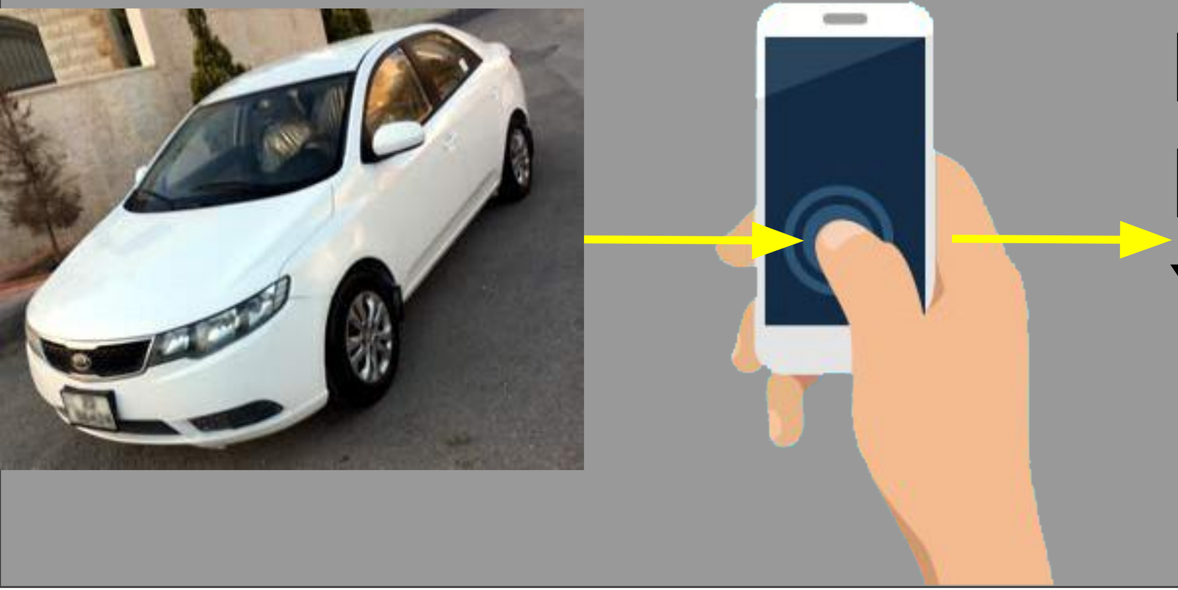
Image recognition by knowledge transfer using deep convolutional neural network

Identifying car makes, models and years.

By Muayyad Alsadi
Supervisor: Prof. Arafat Awajan
Co-Supervisor: Dr. Rawan Ghnemat

The problem

Typing on mobile is very inconvenient annoying compared to just snap a photo



Make: Kia
Model: Cerato
Year: 2009

The flow

- User shoots the item to be sold with his mobile phone
- The application identifies main object in the picture
- Ex. “Car > Honda > Civic > 2006-2012”
- User can adjust auto filled fields ex. Year 2006-2012 into 2008

