

# Object Recognition

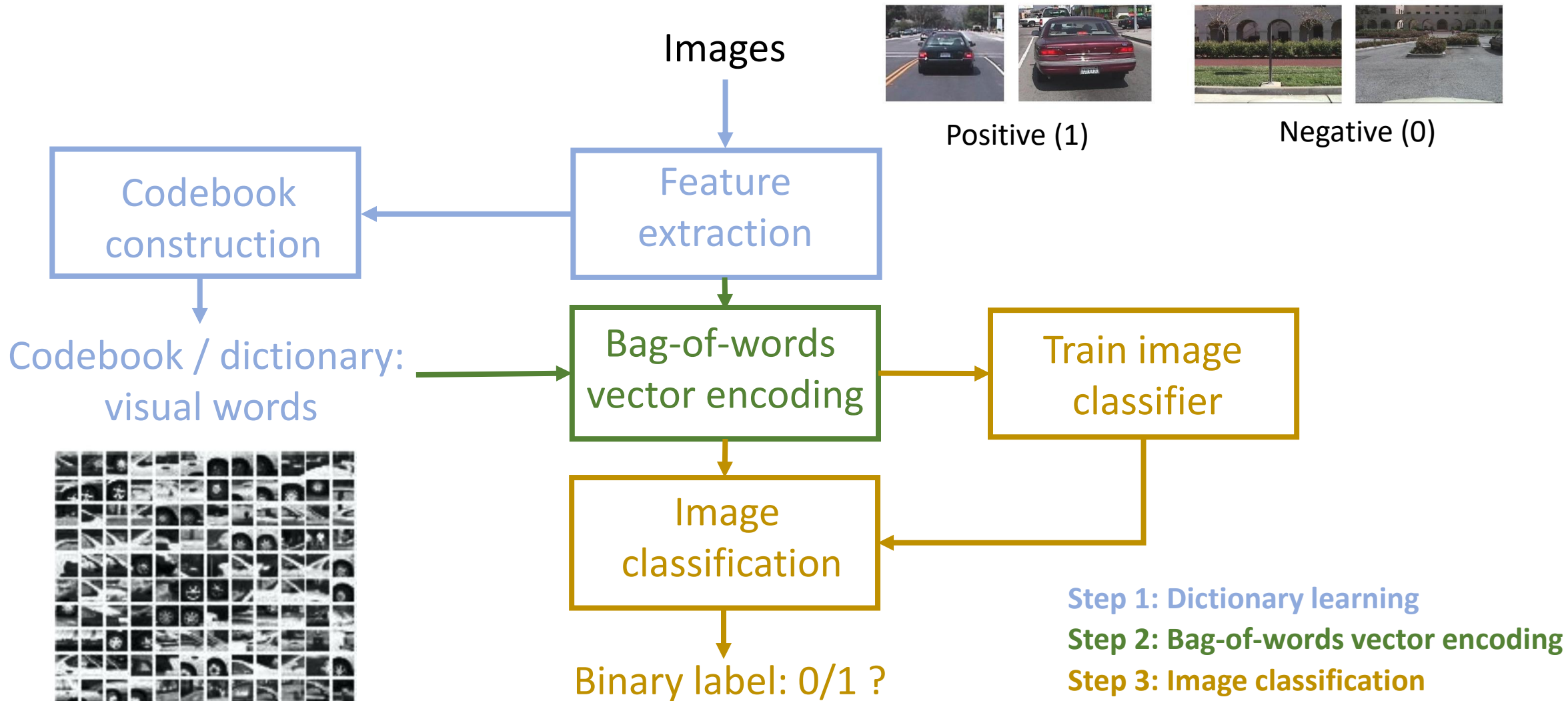
Computer Vision – Exercise 4

Kaifeng Zhao - [kaifeng.zhao@inf.ethz.ch](mailto:kaifeng.zhao@inf.ethz.ch)

# Tasks

- Task 1: Bag of Visual Words (60 pts)
- Task 2: (VGG style) CNN-based image classification (40 pts)

