Damaged Cars Detector

Talal Alnujaiman
Abdulelah Almajed
Fahad Alnafisa

Table of Contents

01 INTRODUCTION

02 PREPROCESS

03 RESULTS

04 CONCLUSION

01

INTRODUCTION

Problem statement

In our project we are trying to analyze and detect the damages in the car body using images, hoping to build a model to assist certain entity in damaged cars capturing using deep learning methods

Tools

Jupyter Notebook

Pandas

Sklearn

Keras

Matplotlib

TensorFlow

NumPy

Seaborn

Methodology

In our project, we decided to check for car damages in three separate ways, where we first determine whether or not the car is damaged, then we move forward to locate the place of the damage(front, side, or rear) and finally we try to evaluate the severity of that damage(minor, moderate, and severe)

❖ Damage check
❖ Damage location check
❖ Damage severity check

02

PREPROCESS

Preprocess

Reshaping
180x180

Normalization
1./255

Augmentation

03

RESULTS

Damage Check

Damage Check Data

2540
Not Damaged

2540
Damaged

Baseline Models: Properties

Simple NN

- Hidden layer: 1
- Dense: 40
- Activation: ReLU
- Optimizer: sgd

NN

- Hidden layer: 2
- Dense: 1000
- Activation: ReLU
- Optimizer: Adam

Simple CNN

- Conv2D: 2
- Filters: 16 - 32
- Hidden layer: 1
- Dense: 500
- Activation: ReLU
- Optimizer: Adam

Baseline Models: Scores

Simple NN

- Train Accuracy: 49.98%
- Validation Accuracy: 50.00%

NN

- Train Accuracy: 70.45%
- Validation Accuracy: 69.09%

Simple CNN

- Train Accuracy: 97.56%
- Validation Accuracy: 74.21%

Main Models

Acc/Models	CNN	VGG16	VGG19	INCEPTION V3	MOBILENET V2
training	89.25%	95.97%	96.63%	62.50%	90.21%
Validation	85.43%	93.31%	91.73%	65.67%	85.83%
test	-	94.29%	-	-	-

BEST

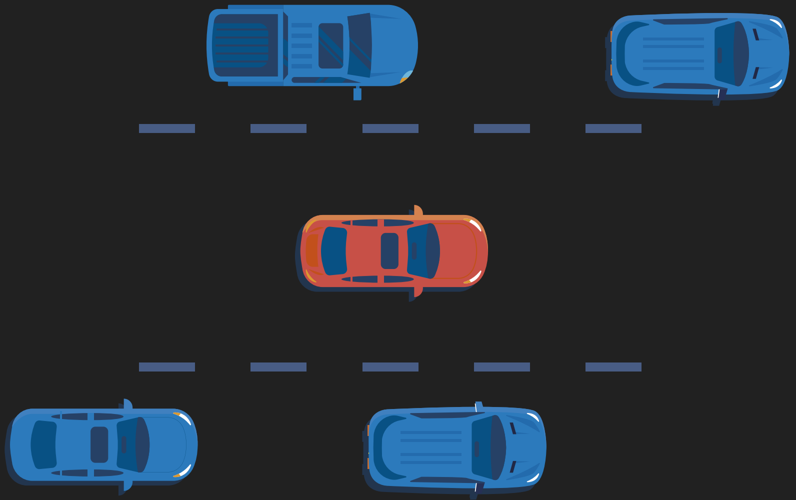
BEST MODEL
Transfer Learning: VGG16

Accuracy:

