# Unsupervised Deep Learning

Mohammad Sabokrou (Researcher at IPM)

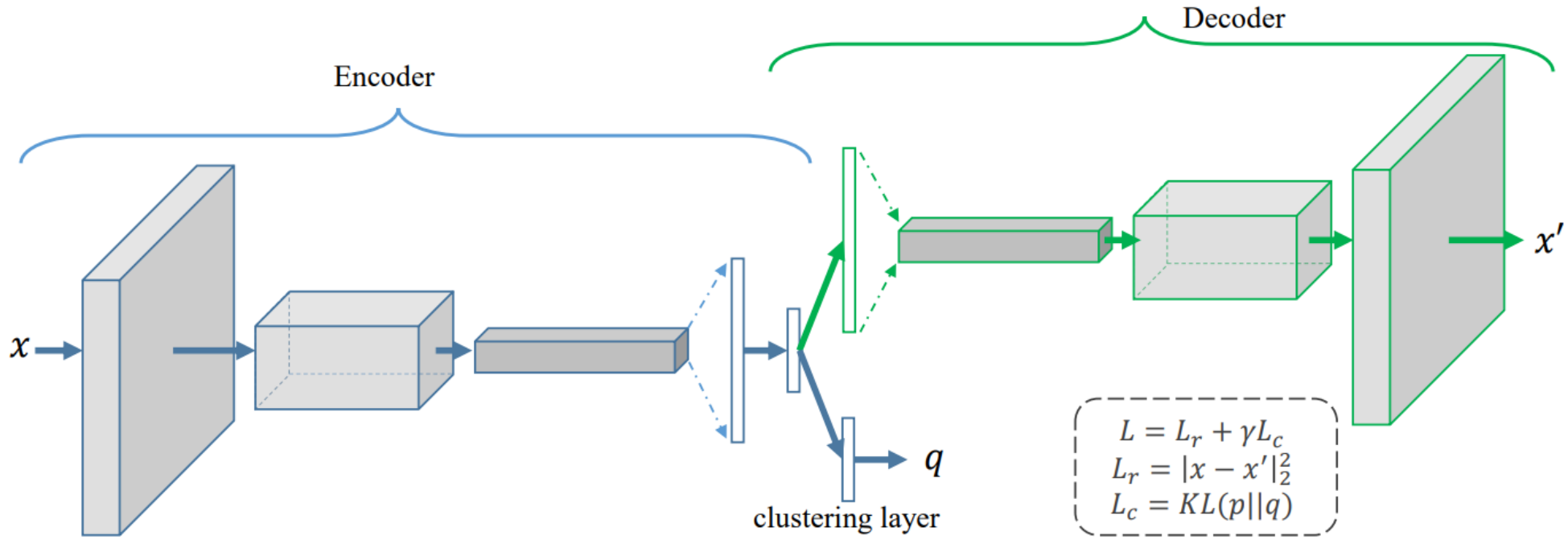
- Deep learning for Unsupervised tasks.

- Unsupervised Learning of Visual Representations.

# Some unsupervised task

- Clustering
- Novelty Detection (One-class calssification)
- Data Generating (Is discussed by Mr. Khalooie)

