Pretext tasks
1. Predicting Rotations

September Schedule

Search

September 2023

Today

Try the new Outlook

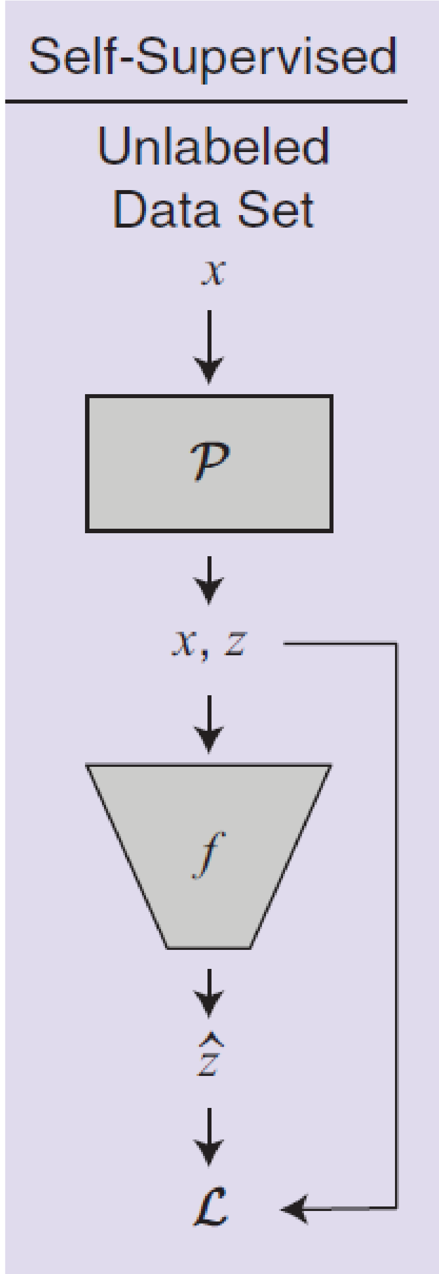
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
04-09	5 Teachers Day / Blood	6 91st Governing Body	7	8 Happy Healthy Hour	9	10
11	12	13	14	15	16	17
18 25th Foundation Day	19 Ganesh Chaturthi (Hc	20	21	22	23	24
25	26				30	01-10
2 Gandhi Jayanthi (Holi	3				7	8

5 classes in Sep

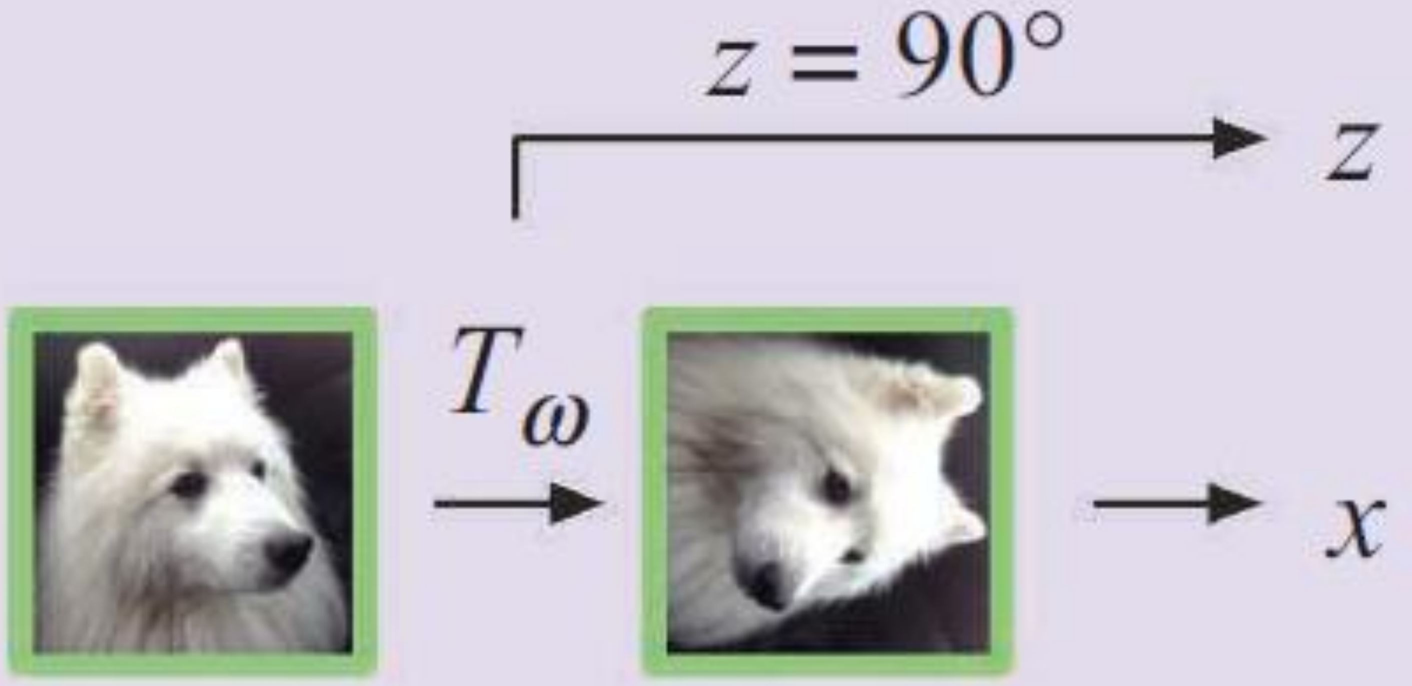
Mid-Term Exam Week

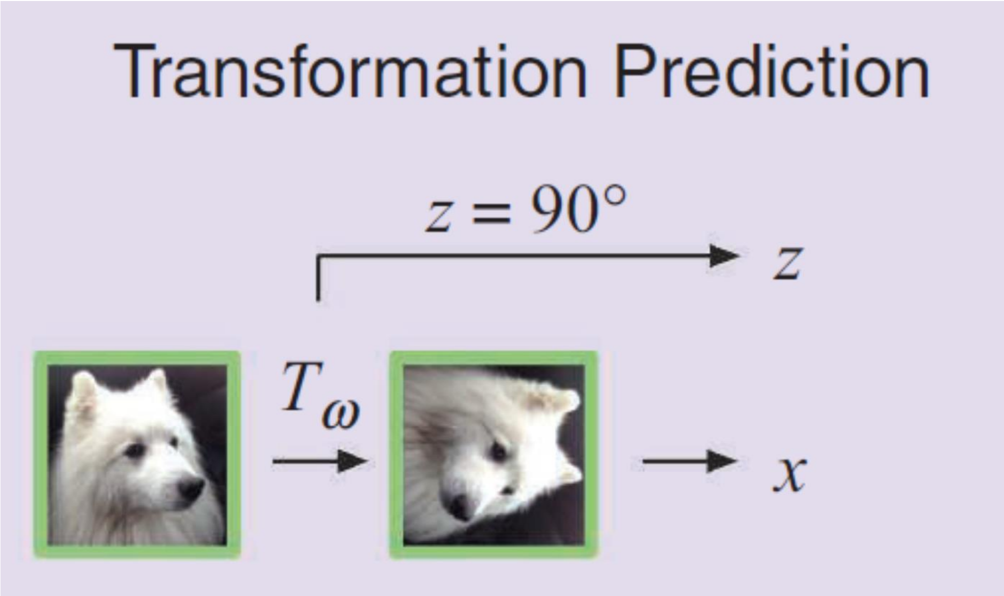
Mid-Term Break

1 2	SELF-PREDICTION	INNATE RELATIONSHIP (Context-based)	1. ROTATION 2. RELATIVE POSITION	IMAGE
3	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	1. Instance Discrimination 2. SimCLR [Contrastive Loss] 3. Theory – Guarantees / Bounds	IMAGE
4	CONTRASTIVE LEARNING	INTER-SAMPLE CLASSIFICATION	Contrastive Predictive Coding (CPC), [NCE, InfoNCE Loss]	AUDIO/ SPEECH
5	SELF-PREDICTION	GENERATIVE (VAE)	1. AE – Variational Bayes 2. VQ-VAE + AR	IMAGE AUDIO/ SPEECH
6	SELF-PREDICTION	GENERATIVE (AR)	1. AR-LM – GPT 2. Masked-LM – BERT	LANGUAGE
7	SELF-PREDICTION	MASKED-GEN (Masked LM for ASR)	1. Wav2Vec / 2.0 2. HuBERT	AUDIO/ SPEECH



Transformation Prediction





Algorithm 2. The pseudolabel generation process $\mathcal{P}$ for TP.

Input: Unlabeled data set $D_s = \{x_i^{(s)}\}_{i=1}^{M}$.
  for i from 1 to M **do**
    Sample $\omega \sim \Omega$
    $x_i \leftarrow T_\omega(x_i^{(s)})$ ▷ Apply transformation to raw input
    $z_i \leftarrow \omega$
  end for
Output: $\{x_i, z_i\}_{i=1}^{M}$.

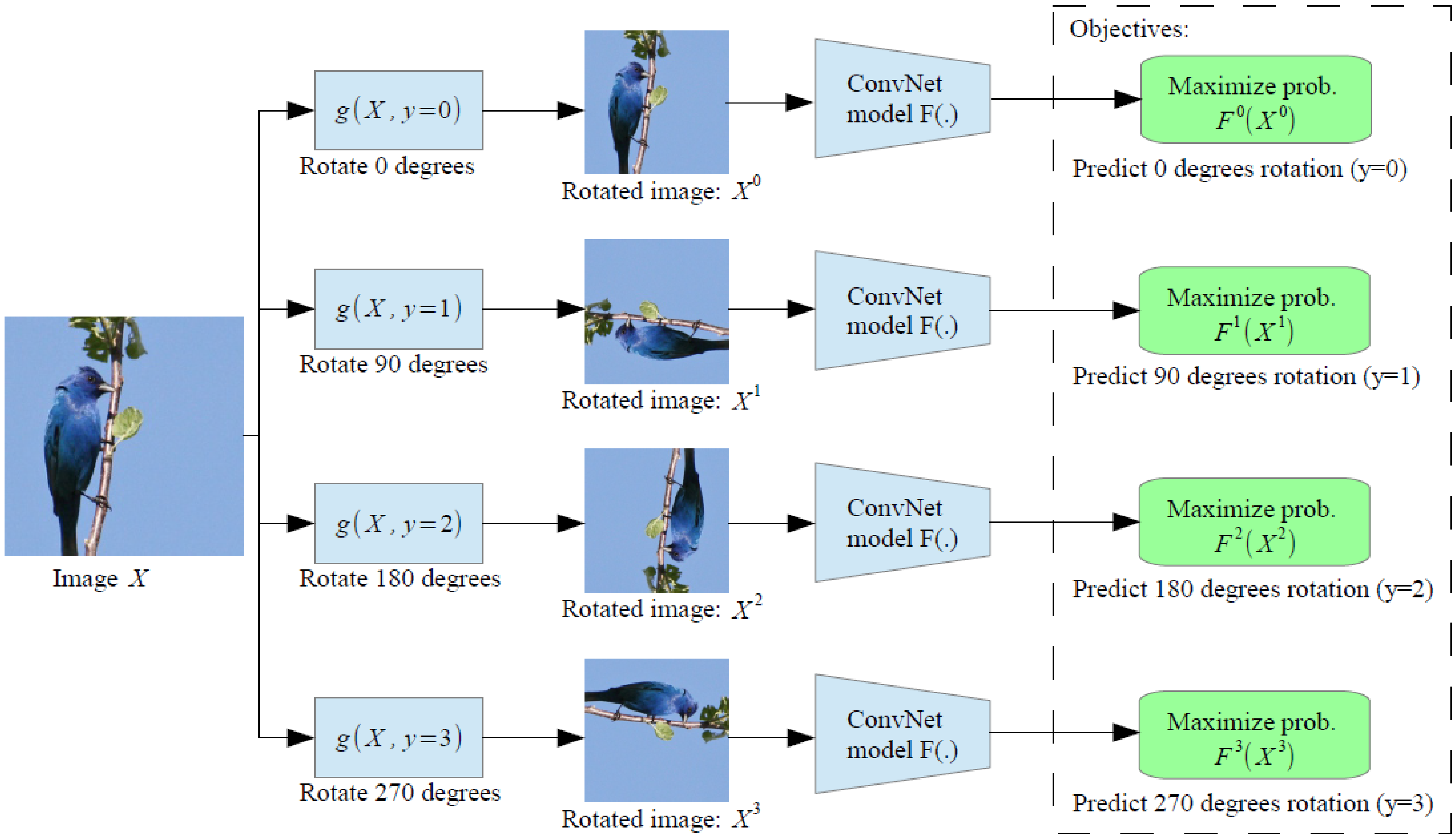
$$\theta^* = \underset{\theta,\gamma}{\operatorname{argmin}} \sum_{(x_i,z_i)\in\mathcal{P}(D_s)} \mathcal{L}_{CE}(k_\gamma(h_\theta(x_i)), z_i).$$