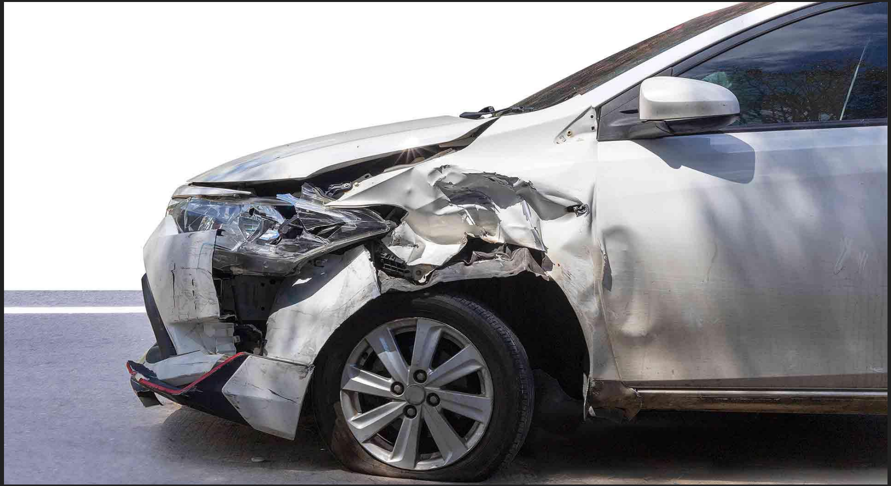
- Train: 95.97%
- Test: 94.29%

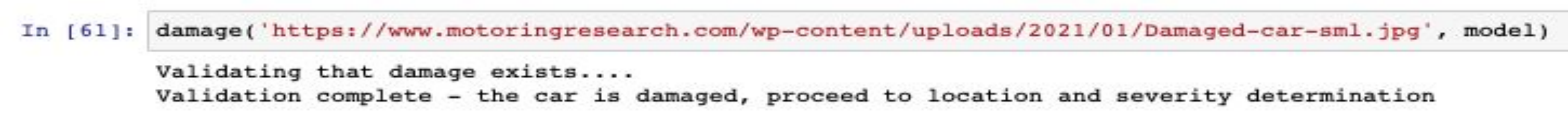
Properties:

- Dense: 100 - 50
- MaxPooling2D
- Dropout: 0.3
- Activation: ReLU
- Optimizer: Adam
- Early stopping at epoch 12
- Epochs = 20



Try Our Model on Damage Check





```
In [61]: damage('https://www.motoringresearch.com/wp-content/uploads/2021/01/Damaged-car-sml.jpg', model)

         Validating that damage exists....
         Validation complete - the car is damaged, proceed to location and severity determination
```

Damage Location Check

Damage Location Check Data

498
Front

320
Side

345
Rear

Baseline Models: Properties

Simple NN

- Hidden layer: 1
- Dense: 40
- Activation: ReLU
- Optimizer: sgd

NN

- Hidden layer: 2
- Dense: 1000
- Activation: ReLU
- Optimizer: Adam

Simple CNN

- Conv2D: 2
- Filters: 16 - 32
- Hidden layer: 1
- Dense: 500
- Activation: ReLU
- Optimizer: Adam

Baseline Models: Scores

Simple NN

- Train Accuracy: 40.96%
- Validation Accuracy: 54.67%

NN

- Train Accuracy: 34.83%
- Validation Accuracy: 54.67%

Simple CNN

- Train Accuracy: 53.36%
- Validation Accuracy: 51.61%

0.7634

Main Models

Acc/Models	CNN	VGG16	VGG19	INCEPTION V3	MOBILENET V2
training	53.36%	41.94%	70.95%	42.34%	33.03%
Validation	51.61	34.95%	71.03%	41.40%	53.33%
test	-	-	76.34%	-	-

BEST

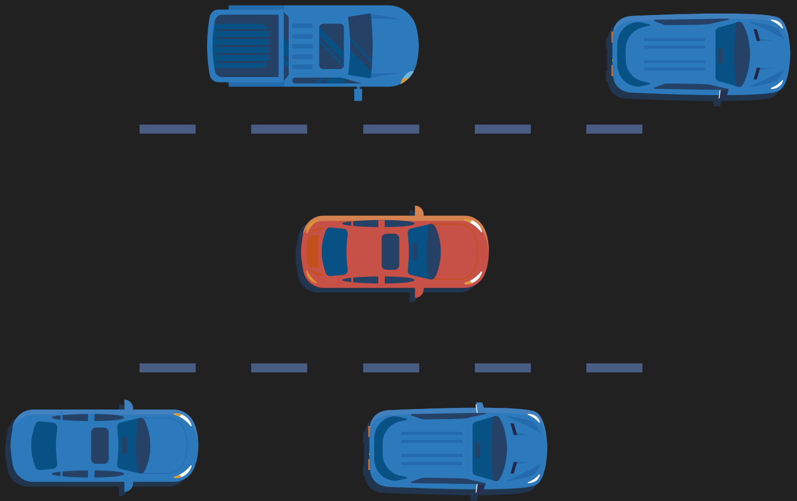
BEST MODEL
Transfer Learning: VGG19

Accuracy:

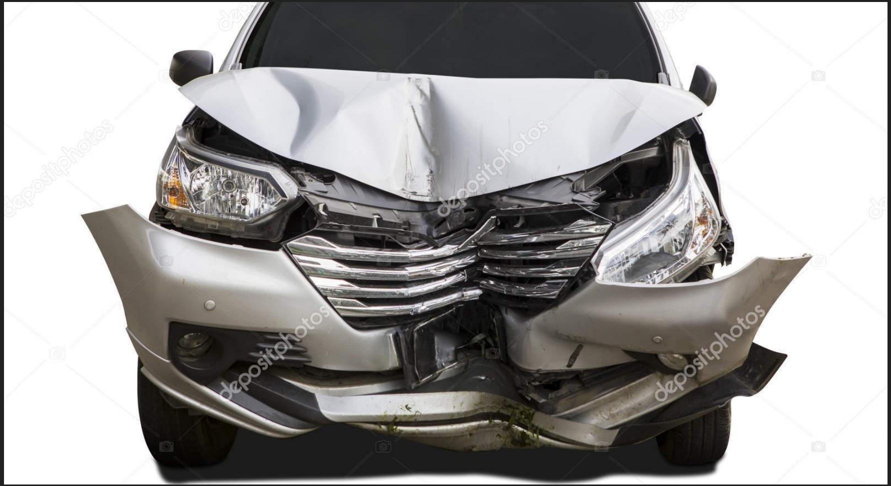
- Train: **70.95%**
- Test: **76.34%**

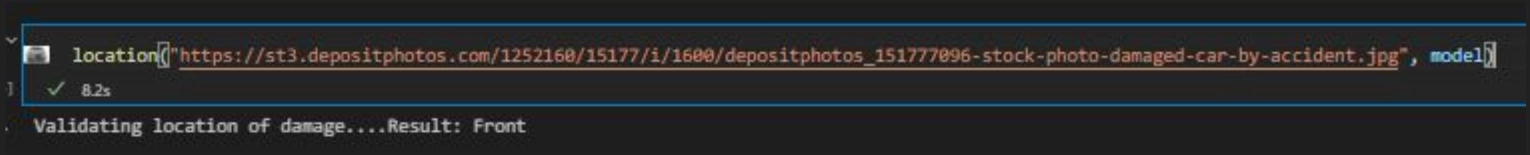
Properties:

- Dense: 100 - 50
- MaxPooling2D
- Dropout: 0.5
- Activation: ReLU
- Loss: Categorical Cross Entropy
- Optimizer: Nadam
- Early stopping at epoch 18
- Epochs = 20



Try Our Model on Damage Check





```
location("https://st3.depositphotos.com/1252160/15177/i/1600/depositphotos_151777096-stock-photo-damaged-car-by-accident.jpg", model)
✓ 8.2s
Validating location of damage....Result: Front
```

Damage Severity Check

Damage Severity Check Data

327
Minor

371
Moderate

455
Severe

Baseline Models: Properties

Simple NN

- Hidden layer: 1
- Dense: 40
- Activation: ReLU
- Optimizer: Adam

NN

- Hidden layer: 1
- Dense: 1500
- Activation: ReLU
- Optimizer: Adam

Simple CNN

- Conv2D: 2
- Filters: 16 - 32
- Hidden layer: 1
- Dense: 500
- Activation: ReLU
- Optimizer: Adam

Baseline Models: Scores

Simple NN

- Train Accuracy: 43.91%
- Validation Accuracy: 33.91%

NN

- Train Accuracy: 73.70%
- Validation Accuracy: 43.48%

Simple CNN

- Train Accuracy: 96.09%
- Validation Accuracy: 33.91%

56.22

Main Models

Acc/Models	CNN	VGG16	VGG19	INCEPTION V3	MOBILENET V2
training	53.36%	41.94%	78.06%	42.34%	33.03%
Validation	51.61	34.95%	85.87%	41.40%	53.33%
test	-	-	73.78%	-	-