# Clustering



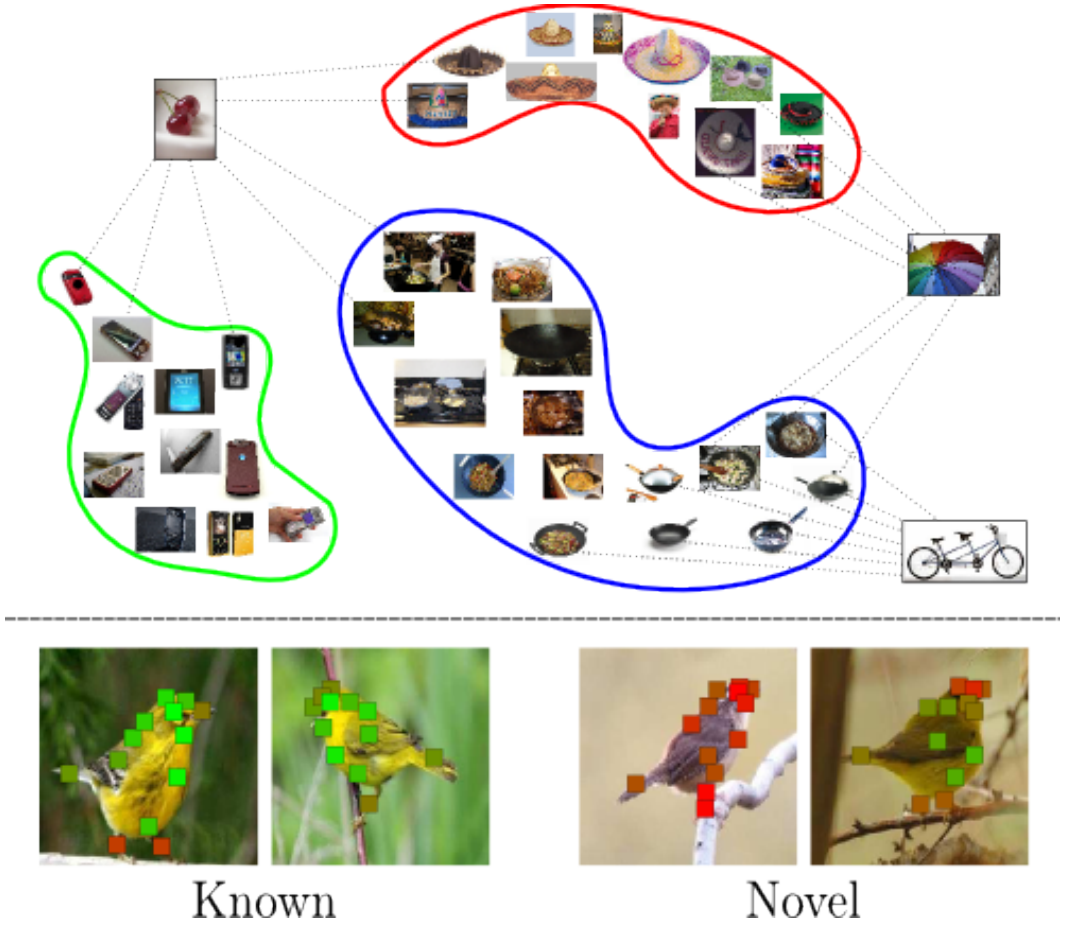

$$L_c = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$L_r = \frac{1}{n} \sum_{i=1}^{n} \|G_{\omega'}(F_\omega(x_i)) - x_i\|_2^2$$

Top 10 scoring elements from one cluster

# One-Class classification

Outlier Detection
Novelty Detection
Rate event Detection
Anomaly Detection
Irregular event(or texture detection)

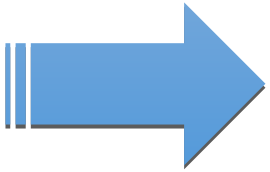


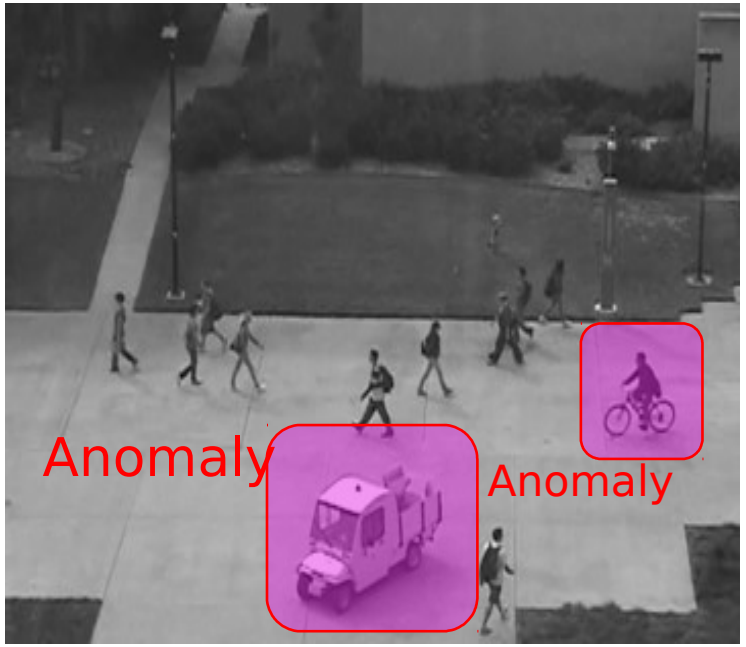

How train a deep network in absence of one class?

