

# Transfer Learning for Convolutional Neural Networks

Paweł Kazimierowicz, Kacper Żuk

## Part I

# Why Transfer Learning is fun and profit?

# What will be talking about?



Transfer learning (Inductive transfer) for Convolutional Neural Networks (ConvNets).

Process where structure or knowledge from a learning problem is used to enhance learning on a related problem<sup>1</sup>.

Example:

Take an image classification model that is capable to correctly label common objects (a dog, a tree or an airplane)

Retrain it to label types of yoghurt on a production lane.

---

<sup>1</sup>West, Jeremy; Ventura, Dan; Warnick, Sean (2007), "A Theoretical Foundation for Inductive Transfer", <http://cpms.byu.edu/springresearch/abstract-entry?id=861>

# First of all - why do we need Transfer Learning?

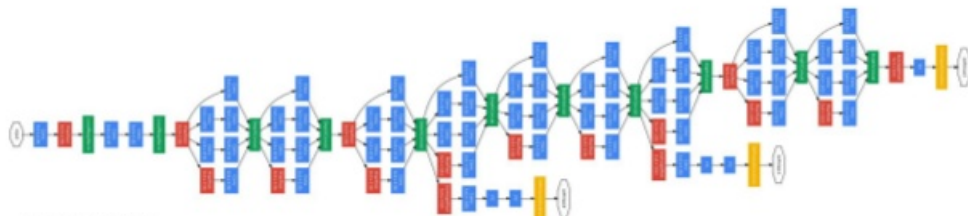


There is a lot of to be designed to build an image classification model:

- first layer:
  - data preprocessing,
  - feature extraction,
- middle layers:
  - what shall be in the middle layers
  - btw, what kind of a network shall it be?
- the last layer:
  - actual classification and labels.

Sounds easy, but in reality it looks something like this...

## Modern ConvNet - GoogLeNet



GoogLeNet (2014)



ResNet-34 (2015)

# But there is a problem...



We usually don't really know what is the best solution to our specific case scenario.

Usual methods consist of trial and error, blind luck, and intuition.

# Big minds have done it better (probably)



Huge research centers and companies have the knowledge and resources to design and train good models:

- they have huge amounts of data (image net is around 14 million of labeled images!)
- these models have been already trained (training on GPU can take weeks)

# Labeling the images takes time and patience...





# Can we just take a pretrained model and use it?



Of course we can, but:

- it will not exactly work,
- there is difference between butterflies, airplanes and printed circuit boards.

# Can we train the defined model on our dataset?



We can take some shortcuts:

- let's say that we have a huge amount of labeled data - that's cool,
- we're no researchers - we can take already designed neural network scheme,
- after training it on our dataset - we might expect good results.

# But what if I don't have a huge dataset?

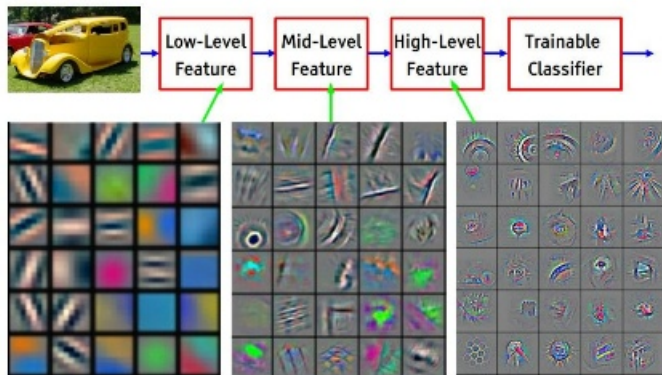


If we have a significantly smaller dataset, we could try:

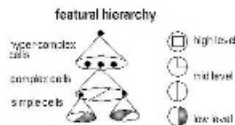
- take the weights from the model as it is already trained,
- retrain it from this point on our dataset,
- depending on how similar our problem is to the original one - that good results.

# How does transfer learning work?

## ConvNet : Interpretation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



# Let's sum this up!



We have three approaches possible to transfer learning:

- take the model as it is - and use it,
- take the model structure and train it on our (huge) dataset,
- take the model and it's weights and train it on our smaller dataset.

