anyoneAI

# Team 01
# Automatic Car Classification


Nahuel Garcia


Joheel Luna


Agustin Genou


Enzo Gianotti


Carlos Prado


Nicolas Passadore

# Structure

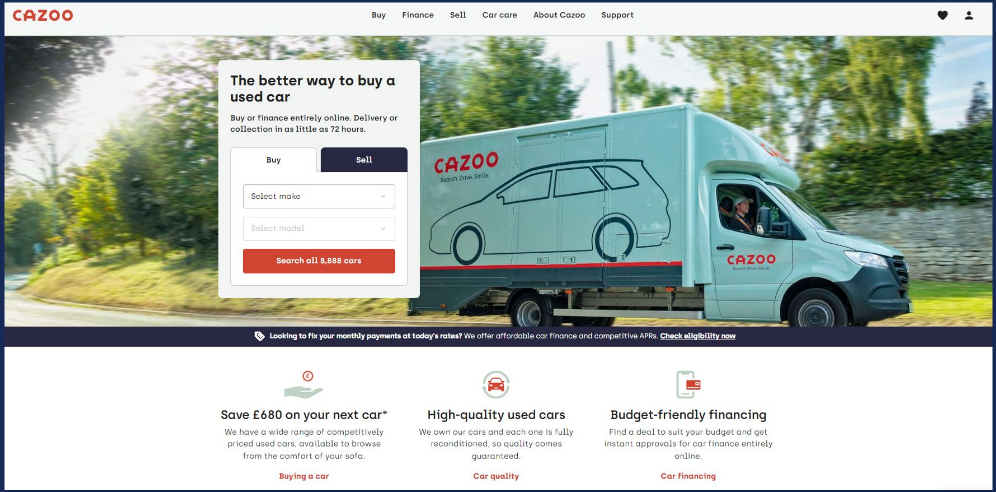
1. **Deliverables**
2. **Definition of the objective of this work**
3. **Literature Review for model Selection**
4. **Workplan**
    a. Designed Workflow
    b. Deployment: Defined Architecture
    c. Definition of profiles
5. **Exploratory Data Analysis and Preprocessing**
    a. Analysis
    b. Problems found
    c. Use of classification models
        - Custom binary Classification Model
        - YoloV7
    d. Remove Background
6. **Main Model Selected**
    a. Metrics
7. **Project DEMO**
8. **Conclusions**
9. **Next Steps**
10. **Q&A**

# Deliverables

- Vehicle classifier Convolutional Neural Network (CNN) for make and model for e-commerce
- Exploratory Data Analysis (EDA)
- Pre-processing, data preparation and training scripts
- Application Programming Interface (API) and basic User Interface (UI)
- Containerized with Docker and ready to be deployed
- Unit tests
- OCR

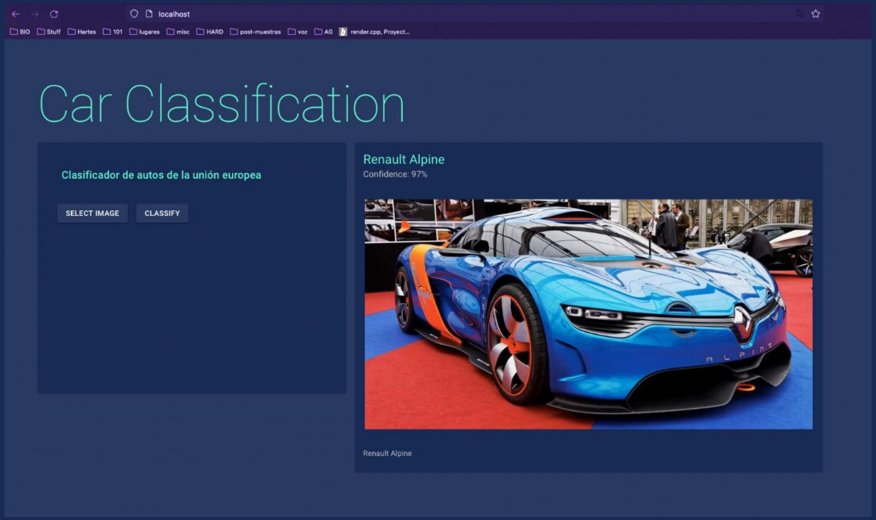
# 1. Objectives Definition

- Online car **<u>marketplaces</u>** need to **process and organize large volumes of images** in their websites.
- **<u>Users</u> need to reduce the time** they spend on their online bidding process.
- **Cazoo is** being forecast to record a **compound annual growth rate of 209 percent** between 2020 and 2022 (Statista).