# One-Class classification

- Straregy1: Just using deep features (Not as an End-2-End network)

- Straregy2:An End-to-End deep neural network for detection target class.





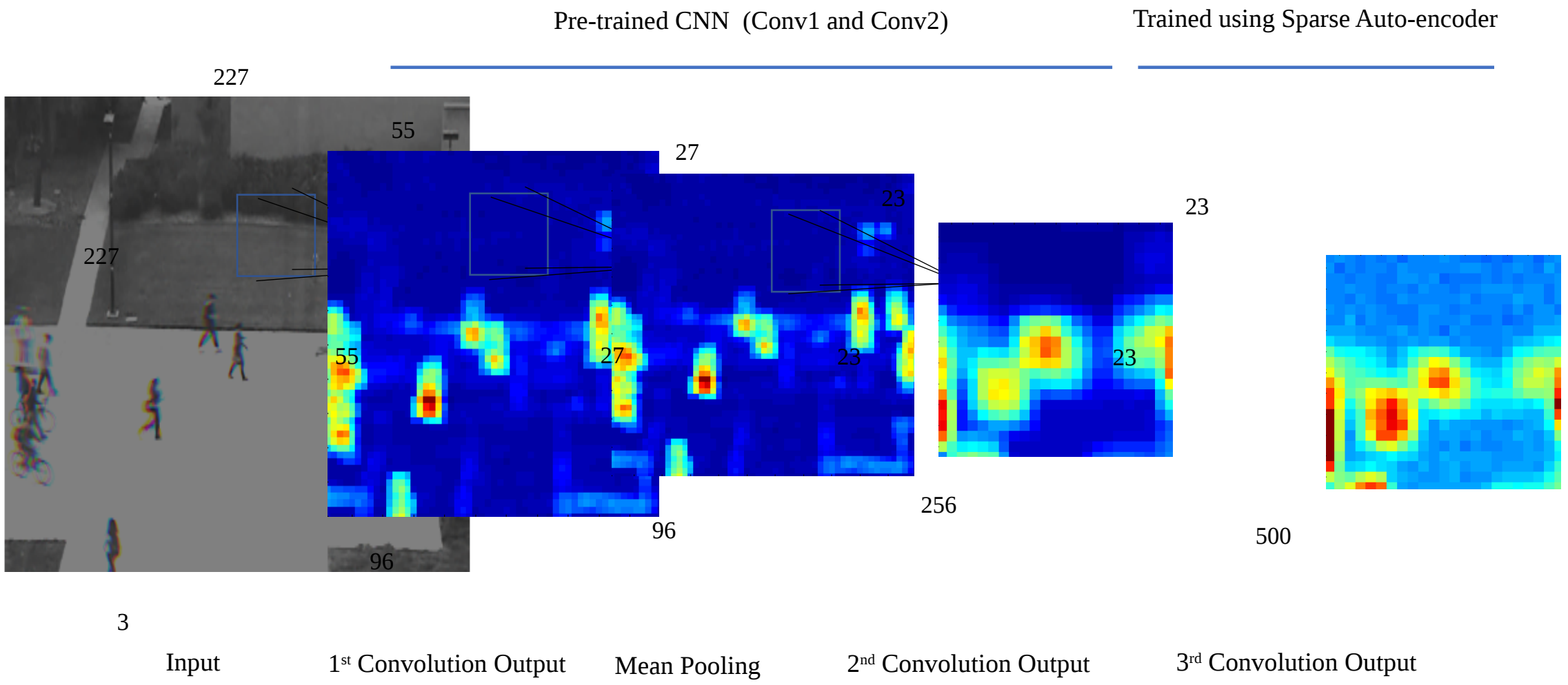

Our goal is to detect and localize anomaly region proposals in video

Now we need a tool for extracting the features of proposals of our video

We have used a pre-trained AlexNet with some changes:

1- We use it up to its 2nd convolution layer

2- We added a new convolutional layer which its kernels are trained using a sparse auto-encoder



