Cardea: A Context–aware and Interactive Visual Privacy Control Framework

Rui Zheng

The Hong Kong University of Science and Technology

rzhengac@connect.ust.hk

October 26, 2016

Agenda

1. Introduction and Related Works
   - Visual Privacy
   - Related Works
2. Convolutional Neural Networks
   - Artificial Neural Networks
   - Convolutional Neural Networks
3. System Design and Implementation
   - Design
   - Model Training
   - Integration
4. System Evaluation
   - Vision Performances
   - Runtime & Energy Consumption
5. Conclusion

Agenda

Rise of visual privacy concern

Pervasive cameras





Protection prototypes

- Respectful Cameras, 2007, cctv, colored markers
- PriSurv, 2008, cctv, RFID tags
- Scanner Darkly, 2013, kinect
- Courteous Glass, 2014, wearable, FIR imagers
- Privacy Tag, 2014, mobile, QR code
- Screenavoider & Placeavoider, 2014, lifelogging, neural networks
- I-Pic, 2016, mobile, neural networks

Research motivation

User studies

- consent mechanism is welcomed when being recorded
- lifeloggers care about the privacy of bystanders
- privacy concerns depend on context: *who, what, when, where, why and how*

- Tamara Denning et al. *"In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies."*
- Roberto Hoyle et al. *"Privacy behaviors of lifeloggers using wearable cameras."*
- Roberto Hoyle et al. *"Sensitive lifelogs: A privacy analysis of photos from wearable cameras."*
- Paarijaat Aditya et al. *"I-pic: A platform for privacy-compliant image capture."*

Research motivation

User studies

- consent mechanism is welcomed when being recorded
- lifeloggers care about the privacy of bystanders
- privacy concerns depend on context: *who, what, when, where, why and how*

- Tamara Denning et al. *"In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies."*
- Roberto Hoyle et al. *"Privacy behaviors of lifeloggers using wearable cameras."*
- Roberto Hoyle et al. *"Sensitive lifelogs: A privacy analysis of photos from wearable cameras."*
- Paarijaat Aditya et al. *"I-pic: A platform for privacy-compliant image capture."*

Research motivation

User studies

- consent mechanism is welcomed when being recorded
- lifeloggers care about the privacy of bystanders
- privacy concerns depend on context: *who, what, when, where, why and how*

- Tamara Denning et al. *"In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies."*
- Roberto Hoyle et al. *"Privacy behaviors of lifeloggers using wearable cameras."*
- Roberto Hoyle et al. *"Sensitive lifelogs: A privacy analysis of photos from wearable cameras."*
- Paarijaat Aditya et al. *"I-pic: A platform for privacy-compliant image capture."*

Research motivation

User studies

- consent mechanism is welcomed when being recorded
- lifeloggers care about the privacy of bystanders
- privacy concerns depend on context: *who*, *what*, *when*, *where*, *why and how*

Limitation of previous solutions

- static policies
- aesthetically awkward
- extra sensors

Privacy design

Axes for design

- problem settings
- technical solution
- enforcement time/level
- protection object
- opt-in vs opt-out

Cardea

- mobile/wearable cameras
- computer vision
- *in situ*/application level
- bystander privacy
- opt-out

Objectives

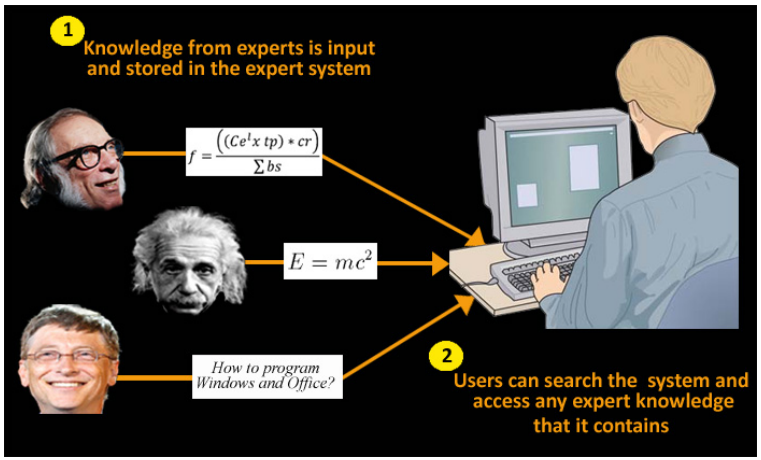
- Context dependent
- Individualized
- Dynamic

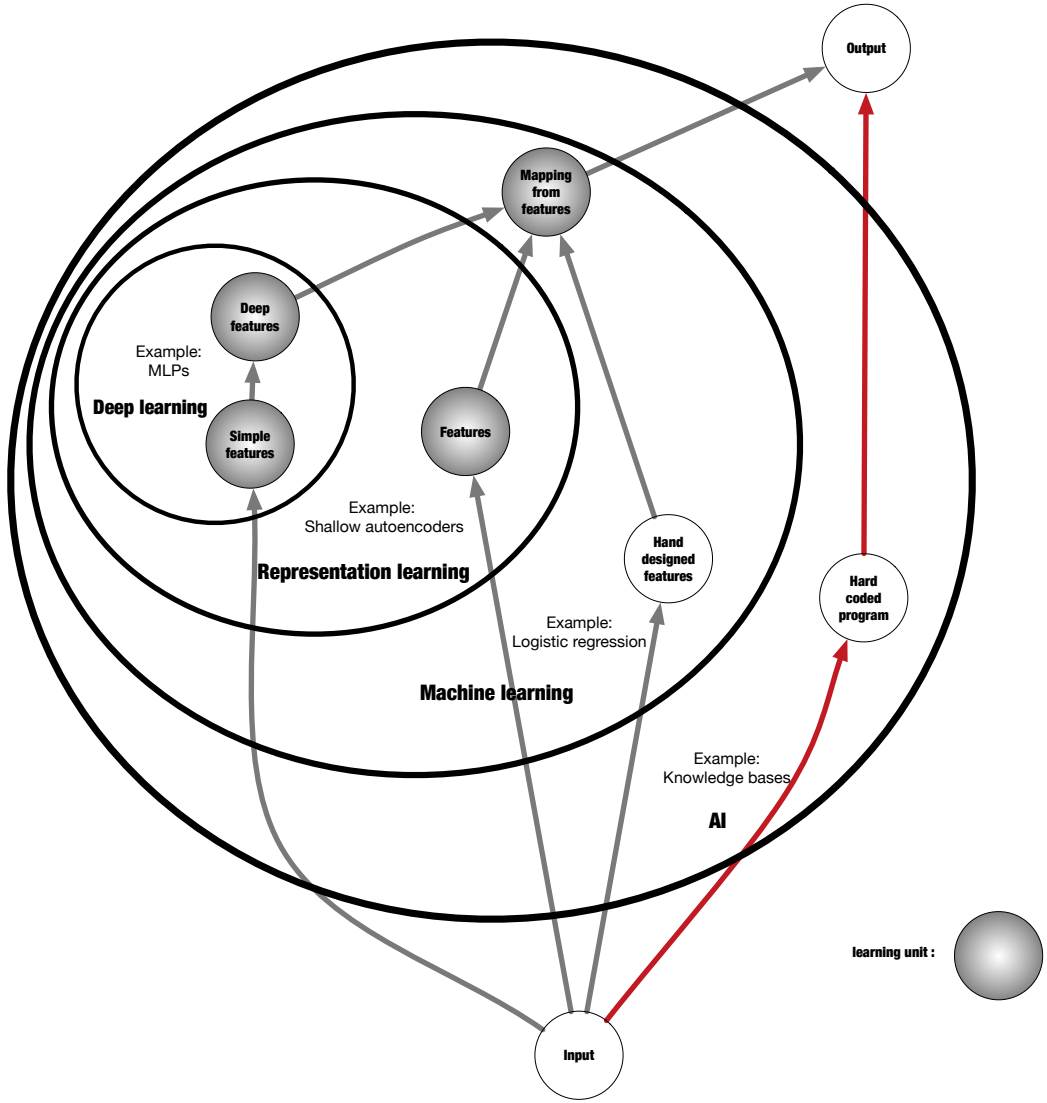
Agenda

Deep learning

Hard coded AI

Requires immense amount of knowledge about the world.





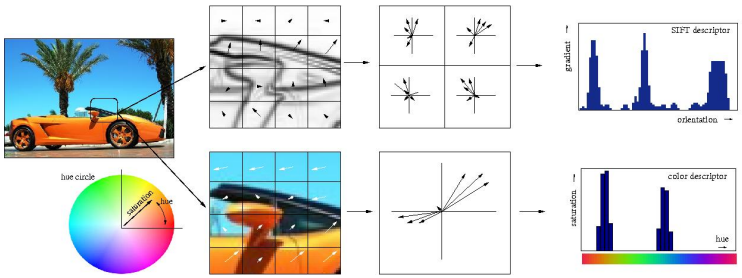
Source:

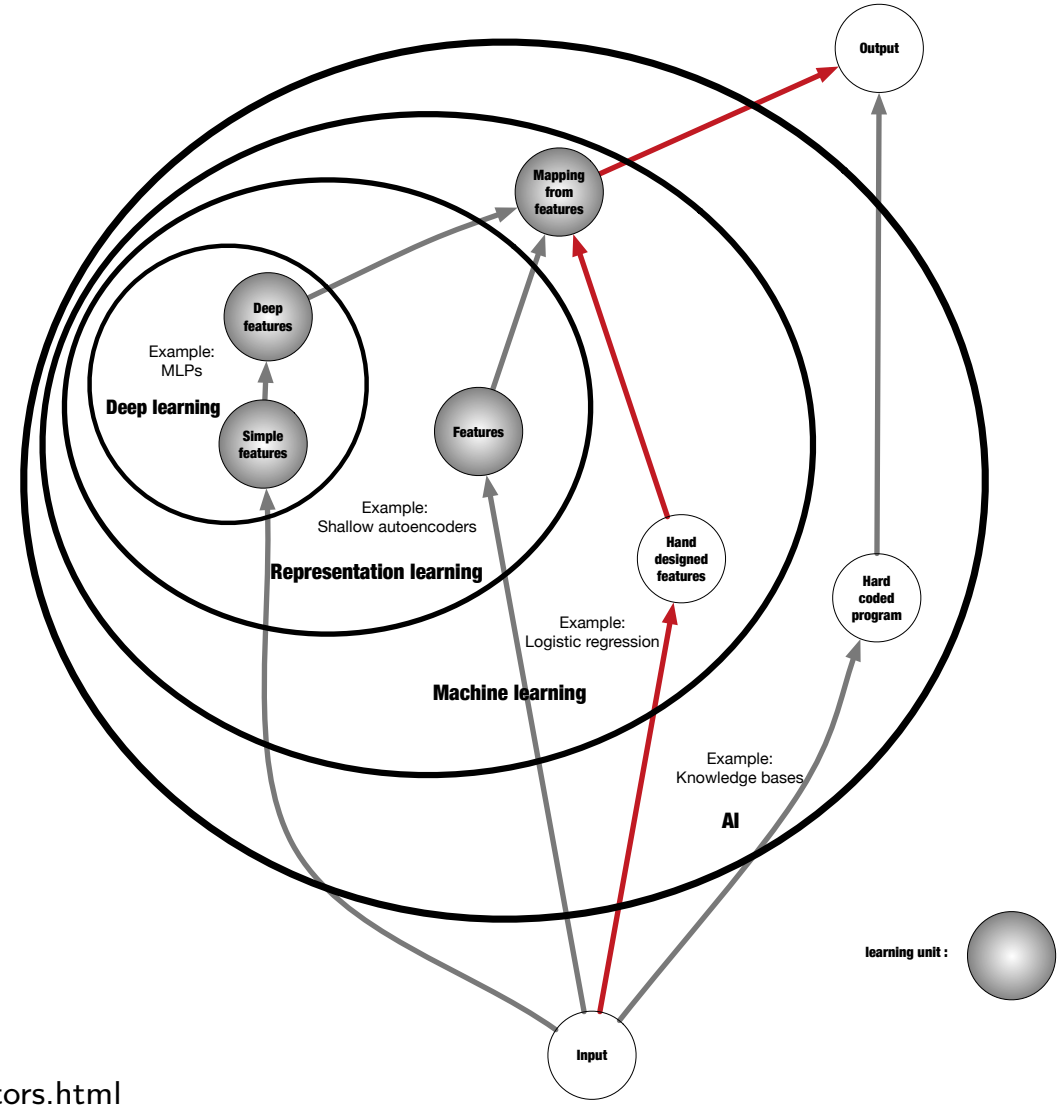
http://www.ictlounge.com/html/expert_systems.htm

Deep learning

Conventional machine learning

Requires domain experts spending years to design effective features.





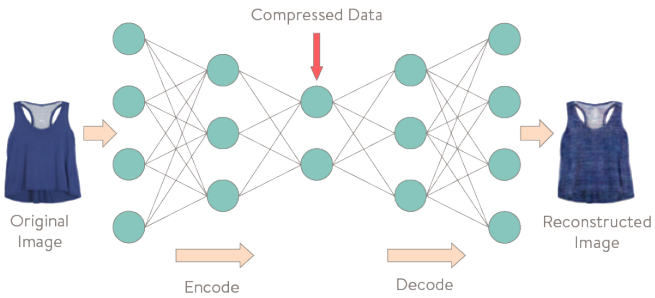
Source:

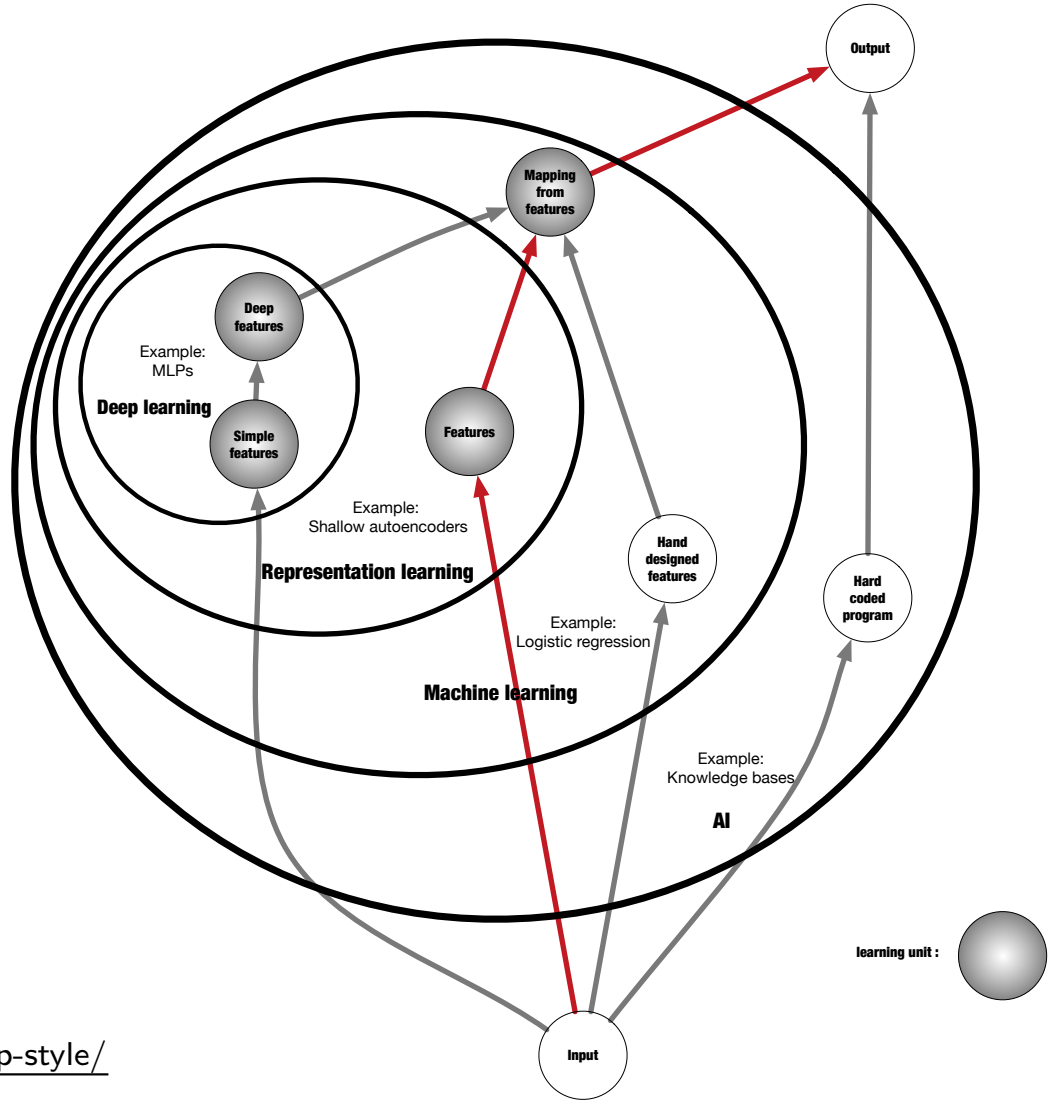
http://lear.inrialpes.fr/people/vandeweijer/color_descriptors.html

Deep learning

Shallow representation learning

Learned features are not powerful enough.