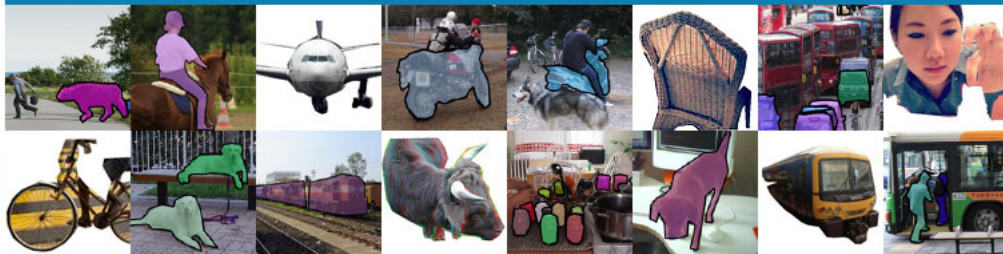
# Examples



For our examples, we've taken Google's MobileNet (<https://arxiv.org/abs/1704.04861>) model, using reference TensorFlow implementation. We've also downloaded weights trained on the COCO dataset (<http://cocodataset.org/#home>). We've only done 1000 steps of training with learning rate equal to 0.003. Our dataset consisted of only 360 labeled photos.

# Examples - COCO

## Dataset examples



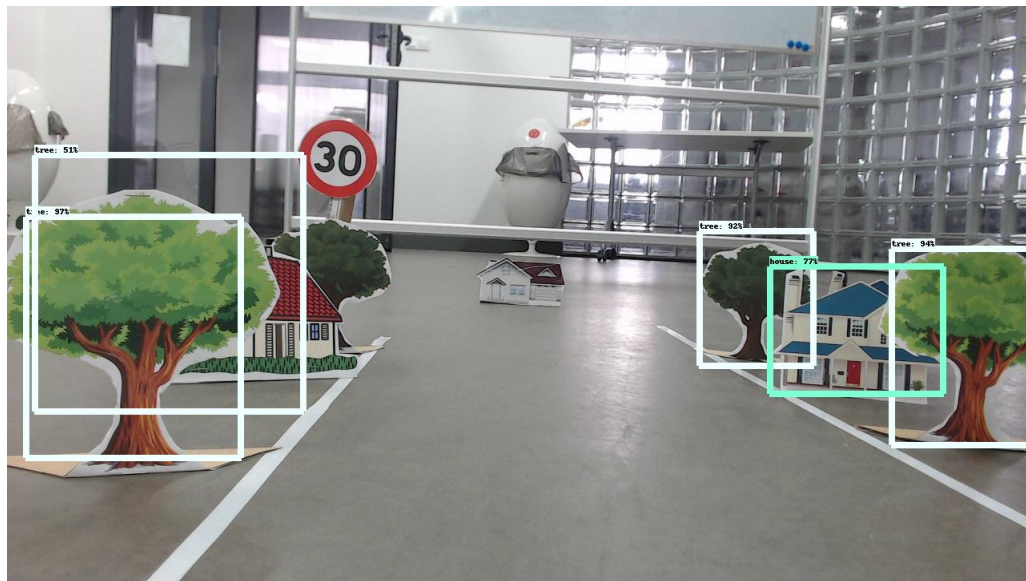
# Examples - first example



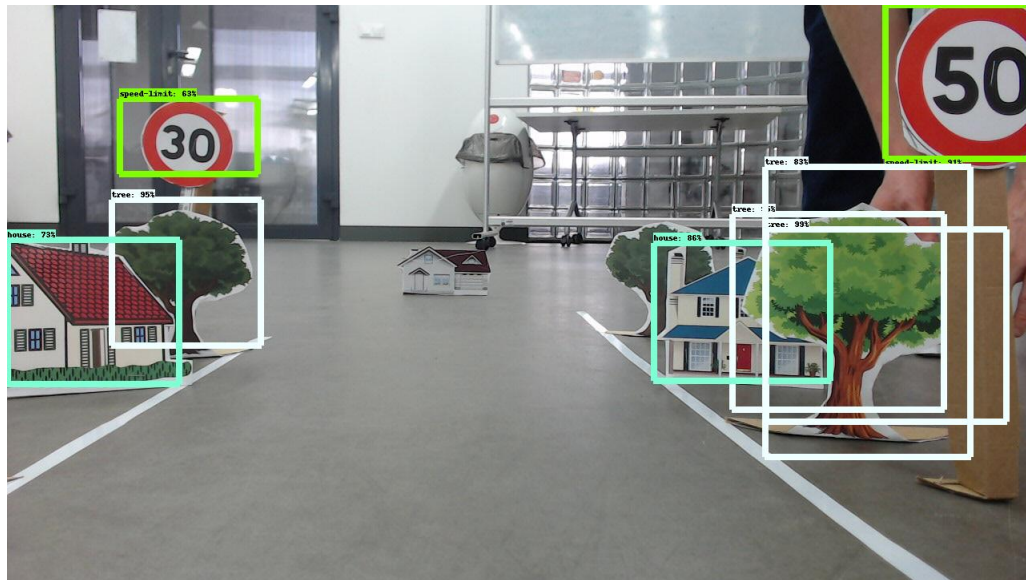
First example used weights trained on the COCO dataset and used this as a base for training on our dataset.