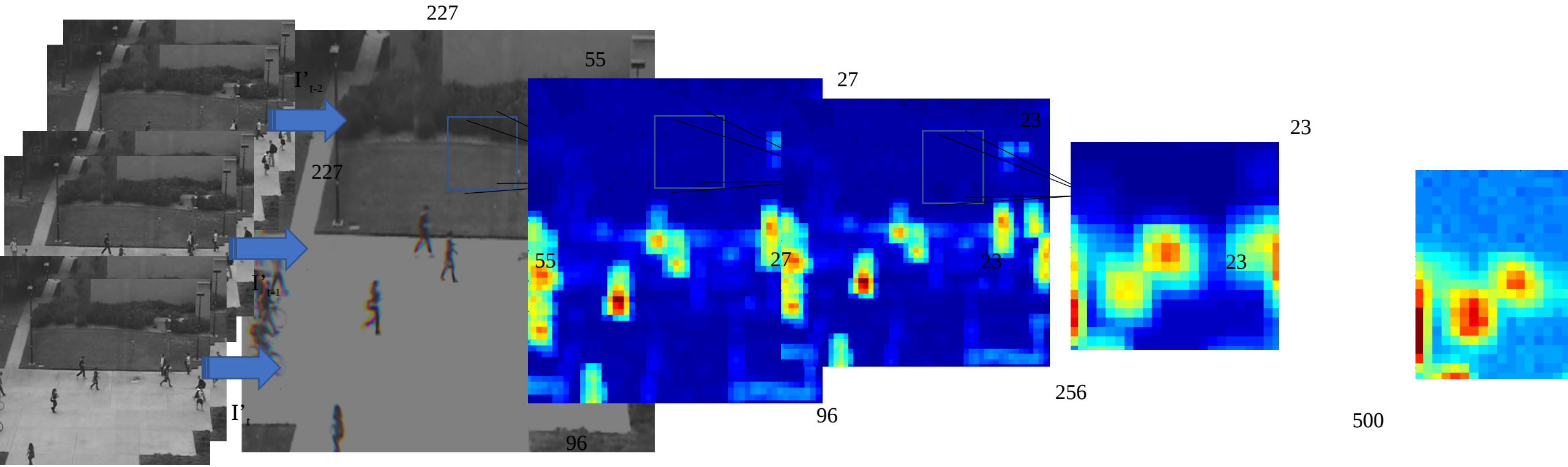
To feed our three mean frames to CNN,

we combine them as a three channel input image and feed it to the CNN

We feed the output of pre-trained AlexNet (2nd conv. Layer) to a sparse auto-encoder to train 3rd conv. layer kernels

