21st century tool for 20th century business

Michal Riha
3088739

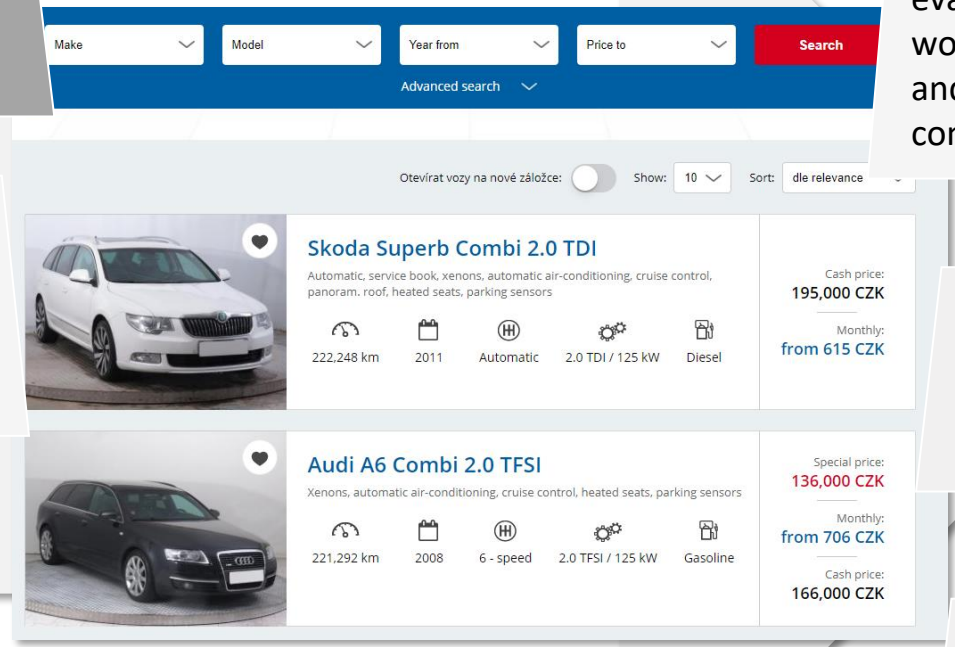
Business focus of the application and users' current pain points

Second-hand car shops processes too opaque and long, NN based pricing app would allow differentiation

All the companies selling second hand cars have a very **complicated** and **opaque** process of **establishing price**

The users **don't know** what they are getting themselves into which causes them to **seek alternatives**

Making the process more transparent would **incentivise** the **sellers** to **prefer one of** the otherwise same looking **services**



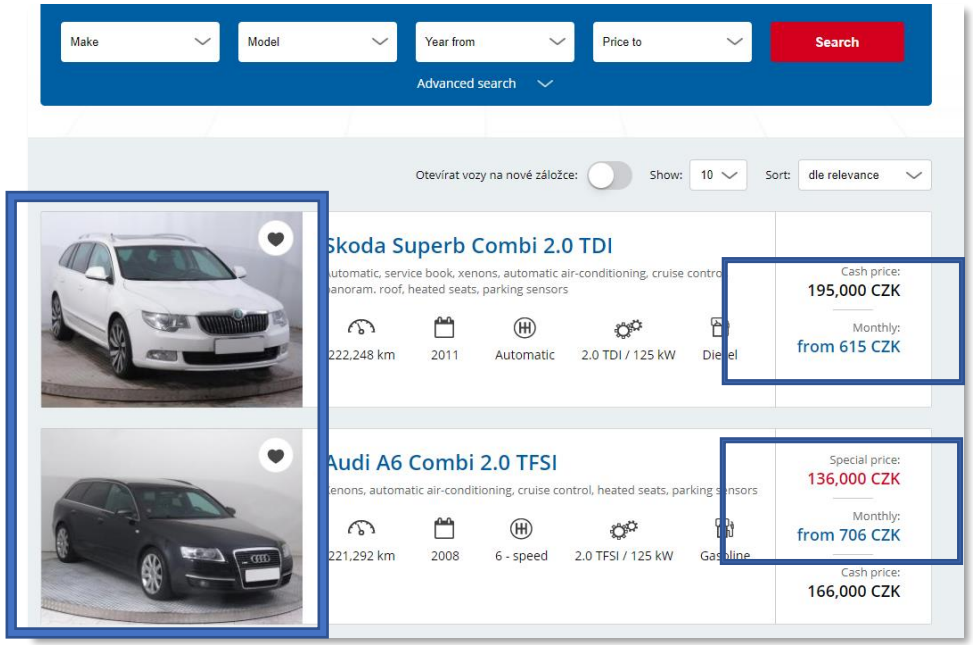
Therefore, usage of a **trained image recognition model** which would be able to evaluate **price levels** of **various models** would make the service **attractive** to users and **shorten** the **customer journey** with the company

The customer would be able to simply **take a photo of the car** at **home** and get an estimate of for **how much** would the **car** be **sold**

The mentioned system would allow company to **position itself** as a **more tech focused** and would **appeal to younger generation**

Data Scraping, Storage and IDE used

AAAuto.cz scraped for car prices and pictures, data stored on Google Drive and processed in Google Colab



Pictures were scraped for the further usage

Prices has been scraped and converted to euros

Data scraping

Proprietary dataset of 4,000 pictures was constructed by **scraping** one of the biggest **Czech second hand car shops** "AAAuto.cz". The cars were later divided into **7 categories** ranging from "€1,000" to "€25,000" according to their **selling prices**

Used storage and IDE

Due to **problems with access** to the remote server of Bocconi (first time login required from campus) the project was carried out **on Google Colab** platform that offers free GPU computing power