# 1.a Analytic Approach

- The business problem can be solved by implementing an **image classification Machine Learning** system that **predicts the make and model** of an uploaded image.
- The system deployed has to **filter out corrupted files and non-vehicle images**.



# 1. “A Systematic Evaluation of Recent Deep Learning Architectures for Fine-Grained Vehicle Classification” by Valev, Sommer and Beyerer

- **Objective:** compare the performance of Convolutional Neural Network (CNN) architectures for classification tasks
- **Dataset:** Stanford Cars-196 dataset.
- **Methodology:** adapt existing CNN for fine-grained vehicle classification transfer learning and data augmentation
- **Evaluation:** compare architectures, approach (from scratch and fine-tuned) and accuracy
- **Results:** DenseNet-161 had the highest accuracy
- **Conclusion:** it’s better to use fine-tuned architectures trained on ImageNet with data augmentation

(a) normal (b) flip (c) rotate (d) blur (e) noise

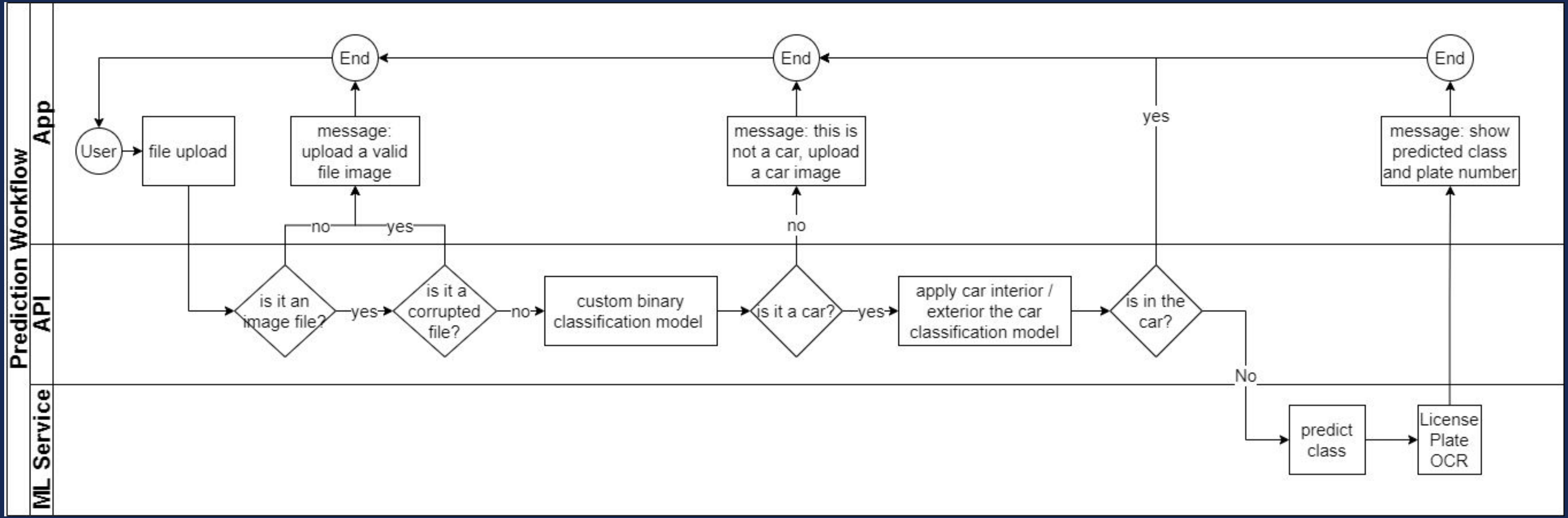
Figure 4: Example of several data augmentation strategies. Note that the blur and noise values have been exaggerated for visualization purposes.