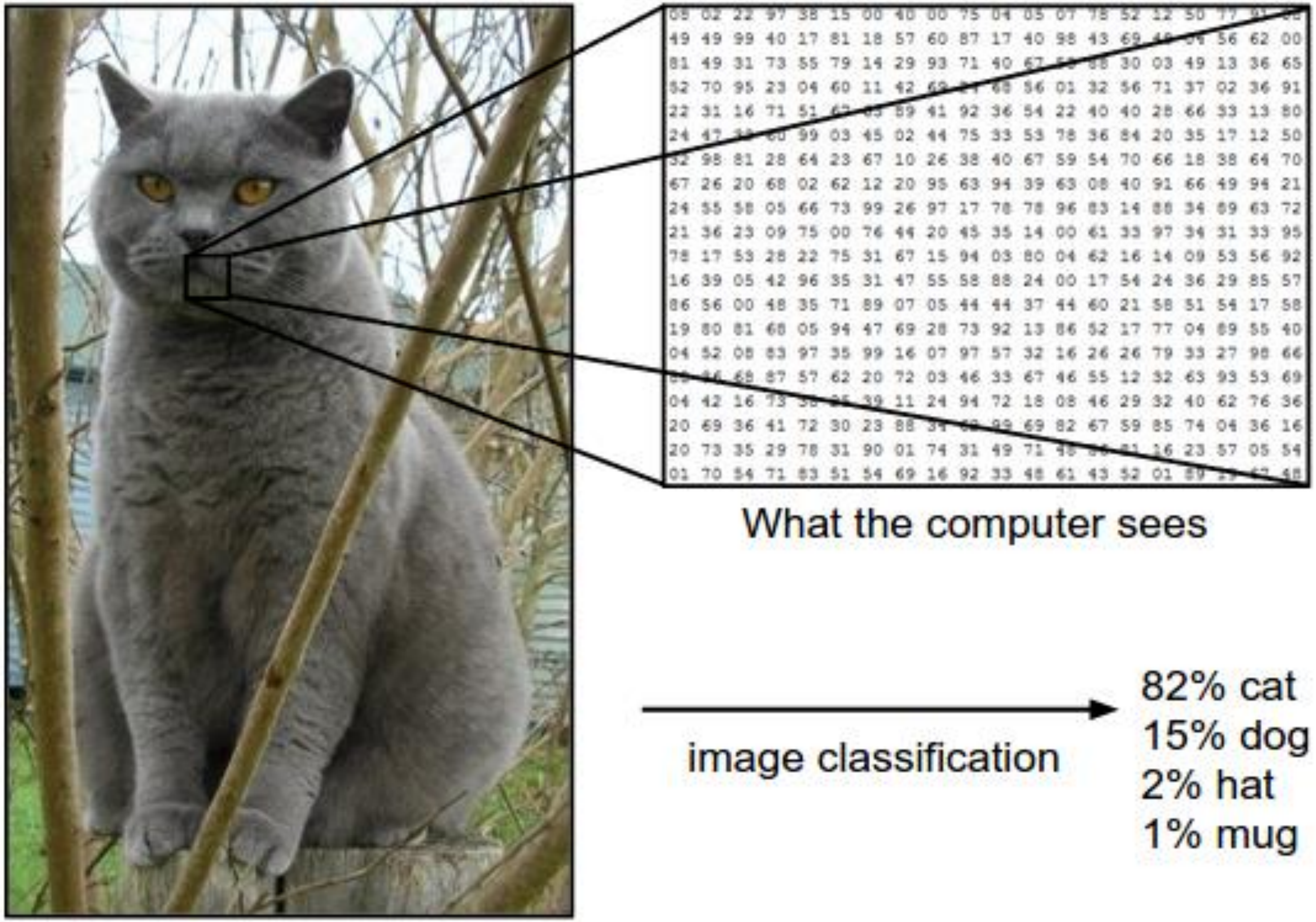
UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS

Spyros Gidaris, Praveer Singh, Nikos Komodakis
University Paris-Est, LIGM
Ecole des Ponts ParisTech
{spyros.gidaris,praveer.singh,nikos.komodakis}@enpc.fr

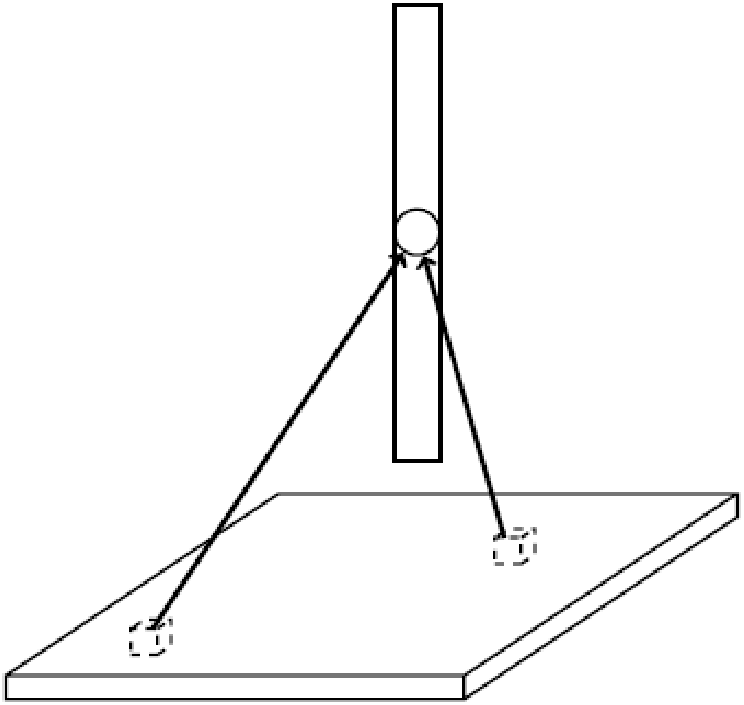
90° rotation | 270° rotation | 180° rotation | 0° rotation | 270° rotation

Figure 1: Images rotated by random multiples of 90 degrees (e.g., 0, 90, 180, or 270 degrees). The core intuition of our self-supervised feature learning approach is that if someone is not aware of the concepts of the objects depicted in the images, he cannot recognize the rotation that was applied to them.


Objectives:
$g(X, y=0)$
Rotate 0 degrees
Rotated image: X^0
ConvNet model F(.)
Maximize prob. $F^0(X^0)$
Predict 0 degrees rotation (y=0)
$g(X, y=1)$
Rotate 90 degrees
Rotated image: X^1
ConvNet model F(.)
Maximize prob. $F^1(X^1)$
Predict 90 degrees rotation (y=1)
Image X
$g(X, y=2)$
Rotate 180 degrees
Rotated image: X^2
ConvNet model F(.)
Maximize prob. $F^2(X^2)$
Predict 180 degrees rotation (y=2)
$g(X, y=3)$
Rotate 270 degrees
Rotated image: X^3
ConvNet model F(.)
Maximize prob. $F^3(X^3)$
Predict 270 degrees rotation (y=3)

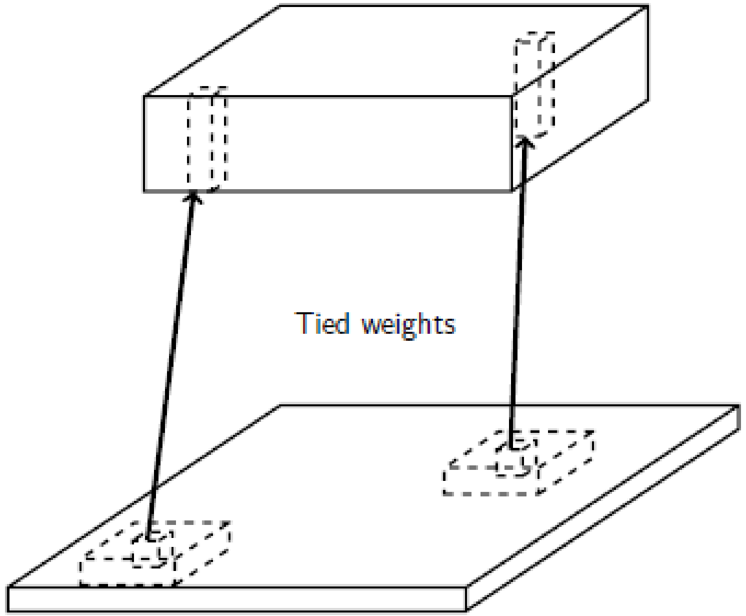

What the computer sees
image classification
82% cat
15% dog
2% hat
1% mug

Fully connected



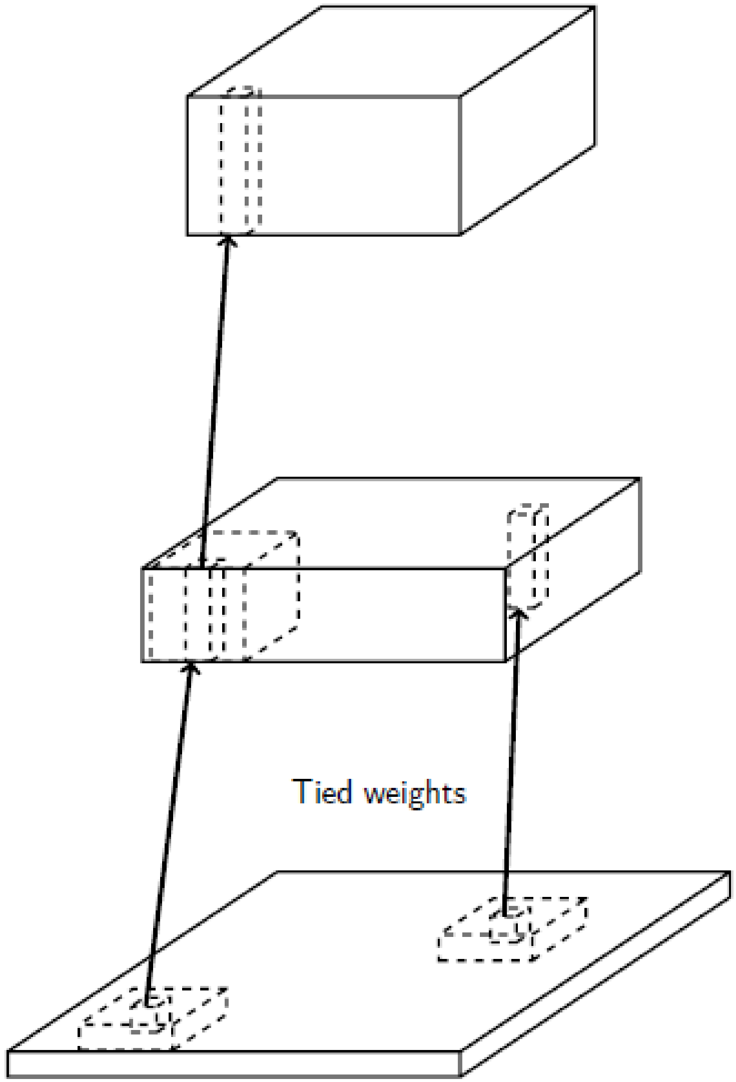
Each hidden unit looks at the entire image.