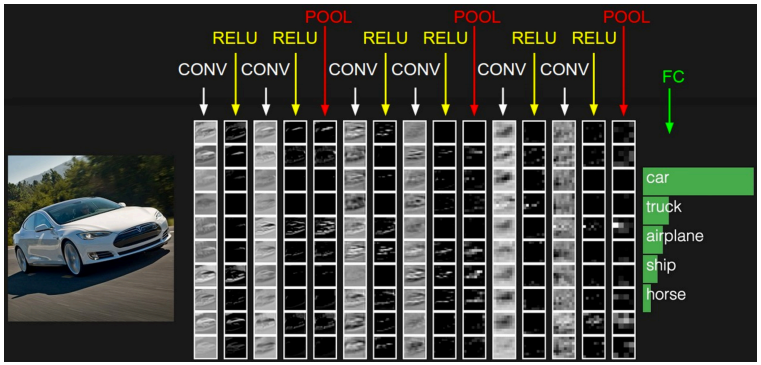
Source:

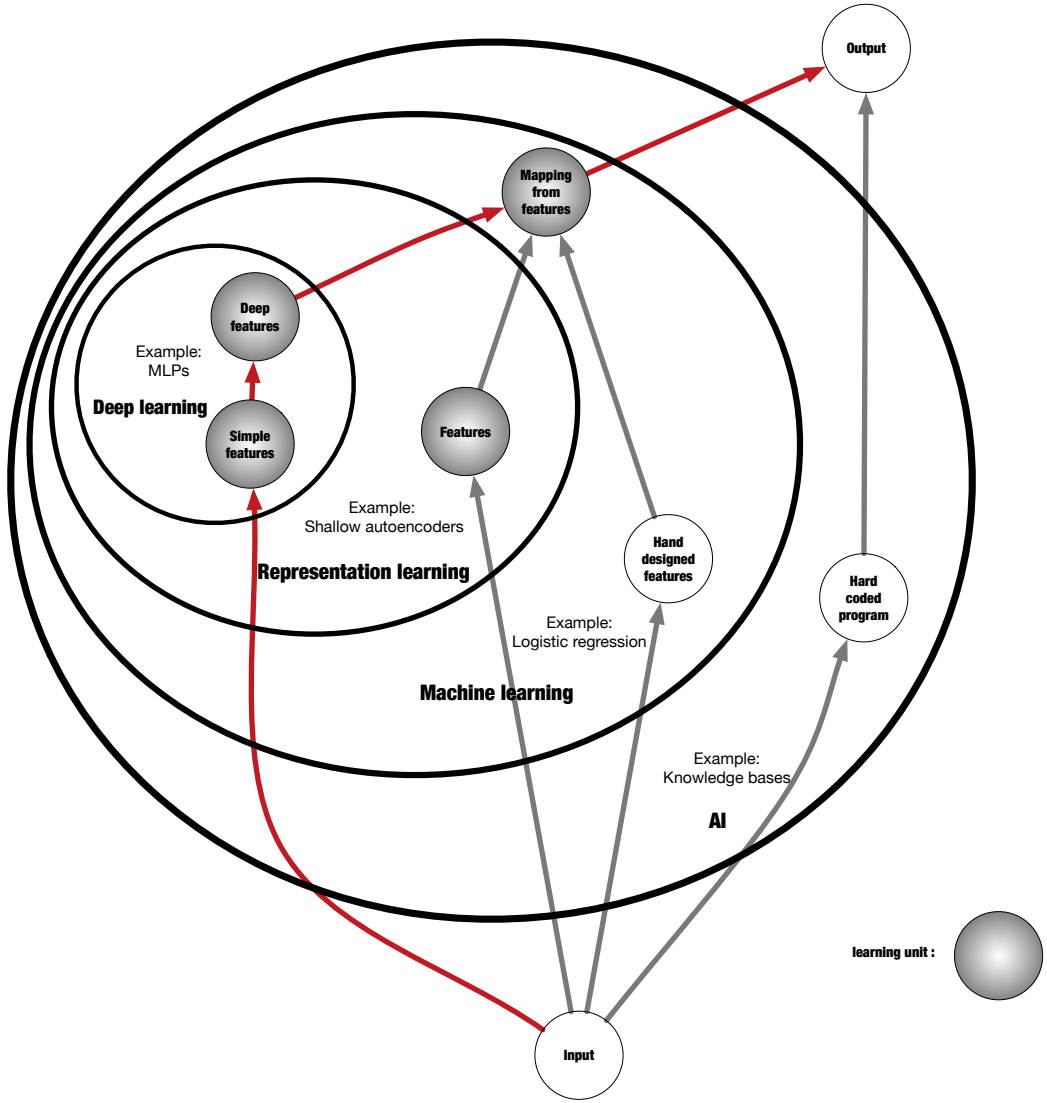
http://multithreaded.stitchfix.com/blog/2015/09/17/deep-style/

Deep learning

Deep learning

Build complex concepts out of simpler concepts. Can learn effective mid-level to high-level features.





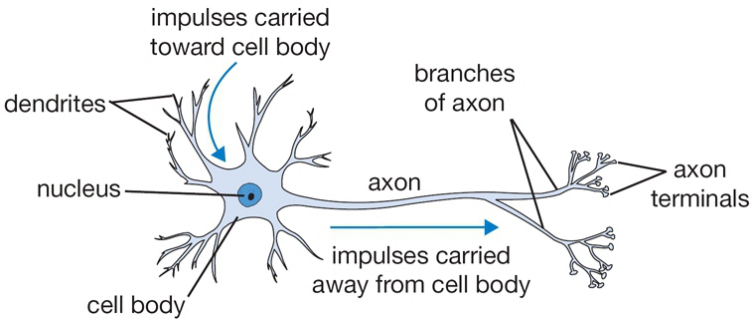
Source:

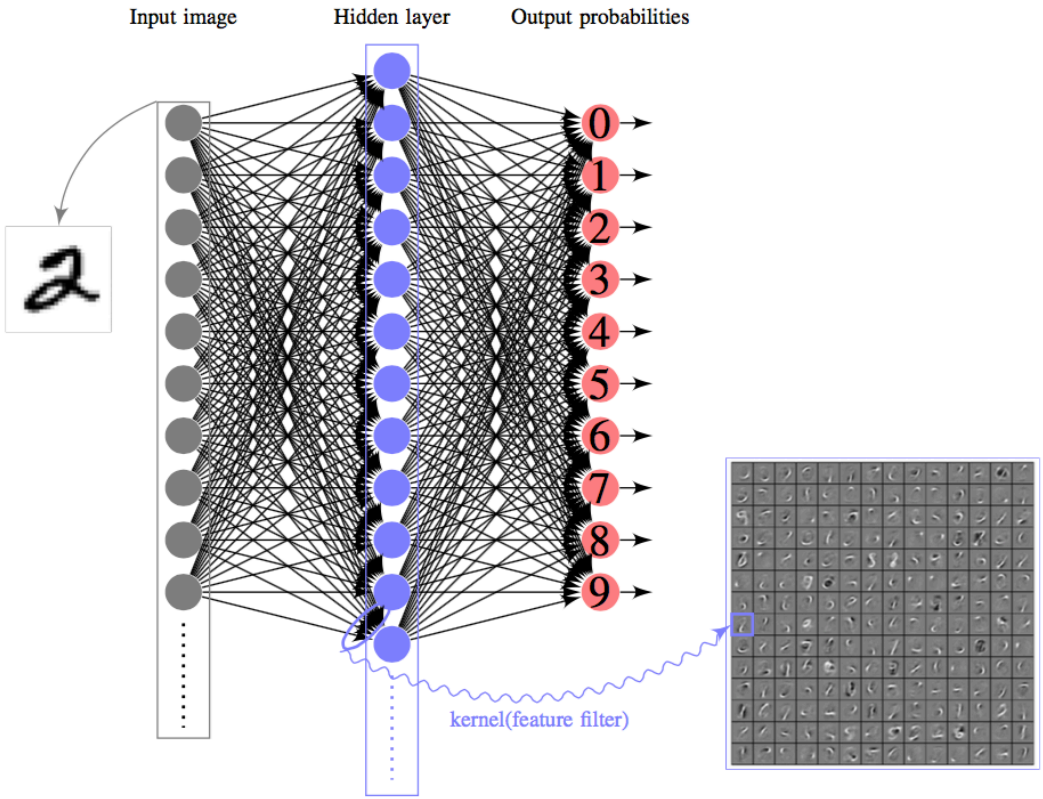
http://cs231n.github.io/convolutional-networks/

Artificial Neural Networks

Network structure

- single neuron
- activation
- loss function





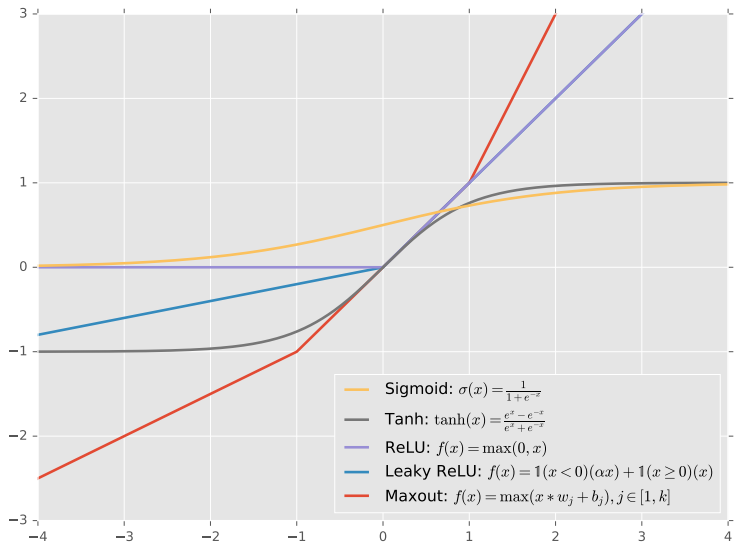
Source:

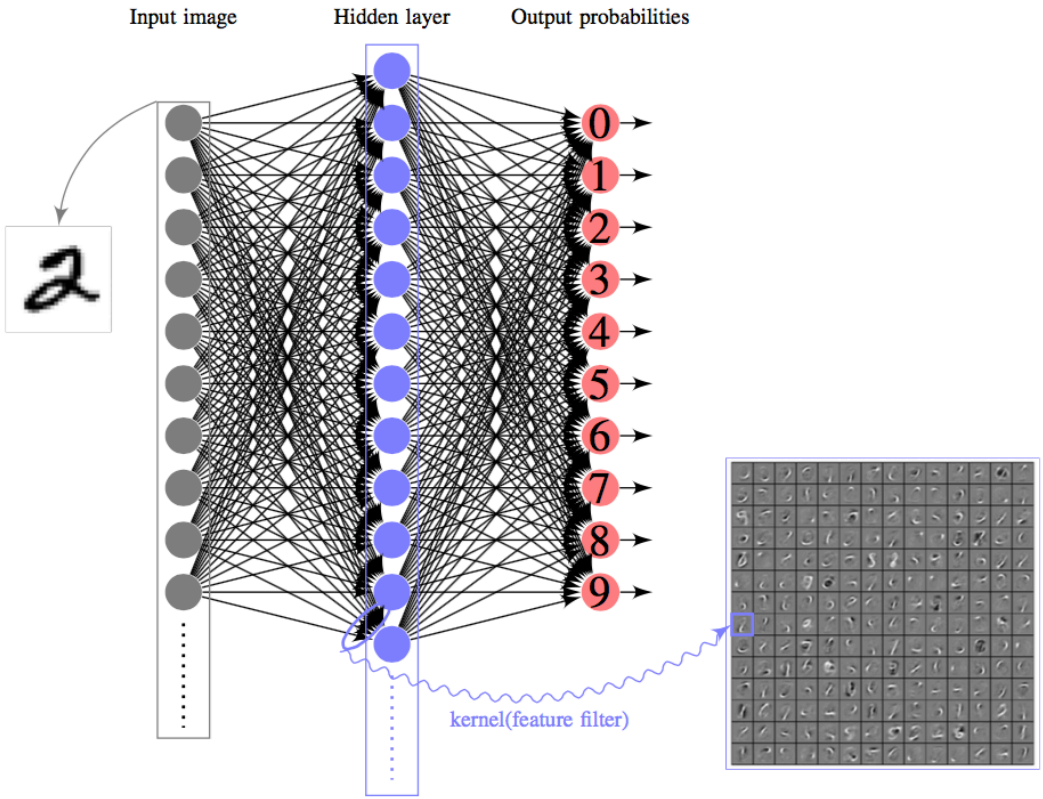
http://cs231n.github.io/neural-networks-1/

Artificial Neural Networks

Network structure

- single neuron
- activation
- loss function

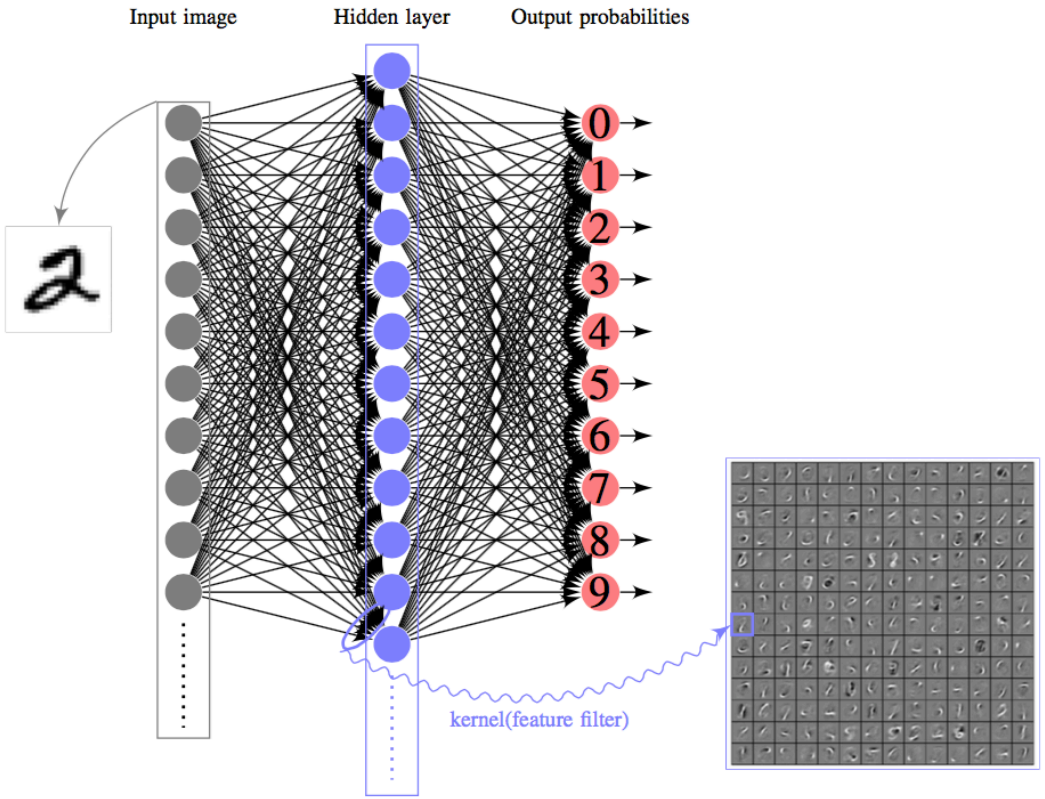




Artificial Neural Networks

Network structure

- single neuron
- activation
- loss function



classification

➥ cross-entropy:

$$L_i = -\log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}})$$

➥ hinge loss:

$$L_i = \sum_{j \neq y_i} \max(0, f_j - f_{y_i} + 1)$$

➥ attribute classfication

regression

➥ L_2 loss:

$$L_i = ||f - y_i||_2^2$$

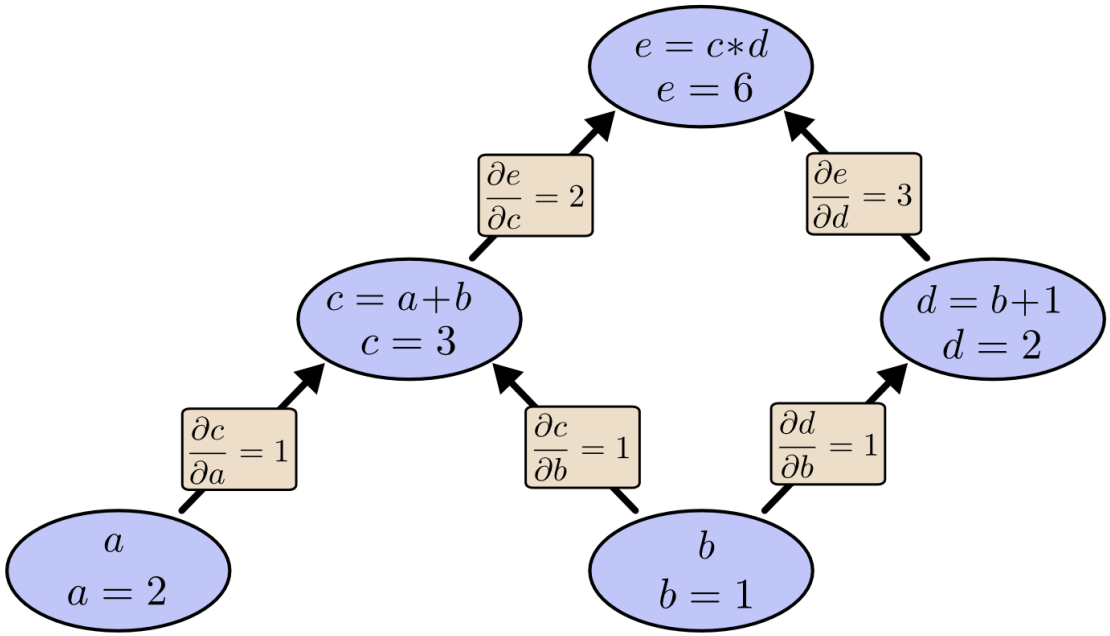
Artificial Neural Networks

Network structure

- single neuron
- activation
- loss function

Training

- back propagation
- data preprocessing
- batch normalization



Source: http://colah.github.io/posts/2015-08-Backprop/

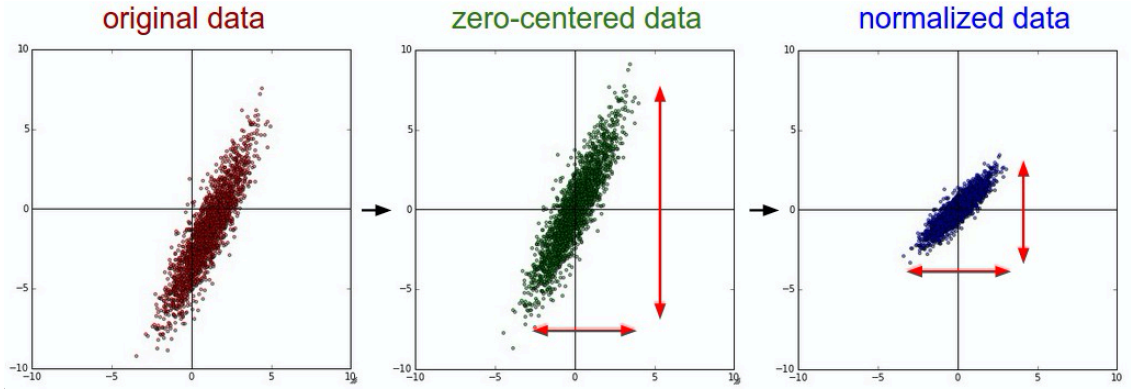
Artificial Neural Networks

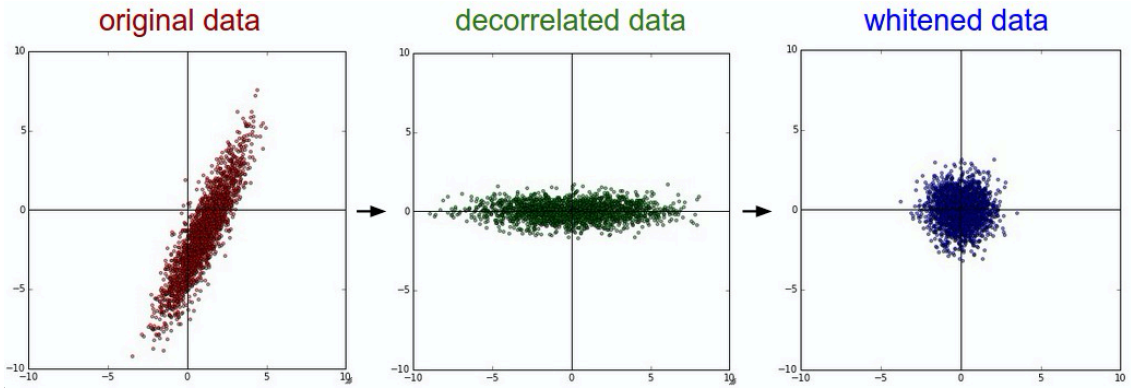
Network structure

- single neuron
- activation
- loss function

Training

- back propagation
- data preprocessing
- batch normalization





Source: http://cs231n.github.io/neural-networks-2/

Artificial Neural Networks

Network structure

- single neuron
- activation
- loss function

Training

- back propagation
- data preprocessing
- batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

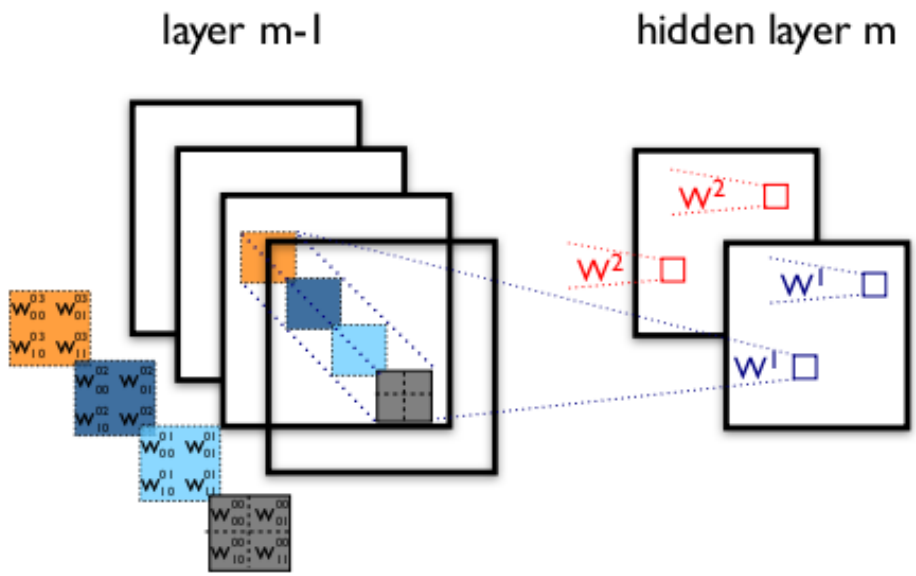
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Sergey Ioffe and Christian Szegedy: https://arxiv.org/abs/1502.03167

What is new about CNN

New layers

- Weight sharing through convolution
- Translation invariance through pooling
- Regularization through dropout

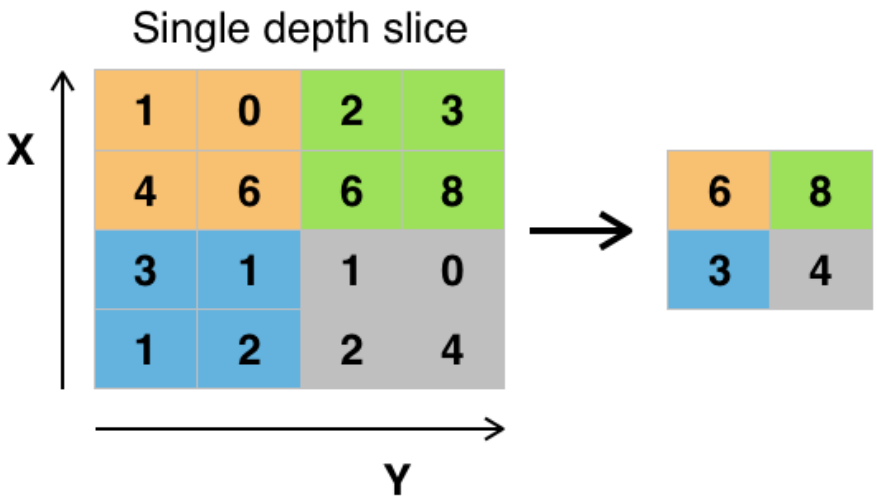


Source: http://deeplearning.net/tutorial/lenet.html

What is new about CNN

New layers

- Weight sharing through convolution
- Translation invariance through pooling
- Regularization through dropout



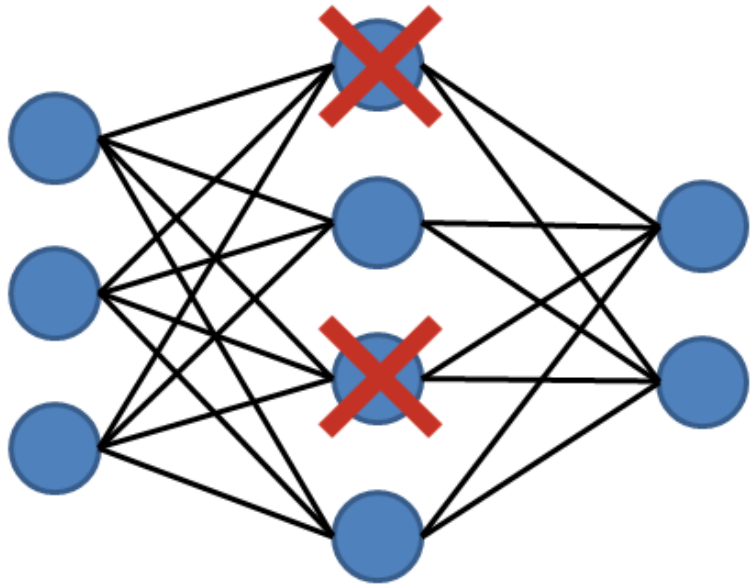
Source:

http://en.wikipedia.org/wiki/Convolutional_neural_network

What is new about CNN

New layers

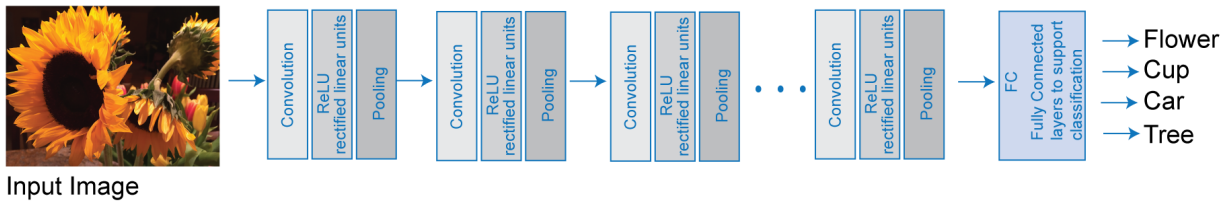
- Weight sharing through convolution
- Translation invariance through pooling
- Regularization through dropout

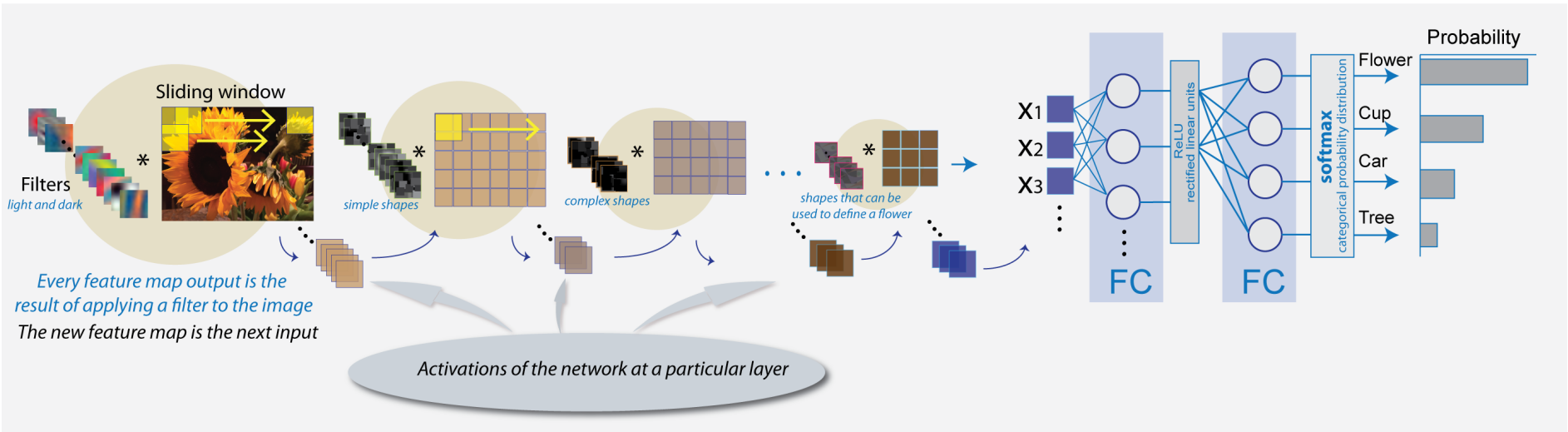


Conventional network structure

Pipeline for classification

$$\texttt{Input} \rightarrow [[\texttt{ConvLayer} \rightarrow \texttt{ReLU}] * N \rightarrow \texttt{PoolLayer?}] * M \rightarrow [\texttt{FC} \rightarrow \texttt{ReLU}] * K \rightarrow \texttt{FC}$$



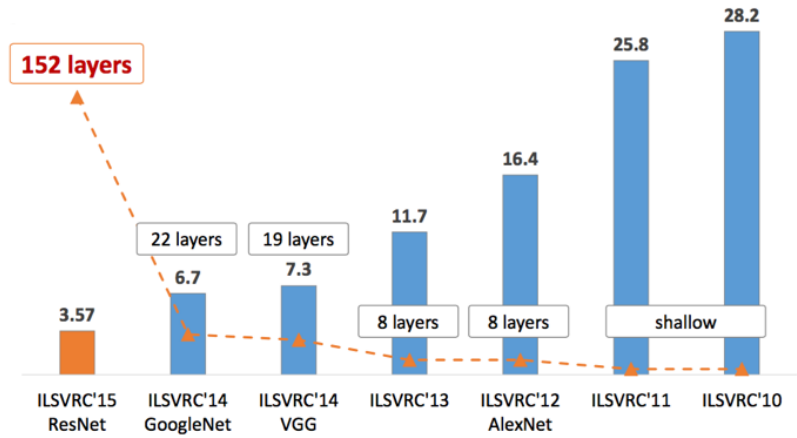


http://www.mathworks.com/help/nnet/convolutional-neural-networks.html

Transfer learning with CNNs

Procedures

1. Train deep CNNs on big dataset



Kaiming He et al.: https://arxiv.org/abs/1512.03385

Transfer learning with CNNs

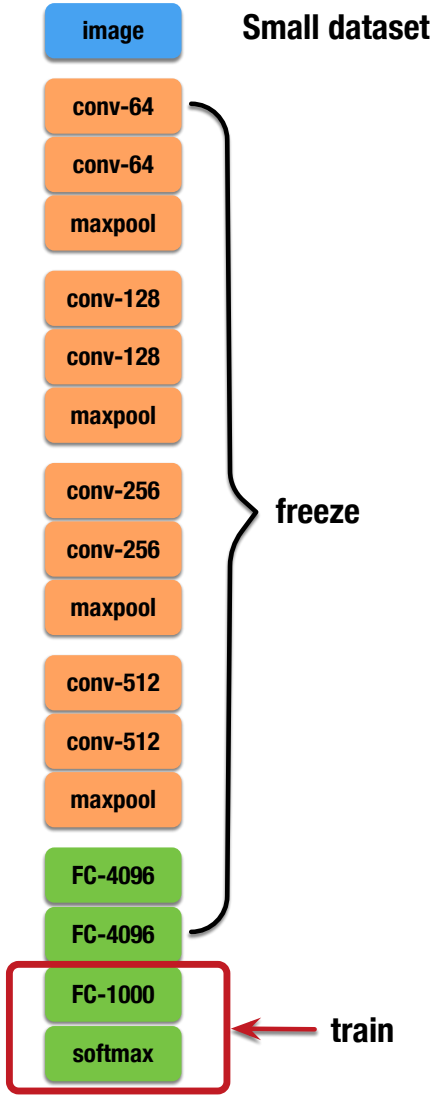
Procedures

1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset

Transfer learning with CNNs

Procedures

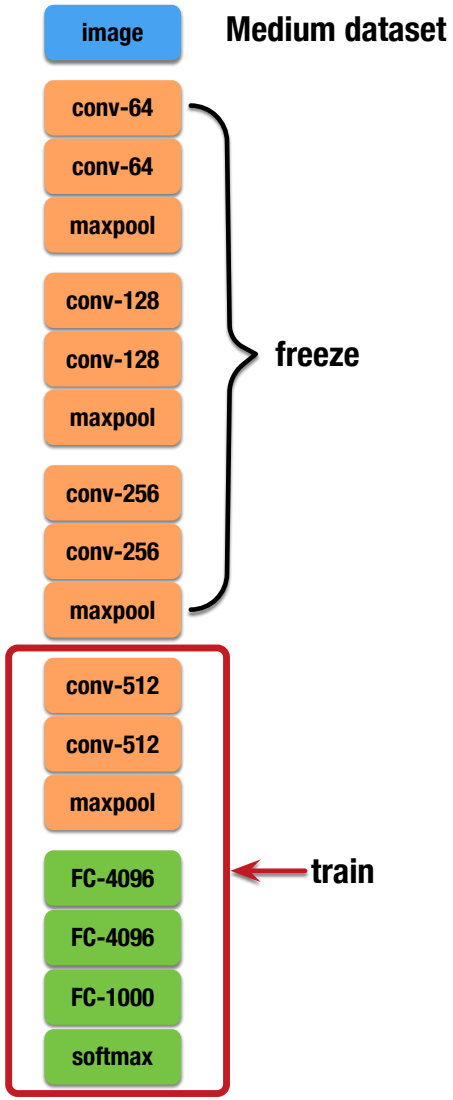
1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset



Transfer learning with CNNs

Procedures

1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset



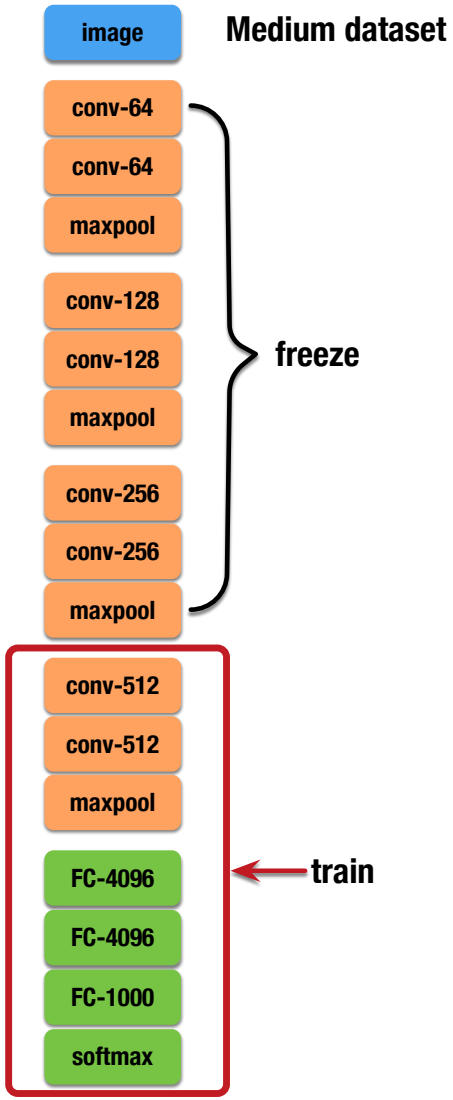
Transfer learning with CNNs

Procedures

1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset

Transferable features

- ConvNet features are more generic in early layers and more dataset-specific in later layers
- Similarity and size of new dataset



Transfer learning with CNNs

Procedures

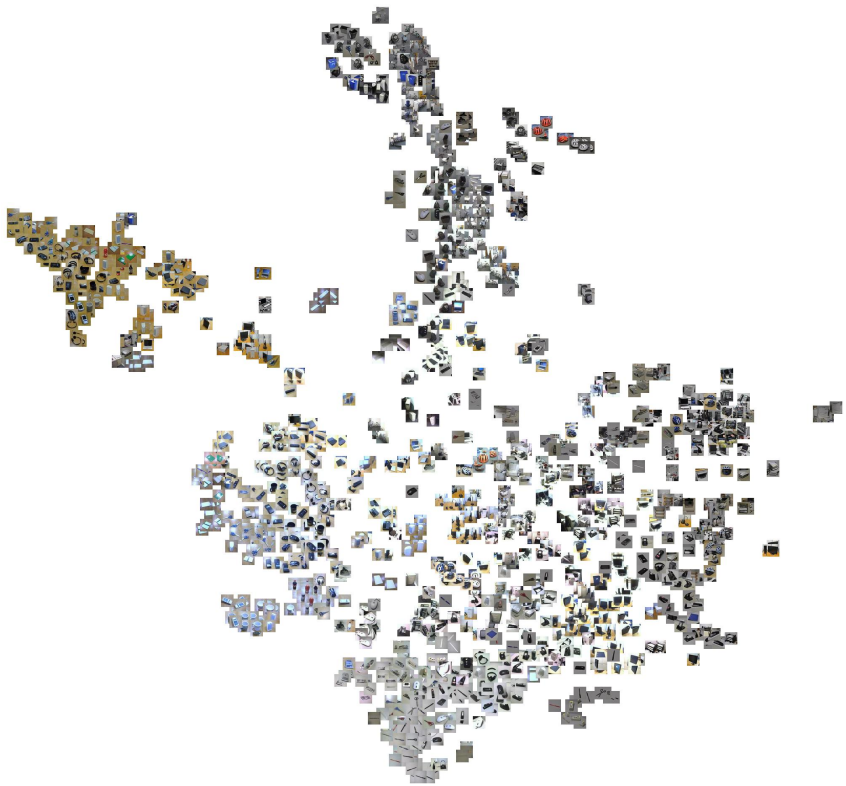
1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset

Transferable features

- ConvNet features are more generic in early layers and more dataset-specific in later layers
- Similarity and size of new dataset

t-SNE visualization

Surf features of Office Dataset



Transfer learning with CNNs

Procedures

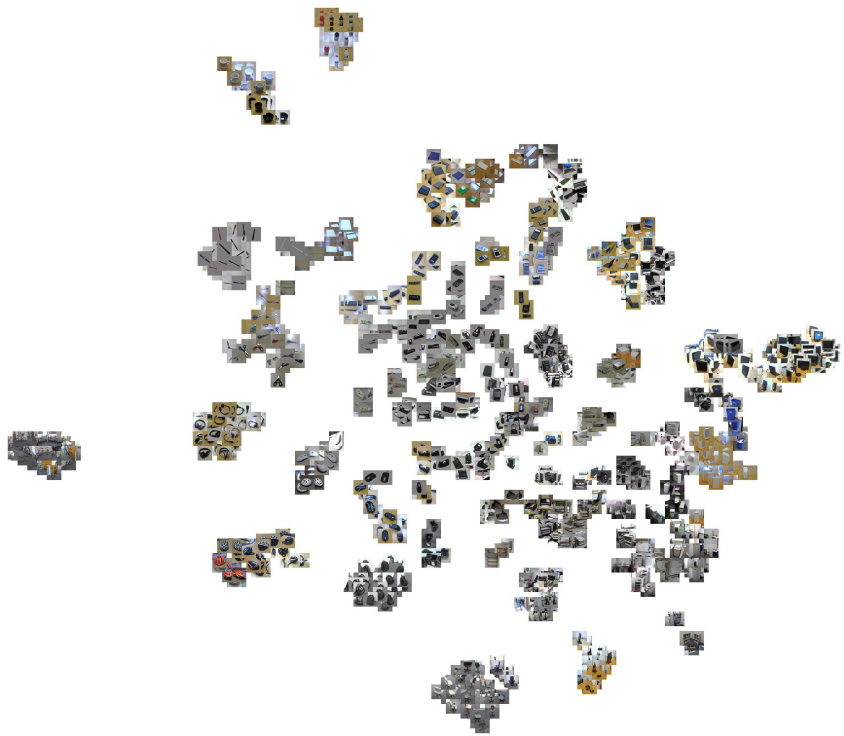
1. Train deep CNNs on big dataset
2. Use pre-trained CNNs as feature extractor, train on new dataset

Transferable features

- ConvNet features are more generic in early layers and more dataset-specific in later layers
- Similarity and size of new dataset

t-SNE visualization

Caffe features of Office Dataset



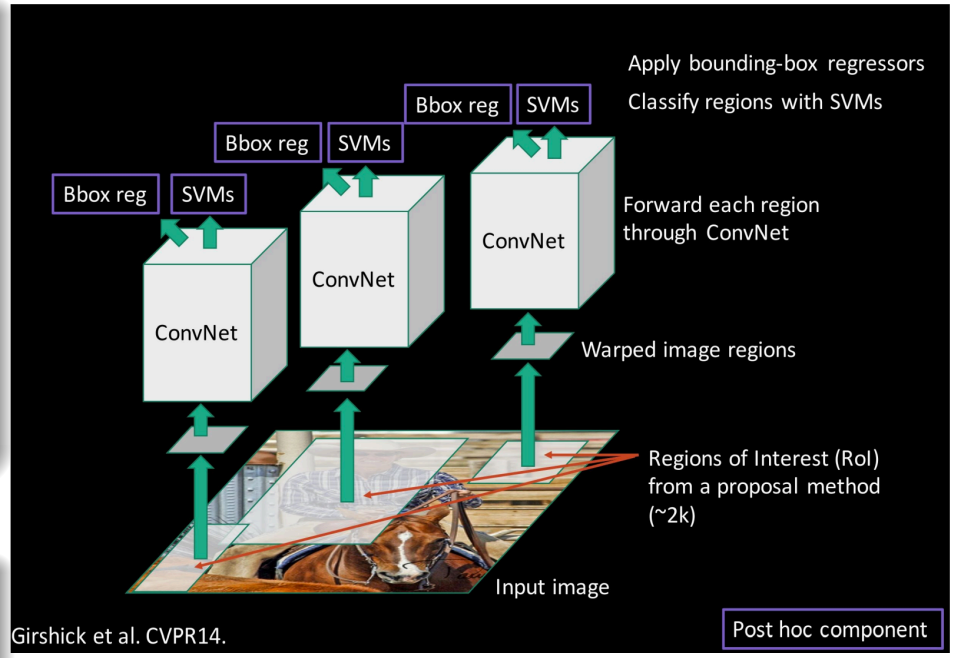
Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)

Drawbacks

- non-shared feature extraction computation
- cache features and post detector training
- external proposals

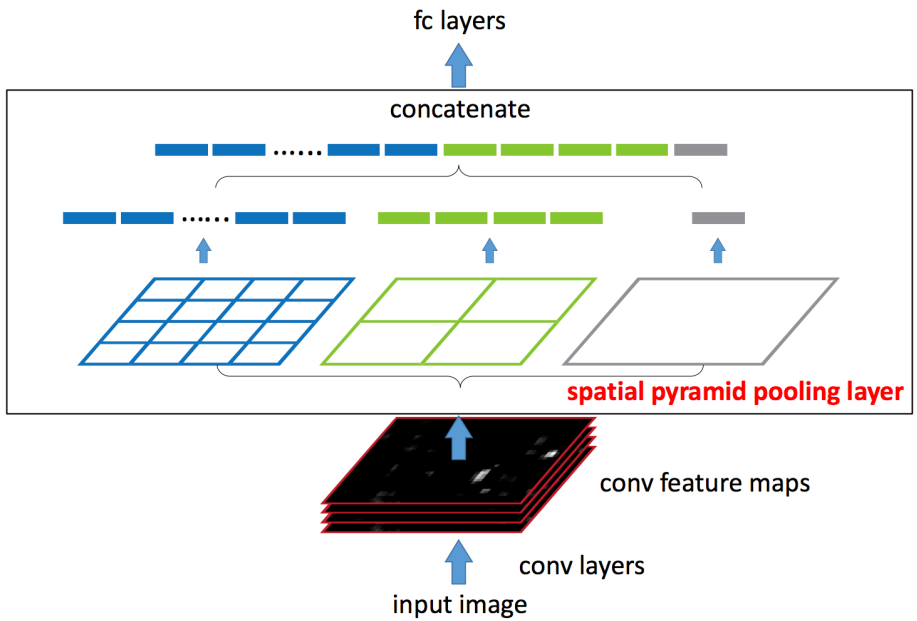


Ross Girshick et al. CVPR 2014

Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)



Drawbacks

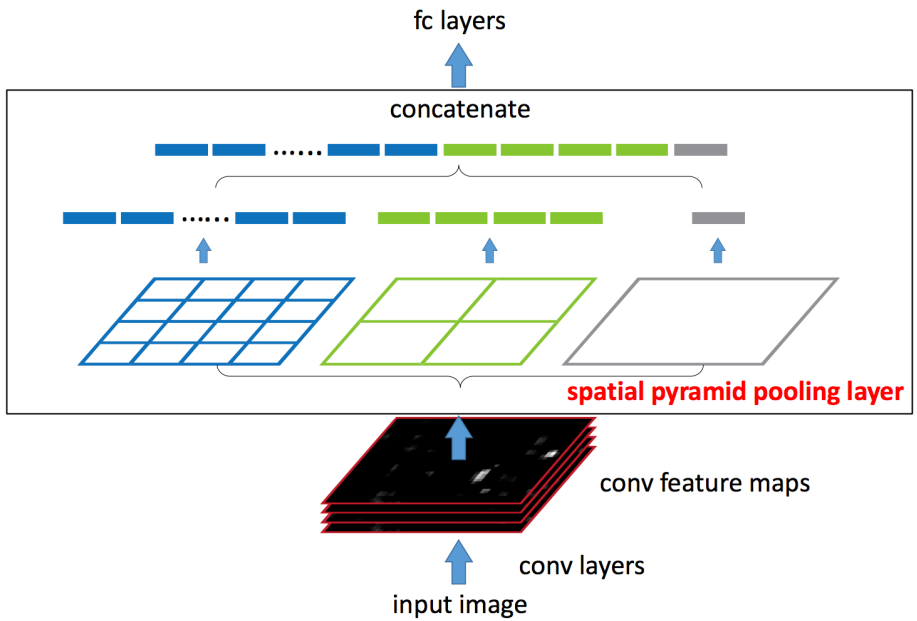
- non-shared feature extraction computation
- cache features and post detector training
- external proposals

Kaiming He et al.: https://arxiv.org/abs/1406.4729

Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)



Drawbacks

- ~~non-shared feature extraction computation~~
- cache features and post detector training
- external proposals

Kaiming He et al.: https://arxiv.org/abs/1406.4729

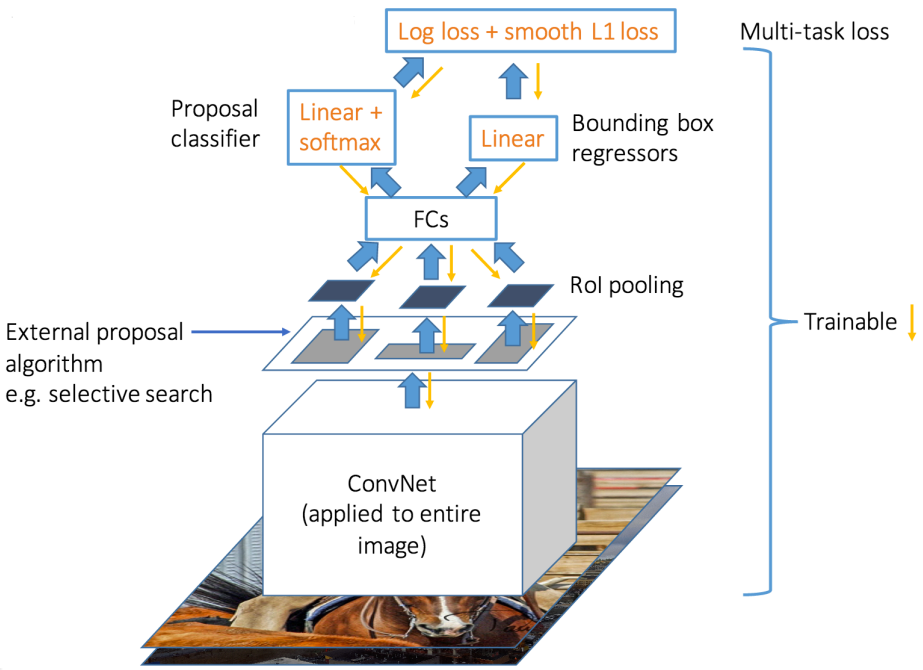
Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)