## 2. "Data Augmentation and Clustering for Vehicle Make/Model Classification" by Nafzi, Brauckmann and Glasmachers

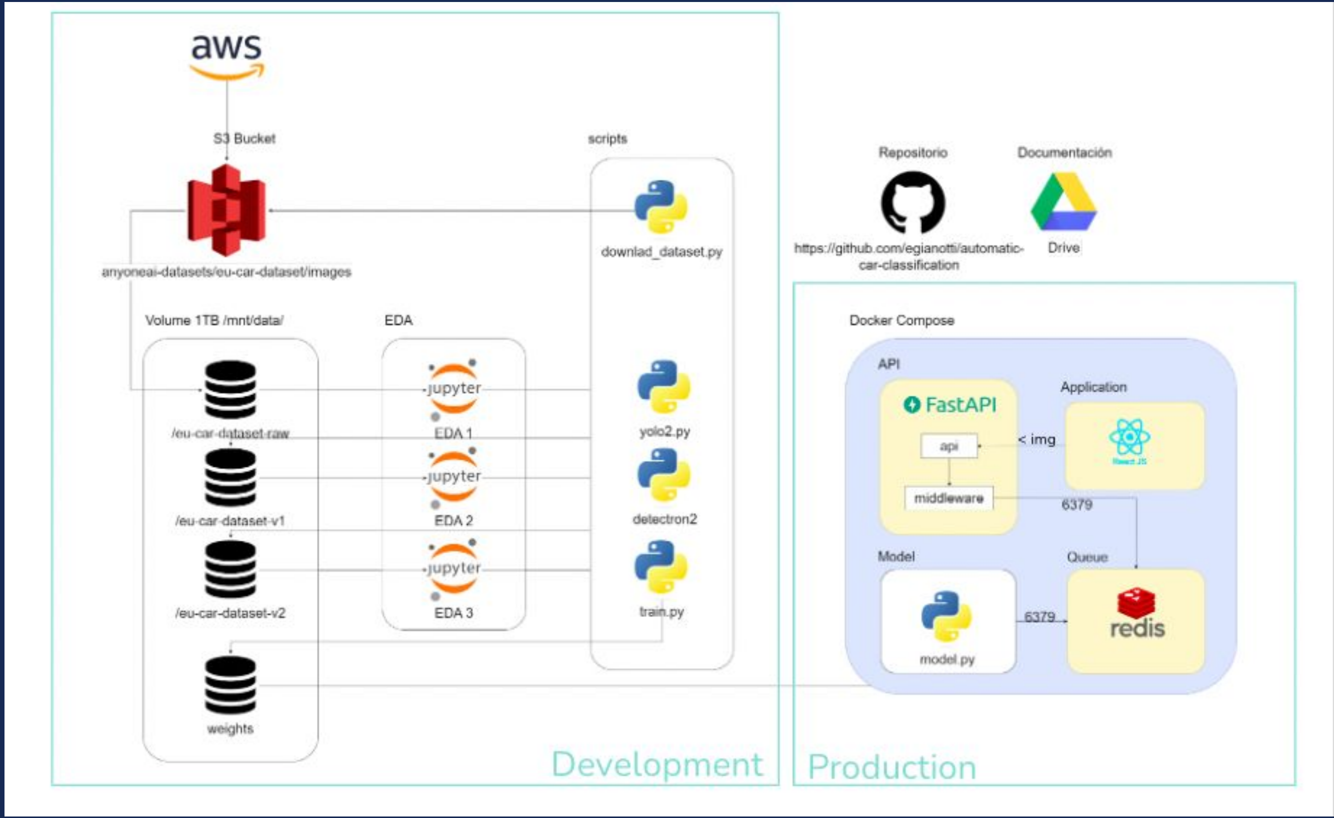
- **Objective:** improve the accuracy of a vehicle classification model by using clustering
- **Dataset:** Stanford data set and video data.
- **Data Augmentation:** web crawler to download data of older car models.
- **Clustering:** matching score of two feature vectors (client score or impostor score) to increase classes for each year
- **CNN-Architecture:** based on Res-Net
- **Conclusion:** clustering is useful for data augmentation with vehicles with different released years and/or different perspectives.