Convolutional



Each column of hidden units looks at a different patch of input.

Stack multiple layers of convolutions



2-D Convolution

What does this convolution kernel do?

$*$

0	1	0
1	4	1
0	1	0

What does this convolution kernel do?

$*$

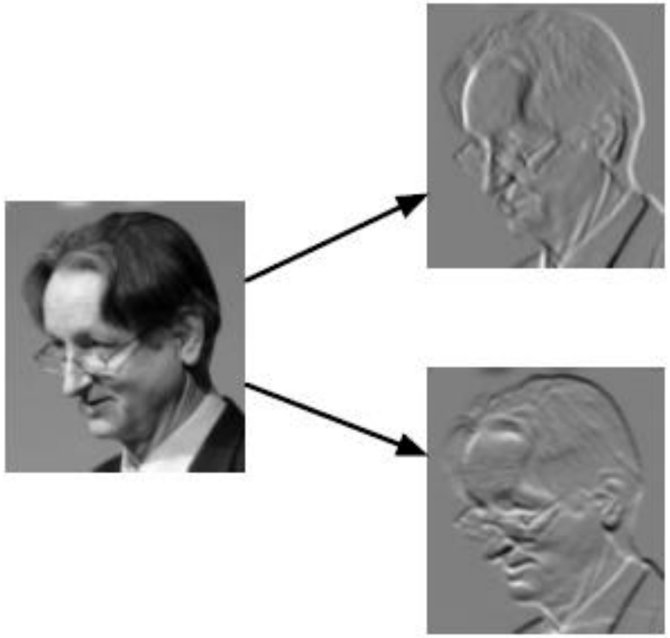
0	-1	0
-1	8	-1
0	-1	0

What does this convolution kernel do?

$*$

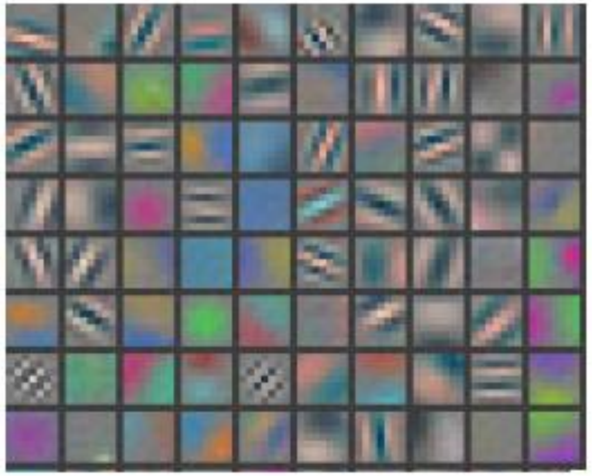
0	-1	0
-1	4	-1
0	-1	0

The convolution layer has a set of filters. Its output is a set of feature maps, each one obtained by convolving the image with a filter.

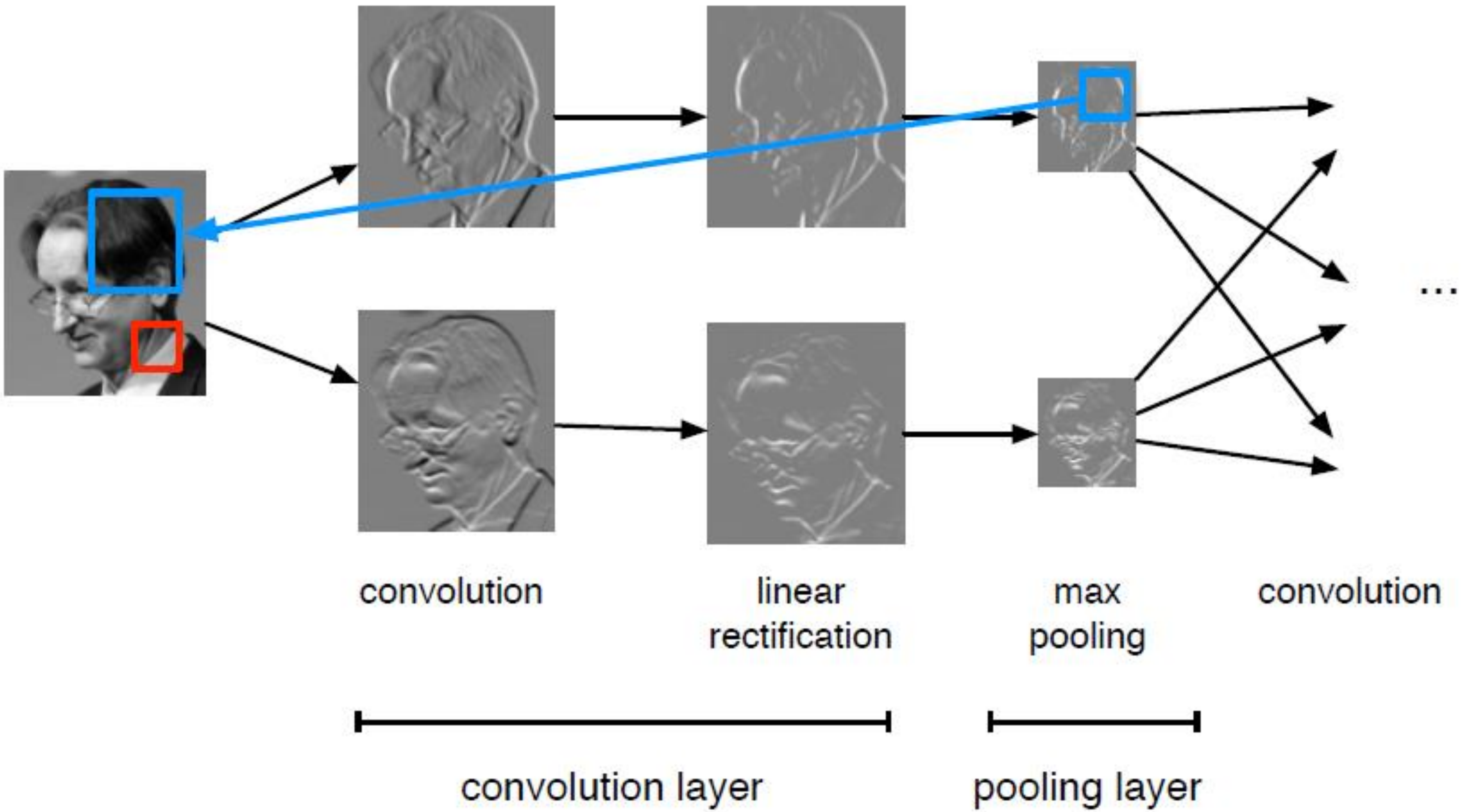


convolution

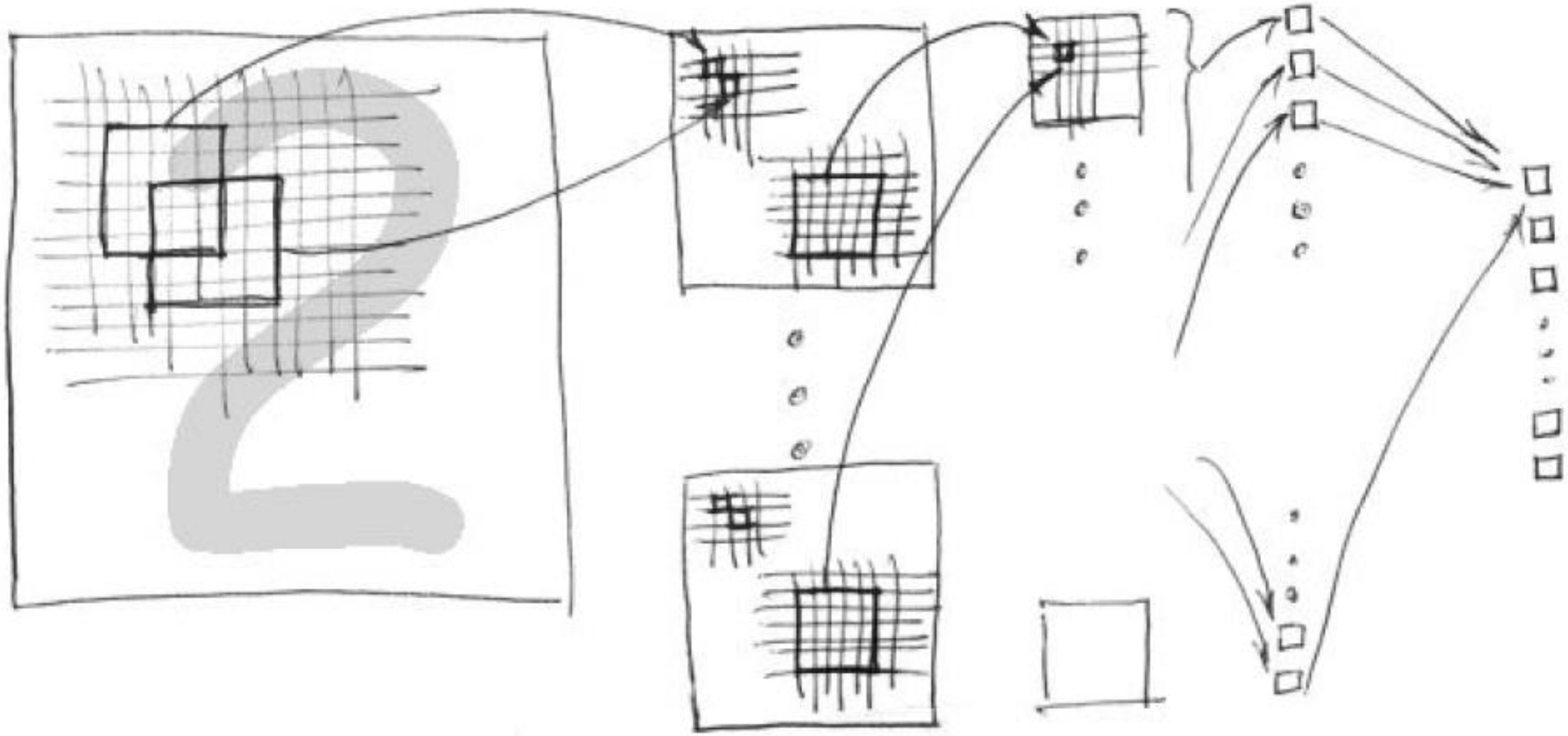
Example first-layer filters



(Zeiler and Fergus, 2013, Visualizing and understanding convolutional networks)