Pre-trained Part

Trained Part




Dt Construction

Mean Combined Input Frames

1st Convolution Output

Mean Pooling

2nd Convolution Output

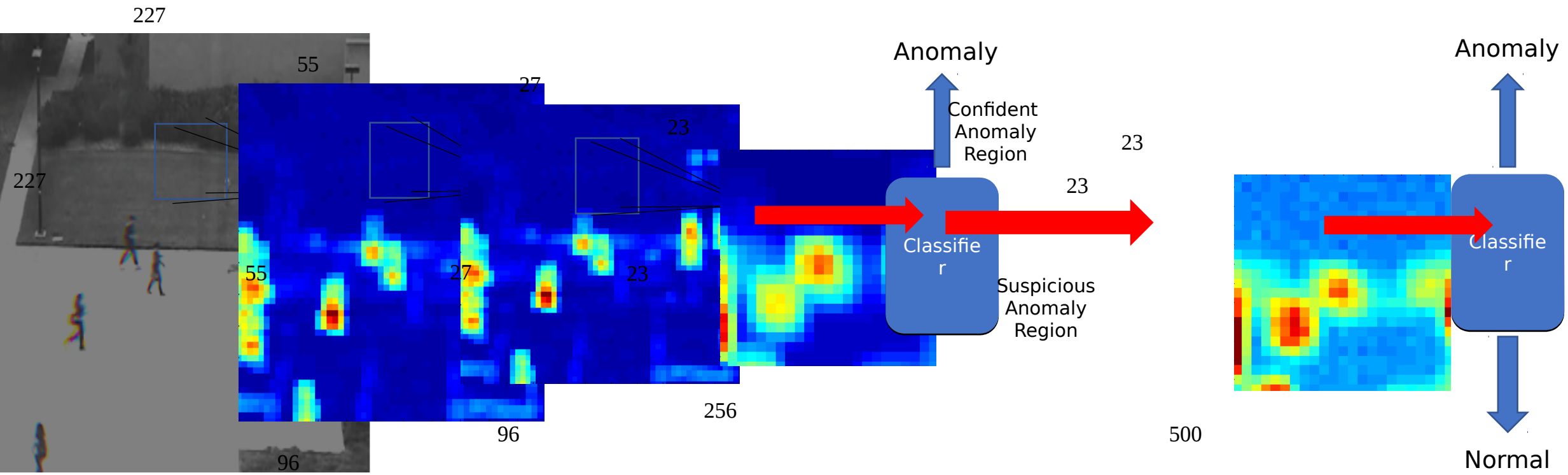
3rd Convolution Output

After training the network (3rd conv.), two Gaussian classifiers are embedded to the network

1st classifier is located between conv. 2 and 3 layers to determine whether a feature vector of conv. 2 output is a confident anomaly or not

2st classifier is located after conv. 3 layer to determine whether a feature vector of conv. 3 output is anomaly or not

Both of classifiers are trained using output of their previous layer (Note that during training only normal frames are seen)