Deep Convolutional Neural Network

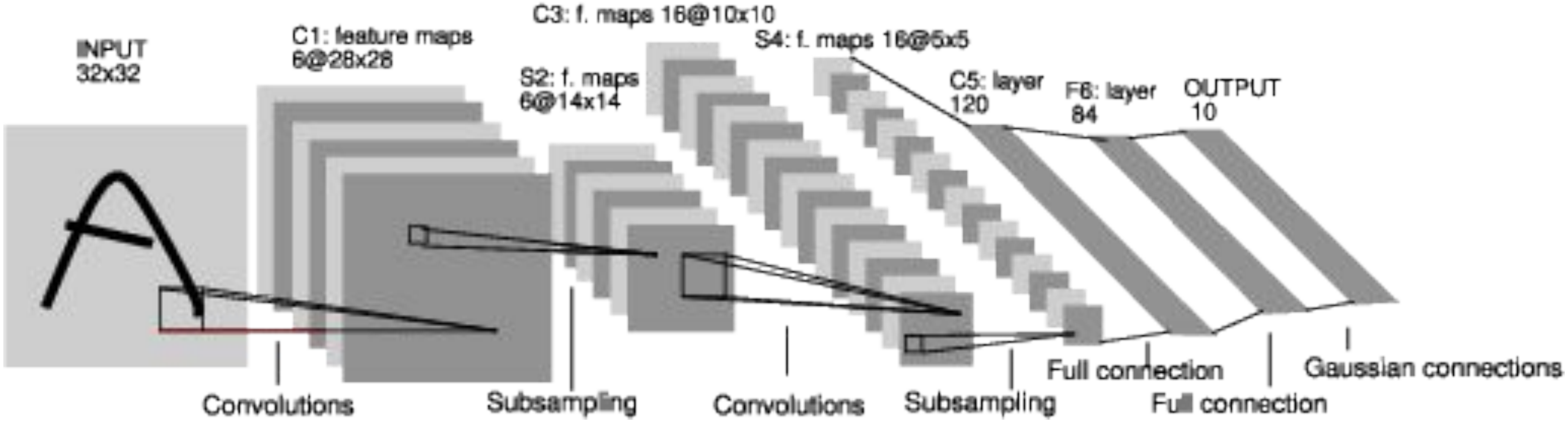
- ConvNets inspired by how visual cortex is assumed to work.
- Pass input image as matrix to multiple trainable convolutional filters of fixed size (kernel size like 5x5)
- Very expensive to train
  - Millions of weights
  - Billions of multiplication-accumulations

Generic or Domain-specific

- Domain specific requires images to be
  - Centered
  - Segmented
  - Same orientation
- Examples:
  - Face recognition
  - Fingerprint scanner

Why Not Domain-Specific

- Different types of items
- Same item different angles
- Different environments (light, background)
- opened/closed
- turned on/off



Simple CNN: LeNet

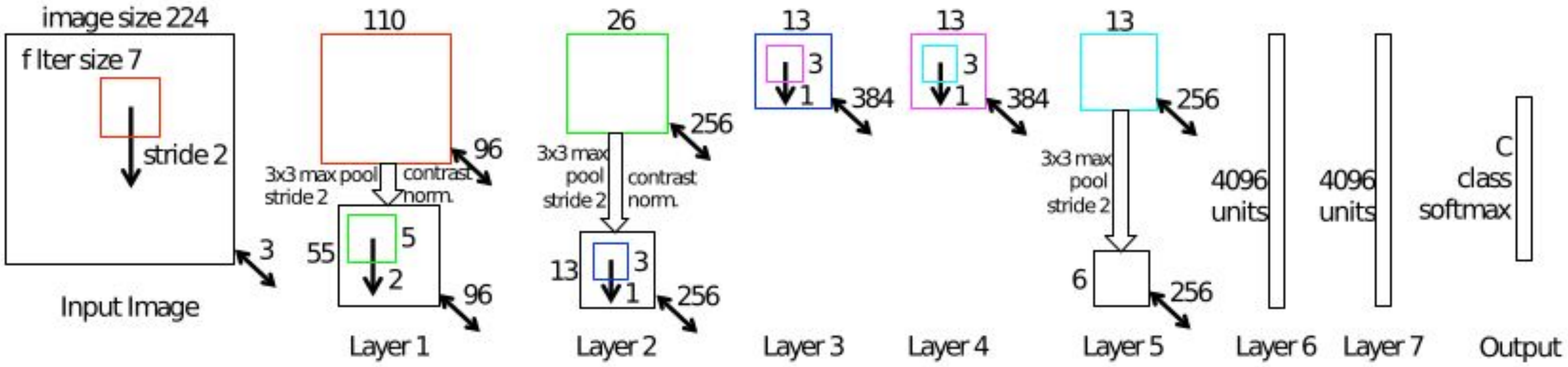


Figure 3.3: ZFNet design
Source: (Zeiler & Fergus, 2014)

