| | ResNet50 | 0.80 | 0.77 | 0.79 |
| | KNN InceptionV3-S | 0.67 | 0.62 | 0.64 |
| | KNN ResNet50-S | 0.67 | 0.62 | 0.63 |

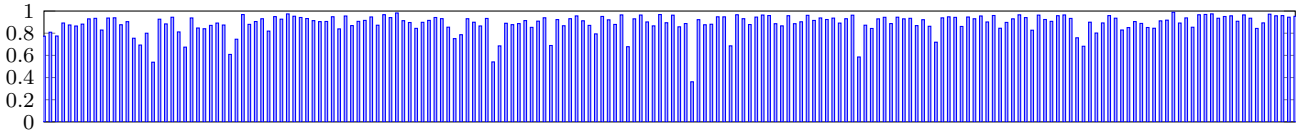
TABLE III
TOP-3 CLASSES IN TERMS OF F1

| Dataset/Arch. | Class with the highest F1 |
|---|---|
| BRCars-196 IV3 | HYUNDAI CRETA (0.99)<br>FIAT 500 (0.99)<br>VOLVO XC40 (0.99) |
| BRCars-196 R50 | JEEP RENEGADE (0.99)<br>CHEVROLET MERIVA (0.98)<br>NISSAN KICKS (0.97) |
| BRCars-196 IV3-S | FIAT MOBI (0.98)<br>FIAT 500 (0.98)<br>CITROEN C4 (0.97) |
| BRCars-196 R50-S | RENAULT KWID (0.99)<br>JEEP RENEGADE (0.97)<br>FIAT MOBI (0.97) |
| BRCars-427 IV3 | CITROEN C4 (0.99)<br>FIAT 500 (0.99)<br>CHERY TIGGO (0.99) |
| BRCars-427 R50 | FIAT 500 (0.98)<br>FIAT MOBI (0.98)<br>TOYOTA YARIS (0.98) |
| BRCars-427 IV3-S | FIAT TORO (0.97)<br>VOLVO XC40 (0.97)<br>FIAT MOBI (0.97) |
| BRCars-427 R50-S | FIAT 500 (1.00)<br>CITROEN C4 (1.00)<br>JEEP RENEGADE (1.00) |

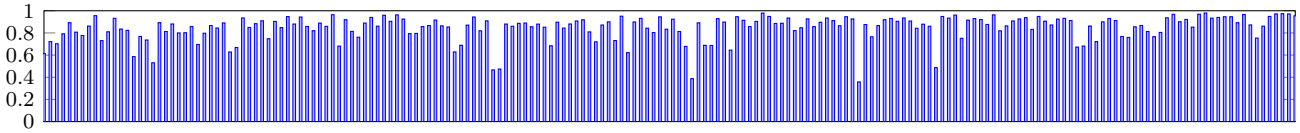
Figure 3 gives an overview of the F1 scores obtained in each class (where each column represents a class). We can see that for both datasets, InceptionV3 and ResNet50 achieved similar results. This similarity is evidenced looking at the pattern formed by the F1 scores on Figure 3. When comparing the columns of Figure 3a which refers to InceptionV3 trained on BRCars-196, with the columns on Figure 3b which refers to the ResNet50 trained on the BRCars-196 dataset, we can observe a pattern, mainly in the classes with the lowest F1. Although it presents slightly lower results, the same pattern is observed in the Figures 3c and 3d, that refer to Siamese networks KNN InceptionV3-S and KNN ResNet50-S, respectively. These similar patterns seem to indicate that the low F1 in these classes may be related to the degree of difficulty of visually distinguishing among the classes. It is also possible to observe, although it is less apparent, a similar pattern comparing the columns of Figure 3e which refers to InceptionV3 trained on BRCars-427, with the columns on Figure 3f which refers to ResNet50 trained on BRCars-427.



