Identifying Car Brands in Noisy Images Using Convolutional Neural Networks

Rohit Amarnath, Aravind Kumar, Mugdhesh Pandkar

Project Report

1 Abstract

The exponential production of cars has resulted in a need for better vehicle classification systems. Such systems include automated highway toll collection, traffic flow control, and perception in self-driving vehicles. These classification systems require improvement in their ability to identify a car based off an image. Most existing research has developed models on clear car images, however, this is not reflective of real world conditions comprised of external factors such as inclement rain, snow, or motion blur. In this paper we seek to classify car brands in noisy and blurry images using convolutional neural networks as a means of better simulating real world conditions. We used the Stanford Cars Dataset, which contains 16,185 images of 196 classes of cars. Our auto-encoder model was able to accept either a noisy/blurred image as input and output an appropriately denoised/deblurred image. Afterwards, training our ResNet architecture for 15 epochs, we were able to identify (predict the make, model and year) of a car image with an accuracy of 82 percent. Our research better tailors models for vehicle classification systems that are consistent with real world conditions.

2 Introduction

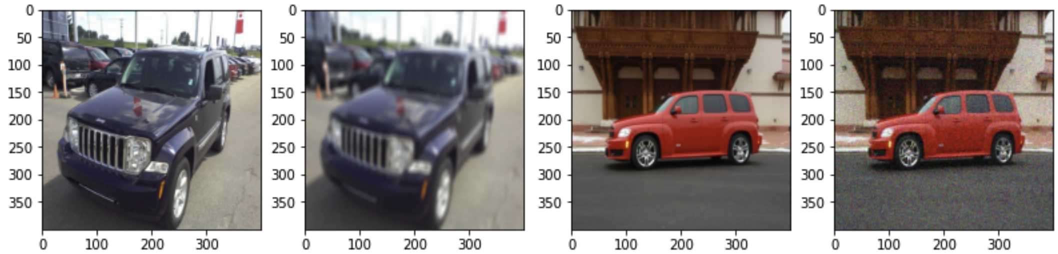
Using the Stanford Cars Dataset[1], our project seeks to more accurately classify passenger vehicle brands in realistic scenarios where the images are not perfectly clear. The work done in this project intends to expand on the research done by Muhammad Butt and his colleagues on classifying vehicles by type through classifying vehicles by brand [2]. While we considered approaches similar to Krause and Liu, due to time constraints, we were unable to fully implement those features. Throughout the project, we focused on pre-processing clear car images, implementing noise/blur functions, developing an autoencoder for denoising/deblurring complex images, and training ResNet for classifying car images. Further details on our methodology and results are below.

3 Methodology

3.1 Noise and Blur Functions

Given that our aim was to better simulate real world conditions by examining imperfect images, our first step was to implement two separate functions that convoluted the clear images from the Stanford Cars Dataset. The first would add noise, and the second would add blur. For both the noise and the blur algorithm, our input were the clear, unedited images from the dataset. We performed pre-processing on these clear car images by resizing them to 400x400 and then applying either randomized noise with a noise-factor of 0.12 or Gaussian blur using a kernel size of (7, 11) and a sigma of (11, 11), respectively. This allowed us to generate train, test, and validation sets of additional noisy and blurry images. Figure 1 demonstrates the effect of pre-processing the images and selectively adding blur/noise.

Submitted to Rutgers University-New Brunswick Deep Learning 01:198:462 (2022). Do not distribute.



(a) Clear image to blurry image

(b) Clear image to noisy image

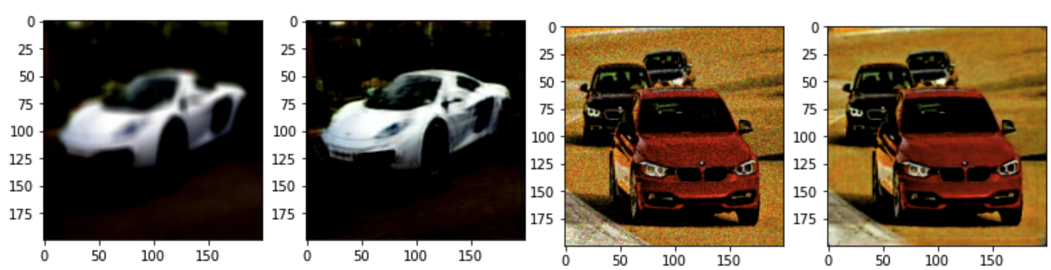
Figure 1: Pre-processing

3.2 Denoising and Deblurring Models

In order to be able to fulfill our overall objective of accurate identification, we logically needed functions that denoised and deblurred images. The denoise function accepts a noisy image from the prior step as input and returns a clearer image with less noise. Similarly, the deblur function accepts a blurry image from the prior step as input and outputs a clearer image with less blur. The technique to both denoise and deblur was a three layer Convolutional AutoEncoder with ReLU activation functions, Adam optimizer and Mean Squared Error (MSE) Loss. The AutoEncoder architecture performs extremely well in tasks that deal with re-configurations of the input; generally, in deep learning, we handle tasks that ask us to classify our input based on a set of output labels. However, the deblurring and denoising stage dealt with an decoding an encoded input, which is better handled by an encoder-decoder architecture. In essence, over the several epochs of training, the model learns the configurations of the input and is able to learn the patterns of blur and noise within the images. Figure 2 demonstrates examples of the input and output of the denoising and deblurring models.