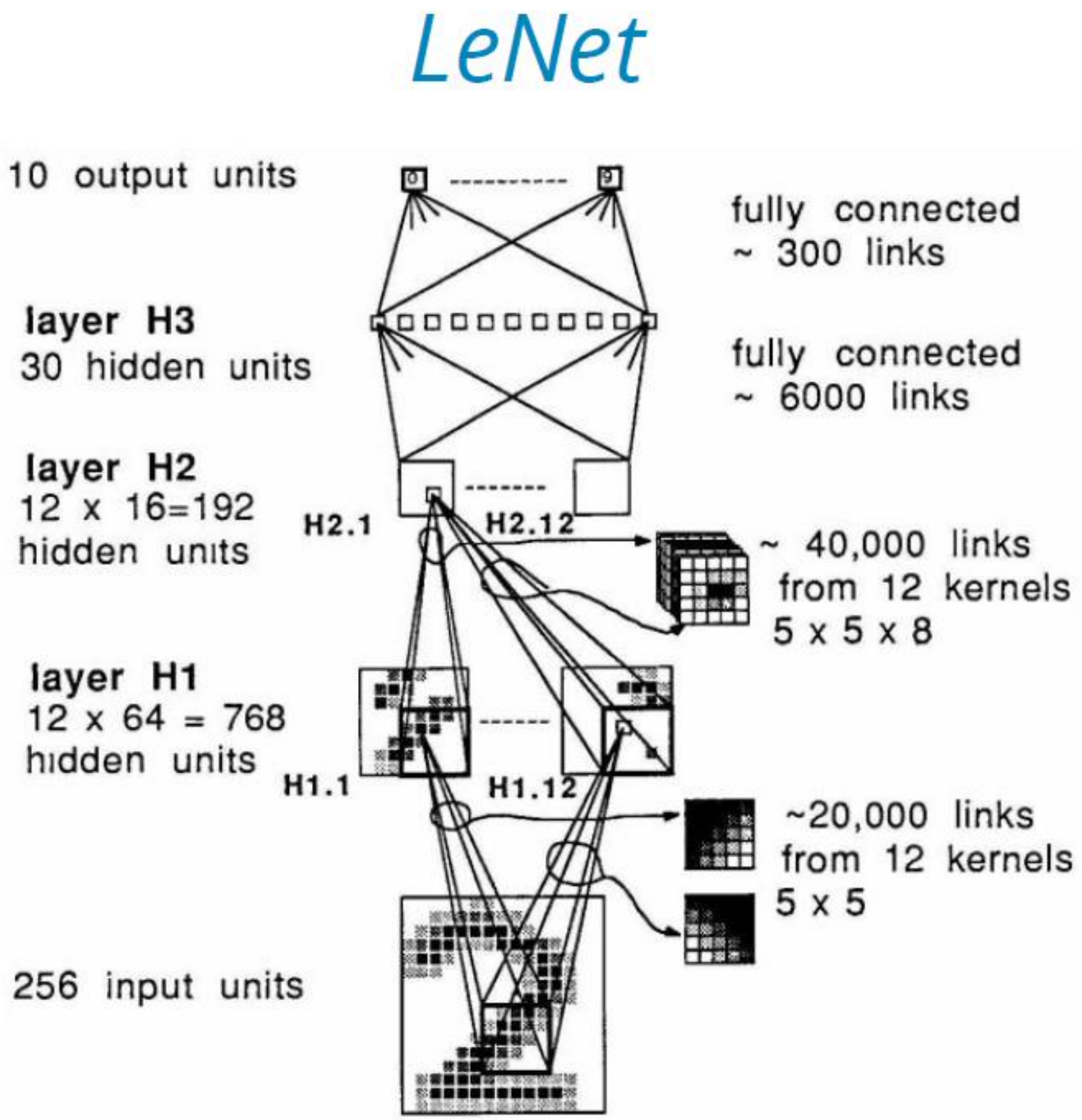

...
convolution
linear
rectification
max
pooling
convolution
convolution layer
pooling layer

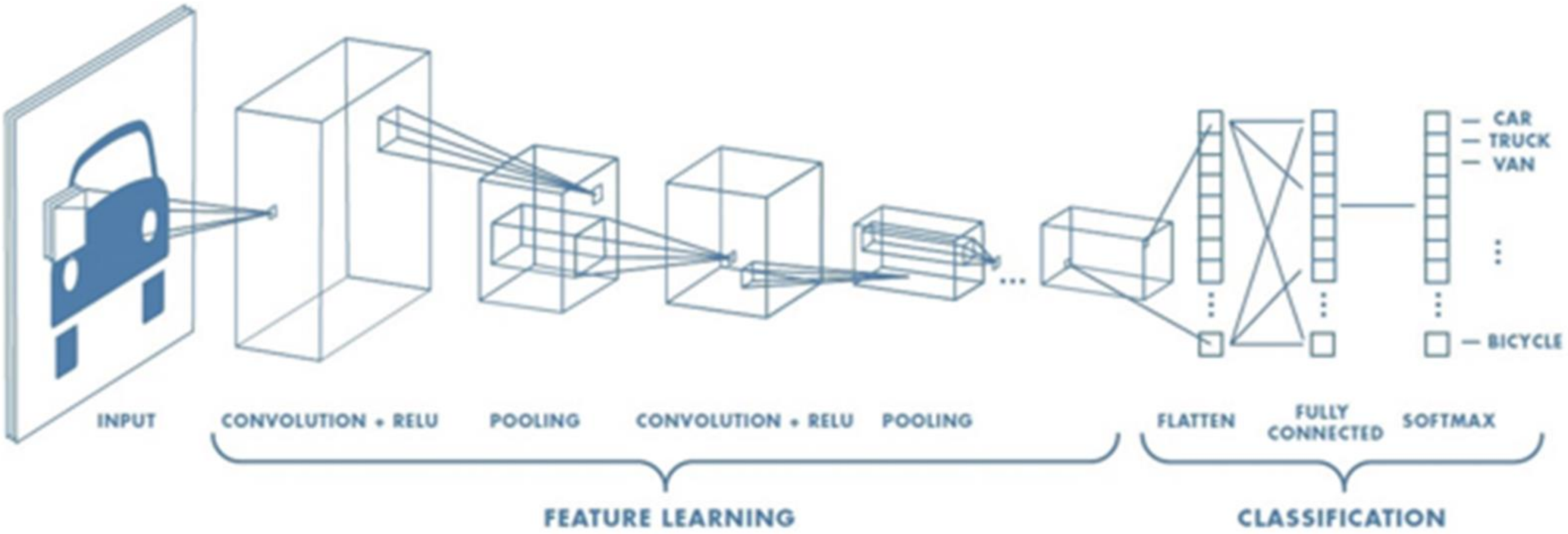
Convolutional Neural Network



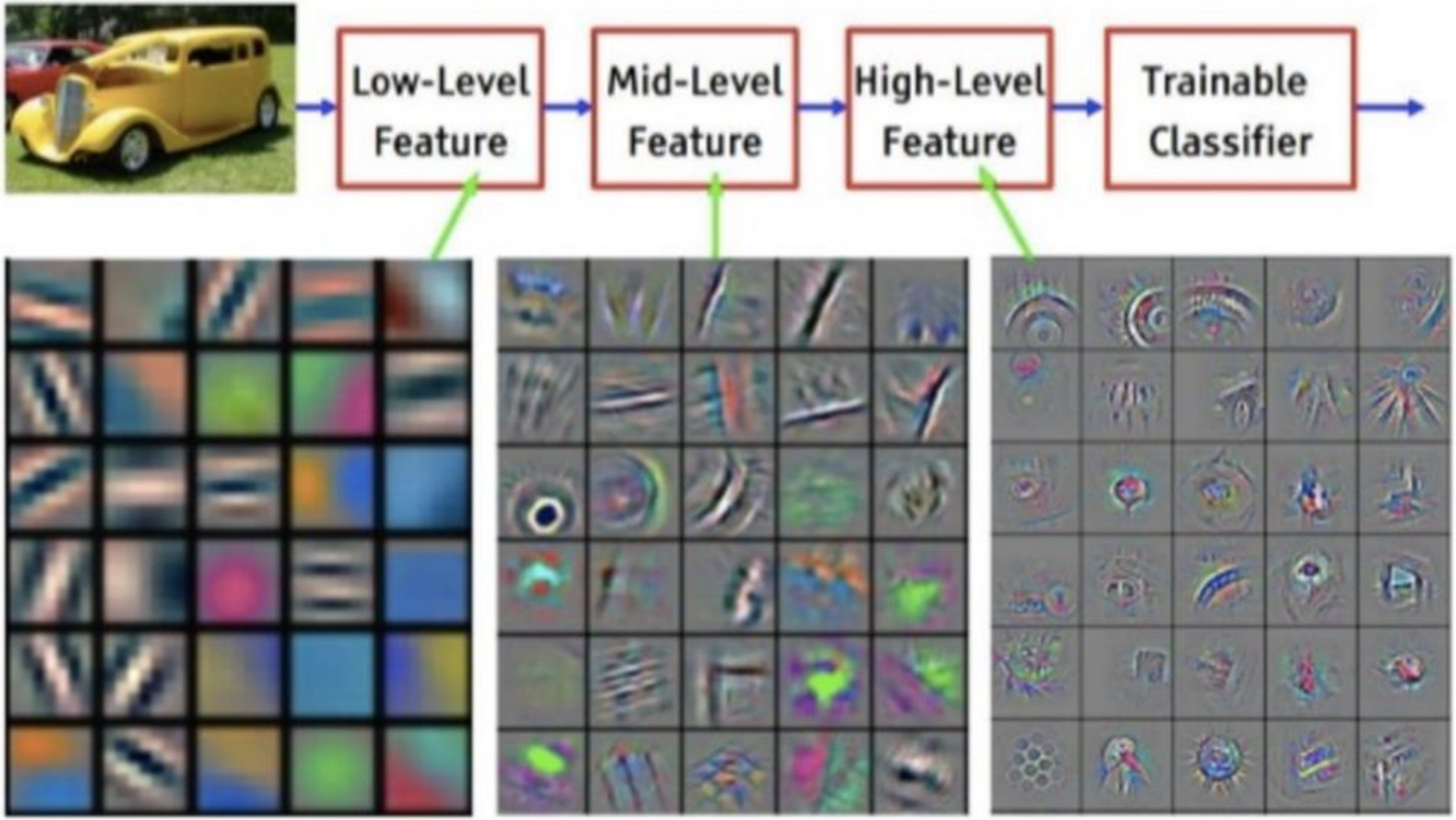
LeNet

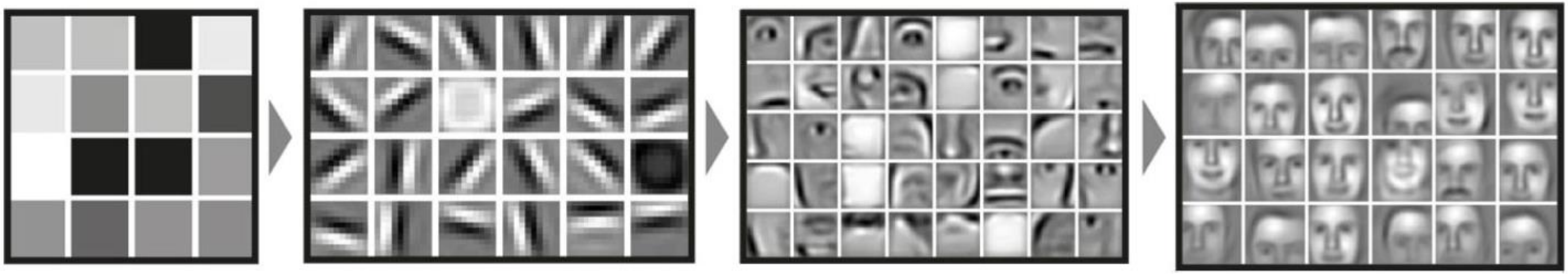
*LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., **1989. Backpropagation applied to handwritten zip code recognition.** Neural computation, 1(4), pp.541-551.*