**Fig. 1.** Examples of the generated classes after data augmentation and clustering by Mercedes-Benz C.

# 3. WorkPlan
## a. Designed WorkFlow



# 3. WorkPlan
## b Designed Architecture

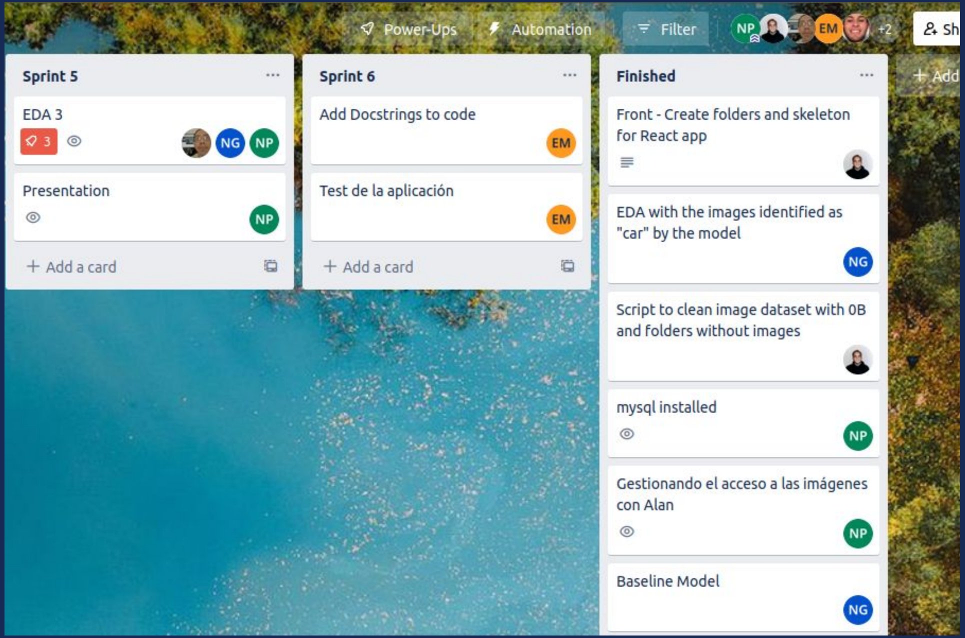



anyoneai.com

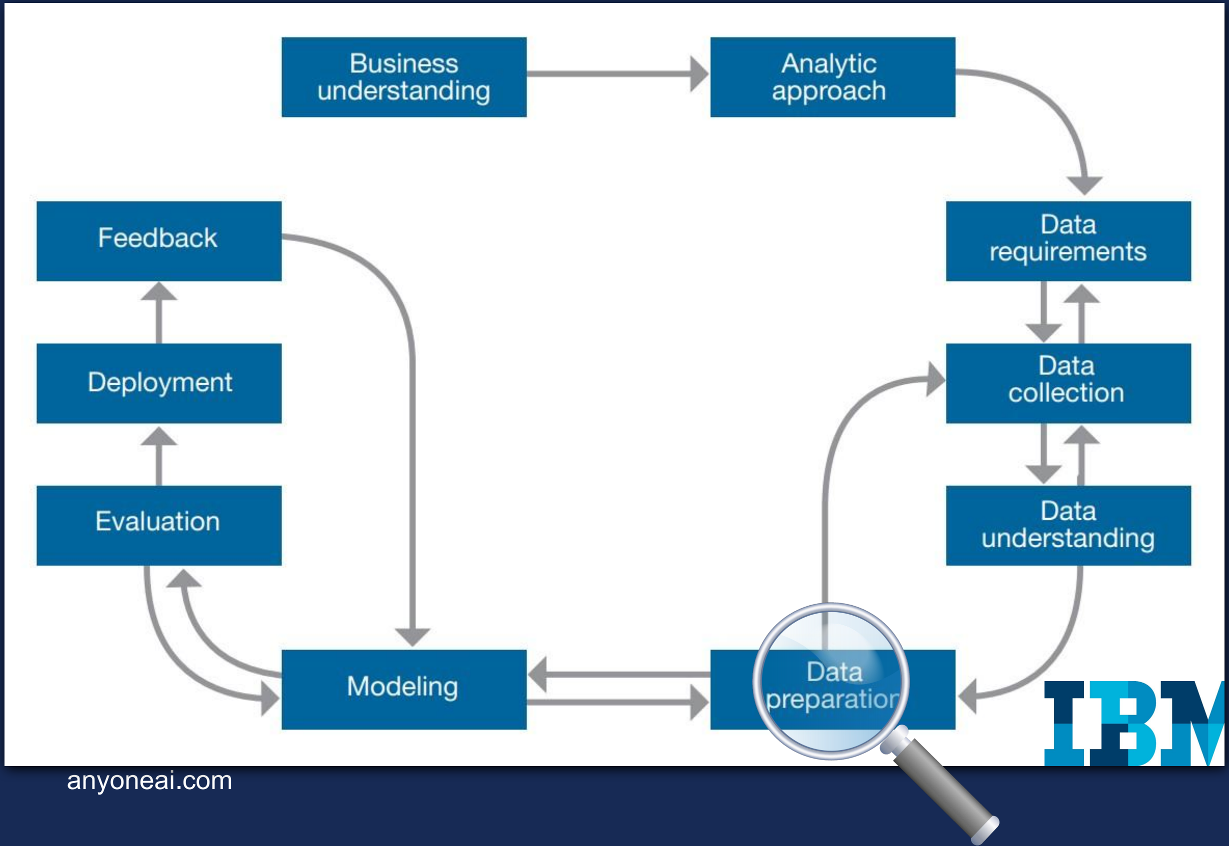
anyoneAI

# 3. WorkPlan
## c Definition of profiles

- Definition:
    - Objectives
    - Tasks and order of their execution