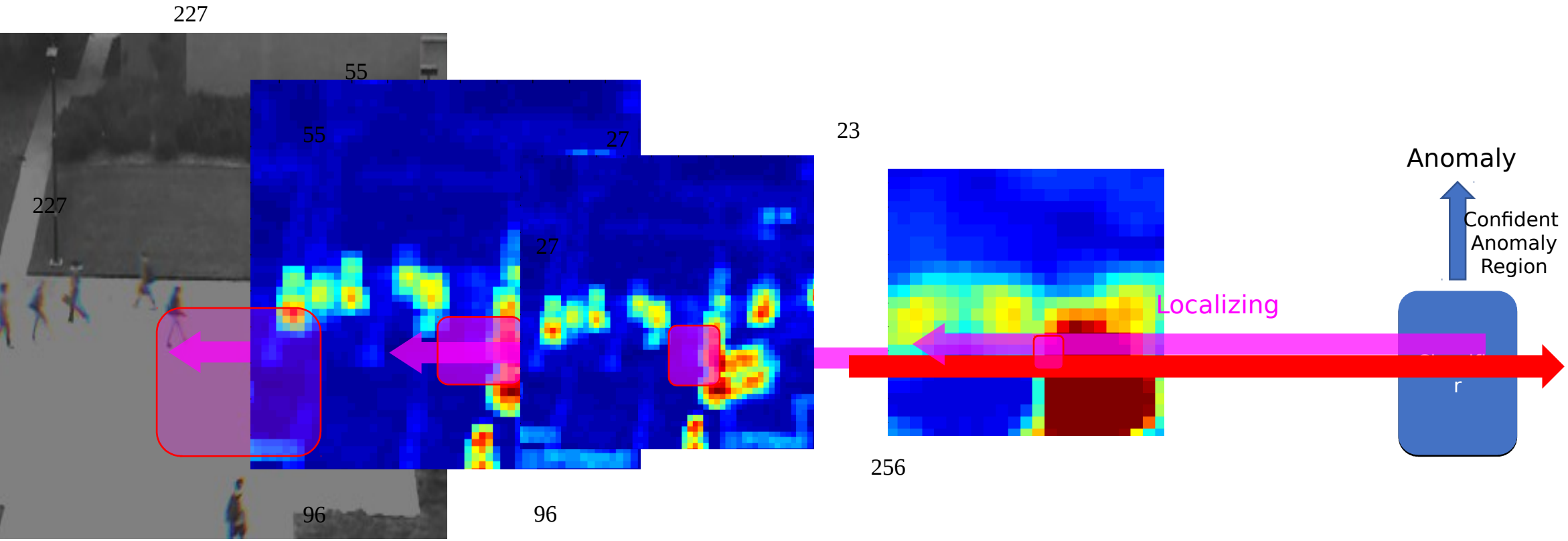


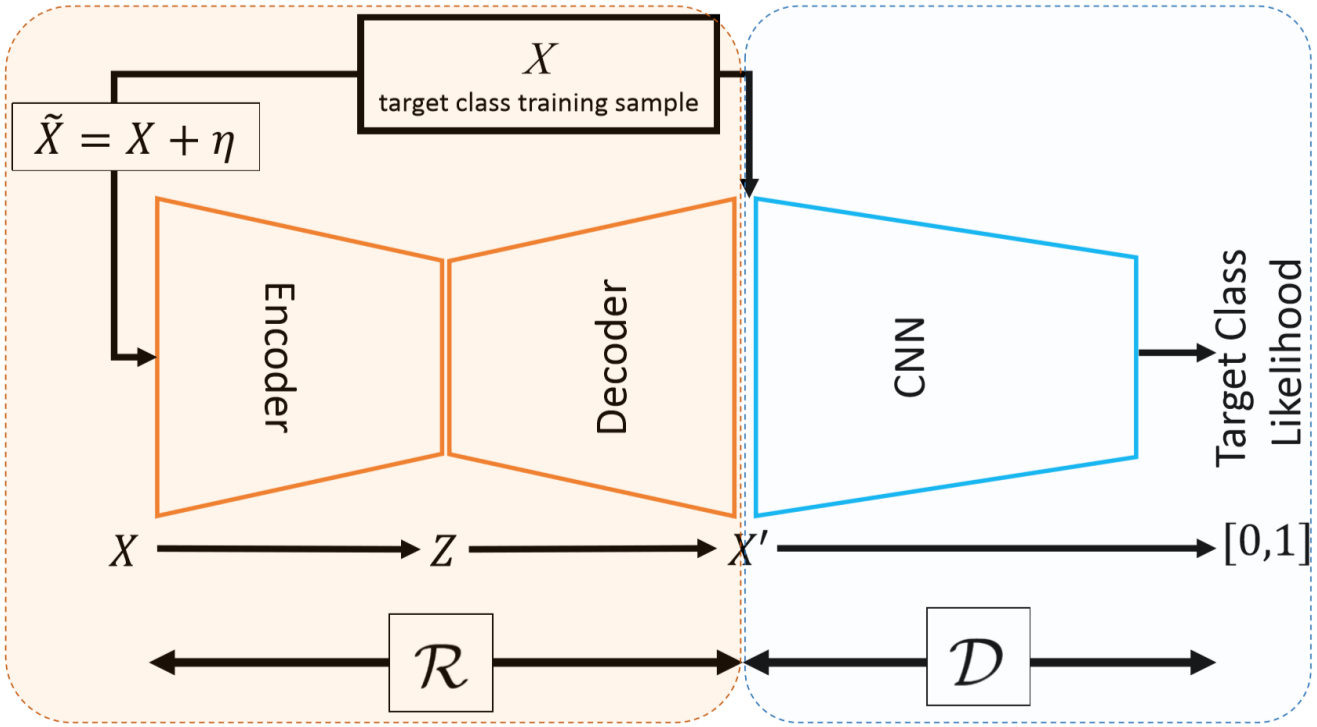

3

Input 1st Convolution Output Mean Pooling 2nd Convolution Output 3rd Convolution Output

When an anomalous feature vector is detected,

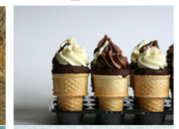
we roll back on CNN to localize anomaly proposal on input frame





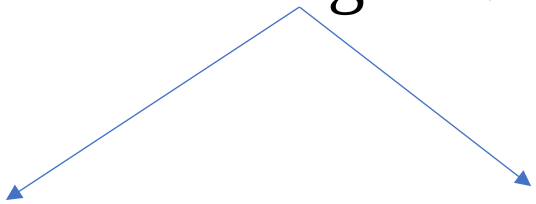

(a)

(b)

| | Noisy Inlier Samples | | Outlier Samples | |
|---|---|---|---|---|
| $X$ |  |  |  |  |
| $\mathcal{R}(X)$ |  |  |  |  |
| $\mathcal{D}(X)$ | 0.75 | 0.72 | 0.53 | 0.27 |
| $\mathcal{D}(\mathcal{R}(X))$ | **0.85** | **0.91** | 0.25 | 0.10 |