# Task 1: Bag of Visual Words



# Dataset

- **Training set:**

- 50 images – with car, back view
- 50 images – without car



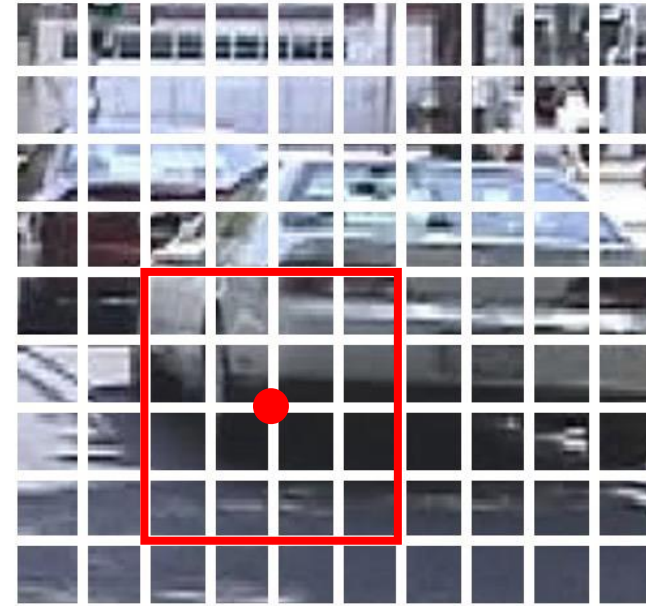
- **Testing set:**

- 49 images – with car, back view
- 50 images – without car



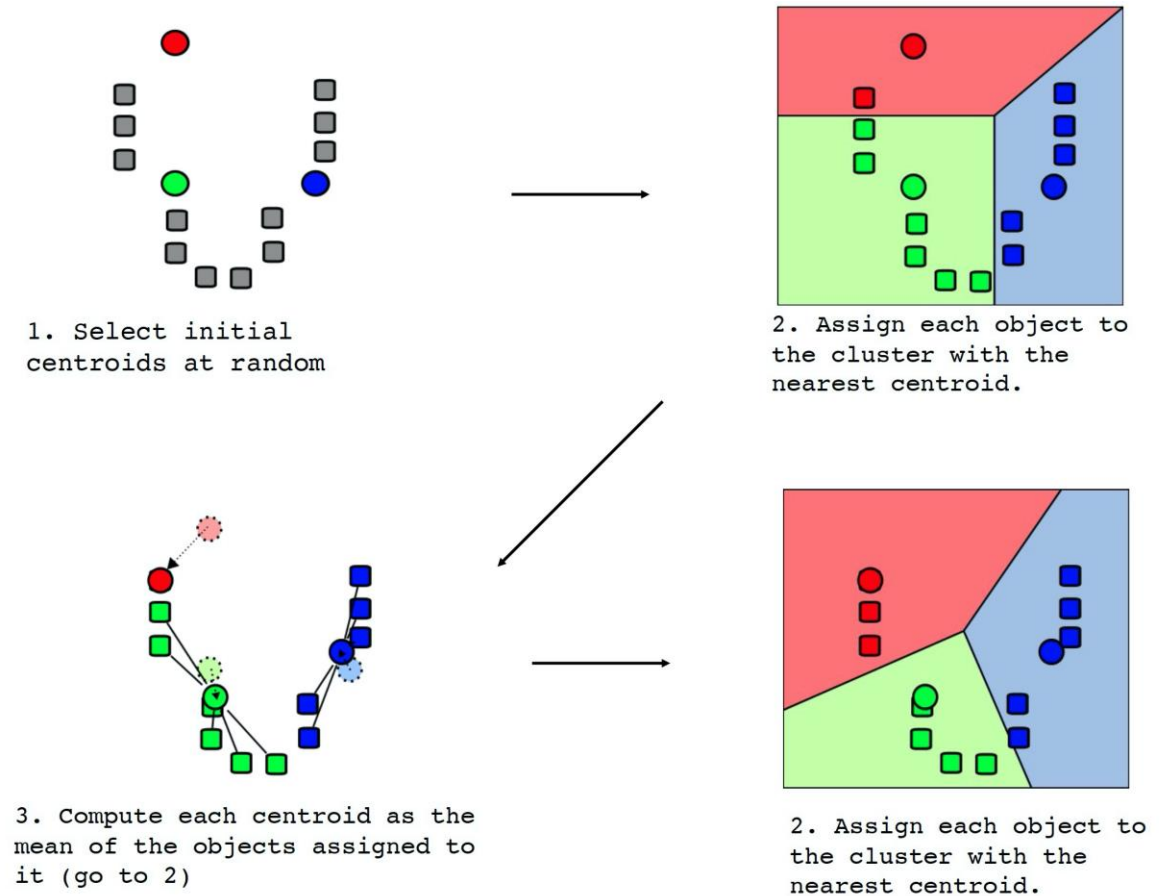
# Local Feature Extraction

- Feature detector:
  - points on a grid
  - dense sampling
  - sample 100 grid points (10x10)
- Feature descriptor:
  - histogram of oriented gradients (HOG) descriptor
  - Defined over 4x4 cells around each grid point
  - Each cell: an 8-bin histogram of gradient orientations
  - → a 128-d feature descriptor for each grid point