# Examples



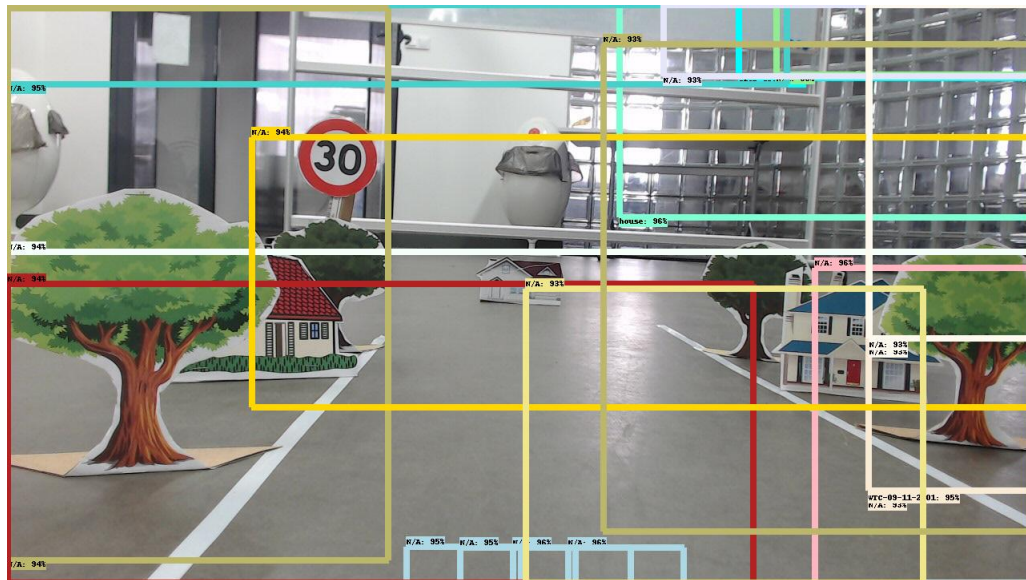
# Examples



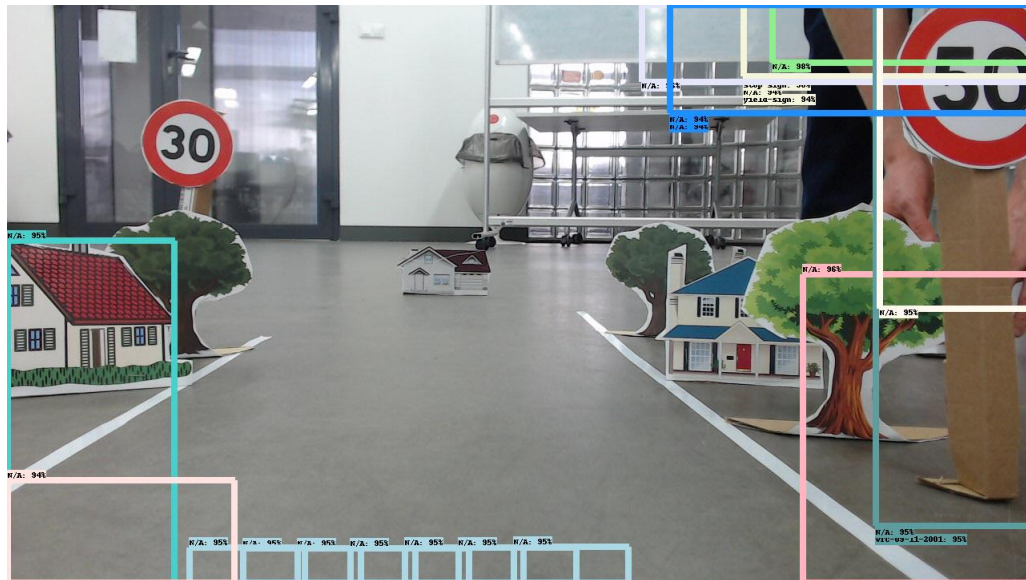
# Examples - second example



Second example used the MobileNet model with weights trained on COCO dataset without additional training on our dataset.



# Examples



# Examples - third example



Third example used the MobileNet model with random initial weights.

# Examples





# Examples





# Examples



Example time!

## Part II

Time for you to build your own *teslas*!

# How to get started easily



Follow the instructions here:

[https://github.com/pkazimierowicz/AI\\_TransferLearning\\_exercise](https://github.com/pkazimierowicz/AI_TransferLearning_exercise)