3.3 Training Algorithm, Test Set and Loss

As we trained our denoising and deblurring models on the training set we recorded the loss of each. We utilized hyper-parameters of weight-decay: 1e-5 and learning-rate: 1e-3, and a batch-size of 32. We trained each model with 15 epochs, with each epoch having 204 iterations, and recorded the loss in each epoch. In our denoising model, we recorded a final loss of 0.1560. On our deblurring model we recorded a final loss of 0.1977. These relatively low losses indicate that a high level of prediction accuracy in our models. After training our models on the training set, we tested our results on the test set. Our results show a high degree of denoising and deblurring, corroborating the low loss achieved.



(a) Pre-processed blurry image to de-blurred image

(b) Pre-processed noisy image to de-noised image

Figure 2: AutoEncoder model

3.4 ResNet Architecture Overview and Linking

Next, we sought to implement our Resnet architecture to classify cars. Our input would be an image of a car, and we predict its make, model, and year. We used transfer learning in order to accomplish this task. Unfortunately, we don't have the computational power nor enough time to train, from scratch, our own model that would be able to learn a car's model, make, and year based on an image. Since Residual Convolutional Neural Networks are trained on the ImageNet dataset, which contains several car images, we realized that using ResNet would be beneficial in efficiently accomplishing our task. We detached the fully connected layer of the pre-trained ResNet model and added a linear layer, allowing extrapolation to a 196 class layer as per the 196 classes of cars in the dataset. We then built a seamless pipeline that would first denoise and deblur the image with our AutoEncoders and then classify the cars with ResNet. Through this linking, we are now able to comprehensively classify cars given a noisy or blurry image.

4 Results

We trained ResNet with 10 epochs on the training and test sets. Initially, in the first epoch, we recorded an accuracy of 22.0956 on the training set and 27 percent on the test set. After training, at the 10th epoch, our model achieved an accuracy of 97.8554 on the training set and 82 percent on the test set. The high level of accuracy achieved represents that given an image as input, the model is able to accurately identify the make, model and year of a car.

5 Discussion

Finally, we sought to test validate the high accuracies obtained in the previous steps We sought to see whether our linked system, where we linked our denoising and deblurring models with Resnet, could correctly classify a car given an input image. To test our overall project, we passed in a random car image (classified as Rolls-Royce Phantom Sedan 2012). Our model was able to accurately identify the make, model, and year of the car with confidence, 11.169 (denoised) and 12.974. The confidence appears to be somewhat low, however given the level of specificity of a car that the model predicts, this confidence level is satisfactory.

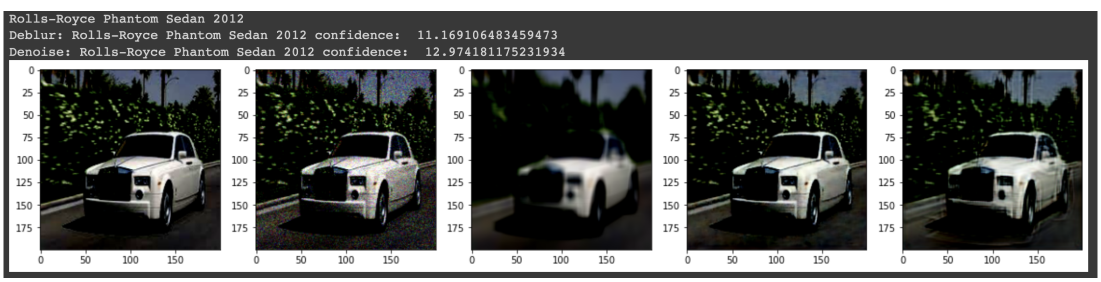


Figure 3: The first line output is the class of the image, second line is predicted class from deblurred model, third line is predicted class from denoised mode. From left to right: original cars dataset image, added noise, added blur, denoised, deblur.

6 Future Direction

Currently, our model only classifies clear images, blurry, and noised images. In the future, this capability could be extended to images that are both noisy blurry, as well as images that include harsh angles of rotation. We may utilize many of the techniques suggested by Krause and colleagues in handling vehicle datasets and potentially converting 2D images into 3D scans that may offer more insight into brand localization [3]. We may also attempt to expand the research done by Xinchen Liu and colleagues on a deep learning approach for urban surveillance that utilizes a two pass process of coarse-to-fine and near-to-distant search techniques [4].

7 Code

A full link to our GitHub repository housing a README, all trained models, and ipynb files can be found at https://github.com/rohitamar/IdentifyingCarBrandsNoisyImagesCNN

References

[1] http://ai.stanford.edu/~jkrause/cars/car_dataset.html

[2] Muhammad Atif Butt, Asad Masood Khattak, Sarmad Shafique, Bashir Hayat, Saima Abid, Ki-Il Kim, Muhammad Waqas Ayub, Ahthasham Sajid, Awais Adnan, "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems", Complexity, vol. 2021, Article ID 6644861, 11 pages, 2021. https://doi.org/10.1155/2021/6644861

[3] 3D Object Representations for Fine-Grained Categorization Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013.http://vision.stanford.edu/pdf/3drr13.pdf

[4] Xinchen Liu, Wu Liu, Tao Mei, Huadong Ma: A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. ECCV (2) 2016: 869-884 (Citation=126)http://xinchenliu.com/papers/2016_ECCV_PVID.pdf