BEST

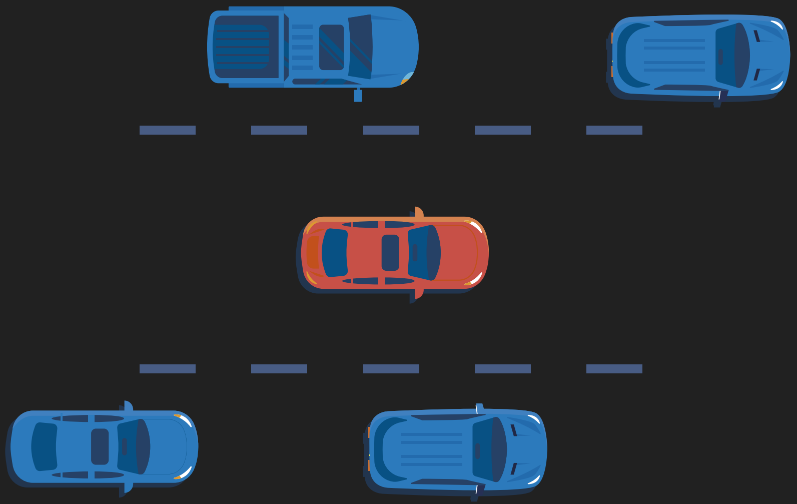
BEST MODEL
Transfer Learning: VGG19

Accuracy:

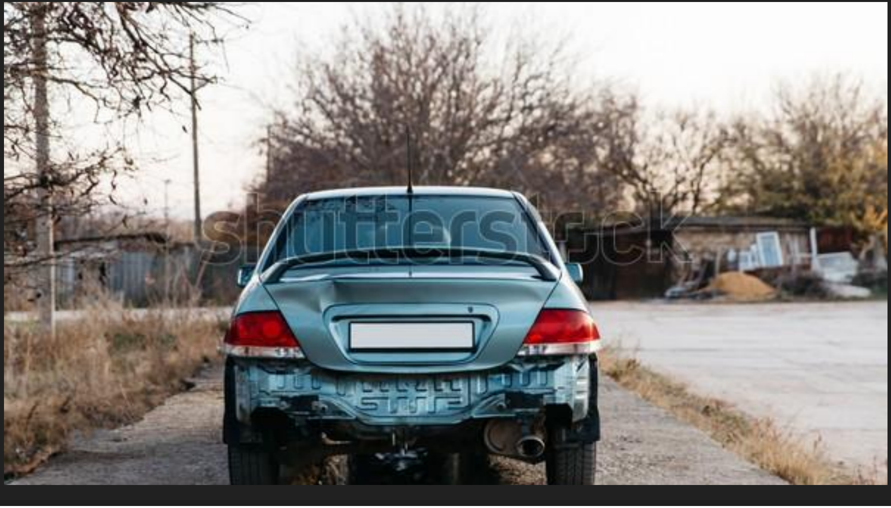
- Train: 78.06%
- Test: 73.78%

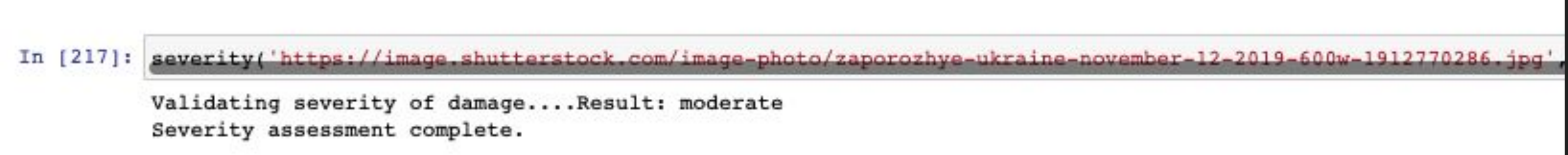
Properties:

- Dense: 100 - 50
- MaxPooling2D
- Dropout: 0.5
- Activation: ReLU
- Loss: Categorical Cross Entropy
- Optimizer: Adam
- Early stopping at epoch 17
- Epochs = 20



Try Our Model on Severity Check





```
In [217]: severity('https://image.shutterstock.com/image-photo/zaporozhye-ukraine-november-12-2019-600w-1912770286.jpg',
          Validating severity of damage....Result: moderate
          Severity assessment complete.
```