    - Task compliance monitoring tools
    - Sprint times
    - Profiles and responsibilities
- Objective:
    - Manage knowledge efficiently
    - Optimize time

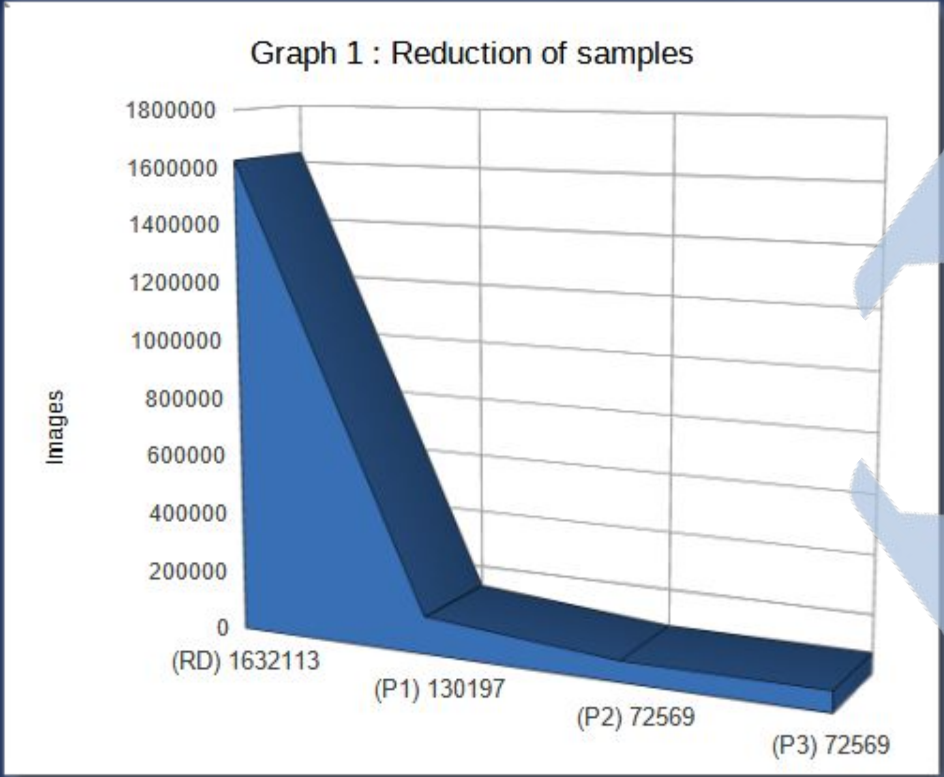


# Fundamental Data Science Methodology



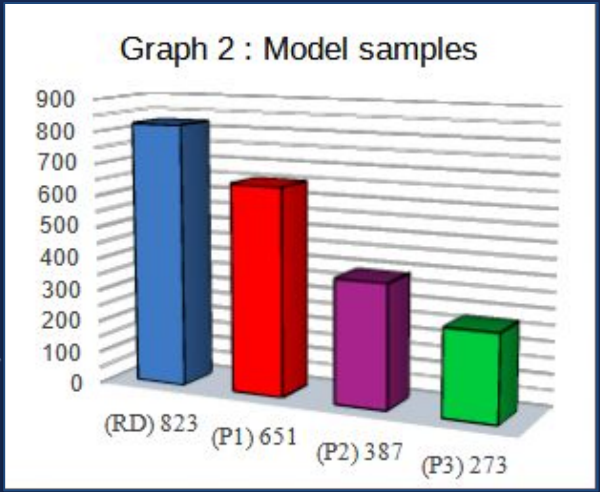

The methodology consists of 10 steps that form an iterative process to discover insights.