Drawbacks

- ~~non-shared feature extraction computation~~
- cache features and post detector training
- external proposals



Ross Girshick: https://arxiv.org/abs/1504.08083

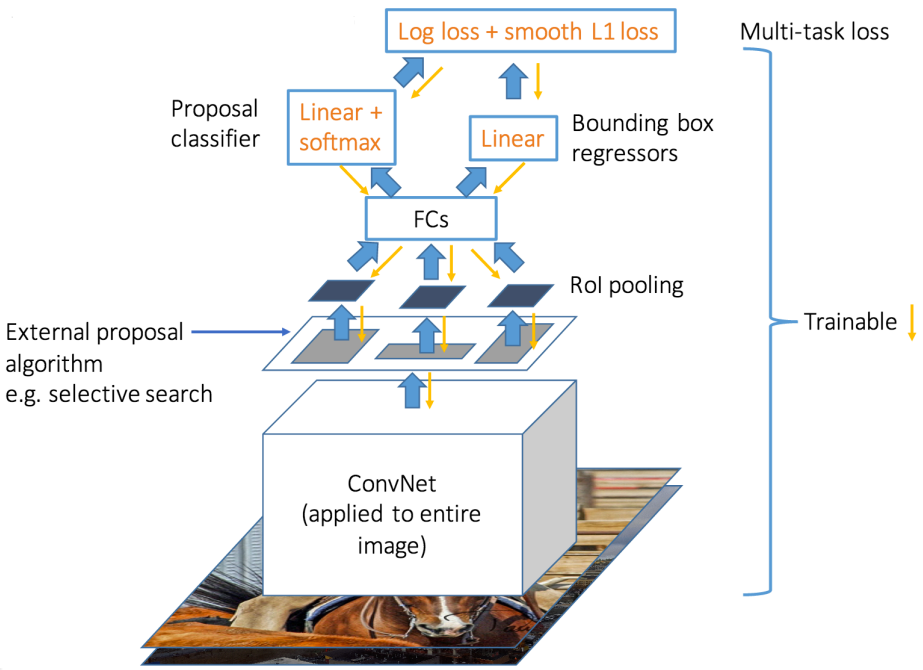
Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)

Drawbacks

- ~~non-shared feature extraction computation~~
- ~~cache features and post detector training~~
- external proposals



Ross Girshick: https://arxiv.org/abs/1504.08083

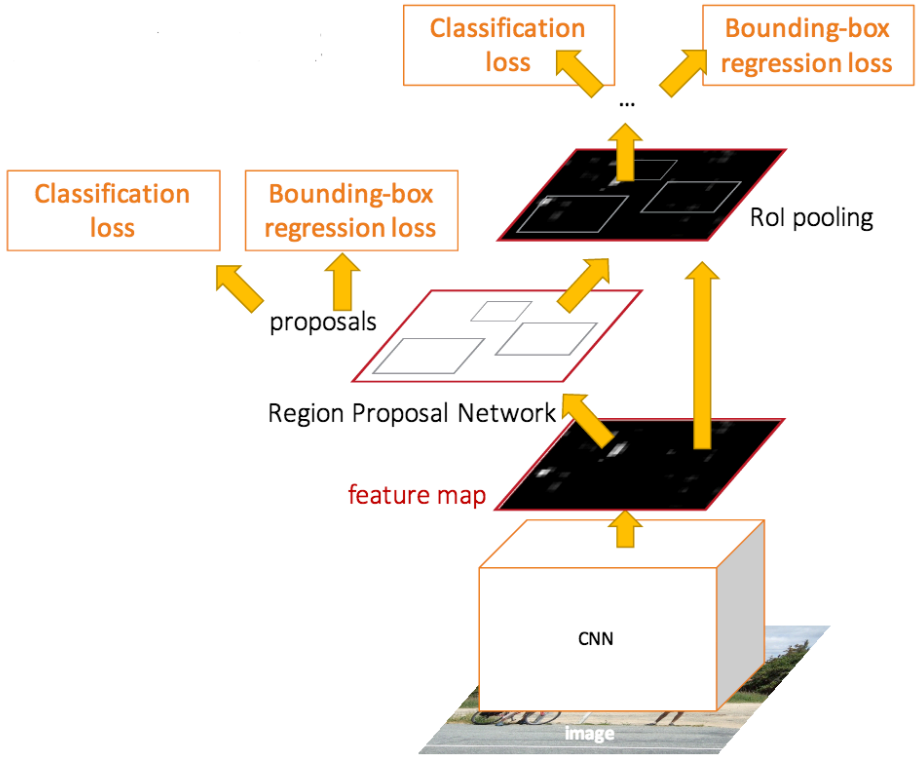
Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)

Drawbacks

- ~~non-shared feature extraction computation~~
- ~~cache features and post detector training~~
- external proposals



Shaoqing Ren et al.: https://arxiv.org/abs/1506.01497

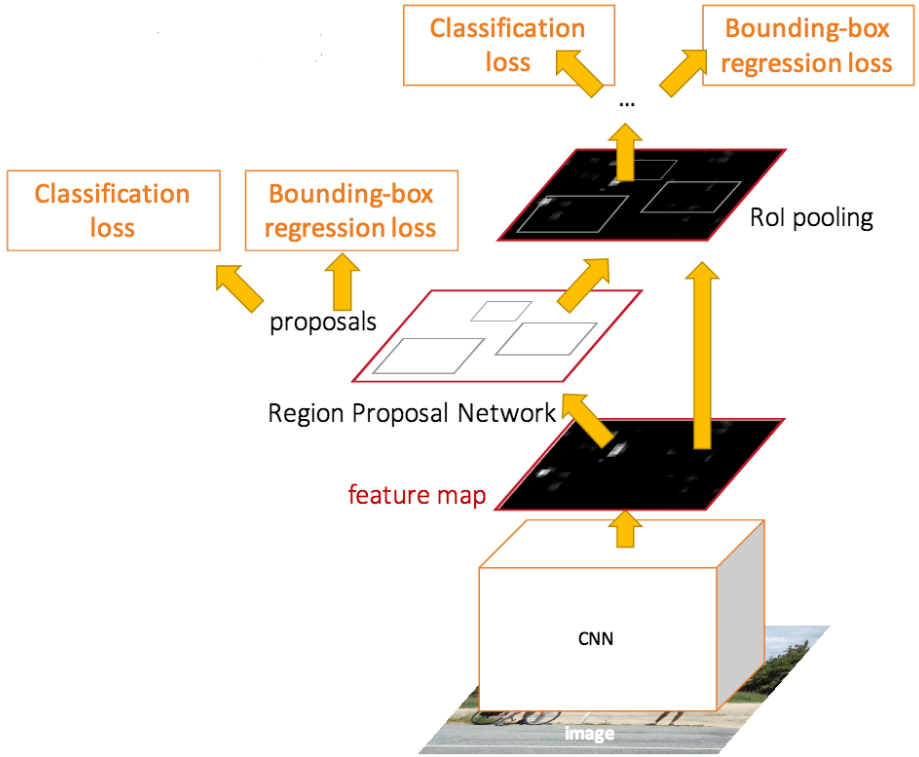
Faster R-CNN

Storyline

- R-CNN
- Spatial pyramid pooling (SPP)
- Fast-RCNN (ROI)
- Faster-RCNN (RPN)

Drawbacks

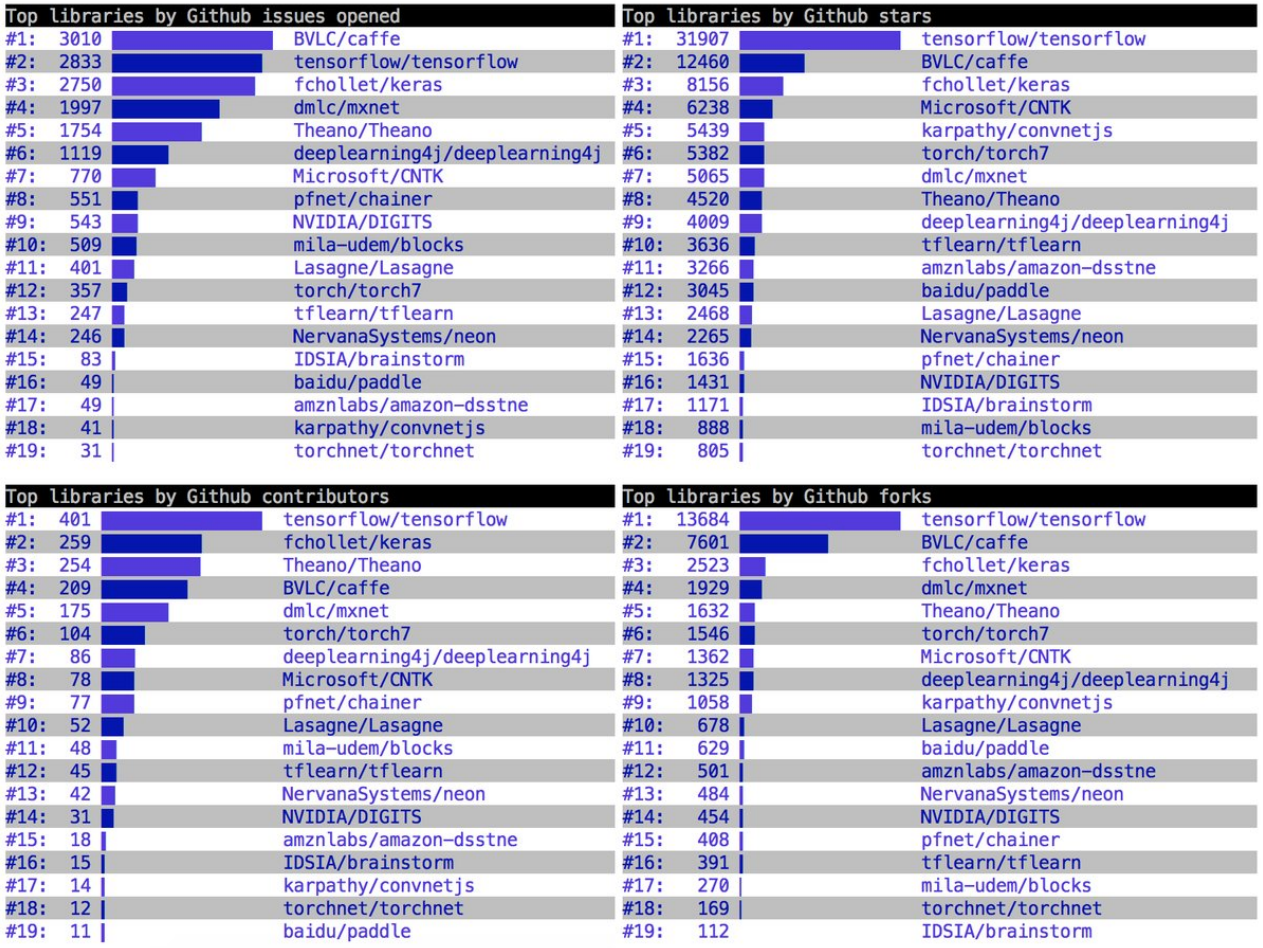
- ~~non-shared feature extraction computation~~
- ~~cache features and post detector training~~
- ~~external proposals~~



Shaoqing Ren et al.: https://arxiv.org/abs/1506.01497

Deep learning frameworks

Landscape, Sep 2016



Top libraries by Github issues opened

#1:	3010	BVLC/caffe
#2:	2833	tensorflow/tensorflow
#3:	2750	fchollet/keras
#4:	1997	dmlc/mxnet
#5:	1754	Theano/Theano
#6:	1119	deeplearning4j/deeplearning4j
#7:	770	Microsoft/CNTK
#8:	551	pfnet/chainer
#9:	543	NVIDIA/DIGITS
#10:	509	mila-udem/blocks
#11:	401	Lasagne/Lasagne
#12:	357	torch/torch7
#13:	247	tflearn/tflearn
#14:	246	NervanaSystems/neon
#15:	83	IDSIA/brainstorm
#16:	49	baidu/paddle
#17:	49	amznlabs/amazon-dsstne
#18:	41	karpathy/convnetjs
#19:	31	torchnet/torchnet

Top libraries by Github stars

#1:	31907	tensorflow/tensorflow
#2:	12460	BVLC/caffe
#3:	8156	fchollet/keras
#4:	6238	Microsoft/CNTK
#5:	5439	karpathy/convnetjs
#6:	5382	torch/torch7
#7:	5065	dmlc/mxnet
#8:	4520	Theano/Theano
#9:	4009	deeplearning4j/deeplearning4j
#10:	3636	tflearn/tflearn
#11:	3266	amznlabs/amazon-dsstne
#12:	3045	baidu/paddle
#13:	2468	Lasagne/Lasagne
#14:	2265	NervanaSystems/neon
#15:	1636	pfnet/chainer
#16:	1431	NVIDIA/DIGITS
#17:	1171	IDSIA/brainstorm
#18:	888	mila-udem/blocks
#19:	805	torchnet/torchnet

Top libraries by Github contributors

#1:	401	tensorflow/tensorflow
#2:	259	fchollet/keras
#3:	254	Theano/Theano
#4:	209	BVLC/caffe
#5:	175	dmlc/mxnet
#6:	104	torch/torch7
#7:	86	deeplearning4j/deeplearning4j
#8:	78	Microsoft/CNTK
#9:	77	pfnet/chainer
#10:	52	Lasagne/Lasagne
#11:	48	mila-udem/blocks
#12:	45	tflearn/tflearn
#13:	42	NervanaSystems/neon
#14:	31	NVIDIA/DIGITS
#15:	18	amznlabs/amazon-dsstne
#16:	15	IDSIA/brainstorm
#17:	14	karpathy/convnetjs
#18:	12	torchnet/torchnet
#19:	11	baidu/paddle

Top libraries by Github forks

#1:	13684	tensorflow/tensorflow
#2:	7601	BVLC/caffe
#3:	2523	fchollet/keras
#4:	1929	dmlc/mxnet
#5:	1632	Theano/Theano
#6:	1546	torch/torch7
#7:	1362	Microsoft/CNTK
#8:	1325	deeplearning4j/deeplearning4j
#9:	1058	karpathy/convnetjs
#10:	678	Lasagne/Lasagne
#11:	629	baidu/paddle
#12:	501	amznlabs/amazon-dsstne
#13:	484	NervanaSystems/neon
#14:	454	NVIDIA/DIGITS
#15:	408	pfnet/chainer
#16:	391	tflearn/tflearn
#17:	270	mila-udem/blocks
#18:	169	torchnet/torchnet
#19:	112	IDSIA/brainstorm