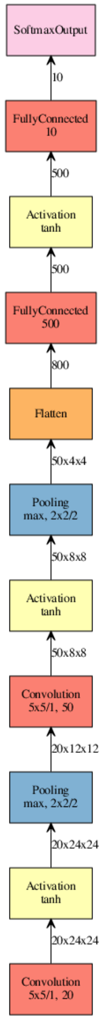


Standard architecture of a CNN

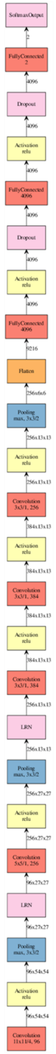

Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier



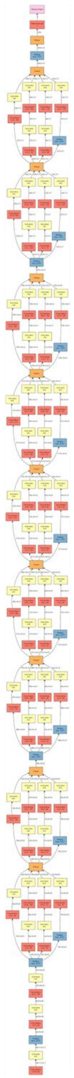
“LeNet”
1998:



“AlexNet”
2012:

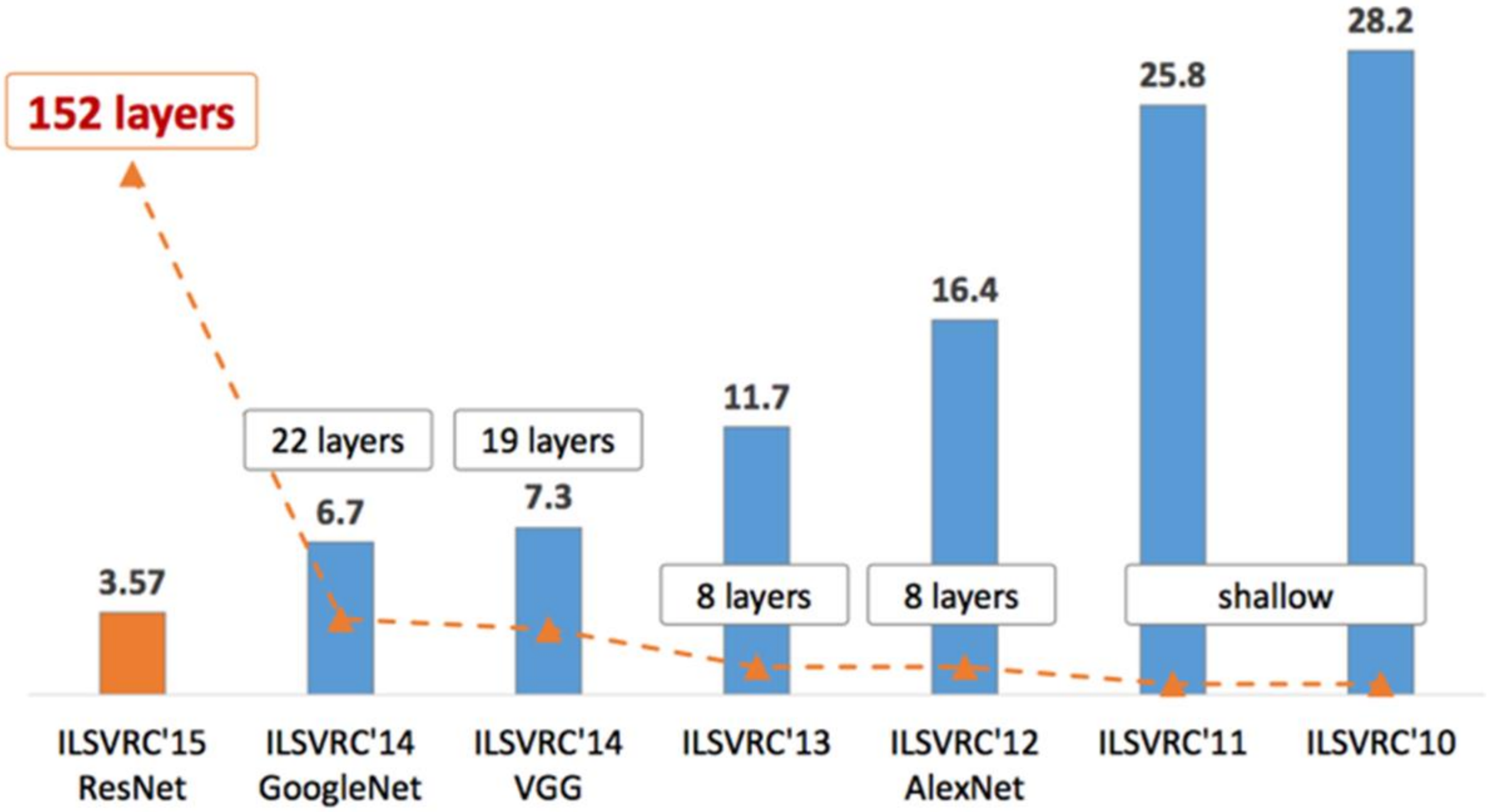


“GoogLeNet”
2014:



“Inception v3”
2015:

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



Back to…

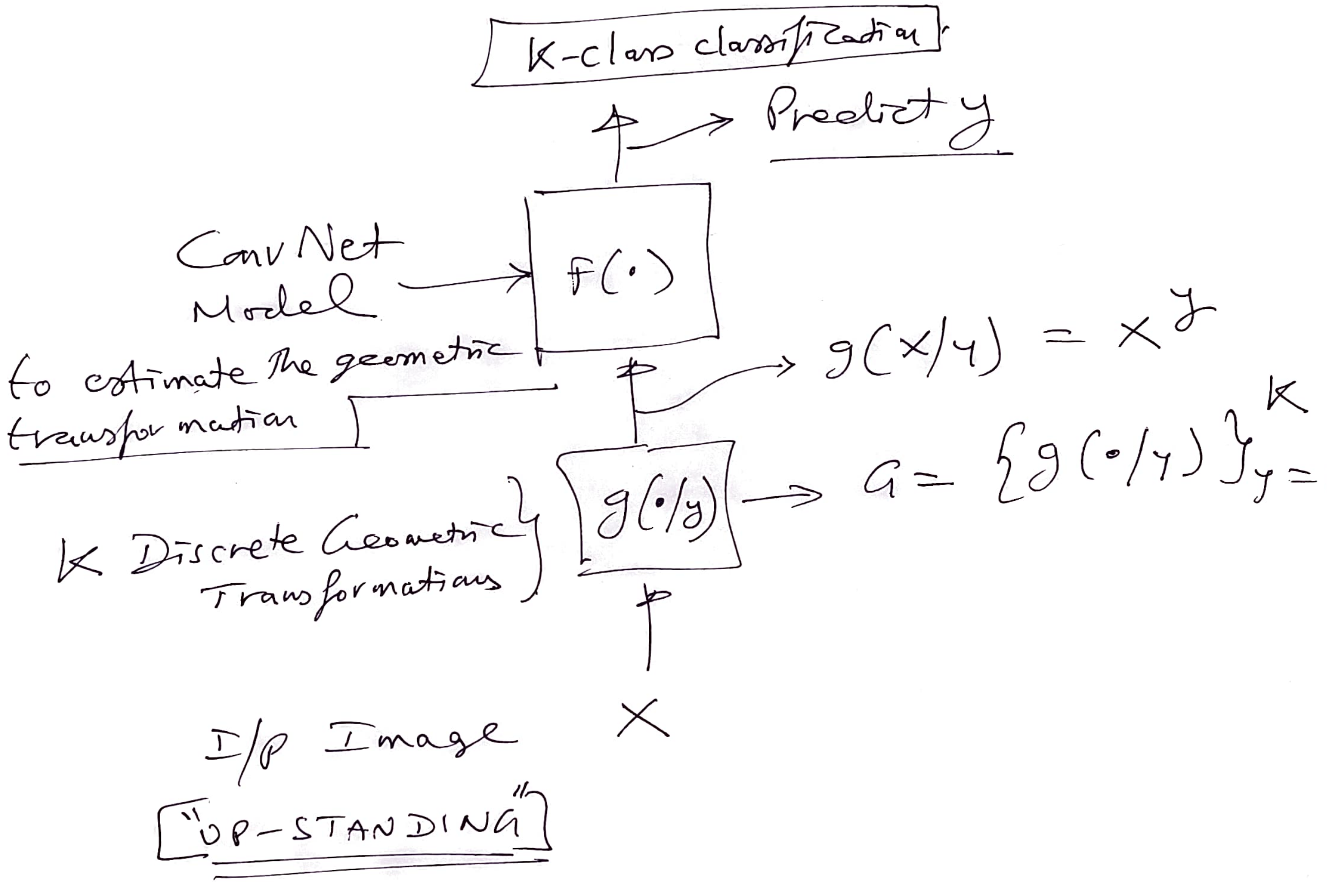
Published as a conference paper at ICLR 2018

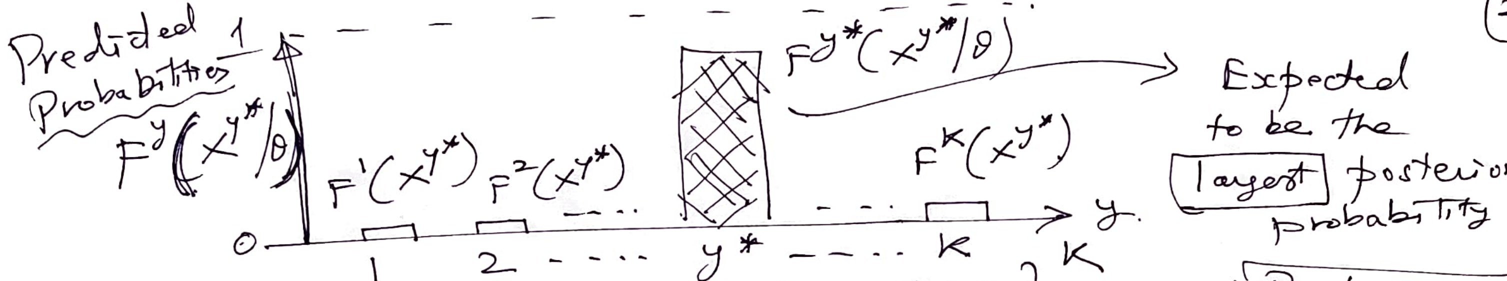
UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS

Spyros Gidaris, Praveer Singh, Nikos Komodakis
University Paris-Est, LIGM
Ecole des Ponts ParisTech
{spyros.gidaris,praveer.singh,nikos.komodakis}@enpc.fr

ICLR-2018

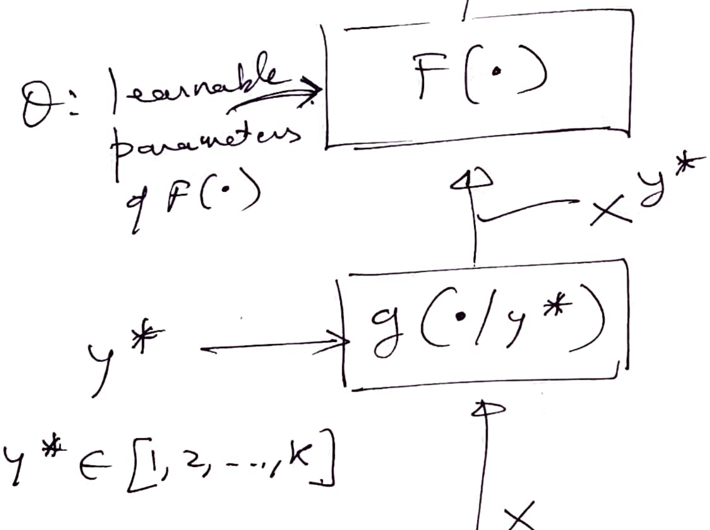
PREDICTING IMAGE ROTATIONS

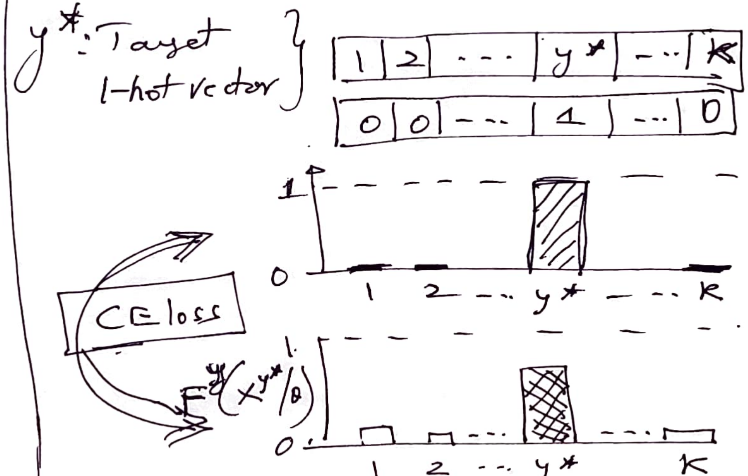




$$F(x^{y^*}/\theta) = \{F^y(x^{y^*}/\theta)\}_{y=1}^{K}$$

Posterior Probability Vector of K-dim.