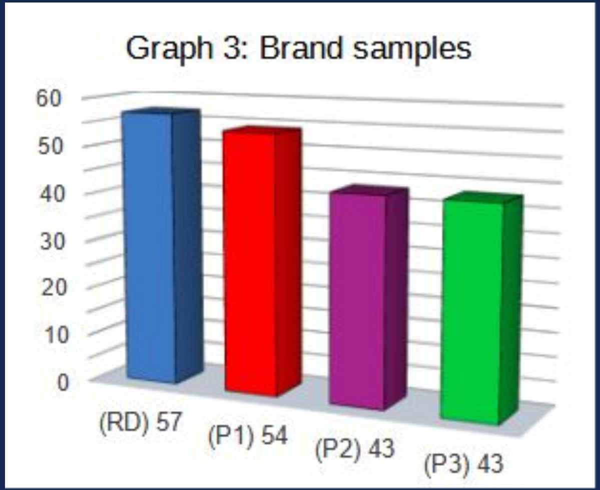
# 4. Exploratory Data Analysis (EDA)




Models

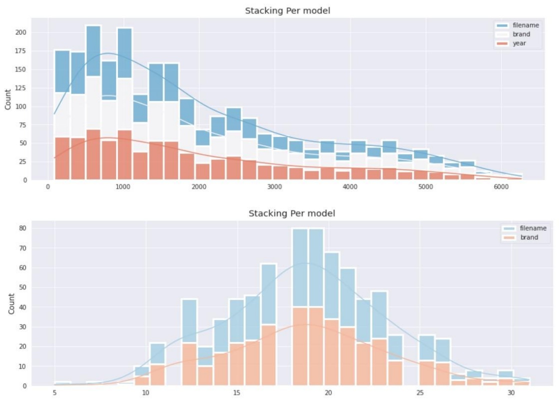
Brands



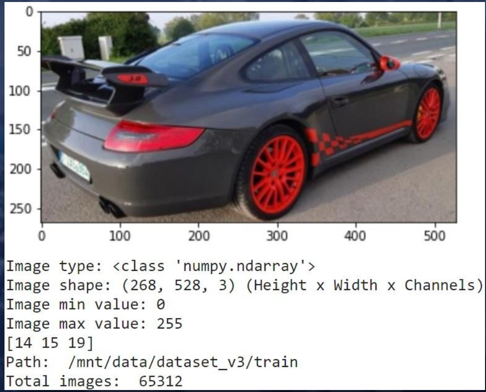




**(P3) = ~72 images 273 models 43 Brands**

# b). Problems Found

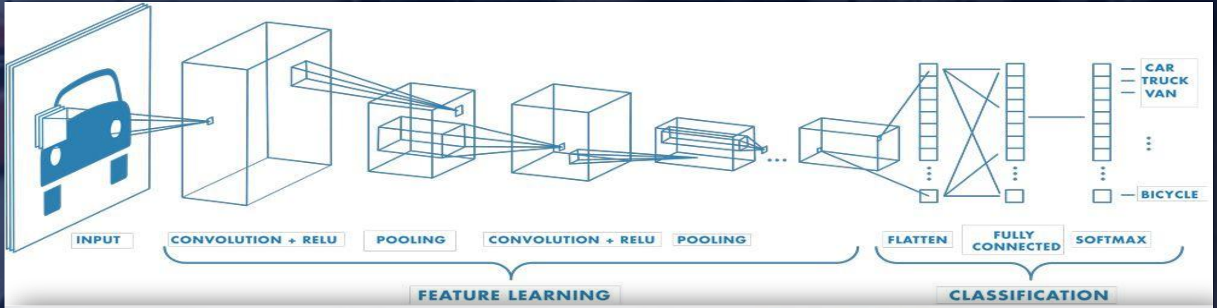


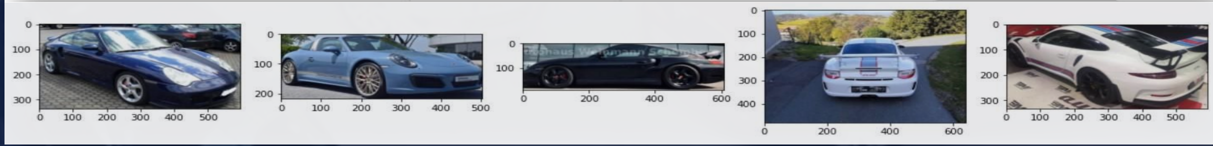

The result of the graphs shows us that there is a positive bias of models, so only 4% of the total images were taken for the final model, improving the final accuracy.