04

CONCLUSION

Obstacles

Overfitting

Time

Collecting Data

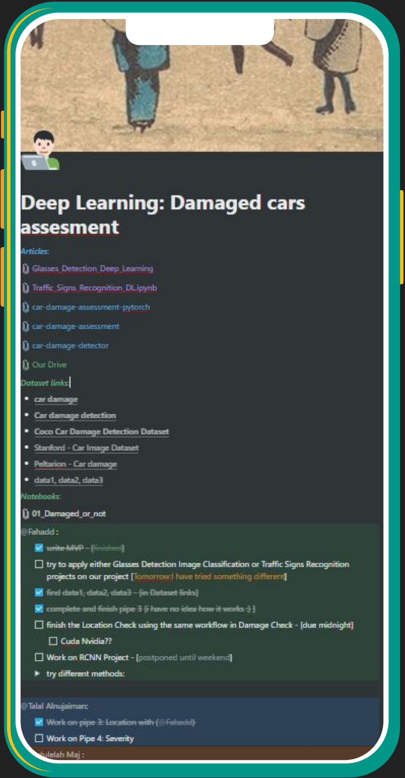
Future work

Web Application

Applying Mask RCNN to detect Damaged Parts

Improve our model

Notion



Deep Learning: Damaged cars assesment

Articles:

- Glasses_Detection_Deep_Learning
- Traffic_Signs_Recognition_DL.ipynb
- car-damage-assessment-pytorch
- car-damage-assessment
- car-damage-detector
- Our Drive

Dataset links:

- car damage
- Car damage detection
- Coco Car Damage Detection Dataset
- Stanford - Car Image Dataset
- Peltarion - Car damage
- data1, data2, data3

Notebooks:

- 01_Damaged_or_not

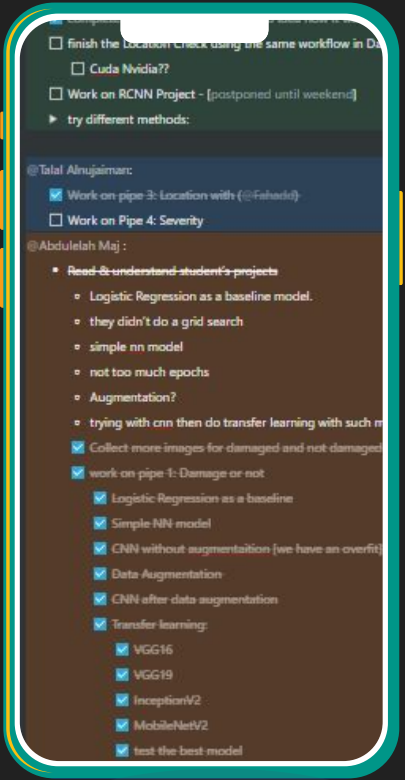
@Fahadd :

- ☑ ~~write MVP - [finished]~~
- □ try to apply either Glasses Detection Image Classification or Traffic Signs Recognition projects on our project [Tomorrow:I have tried something different]
- ☑ ~~find data1, data2, data3 - [in Dataset links]~~
- ☑ ~~complete and finish pipe 3 [i have no idea how it works :)]~~
- □ finish the Location Check using the same workflow in Damage Check - [due midnight]
  - □ Cuda Nvidia??
- □ Work on RCNN Project - [postponed until weekend]
- ▸ try different methods:

@Talal Alnujaiman:

- ☑ ~~Work on pipe 3: Location with (@Fahadd)~~
- □ Work on Pipe 4: Severity

dulelah Maj :



- □ finish the Location Check using the same workflow in Da
  - □ Cuda Nvidia??
- □ Work on RCNN Project - [postponed until weekend]
- ▸ try different methods:

@Talal Alnujaiman:

- ☑ ~~Work on pipe 3: Location with (@Fahadd)~~
- □ Work on Pipe 4: Severity

@Abdulelah Maj :

- ~~Read & understand student's projects~~
  - Logistic Regression as a baseline model.
  - they didn't do a grid search
  - simple nn model
  - not too much epochs
  - Augmentation?
  - trying with cnn then do transfer learning with such m
- ☑ ~~Collect more images for damaged and not damaged~~
- ☑ ~~work on pipe 1: Damage or not~~
  - ☑ ~~Logistic Regression as a baseline~~
  - ☑ ~~Simple NN model~~
  - ☑ ~~CNN without augmentaition (we have an overfit)~~
  - ☑ ~~Data Augmentation~~
  - ☑ ~~CNN after data augmentation~~
  - ☑ ~~Transfer learning:~~
    - ☑ ~~VGG16~~
    - ☑ ~~VGG19~~
    - ☑ ~~InceptionV2~~
    - ☑ ~~MobileNetV2~~
    - ☑ ~~test the best model~~

THANKS

Do you have any questions?