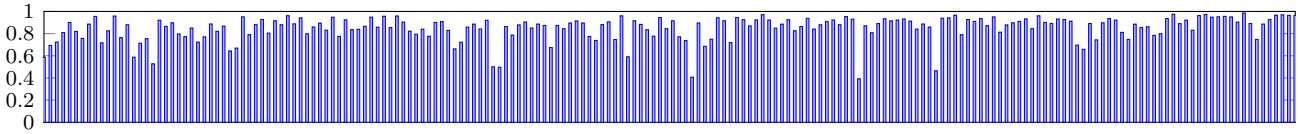

(a) BRCars-196 - InceptionV3



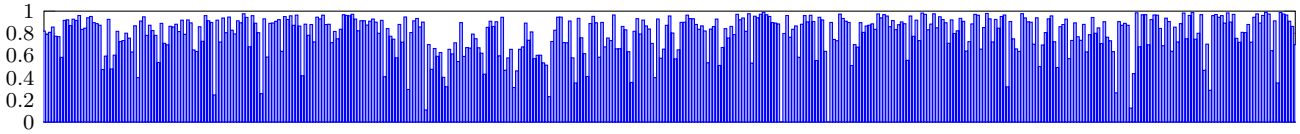

(b) BRCars-196 - ResNet50



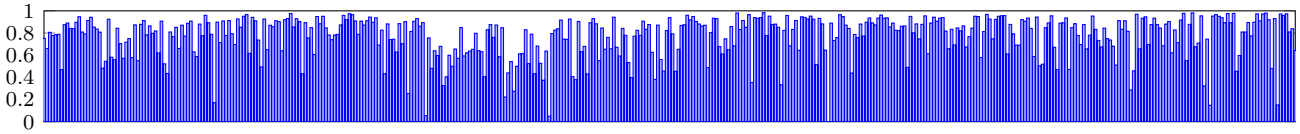

(c) BRCars-196 - KNN InceptionV3-S



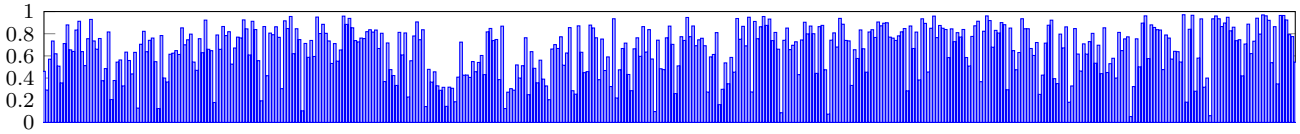

(d) BRCars-196 - KNN ResNet50-S



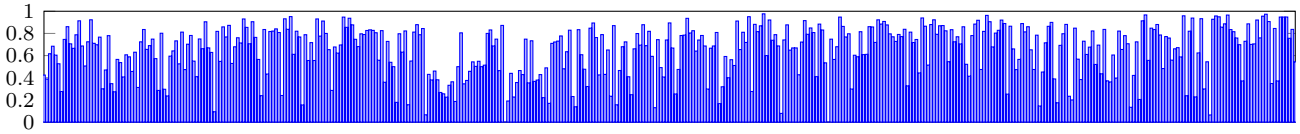

(e) BRCars-427 - InceptionV3




(f) BRCars-427 - ResNet50




(g) BRCars-427 - KNN InceptionV3-S




(h) BRCars-427 - KNN ResNet50-S

Fig. 3. F1 by class for model classification across the different subsets and algorithms.