# Codebook Construction

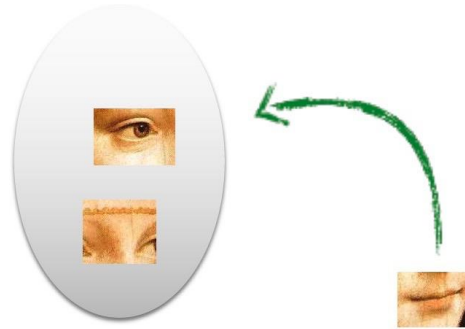
- Codebook/dictionary construction:
  - Clustering by **K-Means**
  - Repeat for a number of iterations
- Cluster center: '**visual words**'
- Ideally: an object part = 1 visual word
- Question: what is the suitable k?



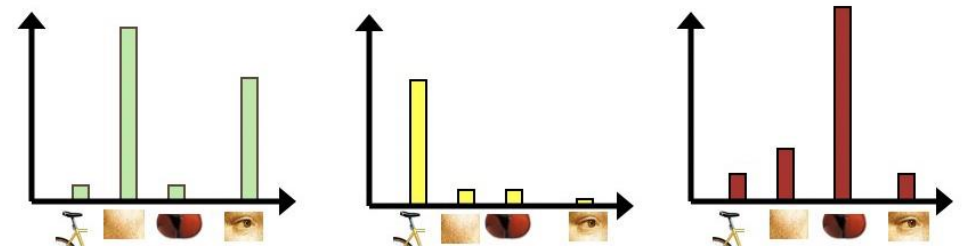
# Bag-of-Words Vector Encoding

Histograms of visual words:

1) Each image feature assigned to a visual word



2) For count # of visual word occurrences

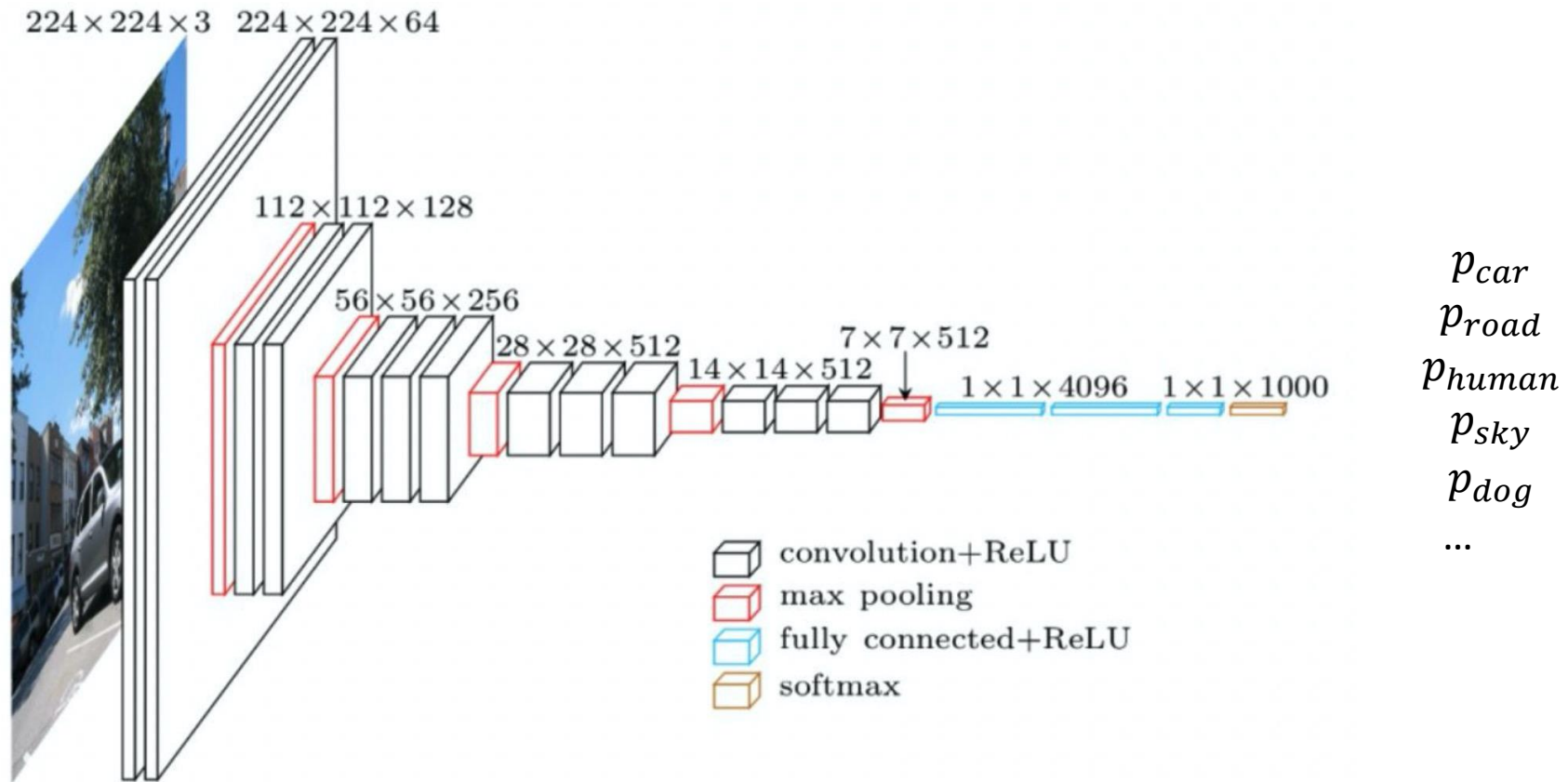


# Nearest neighbor classification

- Training:
  - generate bag-of-words histogram for each training image
- Testing:
  - Given a test image → bag-of-words histogram
  - Find its nearest neighbor training histogram
  - Predict: assign it the category of this nearest training image (0/1)



# Task 2: CNN-based image classification



VGG16 (<https://arxiv.org/pdf/1409.1556.pdf> )

# CIFAR-10 Dataset

- 10 image classes
- 50000 training images
- 10000 testing images
- Image resolution: 32x32