③

CE loss between

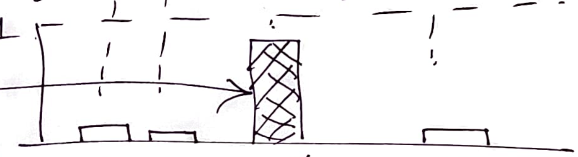
- 1-hot encoding
- &
- $\{F^{y}(x^{y^*}|\theta)\}_{y=1}^{K}$

0	0	..	1	- - -	0
1	2	---	y^*	---	K

CE

1

0



1 2 ... y^* --- K.

Maximize this

$$F^{y^*}(x^{y^*}|\theta)$$

Reduces to

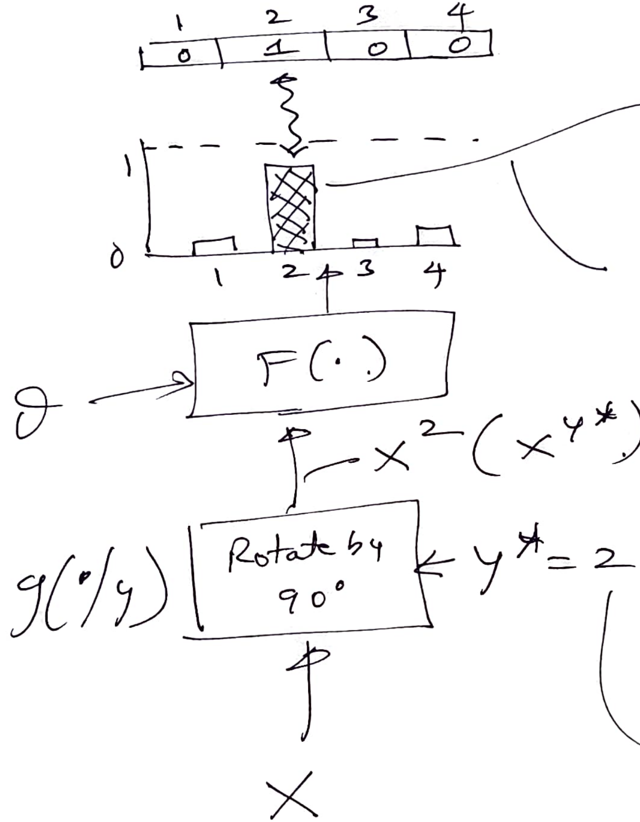
NLL: Negative Log Likelihood

$$\text{Loss} = -\log\left(F^{y^*}(g(x|y^*)|\theta)\right)$$

$K=4$ $y^*=2$

$Rot(x, \phi)$: operator that rotates image x by $\underbrace{0, 90, 180, 270}_{\phi}$ °

$$G=\{g(x|y)\}_{y=1}^{4} \quad \| \quad g(x|y) = Rot(x, (y-1)90)$$



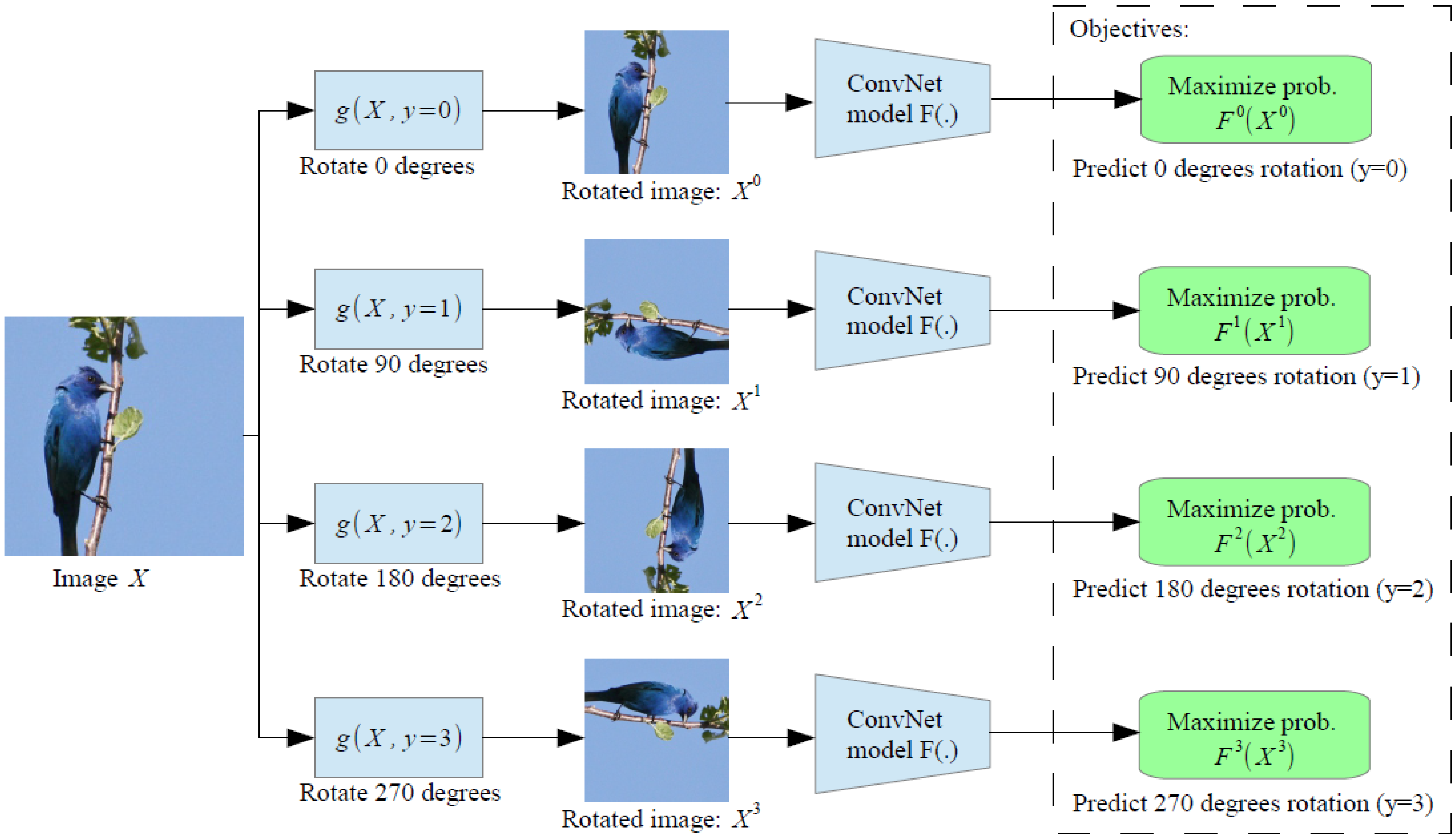
Maximize this

$\log(F^2(x^2/\theta))$

Minimize loss [NLL]

$\boxed{-\log(F^2(x^2/\theta))}$

$$\boxed{\begin{matrix}[1 & 2 & 3 & 4] \\ [0 & 90 & 180 & 270]\end{matrix}}$$


Objectives:
Image X
$g(X, y=0)$
Rotate 0 degrees
Rotated image: X^0
ConvNet model F(.)
Maximize prob. $F^0(X^0)$
Predict 0 degrees rotation (y=0)
$g(X, y=1)$
Rotate 90 degrees
Rotated image: X^1
ConvNet model F(.)
Maximize prob. $F^1(X^1)$
Predict 90 degrees rotation (y=1)
$g(X, y=2)$
Rotate 180 degrees
Rotated image: X^2
ConvNet model F(.)
Maximize prob. $F^2(X^2)$
Predict 180 degrees rotation (y=2)
$g(X, y=3)$
Rotate 270 degrees
Rotated image: X^3
ConvNet model F(.)
Maximize prob. $F^3(X^3)$
Predict 270 degrees rotation (y=3)

On a given unlabelled/unsupervised SSL Trg set

$$D = \{X_i\}_{i=1}^{N} \quad \text{of } N \text{ samples}$$

Find ConvNet Model θ^*

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \text{loss}(x_i, \theta)$$

with associated loss

objective fn: Min Loss $= \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \text{loss}(X_i, \theta)$

where $\text{Loss}(X_i, \theta) = -\frac{1}{K} \sum_{y=1}^{K} \log(F^y(g(x_i|y)|\theta))$

$K = 4$

y	$y-1$	$g(x\|y)$
$y=1$	0	$\text{Rot}(X, 0 \cdot 90) \rightarrow 0°$
$y=2$	1	$\text{Rot}(X, 1 \cdot 90) \rightarrow 90°$
$y=3$	2	$\text{Rot}(X, 2 \cdot 90) \rightarrow 180°$
$y=4$	3	$\text{Rot}(X, 3 \cdot 90) \rightarrow 270°$