TABLE IV
CLASSES WITH LOWEST F1 AND THEIR MOST FREQUENT PREDICTED CLASSES

| Dataset/Arch. | Class with lowest F1 | Most frequently classified as | $2^{nd}$ most frequently classified as |
|---|---|---|---|
| BRCars-196 IV3 | MERCEDES-BENZ C 180 (0.49)<br>MERCEDES-BENZ C 200 (0.57)<br>VOLKSWAGEN POLO (0.59) | MERCEDES-BENZ C 200 (0.44)<br>MERCEDES-BENZ C 180 (0.17)<br>VOLKSWAGEN POLO SEDAN (0.22) | MERCEDES-BENZ C 250 (0.10)<br>MERCEDES-BENZ C 250 (0.08)<br>VOLKSWAGEN VIRTUS (0.17) |
| BRCars-196 R50 | MERCEDES-BENZ C 180 (0.36)<br>VOLKSWAGEN POLO (0.54)<br>BMW 320i (0.54) | MERCEDES-BENZ C 200 (0.55)<br>VOLKSWAGEN VIRTUS (0.28)<br>BMW 328i (0.5) | MERCEDES-BENZ C 250 (0.15)<br>VOLKSWAGEN POLO SEDAN (0.21)<br>BMW X3 (0.01) |
| BRCars-427 IV3 | MERCEDES-BENZ B 170 (0.0)<br>FORD F-4000 (0.0)<br>BMW 316i (0.1) | MERCEDES-BENZ B 180 (0.91)<br>FORD F-250 (0.37)<br>BMW 320i (0.71) | MERCEDES-BENZ B 200 (0.09)<br>FORD F-350 (0.26)<br>BMW 328i (0.16) |
| BRCars-427 R50 | FORD F-4000 (0.0)<br>MERCEDES-BENZ E 500 (0.05)<br>BMW 316i (0.06) | FORD F-250 (0.42)<br>MERCEDES-BENZ E 350 (0.58)<br>BMW320i (0.61) | FORD F-350 (0.26)<br>MERCEDES-BENZ CLK 320 (0.08)<br>BMW 328i (0.35) |
| BRCars-196 KNN IV3-S | MERCEDES-BENZ C 200 (0.36)<br>MERCEDES-BENZ C 180 (0.39)<br>BMW 320i (0.47) | MERCEDES-BENZ B 180 (0.35)<br>MERCEDES-BENZ C 200 (0.32)<br>BMW 328i (0.42) | MERCEDES-BENZ B 250 (0.26)<br>MERCEDES-BENZ C 250 (0.22)<br>BMW X6 (0.03) |
| BRCars-196 KNN R50-S | MERCEDES-BENZ C 200 (0.39)<br>MERCEDES-BENZ C 180 (0.41)<br>MERCEDES-BENZ C 250 (0.46) | MERCEDES-BENZ C 180 (0.32)<br>MERCEDES-BENZ C 200 (0.34)<br>MERCEDES-BEN C 180 (0.28) | MERCEDES-BENZ C 250 (0.22)<br>MERCEDES-BENZ C 250 (0.17)<br>MERCEDES-BENZ C 200 (0.26) |
| BRCars-427 KNN IV3-S | BMW 420i (0.05)<br>MERCEDES-BENZ GLE 43 AMG (0.06)<br>FORD F-4000 (0.07) | BMW 428i(0.35)<br>MERCEDES-BENZ GLE 400 (0.71)<br>FORD F-250 (0.32) | BMW 430i (0.22)<br>MERCEDES-BENZ GLC 250 (0.14)<br>FORD F-350 (0.26) |
| BRCars-427 KNN R50-S | FORD DEL REY (0.0)<br>FORD F-4000 (0.0)<br>MERCEDES-BENZ GLE 43 AMG (0.1) | CHEVROLET OPALA (0.30)<br>FORD F-250 (0.47)<br>MERCEDES-BENZ GLE 400 (0.67) | CHEVROLET CHEVETTE (0.25)<br>FORD F-1000 (0.16)<br>MERCEDES-BENZ GLC 250 (0.14) |

In BRCars-427, the low F1 in certain classes is associated with the small number of images in the 231 additional classes with fewer instances. We found moderate positive correlations around 0.6 between the number of images in the classes and their corresponding F1 scores.

Figures 3g and 3h refer to KNN InceptionV3-S and KNN ResNet50-S on BRCars-427. We can see that both histograms have bars that widely vary in height, representing the high variance of F1 among the classes. According to the results presented on Table IV, which shows the classes with the lowest F1 with the respective classes responsible for the most frequent false positives, in all cases, the predicted false positive class belonged to the same make.

Table IV presents the three classes with the lowest F1 for each architecture trained in the BRCars-196 and BRCars-427 datasets. This information is presented in the *Class with lowest F1* column, and the classes are sorted by F1 in ascending order. For each of the three classes with the lowest F1, we also present the two classes that were most often wrongly predicted (shown in columns 3 and 4). Among the three classes with the lowest F1 scores, *MERCEDES-BENZ* models appear on all trained architectures. This is not surprising, as the different models from this make share a strong visual identity with each other. The table also shows that C 180, C 200, C 250, B 170, B 180, and B 200 are the classes most frequently confused with each other by all classifiers.

In BRCars-427, *FORD F-4000* was one of the three classes with lowest F1 scores for all classifiers. Except for BRCars-427 KNN IV3-S, all other models had an F1 of zero since no instances were correctly predicted into the class. In addition to *FORD F-4000*, two other classes had a F1 score of zero in some architecture – MERCEDES-BENZ B 170 and *FORD DEL RAY*.