# c). Models

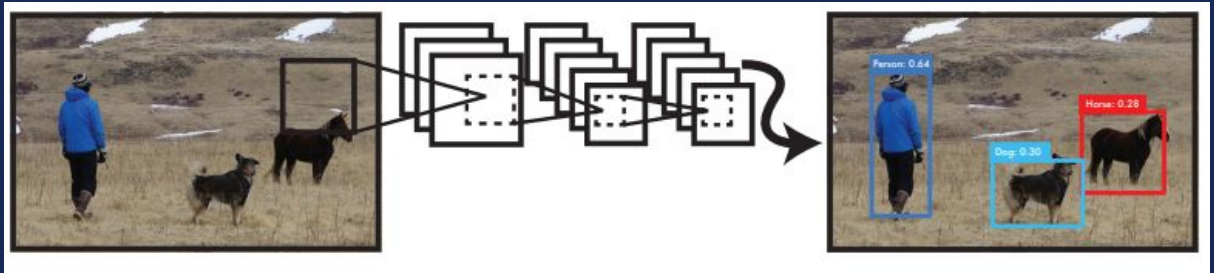
To create our classifier we had to investigate CNNs , which are networks useful for finding patterns in images, object, face and scene recognition



# c. YOLOv7

- Real-time object detection system




- Speed and accuracy
- Clean our dataset from non-vehicles

# d. Remove Background

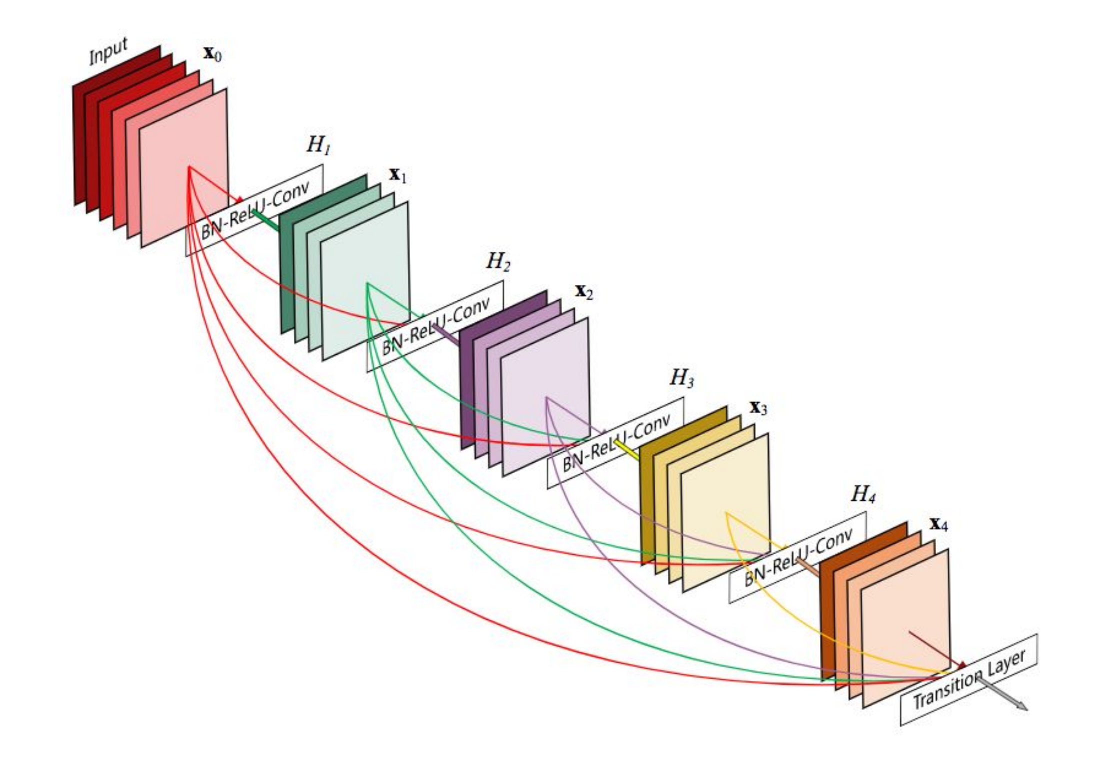
Detectron2

- Improve data quality
- Better model
- Dataset reduced from 27 GB to 14 GB

# 5. Main Model Selected: DenseNet

- A Densenet architecture is composed of Dense blocks.
- A Dense block is a group of layers ***densely*** connected together.
- A single layer has batch normalization, ReLu activation and a 3x3 convolution.
- The feature maps inside each block are the same size
- There is a transition layer between Dense blocks.

# Main Model Selected: DenseNet



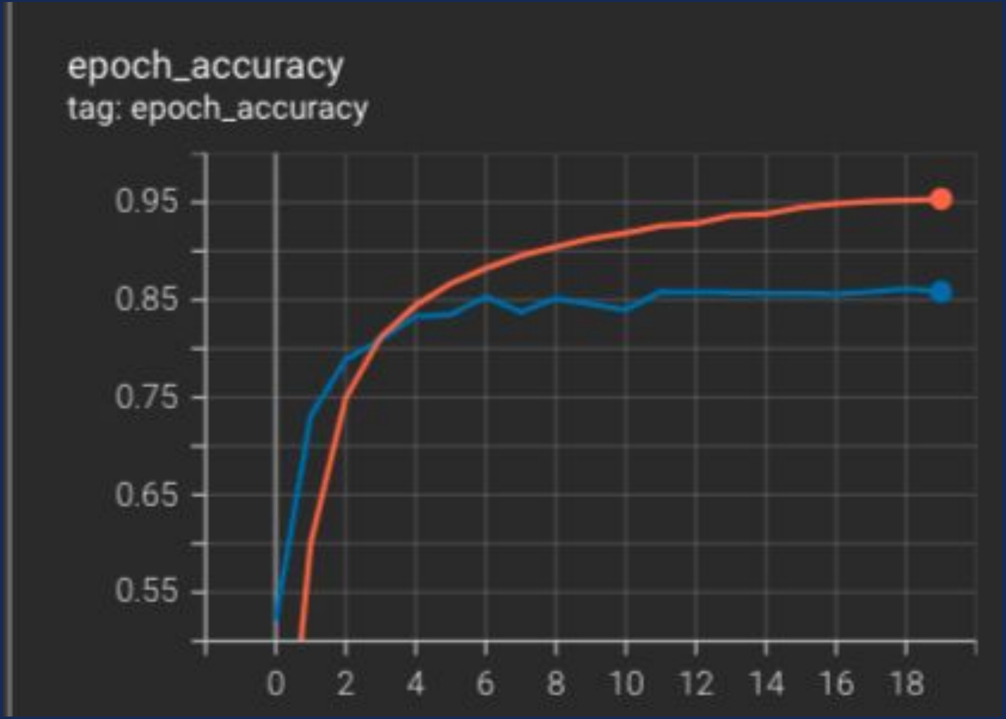

Source: https://arthurdouillard.com/post/densenet/

# 5. Main Model Selected: Model details

- **Densenet201** ~ 14 million parameters, pretrained on imagenet.
- We included a dropout layer and use data augmentation.
- All layers were trained.
- 273 classes
- ~ 72 thousand images without background used for training.

# 5. Main Model Selected:
## a Metrics

- **Test accuracy:** 85%
- **Validation accuracy:** 86%
- **Train accuracy:** 94%
- **Training time:** ~10 hours



# Project Demo

# Conclusions

- With a hardware Mac M1 16GB, the response time of the app was on average between 1 up to 5 seconds.
- The model manages to distinguish between different makes and models of cars.
- The combination of models allows to distinguish the images that are not cars.
- However, using multiple models results in increased response times.
- Despite the low quality of the data, the model predicts with an accuracy greater than 85%.

# Next Steps

- Increase the amount of data by adding images of vehicles.
- Expand the number of classes with vehicles with different released years and/or different perspectives.
- Improve the quality of images with sharpening filters.
- Reduce imbalanced classes in the dataset with more samples using web crawling.
- Experiment with different hyperparameters to improve the model’s accuracy.
- Perform more data augmentation techniques.
- Try other models and architectures.
- Optimize source code to reduce response time.

# Q&A