$$Loss(x_i, \theta) = -\frac{1}{K} \sum_{y=1}^{K} \log(F^y(g(x_i|y)|\theta))$$

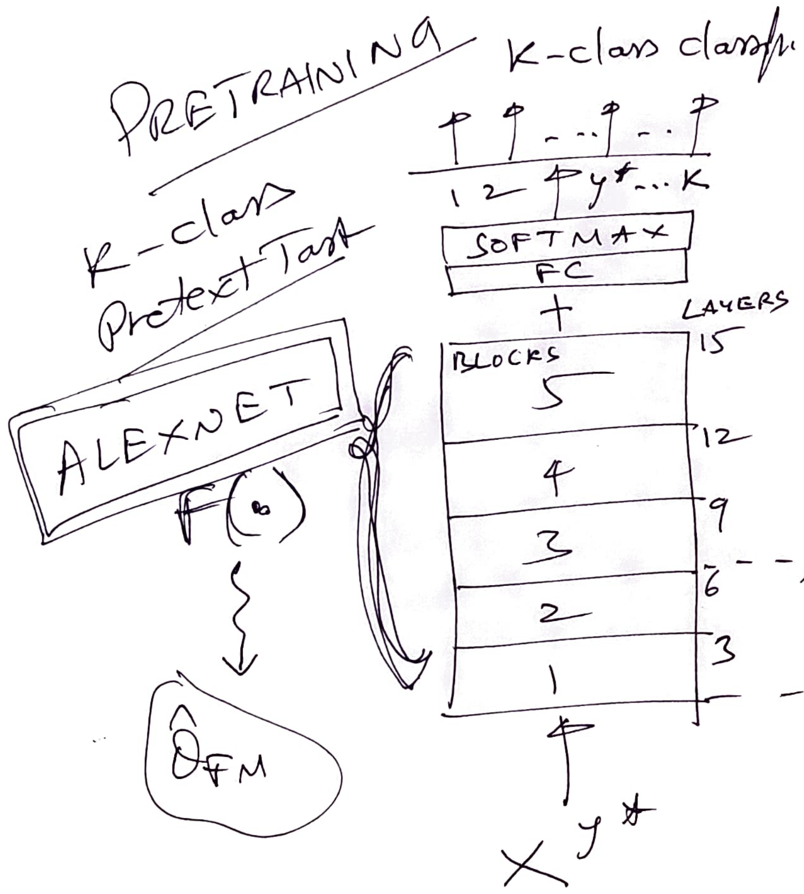
For a $x_i \in D = \{x_i\}_{i=1}^{N}$

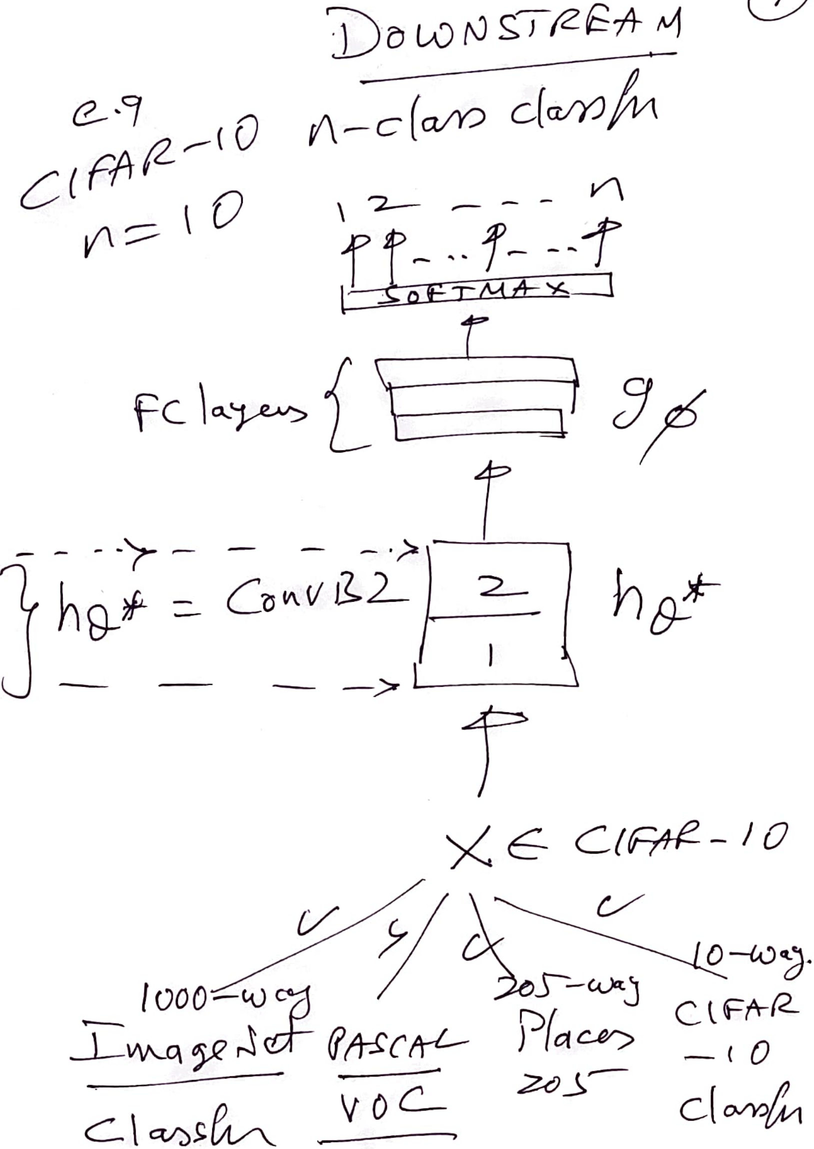
— apply all K geometric transformations [Rotations]

$K = 4 \Rightarrow 0, 90, 180, 270$

— Create $\{x_i^y\} = g(x_i/y)$ $\underline{\forall y = 1, 2, 3, 4}$ $\rightarrow K$

— Compute $\underline{Loss(x_i, \theta)}$ $\forall \{x_i^y\}_{y=1}^{K}$


PRETRAINING
K-class classfn
K-class
Pretext Task
ALEXNET
F(•)
$\hat{\theta}_{FM}$
1 2 y* ... K
SOFTMAX
FC
LAYERS
BLOCKS
5
4
3
2
1
15
12
9
6
3
x^{y^*}
h_{θ^*} = ConvB2


DOWNSTREAM
e.g.
CIFAR-10
n=10
n-class classfn
1 2 ... n
SOFTMAX
FC layers
g_ϕ
2
1
h_{θ^*}
X ∈ CIFAR-10
1000-way
ImageNet
Classfn
PASCAL
VOC
205-way
Places
205
10-way.
CIFAR
-10
classfn

See TABLE-1

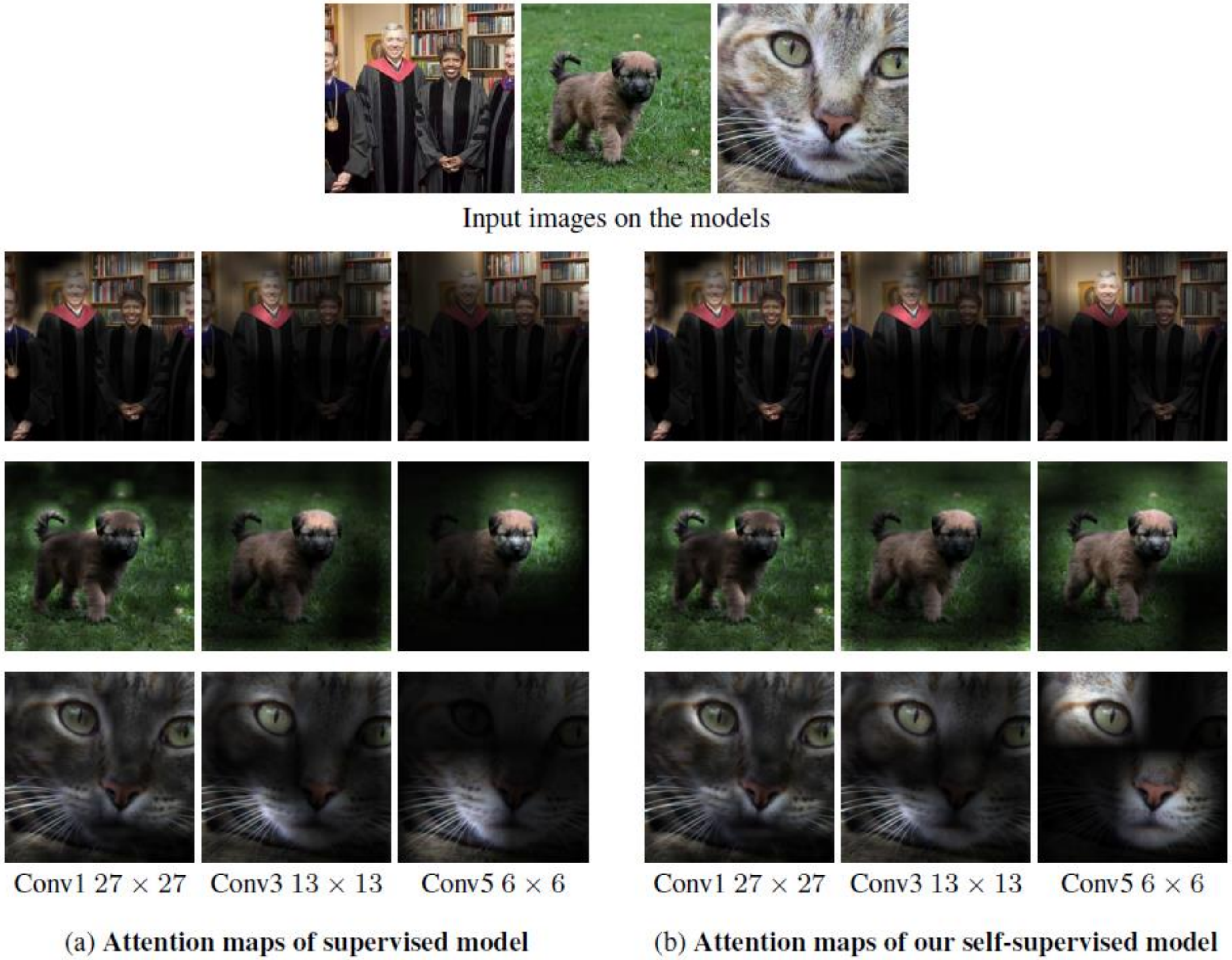
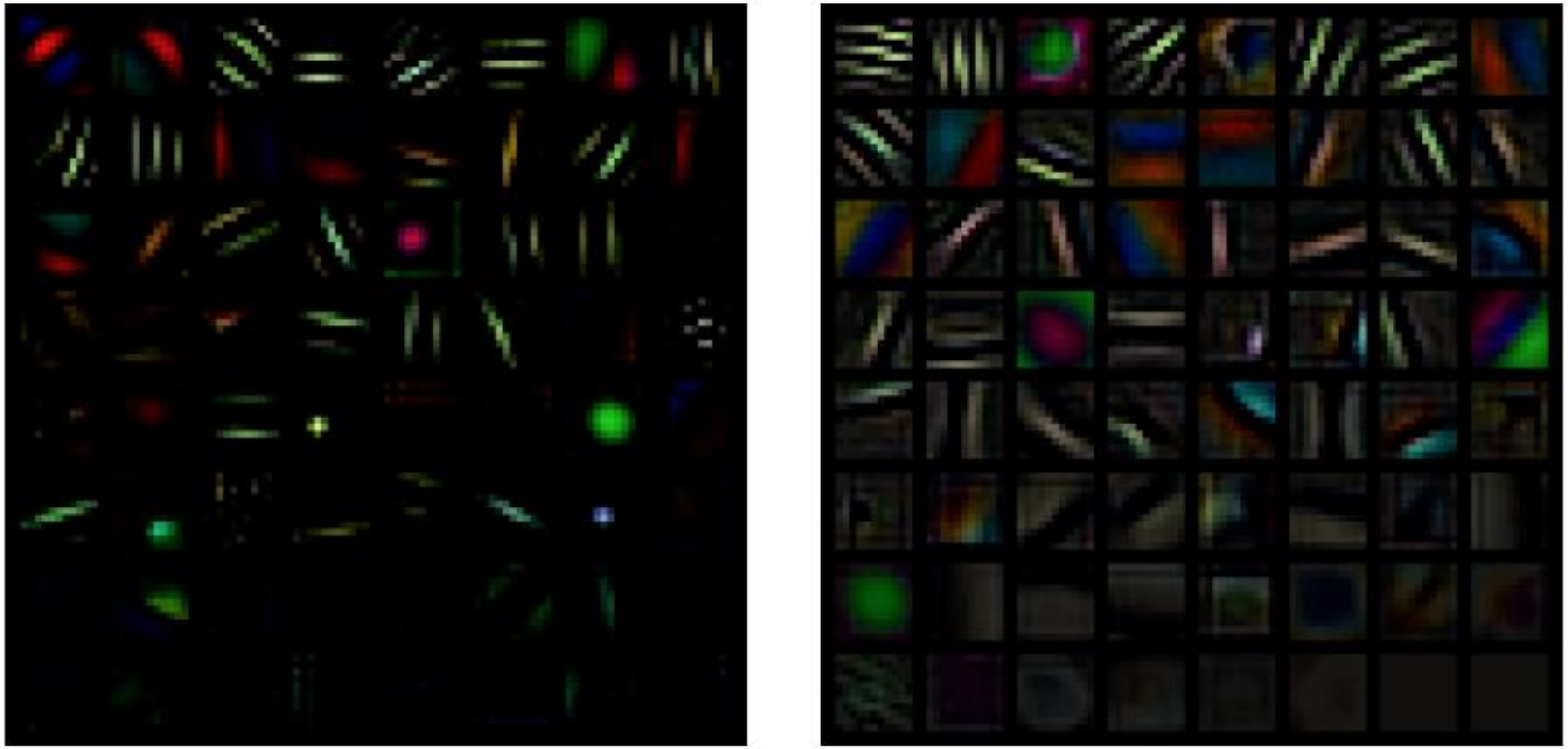


Figure 3: Attention maps generated by an AlexNet model trained **(a)** to recognize objects (supervised), and **(b)** to recognize image rotations (self-supervised). In order to generate the attention map of a conv. layer we first compute the feature maps of this layer, then we raise each feature activation on the power p, and finally we sum the activations at each location of the feature map. For the conv. layers 1, 2, and 3 we used the powers $p = 1$, $p = 2$, and $p = 4$ respectively. For visualization of our self-supervised model's attention maps for all the rotated versions of the images see Figure 6 in appendix A.



