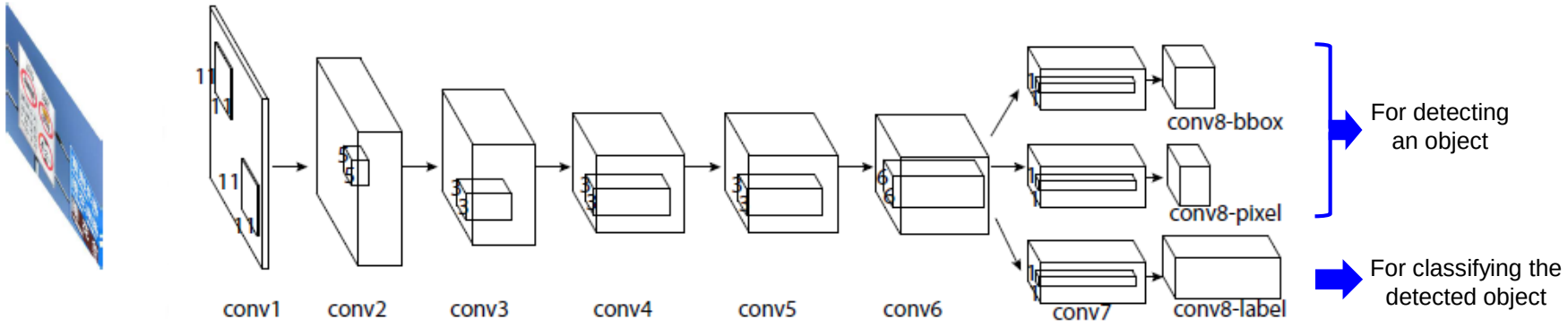
Ways for Speeding Up Tsinghua Tencent 100K-CNN

Sept 27 2017

Tsinghua Tencent 100K-CNN

- **TT100K-CNN Description in Paper†**

†Z. Zhu et al, "Traffic-Sign Detection and Classification in the Wild," CVPR 2016



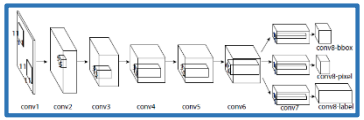
conv8-bbox	Probability of containing an object
conv8-pixel	Bounding box of the object, i.e. x,y coordinate of left-top & right-bottom points of the box
conv8-label	Probability of each class

CNN output in conv8

Note that conv8 is not a final process for making bounding boxes.
Post-processing about conv8 output is needed in order to make final bounding boxes.
This post-processing is not described in both the figure and the paper.

Current Implementation of TT100K-CNN

- **Real Implementation in Publicly Open Code**

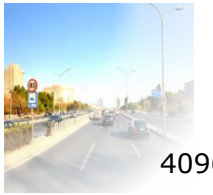


≡ TT100K-CNN : a network whose input size is 1280x1280

Input image
2048x2048
(pixels)

1280x1280

2048x2048



4096x4096

8192x8192

Image Pyramid

TT100K-CNN
1 feedforwarding

TT100K-CNN
4 feedforwarding

TT100K-CNN
16 feedforwarding

TT100K-CNN
64 feedforwarding

Total **85** feedforwarding

CNN output
(output of conv8)

CNN output accumulation

Bounding box generation

(Written in **Python**)

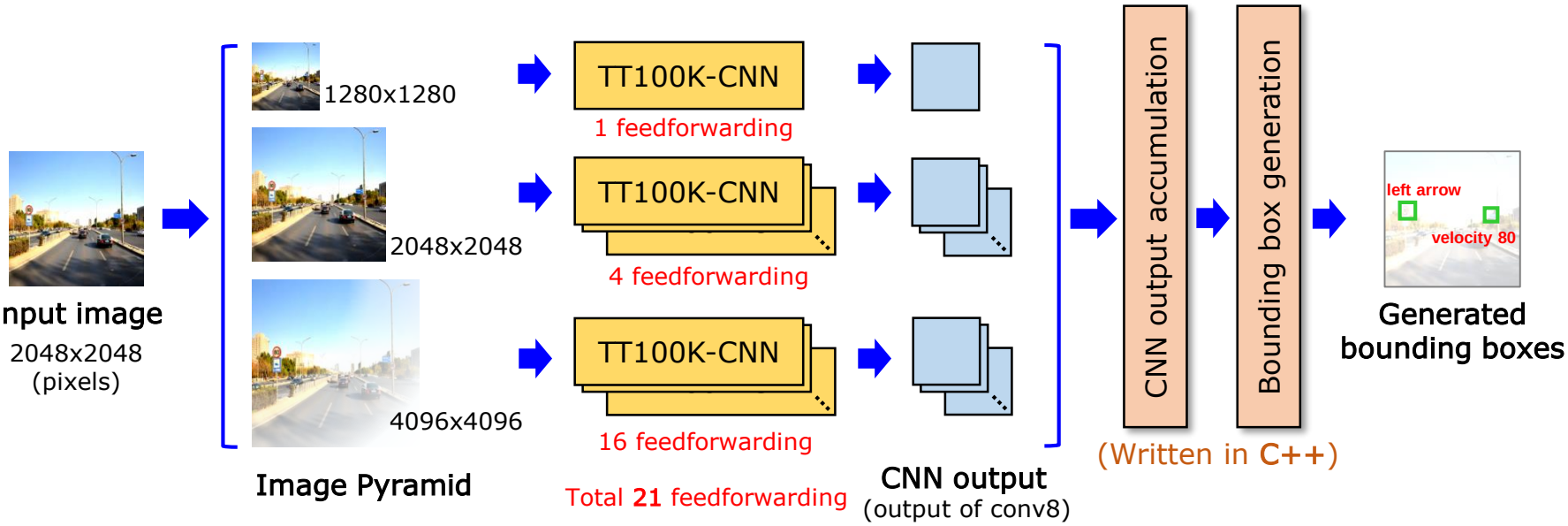
Generated bounding boxes

It takes **27s/image** on a laptop with Nvidia Geforce GTX 1050.

To cover the whole region of a large image in image pyramid, multiple feedforwarding are done for each image larger than 1280x1280.

Speeding Up TT100K-CNN

- **Some Ways for Speeding Up**
  - Adjusting or reducing image pyramid levels
    - ✓ Level 4: 85 feedforwarding → Level 3: 21 feedforwarding
  - Writing a C++ code for post-processing part (which is currently written in a Python code.)



One possible configuration for speeding up