F Chollet: http://twitter.com/fchollet/status/776455778274250752

Agenda

Concern and solution

Design objectives

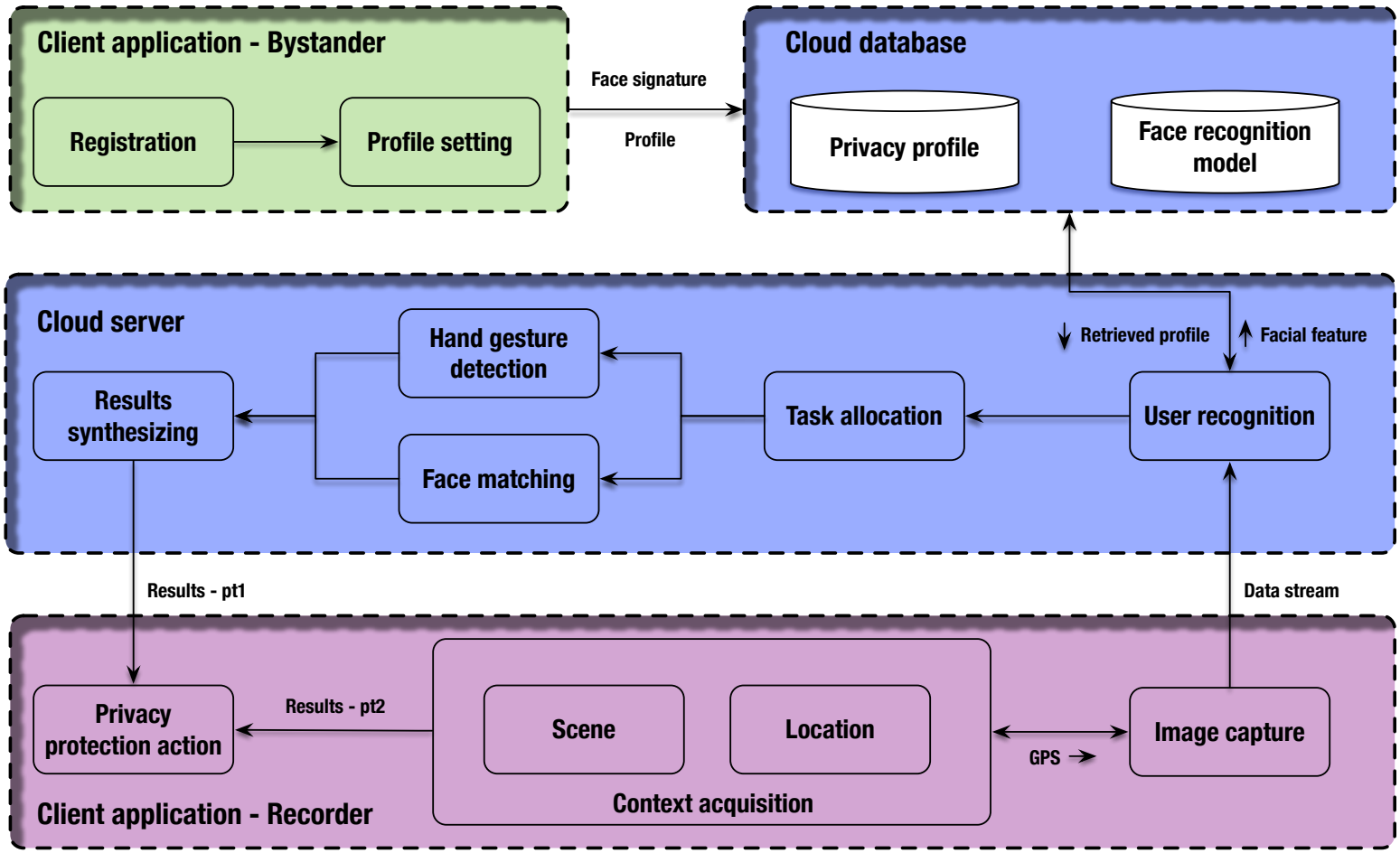
- Context dependent
- Individualized
- Dynamic

What we propose

- GPS location, grouped scene categories and accompanied persons
- Privacy preferences binded with facial features
- Easily update preferences, use gestures (✌ and ✋) to actively speak out in capturing moment

Architecture

Cardea components



Scene classification

Training

- Dataset
- Pre-trained model
- Train classifier

Places2 dataset: http://places2.csail.mit.edu/index.html

Deprecated dataset: 401 scene categories, 10 million training images

Currently: 365 scene categories, 8 million training images

Scene classification

Training

- Dataset
- Pre-trained model
- Train classifier

59 categories, 10 groups, 1 million training images

Scene Group	Scene Category Examples
Eating	bistro/indoor, bistro/outdoor, cafeteria, coffee_shop
Entertainment	bar, discotheque, pub/indoor
Shopping	bazaar/indoor, bazaar/outdoor, clothing_store, general_store/indoor
Work	conference_center, conference_room, cubicle/office, library/indoor
Public	park, street
Mobility	airplane_cabin, airport_terminal, bus_interior, bus_station/indoor
Exhibition	art_gallery, museum/indoor
Religion	cathedral/indoor, cathedral/outdoor, church/indoor, church/outdoor
Illness	hospital, hospital_room, nursing_home
Nudity	bathroom, beach, jacuzzi/indoor, sauna, shower

Scene classification

Training

- Dataset
- Pre-trained model
- Train classifier

Scene classification with provided training data

Team name	Entry description	Top-5 classification error
Hikvision	Model D	0.0901
Hikvision	Model E	0.0908
Hikvision	Model C	0.0939
Hikvision	Model B	0.0948
MW	Model ensemble 2	0.1019
MW	Model ensemble 3	0.1019
MW	Model ensemble 1	0.1023
Hikvision	Model A	0.1026
Trimps-Soushen	With extra data.	0.103
Trimps-Soushen	Ensemble 2	0.1042
SIAT_MMLAB	10 models fusion	0.1043
SIAT_MMLAB	7 models fusion	0.1044
SIAT_MMLAB	fusion with softmax	0.1044
SIAT_MMLAB	learning weights with cnn	0.1044
SIAT_MMLAB	6 models fusion	0.1049
Trimps-Soushen	Ensemble 4	0.1049
Trimps-Soushen	Ensemble 3	0.105

Source: http://places2.csail.mit.edu/results2016.html

Place401-AlexNet

	Validation Set of Places365		Test Set of Places365	
	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.
Places365-AlexNet	53.17%	82.89%	53.31%	82.75%
Places365-GoogLeNet	53.63%	83.88%	53.59%	84.01%
Places365-VGG	**55.24%**	84.91%	**55.19%**	85.01%
Places365-ResNet	54.74%	**85.08%**	54.65%	**85.07%**

Source: http://github.com/metalbubble/places365

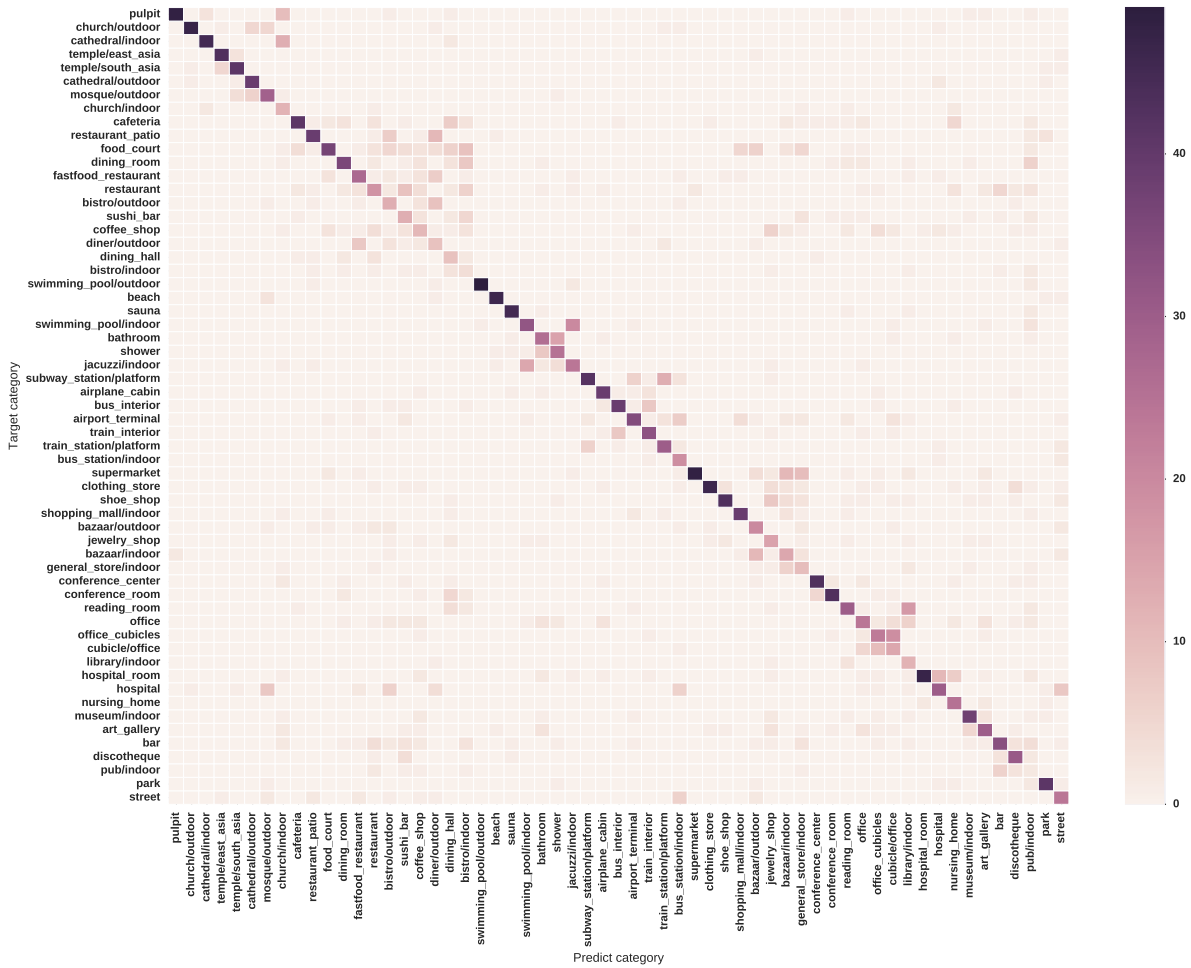
Scene classification

Training

- Dataset
- Pre-trained model
- Train classifier

fc7 layer features, 60.0% validation accuracy for scene categories, 82.8% validation accuracy for scene groups

Confusion Matrix - Category



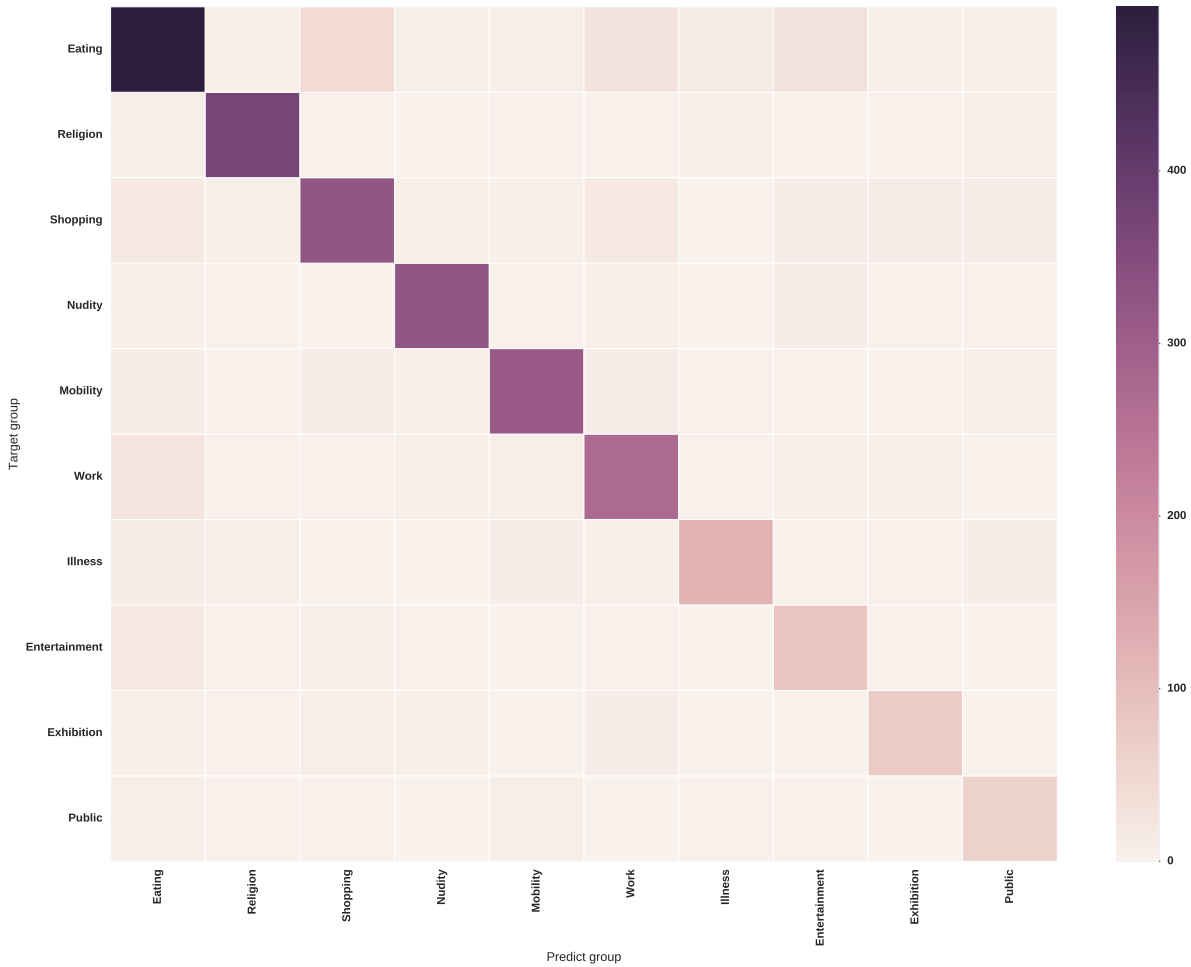
Scene classification

Training

- Dataset
- Pre-trained model
- Train classifier

fc7 layer features, 60.0% validation accuracy for scene categories, 82.8% validation accuracy for scene groups

Confusion Matrix - Group



Scene classification

Prediction examples

Face recognition

VGG16 CNN vs Lightened CNN

- Model size
- Feature performance
- Runtime on Android

(VGG16 CNN) Omkar M. Parkhi et al.:

http://www.robots.ox.ac.uk/ vgg/software/vgg_face/

(Lightened CNN) Xiang Wu et al.:

https://arxiv.org/abs/1511.02683

500MB vs 30MB

Face recognition

VGG16 CNN vs Lightened CNN

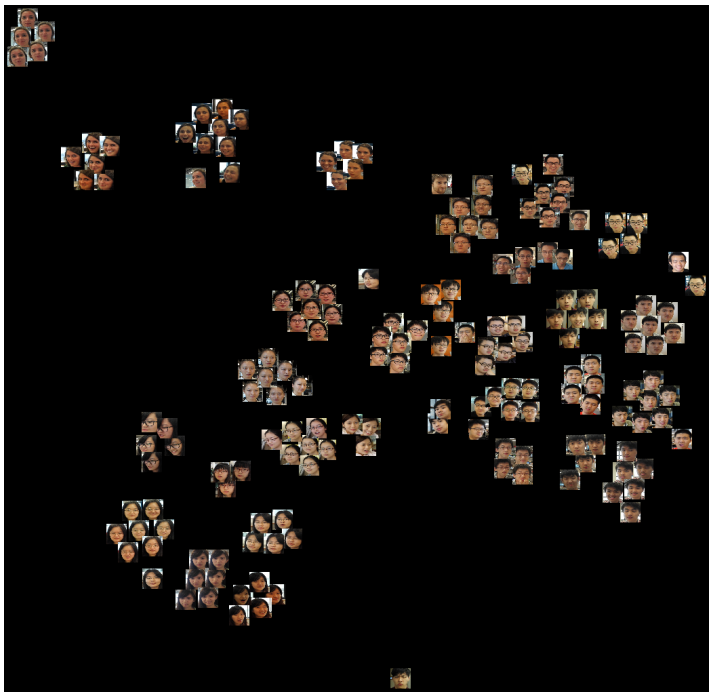
- Model size
- Feature performance
- Runtime on Android

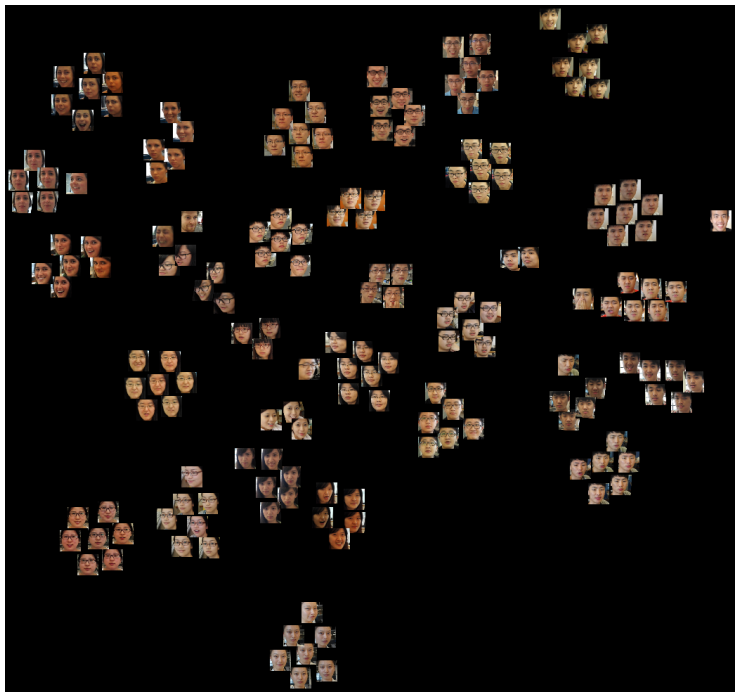
(VGG16 CNN) Omkar M. Parkhi et al.:

http://www.robots.ox.ac.uk/ vgg/software/vgg_face/

(Lightened CNN) Xiang Wu et al.:

https://arxiv.org/abs/1511.02683





Face recognition

VGG16 CNN vs Lightened CNN

- Model size
- Feature performance
- Runtime on Android

(VGG16 CNN) Omkar M. Parkhi et al.:

http://www.robots.ox.ac.uk/ vgg/software/vgg_face/

(Lightened CNN) Xiang Wu et al.:

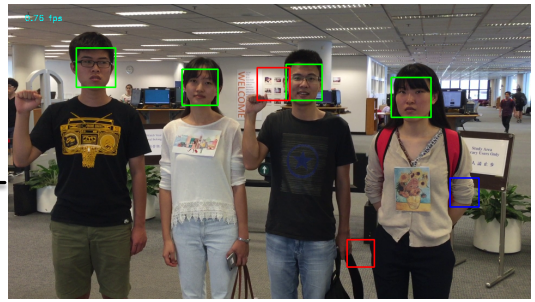
https://arxiv.org/abs/1511.02683

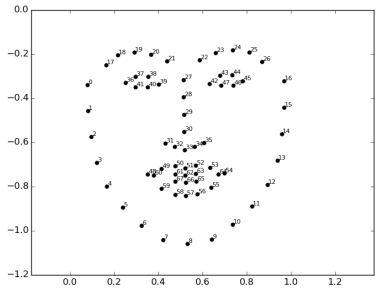
Table: Time of single facial feature extraction and batch facial feature extraction (10 faces).