Colab was used with **data uploaded** to the **Google Drive** as can be seen in the provided notebook

Libraries and Neural Networks Architecture Used, Outcome of the Model

Keras and Fastai used, ResNet152 identified as best-performer with 60% accuracy on the validation set

Used architecture

Two **different libraries** were implemented to tackle the computer vision problem, **Keras** and **Fast.ai**

In both of them, after a verifying the other architectures, **ResNet** with **152 layers** was established as the **best performing** setup of the neural net

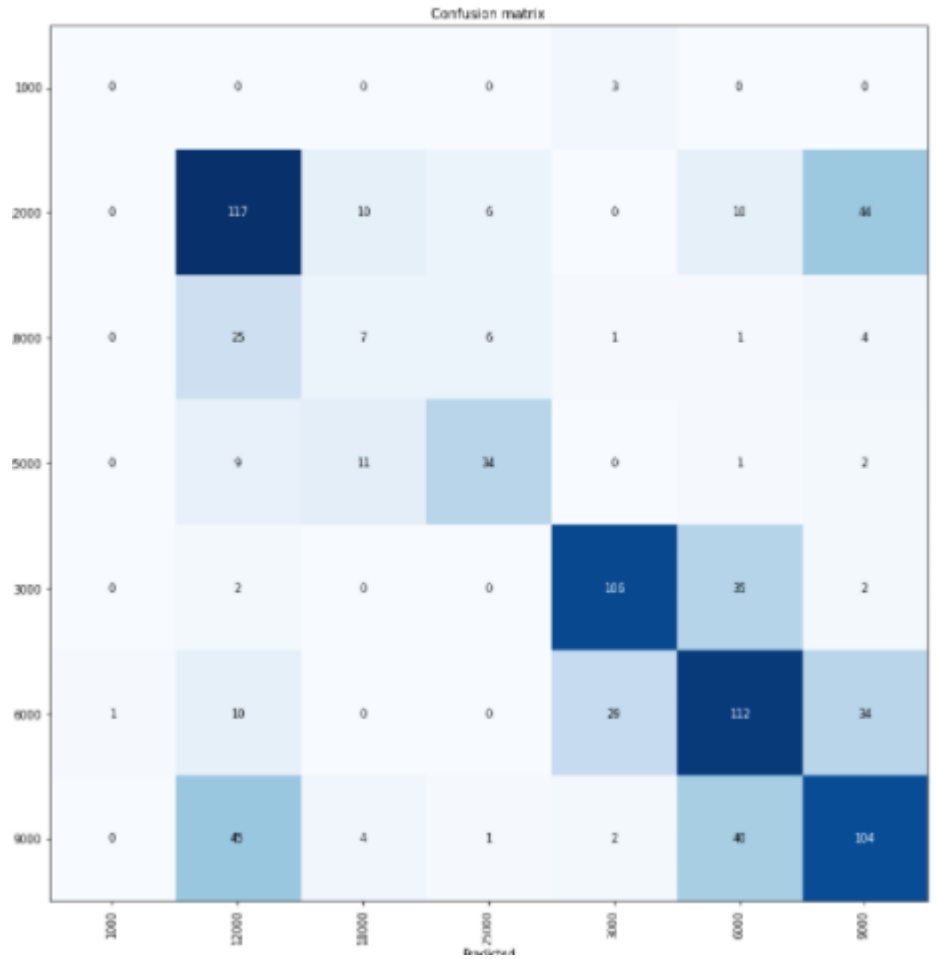
Due to the **small size** of the **dataset** at hand, pictures were **augmented** before the usage to increase variation of the different elements present

Outcome

The **reason** for dual implementation of the same architecture was the **unsatisfactory performance** of the model, which achieved **maximal accuracy** of **60%** on the validation set from the dataset at hand with Fast.ai

Low performance was mainly due to the **large number** of **classes** (7) with respect to the amount of data collected (4,000 pictures scraped)

The algorithm could be improved by further scraping of the different attributes of the cars (such as engine types etc.) and combination of the vision model with the categorical data input



epoch	train_loss	valid_loss	accuracy	time
0	0.758390	1.004593	0.575795	02:02
1	0.754611	1.009688	0.580685	02:02

Image Generation for Better Customer Experience

WGAN was trained on the same data, 7hrs of training was not enough for fully recognizable images

Used architecture & Outcomes

Fast.ai implementation of the GAN model was used for the training of the network

The instability of the Colab environment led to breaks in the training that caused multiple restarts, therefore, the longest period without any breaks of 7hrs was achieved

Results can be seen on the left side (the model started picking up the right patterns of the car designs but was unable to converge in the time period given)

Conclusions and Potential Next Steps for Improvement

Model with mixed data sources on top of the computer vision is suggested to improve the performance

Conclusions

Due to the **low amount** of **data** available and the usage of the **simple model** only **containing** the **computer vision** neural network, the resulting accuracy after fine-tuning the pretrained ResNet152 model ended on **approximately 60%**

When it comes to the **image generation** through **W-GAN**, the **low computing power** and the **environment instability** did not allow for proper high resolution generation of new models from the particular categories, nevertheless it was proven that the GAN is able to learn the patterns of the objects in the dataset

Next Steps

The algorithm **could** be **improved** by further scraping of the **different attributes** of the **cars** (such as engine types etc.) and **combining them** with the vision model

Such a combined model with **mixed input** would be without a doubt capable of **increasing** the **accuracy** significantly enough for actual usage

Also, **more websites** than the one used for the project would have to be scraped in order to improve the intended outcome