Figure 4 presents some examples of the classes that were most often wrongly predicted (from Table IV). We observe that these models are very similar. Among these classes, *VOLKSWAGEN POLO* is a special case for two reasons: $(i)$ there are two classes for *VOLKSWAGEN POLO* – one for the sedan and another for the hatch version. Other models did not have such a separation; $(ii)$ the new *VOLKSWAGEN POLO* is only available in the hatch version. However, *VOLKSWAGEN VIRTUS* is a sedan model very similar to *VOLKSWAGEN POLO* (their front perspectives are identical). This way, the new *VOLKSWAGEN POLO* and the old *VOLKSWAGEN POLO HATCH* belong to the same class (although their front has changed), while *VOLKSWAGEN VIRTUS* and *VOLKSWAGE POLO SEDAN*, are two distinct classes.

**Comparison with other datasets**. In order to enable a comparison of the classification performance on BRCars and existing vehicle datasets, we also trained ResNet50 and InceptionV3 on the Cars and CompCars datasets. The results are in Table V.

TABLE V
RESULTS ON EXISTING DATASETS

| Dataset | Architecture | Precision | Recall | F1 |
|---|---|---|---|---|
| Cars | InceptionV3 | **0.85** | **0.84** | **0.83** |
| | ResNet50 | 0.83 | 0.81 | 0.81 |
| CompCars | InceptionV3 | **0.85** | **0.82** | **0.82** |
| | ResNet50 | 0.84 | 0.78 | 0.79 |

The scores were better than the scores in BRCars-427 (which is more unbalanced) and worse than in BRCars-196. This seems to indicate that the algorithms are robust in

dealing with unbalanced data. It may also be possible that the larger volume of images may have contributed for a better generalization in our datasets.


Fig. 4. Examples of car models that are often misclassified – the cars in each row are often confused.

## V. Conclusion

This paper introduced BRCars, a dataset of Brazilian car images for FGVC tasks. BRCars contains 300,325 images belonging to 52,505 car advertisements of 427 car models. Compared to existing datasets for the FGVC task in the context of vehicles, our dataset is characterized by a lack of standardization with regards to perspectives (containing both external and internal images). Also, our classes are unbalanced, *i.e.,* some car models have more images than others. We believe that these characteristics are more representative of how images are presented in a number of practical applications, including transport monitoring, surveillance, self-inspection for car insurance, and automatic ad verification for advertising websites.

We believe that there are several possible usages for the BRCars dataset. In this first work, we focused on building a dataset for fine-grained classification, emphasizing external and cockpit perspectives with the goal of enabling experiments that can more closely replicate the real world.

In this paper, we did not train separate classifiers for internal and external images. This could be done as future work to assess how having more homogeneous classes affects the results. Additionally, one could add other perspectives, such as the car engine or other specific parts of the car interior.

## Acknowledgments

This work was partially supported by CNPq/Brazil, and by CAPES Finance Code 001.

## References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.
[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *(IJCV)*, vol. 88, no. 2, 2010.
[3] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *IEEE (ICCV)*, 2017, pp. 5209–5217.
[4] F. Zhang, M. Li, G. Zhai, and Y. Liu, "Multi-branch and multi-scale attention learning for fine-grained visual categorization," 2020.
[5] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Comput Intell Neurosci*, vol. 2017, 2017.
[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," no. CNS-TR-2011-001, 2011.
[7] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," Tech. Rep., 2013.
[8] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *IEEE (ICCV)*, 2013, pp. 554–561.
[9] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *IEEE CVPR*, 2015.
[10] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *IJCNN*. IEEE, 2018.
[11] I. O. De Oliveira, R. Laroca, D. Menotti, K. V. O. Fonseca, and R. Minetto, "Vehicle-rear: A new dataset to explore feature fusion for vehicle identification using convolutional neural networks," *IEEE Access*, vol. 9, 2021.
[12] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *IEEE (ICCV)*, December 2015.
[13] Y. Bai, F. Gao, Y. Lou, S. Wang, T. Huang, and L.-Y. Duan, "Incorporating intra-class variance to fine-grained visual recognition," 2017.
[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *Image*, vol. 2, p. T2, 2021.
[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR*, 2015.
[17] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.
[18] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *(IJCV)*, vol. 124, no. 2, pp. 237–254, 2017.
[19] J. V. A. de Souza, L. E. S. E. Oliveira, Y. B. Gumiel, D. R. Carvalho, and C. M. C. Moro, "Exploiting siamese neural networks on short text similarity tasks for multiple domains and languages," in *PROPOR*, 2020.
[20] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.