(a) **Supervised** (b) **Self-supervised to recognize rotations**

Figure 4: First layer filters learned by a AlexNet model trained on **(a)** the supervised object recognition task and **(b)** the self-supervised task of recognizing rotated images. We observe that the filters learned by the self-supervised task are mostly oriented edge filters on various frequencies and, remarkably, they seem to have more variety than those learned on the supervised task.

Table 1: Evaluation of the unsupervised learned features by measuring the classification accuracy that they achieve when we train a non-linear object classifier on top of them. The reported results are from CIFAR-10. The size of the ConvB1 feature maps is $96 \times 16 \times 16$ and the size of the rest feature maps is $192 \times 8 \times 8$.

Model	ConvB1	ConvB2	ConvB3	ConvB4	ConvB5
RotNet with 3 conv. blocks	85.45	88.26	62.09	-	-
RotNet with 4 conv. blocks	85.07	89.06	86.21	61.73	-
RotNet with 5 conv. blocks	85.04	**89.76**	86.82	74.50	50.37

Table 2: Exploring the quality of the self-supervised learned features w.r.t. the number of recognized rotations. For all the entries we trained a non-linear classifier with 3 fully connected layers (similar to Table 1) on top of the feature maps generated by the 2nd conv. block of a RotNet model with 4 conv. blocks in total. The reported results are from CIFAR-10.

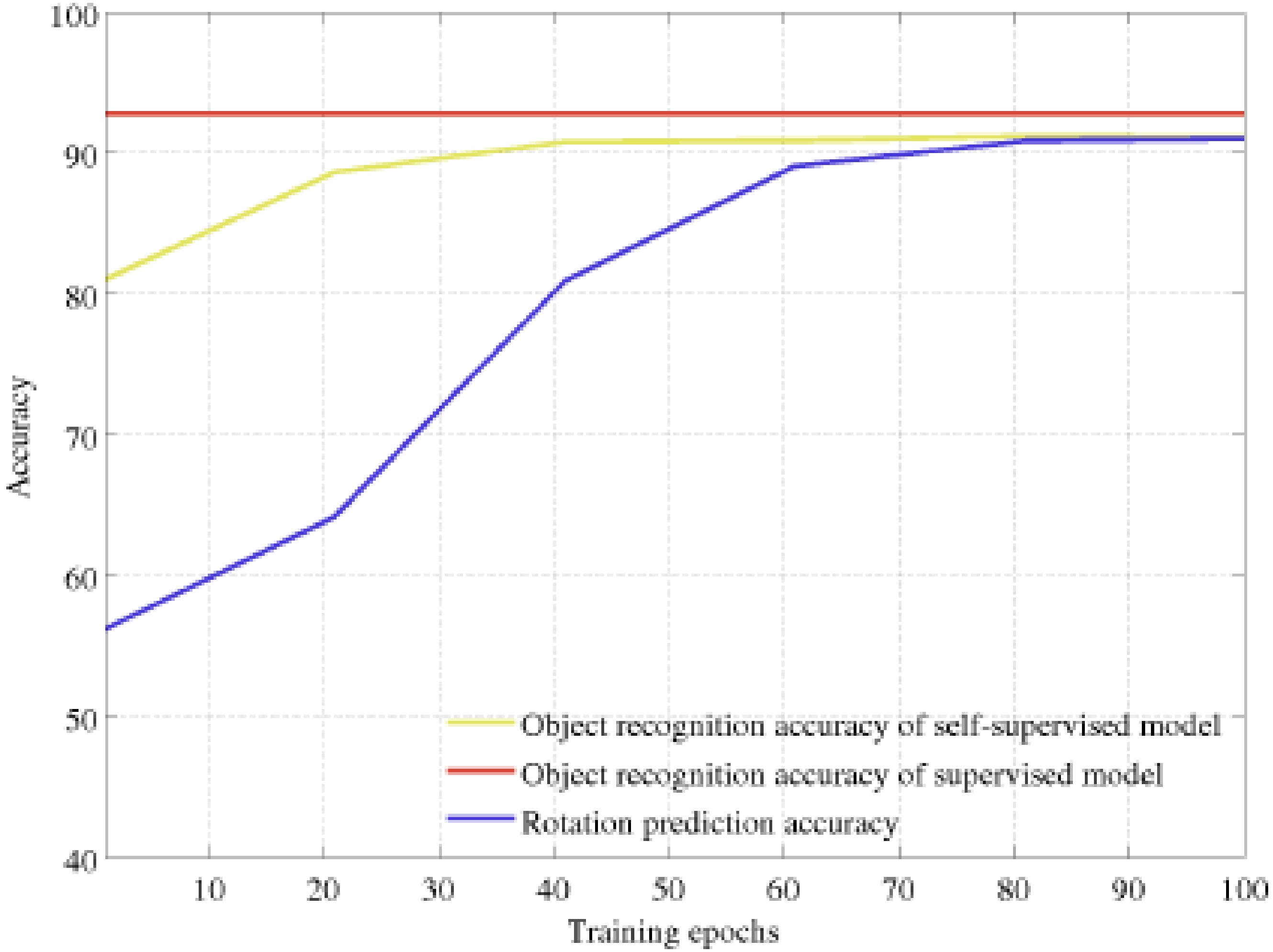
# Rotations	Rotations	CIFAR-10 Classification Accuracy
4	$0^\circ, 90^\circ, 180^\circ, 270^\circ$	**89.06**
8	$0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$	88.51
2	$0^\circ, 180^\circ$	87.46
2	$90^\circ, 270^\circ$	85.52

Table 3: Evaluation of unsupervised feature learning methods on CIFAR-10. The *Supervised NIN* and the *(Ours) RotNet + conv* entries have exactly the same architecture but the first was trained fully supervised while on the second the first 2 conv. blocks were trained unsupervised with our rotation prediction task and the 3rd block only was trained in a supervised manner. In the *Random Init. + conv* entry a conv. classifier (similar to that of *(Ours) RotNet + conv*) is trained on top of two NIN conv. blocks that are randomly initialized and stay frozen. Note that each of the prior approaches has a different ConvNet architecture and thus the comparison with them is just indicative.

Method	Accuracy
Supervised NIN	92.80
Random Init. + conv	72.50
(Ours) RotNet + non-linear	89.06
(Ours) RotNet + conv	**91.16**
(Ours) RotNet + non-linear (fine-tuned)	91.73
(Ours) RotNet + conv (fine-tuned)	92.17
Roto-Scat + SVM Oyallon & Mallat (2015)	82.3
ExemplarCNN Dosovitskiy et al. (2014)	84.3
DCGAN Radford et al. (2015)	82.8
Scattering Oyallon et al. (2017)	84.7

Table 9: **Per class CIFAR-10 classification accuracy.**

Classes	aero	car	bird	cat	deer	dog	frog	horse	ship	truck
Supervised	**93.7**	**96.3**	**89.4**	82.4	**93.6**	**89.7**	**95.0**	**94.3**	**95.7**	**95.2**
(Ours) RotNet	91.7	95.8	87.1	**83.5**	91.5	85.3	94.2	91.9	**95.7**	94.2



(a)

Figure 5: (**a**) Plot with the rotation prediction accuracy and object recognition accuracy as a function of the training epochs used for solving the rotation prediction task. The red curve is the object recognition accuracy of a fully supervised model (a NIN model), which is independent from the training epochs on the rotation prediction task. The yellow curve is the object recognition accuracy of an object classifier trained on top of feature maps learned by a *RotNet* model at different snapshots of the training procedure. (**b**) Accuracy as a function of the number of training examples per category

Visualizing attention maps of rotated images

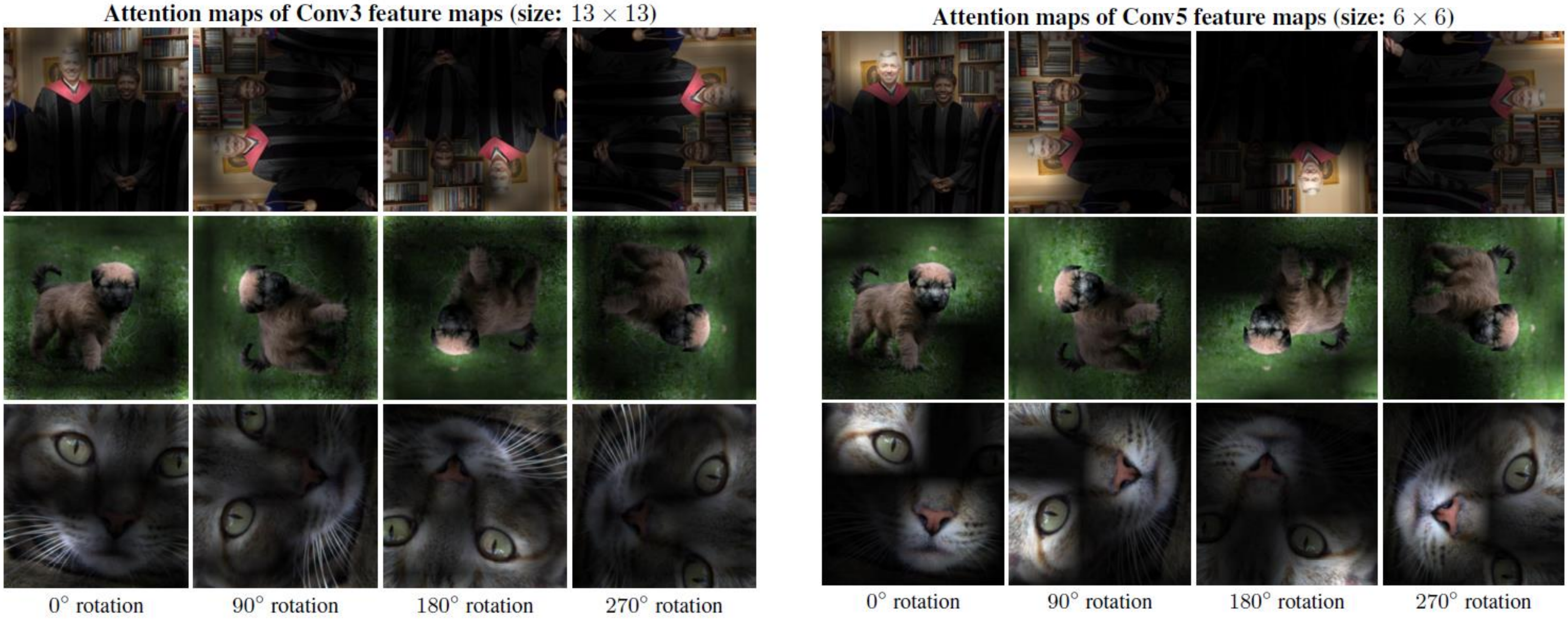


Figure 6: Attention maps of the Conv3 and Conv5 feature maps generated by an AlexNet model trained on the self-supervised task of recognizing image rotations. Here we present the attention maps generated for all the 4 rotated copies of an image.

Thank you !!