More complex CNN: ZFNet

Different CNNs and their cost and accuracy

Table 3.7: Comparing all versions of Inception

Model Name	layers	weights	mults	Top-1 accuracy	Top-5 accuracy
Inception v1	22	6.6M	1,498M	69.8	89.6
Inception v2	42	11M	1,934M	73.9	91.8
Inception v3	53	27M	5,719M	78.0	93.9
Inception v4	81	46M	13,882M	80.2	95.2
Inception ResNet V2	130	59M	14,882M	80.4	95.3

Constraints

- Limited Budget and Resources
  - Train on commodity hardware for short period of time
  - No expensive specialized hardware (GPUs)
- Expandable model
  - Ability to add more classes

Re-using Off the shelf models

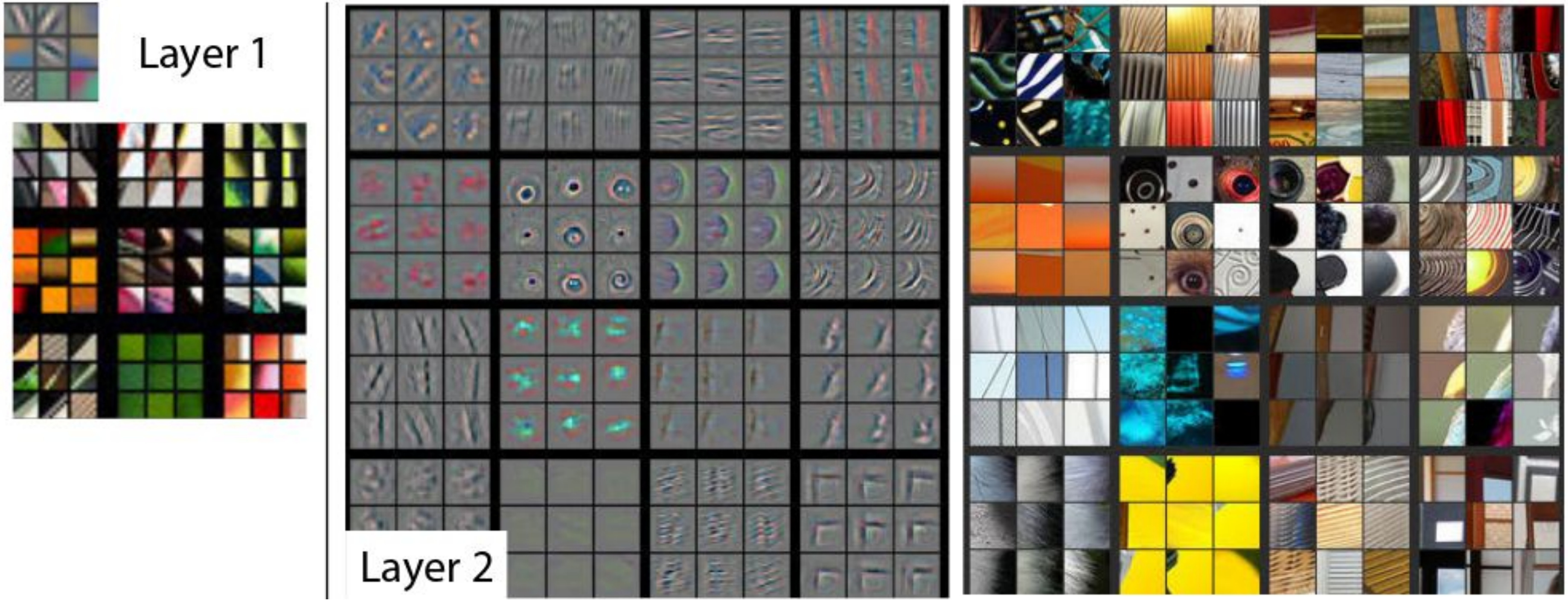
- Models for ImageNet 1K-class task
- Submitted to Large Scale Visual Recognition Challenge (ILSVRC)
- Trained on multiple high-end specialized GPUs
- Trained for many weeks or more than a month
- Publically available