**airplane**



**automobile**



**bird**



**cat**



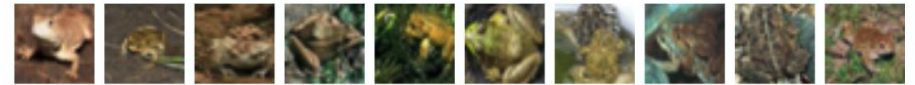
**deer**



**dog**



**frog**



**horse**



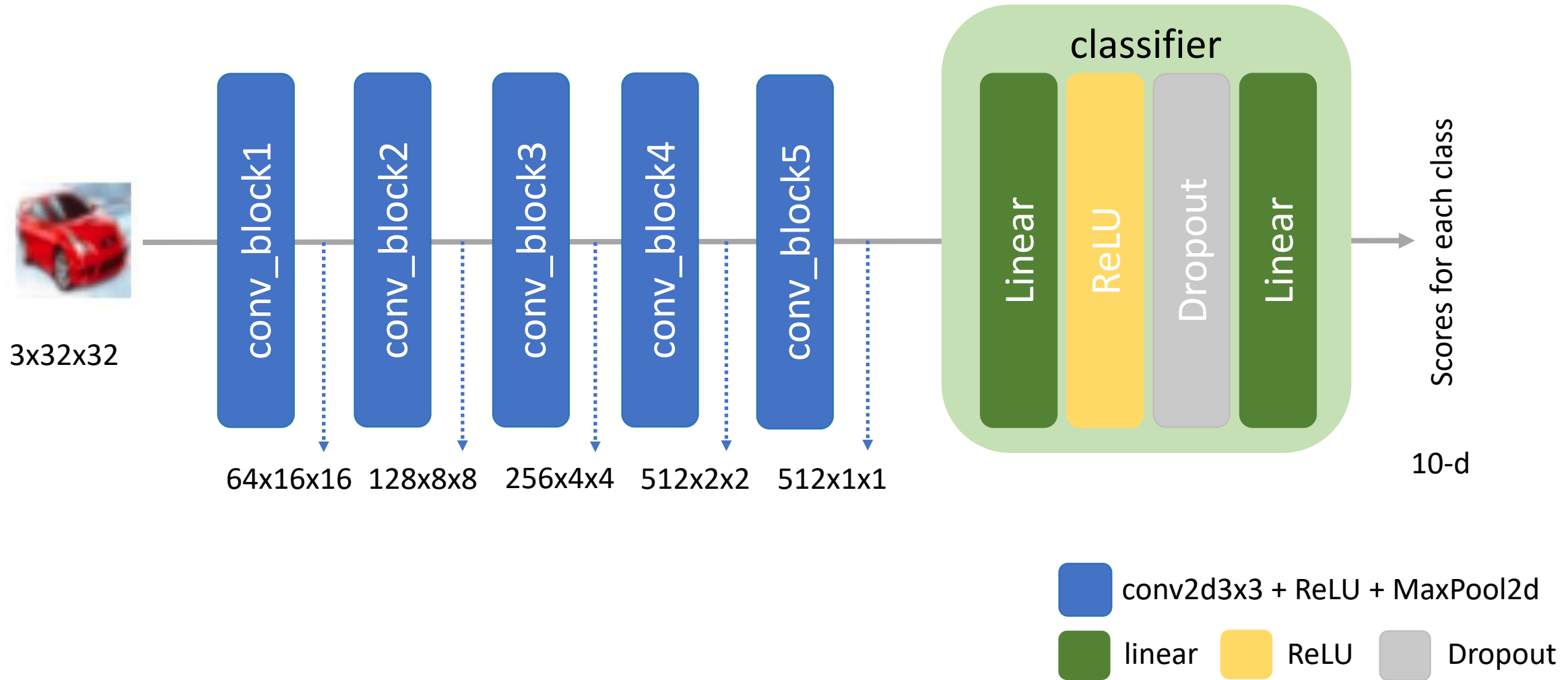
**ship**



**truck**



# Simplified VGG



# Basic Modules

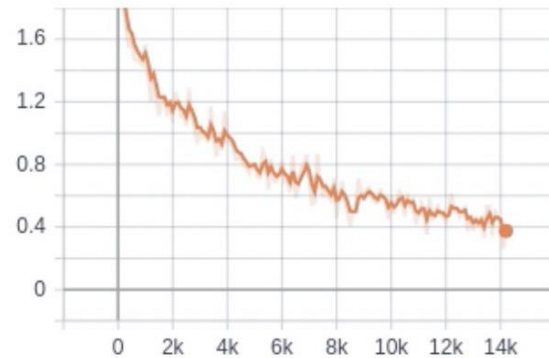
- Conv2d
  - <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>
- MaxPool2d
  - <https://pytorch.org/docs/stable/generated/torch.nn.MaxPool2d.html>
- ReLU
  - <https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html>
- Linear
  - <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html>

# Training

- Loss: cross\_entropy
- Train models / logs saved in `runs/xxxxxx`
  - `xxxxxx`: a random ID for each experiment
- Check Tensorboard logs:
  - `tensorboard -logdir runs`

train

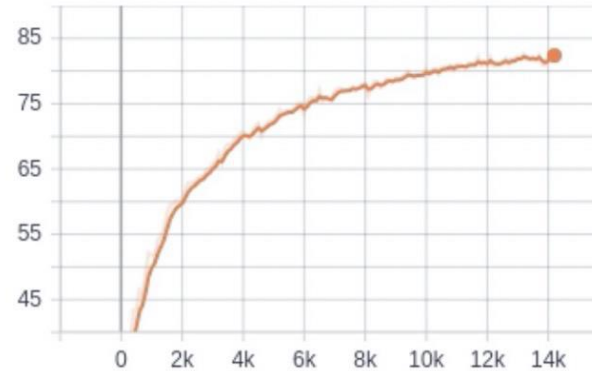
loss  
tag: train/loss



Training loss (40 epochs)

val

acc  
tag: val/acc



Validation accuracy (40 epochs)

# Hand-in

- Assignment 4 is due **10.11.2023** noon (11:59 am)
- All source code (excluding data folder)
- `runs/xxxxxx` folder for VGG model training:
  - `events.out.tfevents...`
  - `last_model.pkl`
  - `params.json`
  - `run_XXXX_XX_XX_XX_XX_XX.log`
- A short report explaining:
  - implementations / results / tensorboard screenshot
- **Please zip all files into one single file for submission**