Team: Keenan Willison and Sean Salvador

Elevator Pitch/ Proposal: Train a Convolutional Neural Network to classify car damages into different classes of severity (minor/moderate/severe, as well as draw bounding boxes around the damages to assist insurance companies in automating claim processing and providing quick initial assessments.

Methods: Develop and train a CNN to best classify the car damages. For practicality and to avoid creating an entire CNN from scratch, we will employ transfer learning and reuse/freeze many of the initial layers for feature extraction, and then experiment with different layers/architecture on top. Given that our dataset classifies car damages into different classes of severity, our output layer will utilize softmax as the activation function. Therefore, we will also initially use categorical cross-entropy as our loss function. For the model itself, we will experiment with a few different well-established architectures such as VGG-16 (which we've used on the homeworks). However, since VGG16 is not trained on car damage labels - we would have to fine-tune the last few layers. Other architectures we will experiment with include ResNet and EfficientNet. *Moreover, we also aim to explore adding bounding boxes to identify where the damage/damages is located - but this would require manual/one-shot labeling or some form of semi-supervised or self-supervised learning. *We'd also like to do some experimentation with estimating the cost of damages which is definitely a huge undertaking with the amount of variables that goes into it.

Data Source: We have found some preliminary data sources from Github and Kaggle: https://cardd-ustc.github.io/ ; https://www.kaggle.com/datasets/prajwalbhamere/car-damage-severity-dataset?utm_source=chatgpt.com

Code Resources: Using Sean's Medium Subscription, we will use many of the website's articles on CNN's as inspiration and a guideline for our project. For debugging purposes, we might utilize ChatGPT to assist.

What's New: We will utilize many of the methods we've learned in class to improve the quality and generalizability of our model. For instance, we will add invariance to make our CNN more resilient/smarter and less sensitive to changes in input. Thus we will perform data augmentation to randomly modify our car image data (rotation/noise/reflections/translations etc.). This is especially important for our model since after viewing the datasets we noticed a few things – the lighting, angles, and overall quality of our images varies considerably. In general, we will experiment with different implementations of the last few layers of the pretrained CNN :adjusting the amount of layers, adjusting different types of layers (different combinations of convolutional, pooling, dense layers and also dropout) and adjusting the width of layers (amount of neurons). other hyperparameters to try include Regularization (L1/L2) and learning step adjustments etc.

Plan:
Milestone 1 (April 4): Data Preparation & Initial Model Training

- Download, clean, and preprocess the car damage dataset.
- Conduct small baseline experiments with different pre-trained CNN architectures and see which perform best out of box on a small sample of the data.
- Evaluate initial performance and identify areas for improvement.

Milestone 2 (April 18): Model Optimization & Testing

- Using the most effective models, each of us will experiment with running different architecture adjustments and fine-tuning individual pre-trained models.
- Conduct thorough testing to improve classification accuracy.
- Potentially experiment with a stacking approach in which both of our models are used and we train an additional meta classifier to pick which model to use depending on the image.
- Begin working on the short summary paper. (start drafting April 23ish)

Final Deliverables (April 30):

- A trained model capable of classifying car damage severity.
- A well-documented summary paper detailing methods, results, and findings.
- A lightning talk presentation summarizing key insights and project takeaways.

Proposed demonstration/evaluation: We will visualize model predictions on our validation set as well as new unseen real-world accident images that we collect to see how well it identifies damage severity. We will evaluate overall model multi-class classification success with a basic accuracy metric as well as Precision, Recall and F1-Score to evaluate success within each class.

Experiments:

1. Different Model Architectures - experiment with different well-known CNN models and see which model's based feature extractors work best for assessing car damages.We will also experiment with full fine-tuning of each models and transfer learning (freezing some of the earlier layers)
2. Data Augmentation/Architecture adjustments - using different combinations of data augmentation to see which performs success. For instance, combinations of lighting, random noise, translation and rotation augmentations to improve invariance. Moreover, experimenting with different adjustments for improving model generalizability such as more layers/different combos of layers (convolutional - also experiment with filter + kernel sizes, pooling, dense, dropout) and general hyperparameter tuning such as learning rate and regularization strategies.