Knowledge Transfer

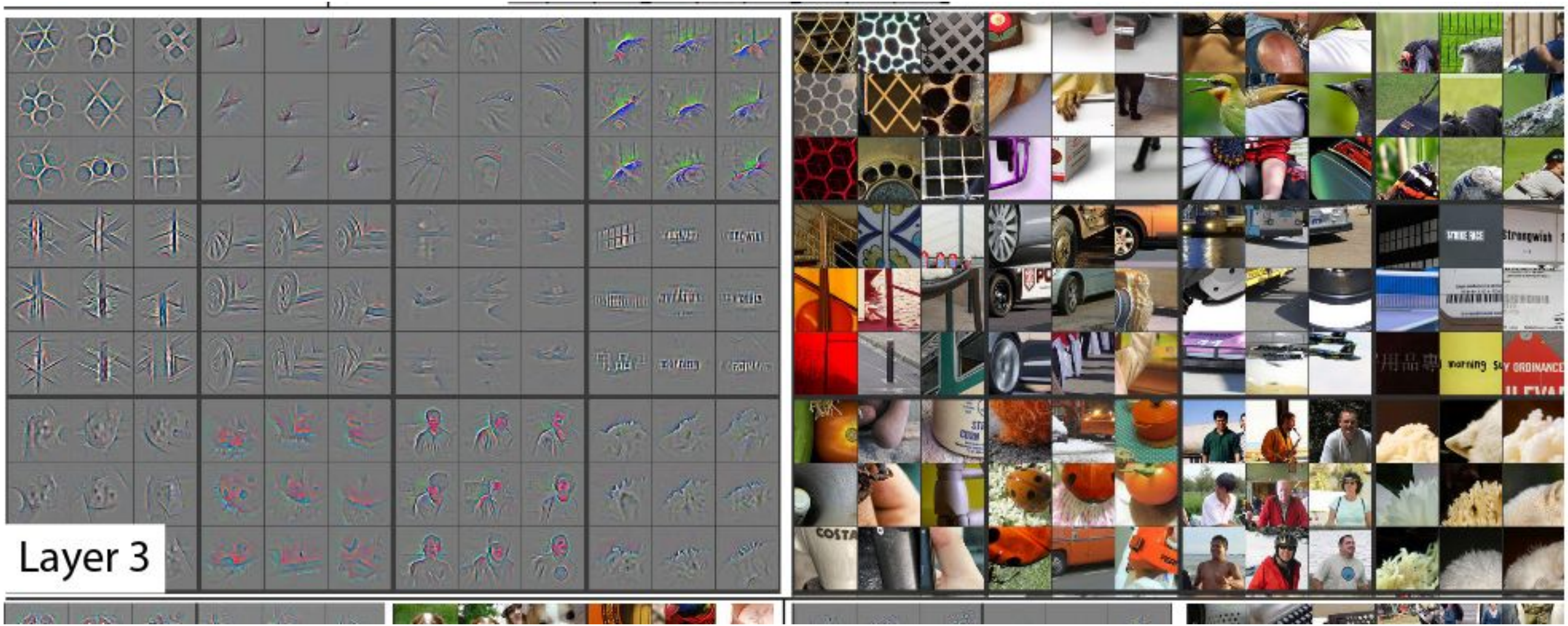
- Re-use off-the-shelf state-of-the-art model (any model for ImagetNet 1K task)
- Transfer knowledge from that model into our task
- One need to address:
  - What to transfer?
  - How?
  - When and why it would work?

