Predicting a vehicles speed using dashcam footage

A deep learning approach

Florian Wolf, Franz Herbst

Machine Learning using Matlab
Universität Konstanz

January 19, 2021

Table of content

The “comma ai speed challenge”[1]

Motivation

- Here are some motivational words needed

Data collection:

- “comma ai speed challenge” provides two videos:
  - Train video: 24000 frames, shoot at 20 frames per second, including ground truths
  - Test video: 10798 frames, shoot at 20 frames per second, no ground truths, used to applications
- Split train video after 80% with hard cut off (ability the generalize), to get train and test datasets

Initial assumptions

- Use mean squared error (MSE) as a performance measure
- How to evaluate a prediction? Assumptions:
  - MSE ≤ 10: good
  - MSE ≤ 5: better
  - MSE ≤ 3: correct

[1] https://github.com/commaai/speedchallenge

Preprocessing

- Frame size of $(640, 480, 3)$ pixels
- Cut off last 60 pixels, to remove black frame inside the car
- Sample down the frame to half its size, due to computational limitations

IMAGES with arrows are needed

Optical flow using “Farneback pyramid method” [2]

- Global method to solve the optical flow equation

$$\partial_x f \cdot V_x + \partial_y f \cdot V_y + \partial_t f = 0$$

  for an image sequence $(f_t)_t$ with $f_t : \Omega \to \mathbb{R}^3$, for all t, and the (dense) flow field $V : \Omega \to \mathbb{R}^2, \omega \mapsto (V_x(\omega), V_y(\omega))$.
- Uses a downsampling pyramid, to solve the equation for different resolutions of the image
- Parameters for the Farneback method

$$\begin{aligned} \text{pyramid levels} &:= 3 \\ \text{pyramid scaling} &:= 0.5 \\ \text{window size} &:= 6 \\ \text{pixel neighborhood size} &:= 5 \\ \text{SD of the gaussian filter} &:= 1.1 \end{aligned}$$

- Result: **Flow field with** $(160, 105, 3)$ **pixels**

Visualization of the flow field

- Flow field is a two-dimensional vector field
- RGB representation via
  - Transform flow field into polar coordinates $(V_x, V_y) \to (r, \varphi)$
  - Normalize magnitudes r for third channel
  - Values of the second channel are set to 255
  - Multiply angle φ by factor $\frac{180}{2\pi}$ for first channel
- Sample down the size again, to speed up the training

IMAGE REQUIRED

Convolutional neural network and initial architecture

Method selection

- Speed prediction is a **non-linear regression** task $\rightsquigarrow$ Neural network
- Use convolution layers to perform feature extraction $\rightsquigarrow$ **convolutional neural network** (CNN)

Initial architecture

- Paper of NVIDIA work group [1] of a CNN for self-driving cars
- Enough complexity and layers to handle the task and lots of possibilities to fine-tune it
  IMAGE OF THE MODEL
- Initial results with the raw model: MSE of under 3 on the training set and around 18-20 on the testing set
  $\Rightarrow$ Improvements needed

Batch Normalization, Dropout layers and pooling

- Test

Literature I

Mariusz Bojarski et al. “End to End Learning for Self-Driving Cars”. In: (Apr. 2016). URL: https://arxiv.org/pdf/1604.07316v1.pdf.

Gunnar Farnebäck. “Two-Frame Motion Estimation Based on Polynomial Expansion”. In: *Scandinavian Conference on Image Analysis* (2003), pp. 363–370.

Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: (Feb. 2015). URL: https://arxiv.org/pdf/1502.03167.pdf.

Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a html.