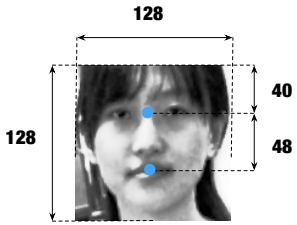
	Xiaomi Mi 3W Snapdragon 800 2GB RAM	Galaxy Note 4 Snapdragon 805 3GB RAM	Xiaomi Mi 5 Snapdragon 820 4GB RAM
1 VGG CNN	N/A	N/A	$\sim$ 2780 ms
10 VGG CNN	N/A	N/A	$\sim$ 26740 ms
1 Lightened CNN	$\sim$ 508 ms	$\sim$ 330 ms	$\sim$ 303 ms
10 Lightened CNN	$\sim$ 6602 ms	$\sim$ 3071 ms	$\sim$ 2031 ms

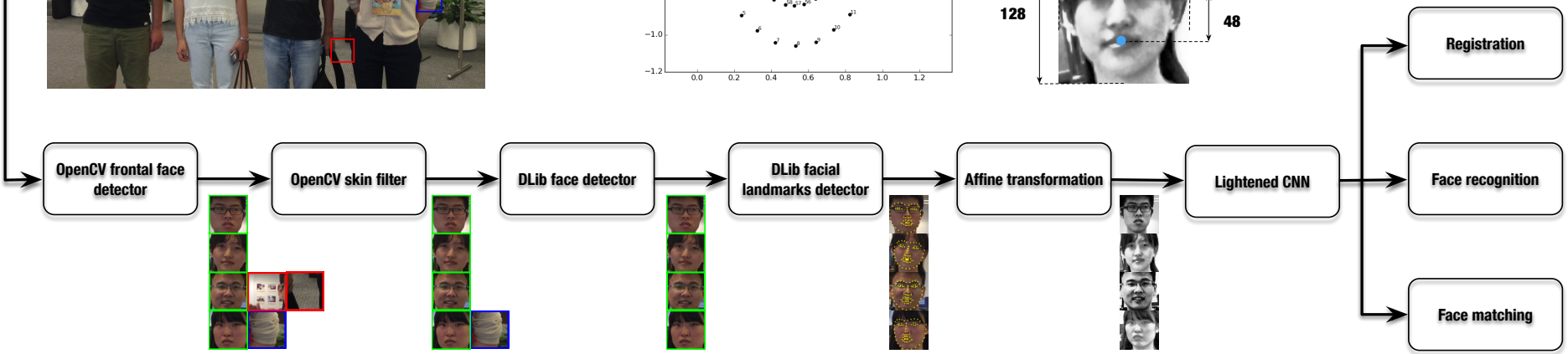
Face recognition

Detection and alignment









Recognition model

SVM, probability threshold T_p at prediction time

Face recognition

Face matching algorithm

Distance metric: `cosine`

1: initialize P's feature f_0, Bob's feature $f_i, i \in 1, \ldots, N$, distance threshold T_d, hit ratio threshold T_r
2: $m \leftarrow 0$ // number of hits
3: **for** $i = 1$ to N **do**
4: $\quad d_i \leftarrow dis(f_0, f_i)$ // $dis(x, y)$ returns the distance between x and y
5: $\quad$ **if** $d_i \leq T_d$ **then**
6: $\quad\quad m \leftarrow m + 1$
7: $\quad$ **end if**
8: **end for**
9: **if** $m/N \geq T_r$ **then**
10: $\quad$ **return** true // P is Bob
11: **else**
12: $\quad$ **return** false // P and Bob are two persons
13: **end if**

Gesture recognition

Training

- Dataset
- Pre-trained model
- Train detector

VGG hand dataset:
5628 images with 13050 "natural" gesture instances,
http://www.robots.ox.ac.uk:5000/~vgg/research

Self prepared dataset: 4712 images with "yes" gestures and 3503 images with "no" gestures

Composed dataset

Gesture recognition

Training

- Dataset
- Pre-trained model
- Train detector

VGG hand dataset:
5628 images with 13050 "natural" gesture instances,
http://www.robots.ox.ac.uk:5000/~vgg/research

Self prepared dataset: 4712 images with "yes" gestures and 3503 images with "no" gestures

VGG16 ImageNet model

Composed dataset

Gesture recognition

Training

- Dataset
- Pre-trained model
- Train detector

VGG hand dataset:
5628 images with 13050 "natural" gesture instances,
http://www.robots.ox.ac.uk:5000/~vgg/research

Self prepared dataset: 4712 images with "yes" gestures and 3503 images with "no" gestures

VGG16 ImageNet model

Fine tune "conv3_1" and up layers, approximate joint optimization

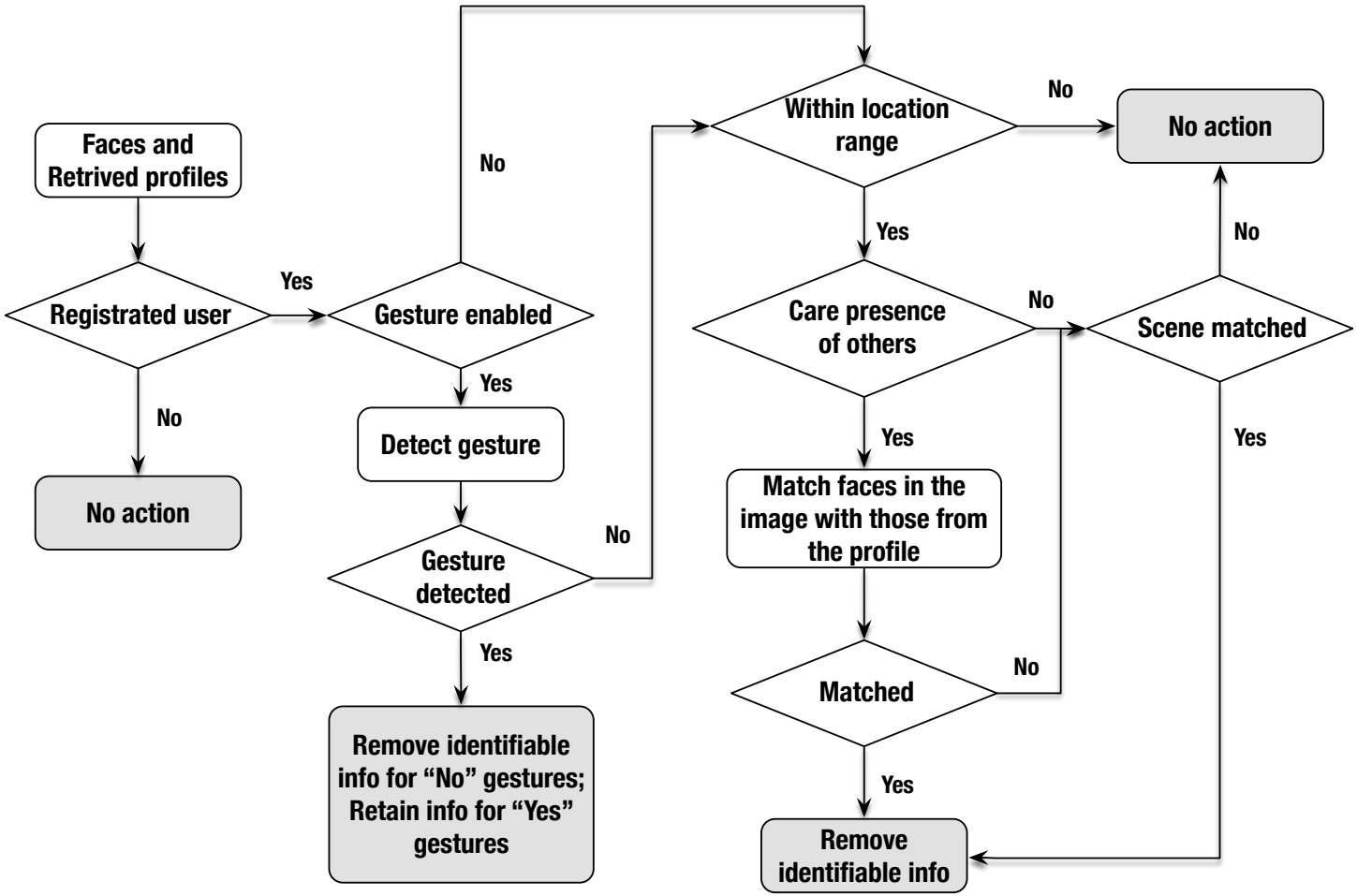
Composed dataset

Gesture recognition

Prediction examples

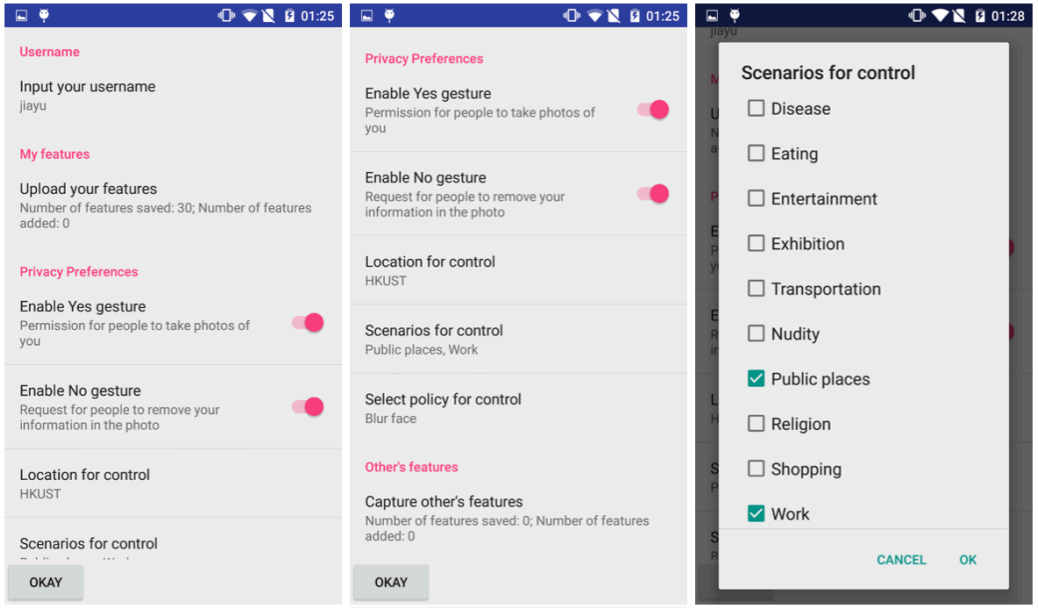
Decision flow

Decision path of protection action



Decision flow

Example



(a) Registration and profile updating interface

(b) Privacy protection example

Figure: Cardea user interface and privacy protection results. In (a), *jiayu* registers as a Cardea user by extracting and uploading her face features. She specifies HKUST, two scene groups for privacy protection. She also enables "yes" and "no" gestures. In (b), a picture is taken in HKUST. 3 registered users, one "yes" and one "no" gesture are recognized. The scene is correctly predicted as "Public". *jiayu*'s face is not blurred due to her "yes" gesture. Prediction probabilities are also shown in (b).

System overview

On mobile

- face detection, alignment
- face feature extraction
- scene classification

OpenCV, Dlib, Caffe-android-lib

On cloud

- face recognition
- face recognition model updating
- gesture recognition

Py-faster-rcnn, LIBSVM,

multiprocessing/multithreading

Demo video link: http://bit.ly/2egw2H6

Agenda

Scene classification

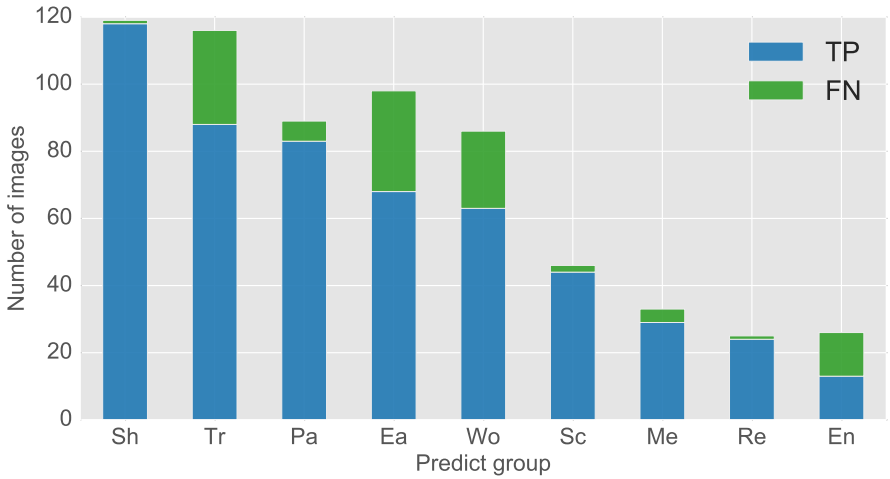
Scene group modification

Group name	Abbreviatin	Scenes #	Examples
Shopping	Sh	20	clothing store, market, supermarket
Travelling	Tr	9	airport, bus station, subway platform
Park & street	Pa	12	downtown, park, street, alley
Eating & drinking	Ea	18	bar, bistro, cafeteria, coffee shop, fastfood restaurant, food court
Working & study	Wo	9	classroom, conference center, library, office, reading room
Scantily clad	Sc	12	beach, swimming pool, water park
Medical care	Me	2	hospital room, nursing home
Religion	Re	11	cathedral, chapel, church, temple
Entertainment	En	5	amusement park, ballroom, discotheque
All		*98*	

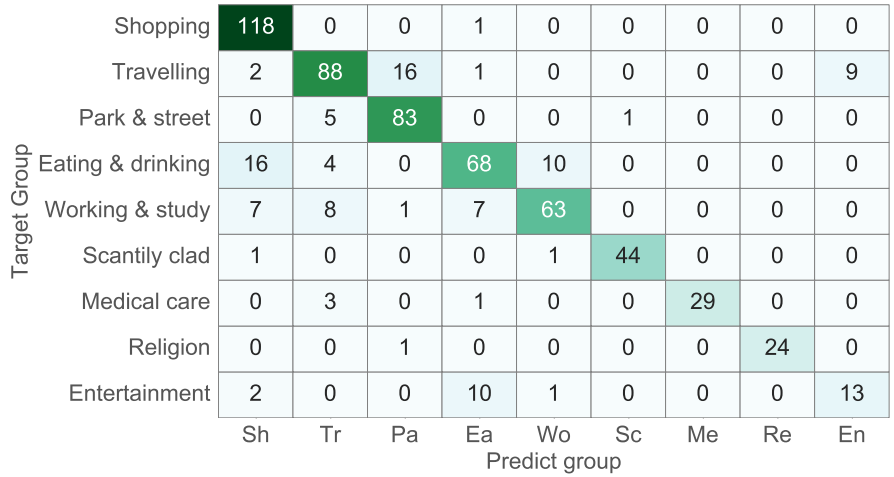
Add more scenes, remove "Exhibition" group. Train the new model, 55% and 83% validation accuracy. Evaluation results are based on new model.