Why Knowledge Transfer work

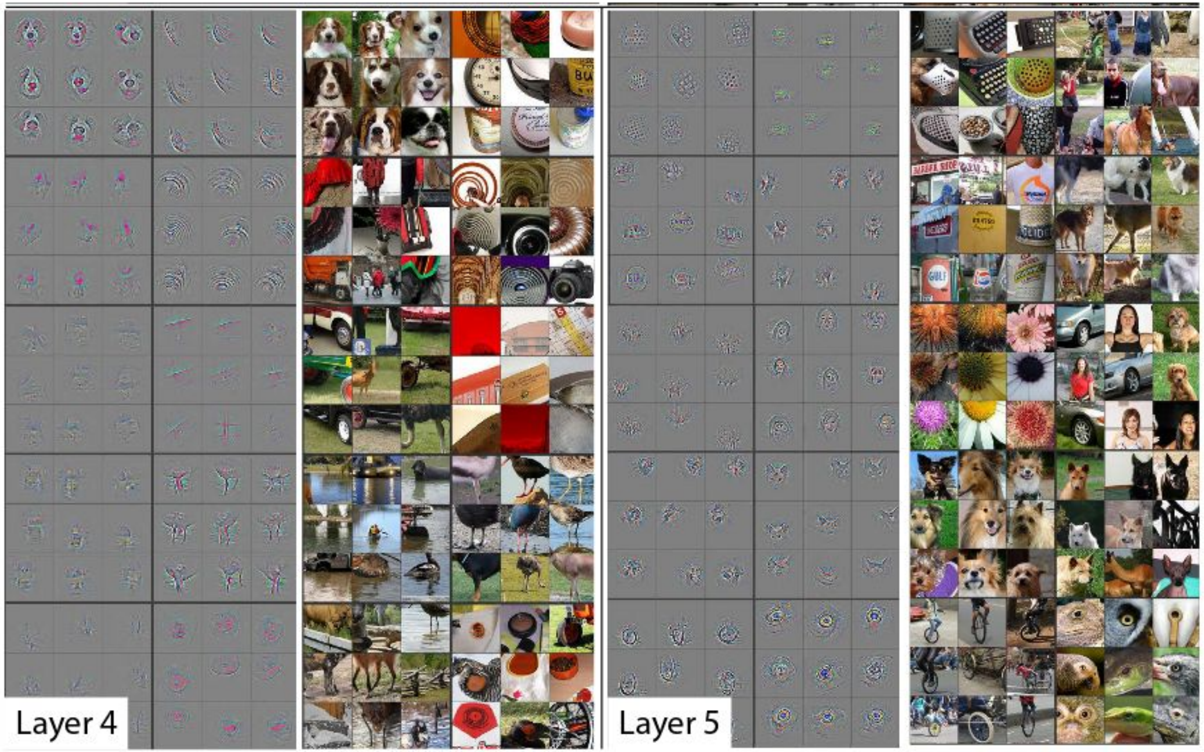
- Several papers showed that:
  - First layers activate primitive shapes
  - Deeper layers activates higher level more composite shapes
- Primitive shapes and intermediate filters can be reused to activate other tasks, only decision making layers are to be customized.



First layers activates primitive shapes and textures
(Zeiler & Fergus, 2014)



Intermediate layers



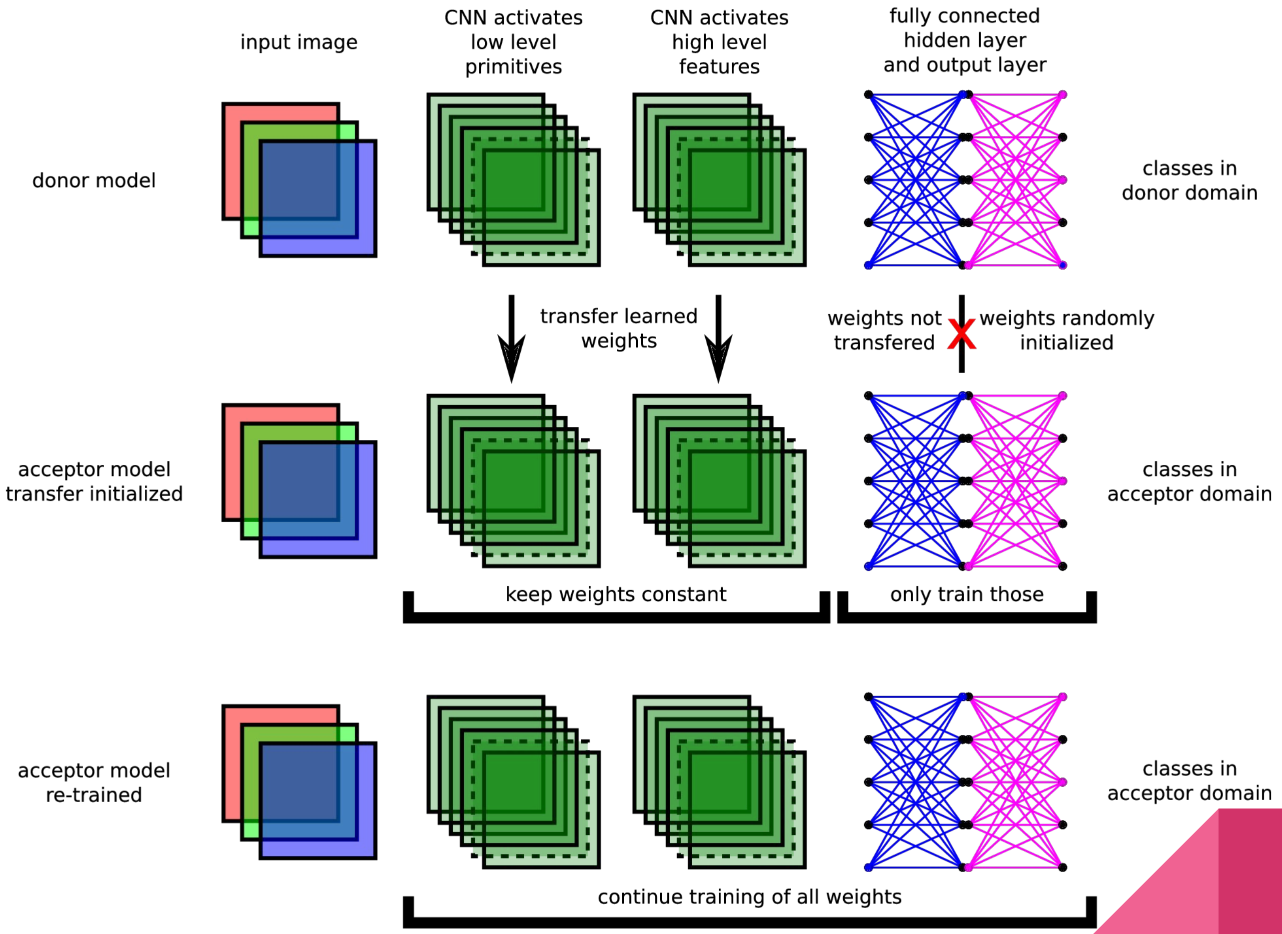
Deeper layers

How?

fine-tuning

- Taking a trained model
- replace last last layer (model head) with random layer
- Train last layer only

How?


input image
CNN activates low level primitives
CNN activates high level features
fully connected hidden layer and output layer
donor model
classes in donor domain
transfer learned weights
weights not transfered
weights randomly initialized
acceptor model transfer initialized
classes in acceptor domain
keep weights constant
only train those
acceptor model re-trained
classes in acceptor domain
continue training of all weights

How?

fine-tuning

- Taking a trained model
- replace last last layer (model head) with random layer
- Train last layer only, keep previous layers frozen.

Not enough? Add extra layers or unfreeze more layer and train them

Noisy Dataset

- Use previously uploaded photos on OpenSooq.com
- User-Generated Content is noisy
- Some images are:
  - registration papers
  - inspection papers
  - Does not show any car identity

Proposed Procedures

- Use Inception V1 trained on Imagenet 1K as source model (donor)
- Use fine-tuning knowledge transfer to train a model to clear dataset
  - Car sides
  - Craft weights for non-cars
- Start by most common car models
- Adaptive batch size to increase speed
- Expand classes by injecting confusion and oversampling of confusion

Results for 229 car models

- Top 1 accuracy: 81.17%
- Top 2 accuracy: 88.20%
- Top 3 accuracy: 90.94%
- Top 5 accuracy: 93.67%
- Precision: 80.89%
- Recall: 79.70%
- F1-Score: 80.29%

Recorded Demo

Live Demo

Thank you