Scene classification

Group recall



(a) Recall



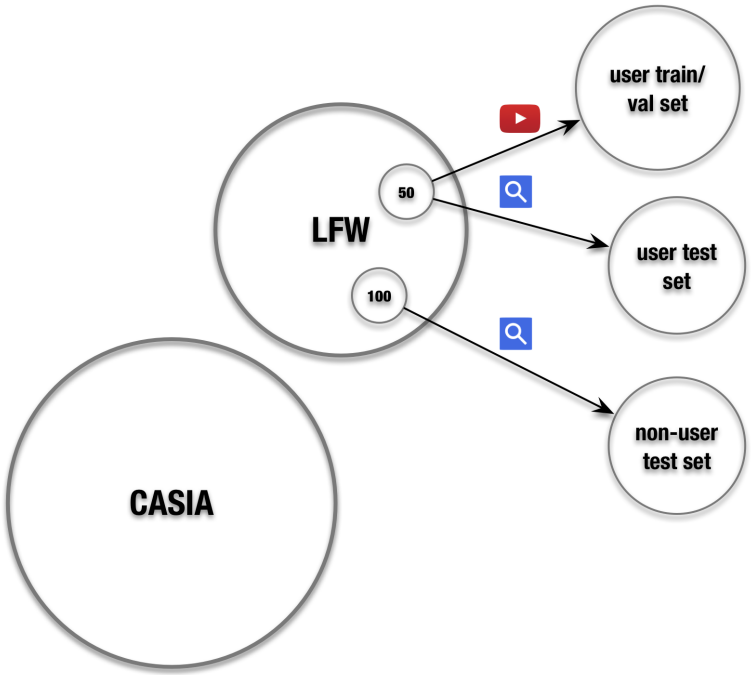
(b) Confusion matrix

Figure: Scene classification evaluation results.

8 volunteers take 759 images in "in the wild", with 638 images are selected, covering 9 scene groups shown earlier.

Face recognition and matching

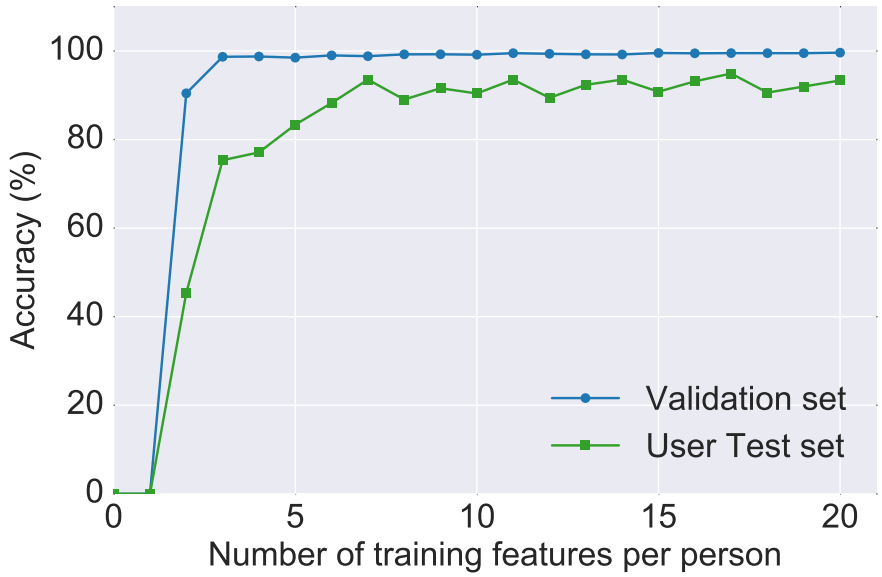
User / Non-user test set



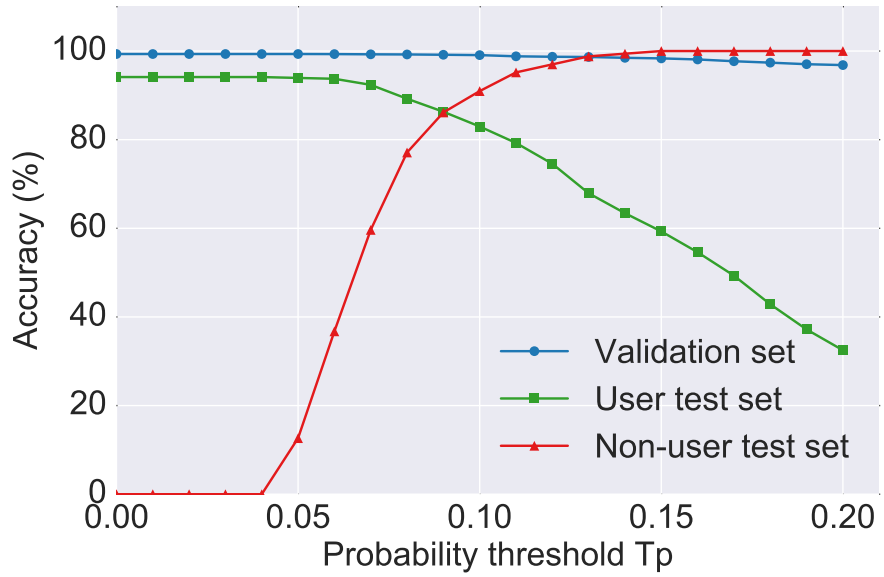
Select 50 subjects from LFW dataset, 5042 features for train and validation, 511 features as **user test set**, 166 features from 100 other subjects as **non-user test set**. (CASIA WebFace Database (Lightene CNN) does not overlap with LFW dataset).

Face recognition and matching

Recognition accuracy with T_p



(a) Training accuracy

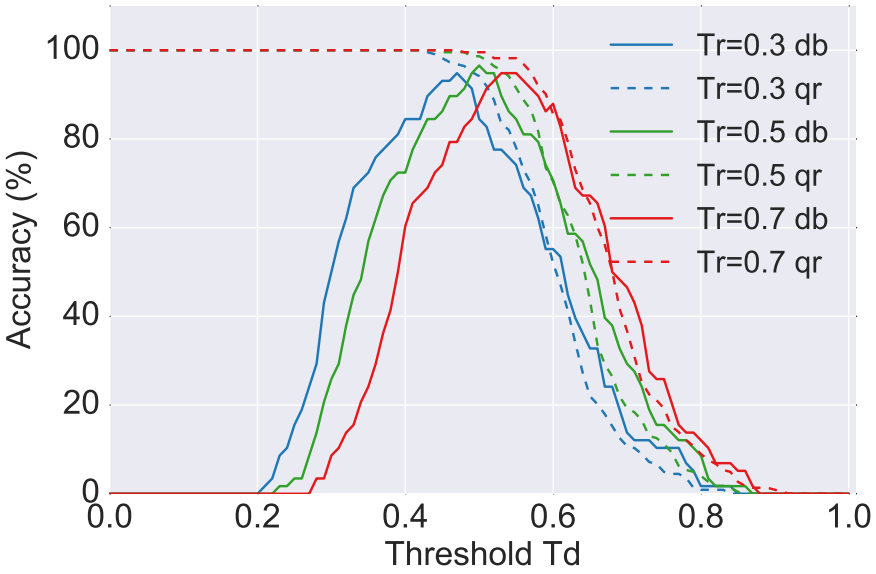


(b) Testing accuracy with threshold T_p

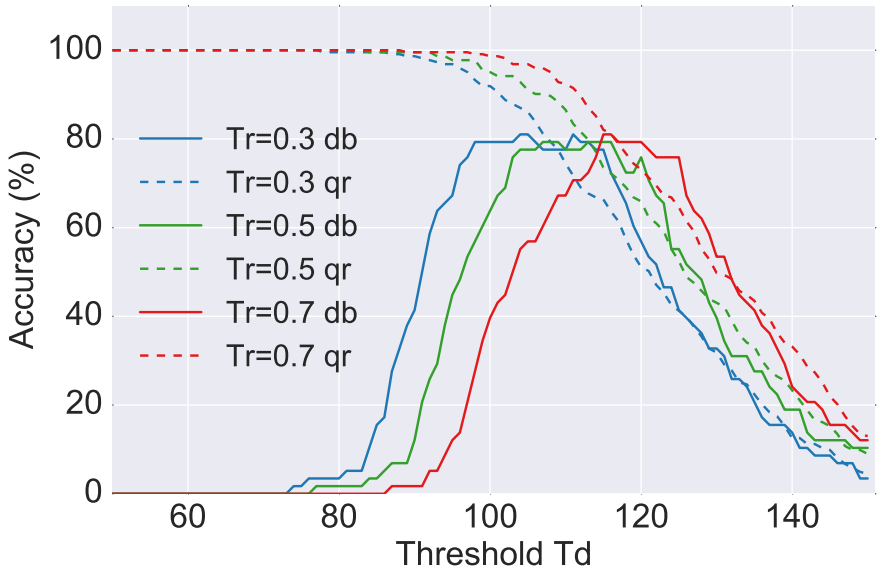
Select 50 subjects from LFW dataset, 5042 features for train and validation, 511 features as **user test set**, 166 features from 100 other subjects as **non-user test set**. (CASIA WebFace Database (Lightene CNN) does not overlap with LFW dataset).

Face recognition and matching

Matching accuracy with T_d & T_r



(a) Cosine distance

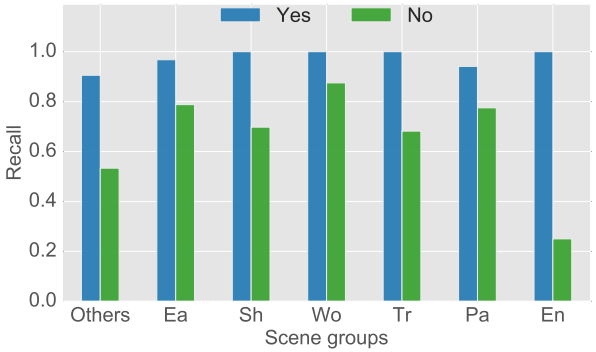


(b) Euclidean distance

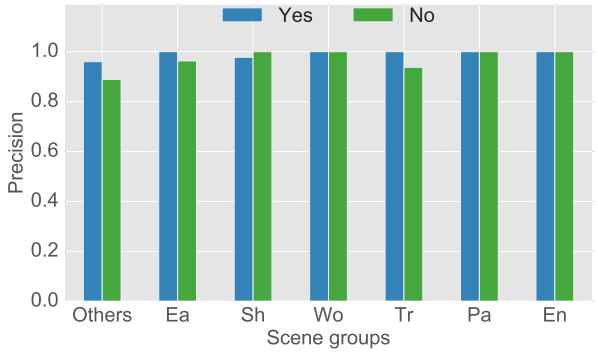
Select 23 subjects from **user test set** who has more than 10 features, use these 230 features as **database features** , the other 281 features as **query features**. The 23 subjects are **database subjects** (solid line), while the other 27 subjects are **non-database subjects** (dashed line).

Gesture recognition

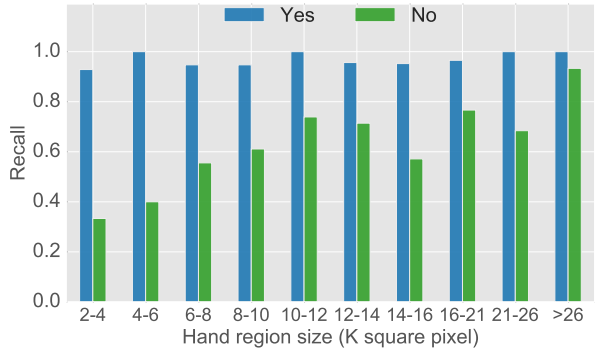
Recognition performance to scene and gesture region size



(a) Recall for different scenes



(b) Precision for different scenes



(c) Recall for different hand region sizes

High precision, recall for "no" gesture needs to be improved. More training data with smaller size.

Overall protection accuracy

Table: Cardea's overall privacy protection performance.

Overall accuracy	**86.4%**	Protection accuracy	80.4%
		No protection accuracy	91.0%
Face recognition accuracy	98.5%	"Yes" gesture recall	97.9%
scene classification recall	77.7%	"No" gesture recall	77.3%

5 volunteers to register as Cardea users, set their privacy profiles. 224 images for evaluation.

Runtime

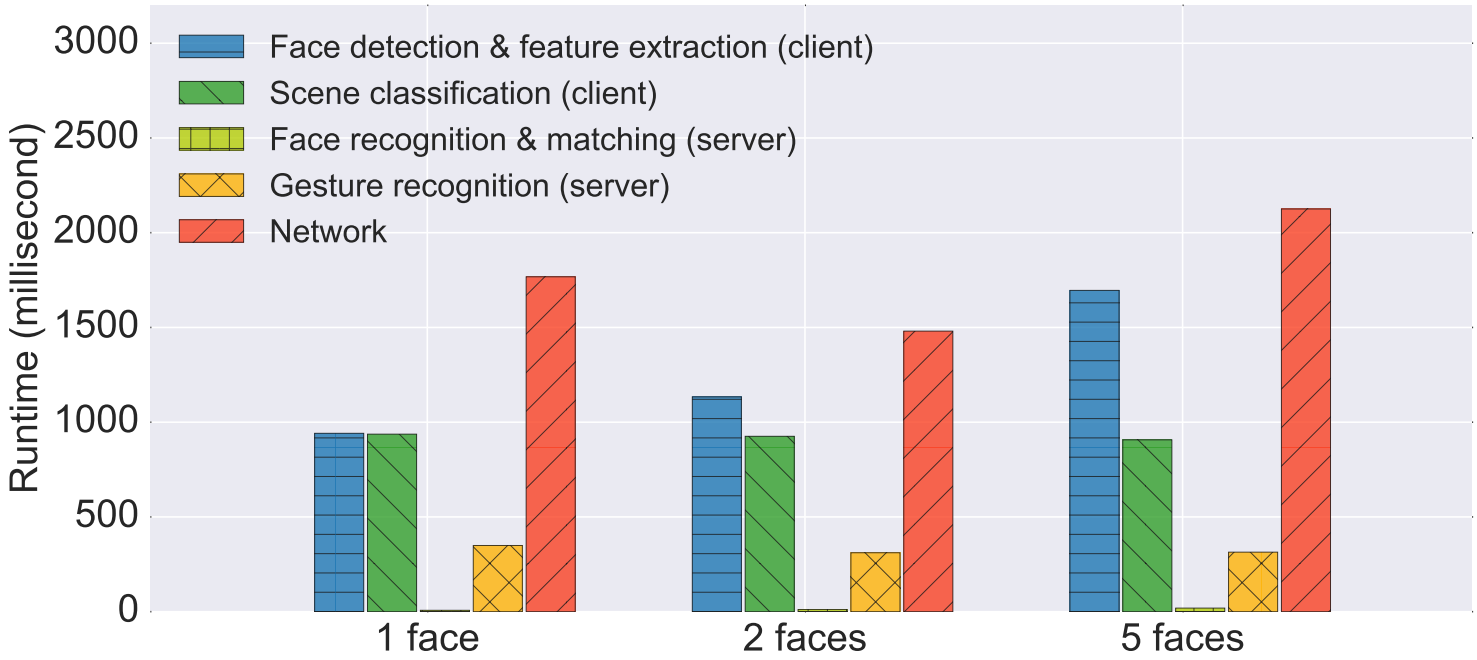


Figure: Task level runtime of Cardea.

client: Samsung Galaxy Note 4, 4×2.7 GHz Krait 450 CPU, Qualcomm Snapdragon 805 Chipset, 3GB RAM, and 16 MP, f/2.2 Camera.

server: Intel i7–5820K CPU, 16GB RAM, GeForce 980Ti Graphic Card (6GB RAM).

network: eduroam.

Energy consumption

Table: Energy consumption of Cardea with different number of faces.

	Face feature extraction	Whole process (uAh)	# of images
1 face	217.2 (std 3.4)	1134.5 (std 45.9)	$\sim$ 2800
2 faces	344.1 (std 13.1)	1276.7 (std 113.8)	$\sim$ 2500
5 faces	692.6 (std 36.5)	1641.1 (std 66.0)	$\sim$ 2000

Agenda

Summary

- Design
  - context awareness
  - interactive control
- Implementation
  - deep neural networks
  - deploy on android
- Feasibility
  - micro benchmarks
  - overall performance
  - run time & energy consumption

Thanks for your attendance !

Q&A ?