# Example of output

# Unsupervised Learning of Visual Representations



Defining a pretext task

modeling the **probability** distribution of given **data**

# Defining a pretext task

- Context Encoders: Feature Learning by Inpainting
- Unsupervised feature learning by augmenting single images
- Learning of Visual Representations using Videos

# Context Encoders: Feature Learning by Inpainting

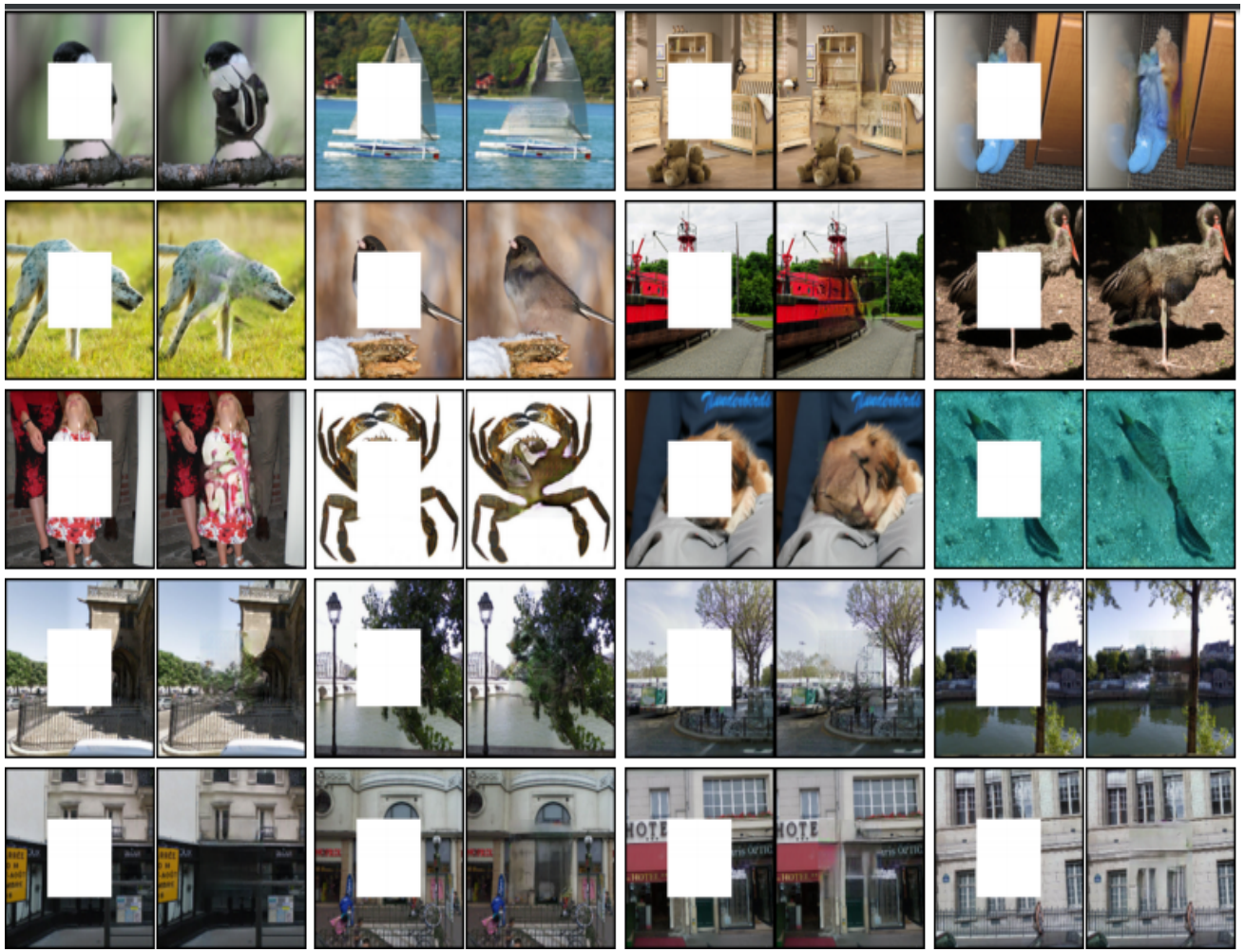
(a) Input context



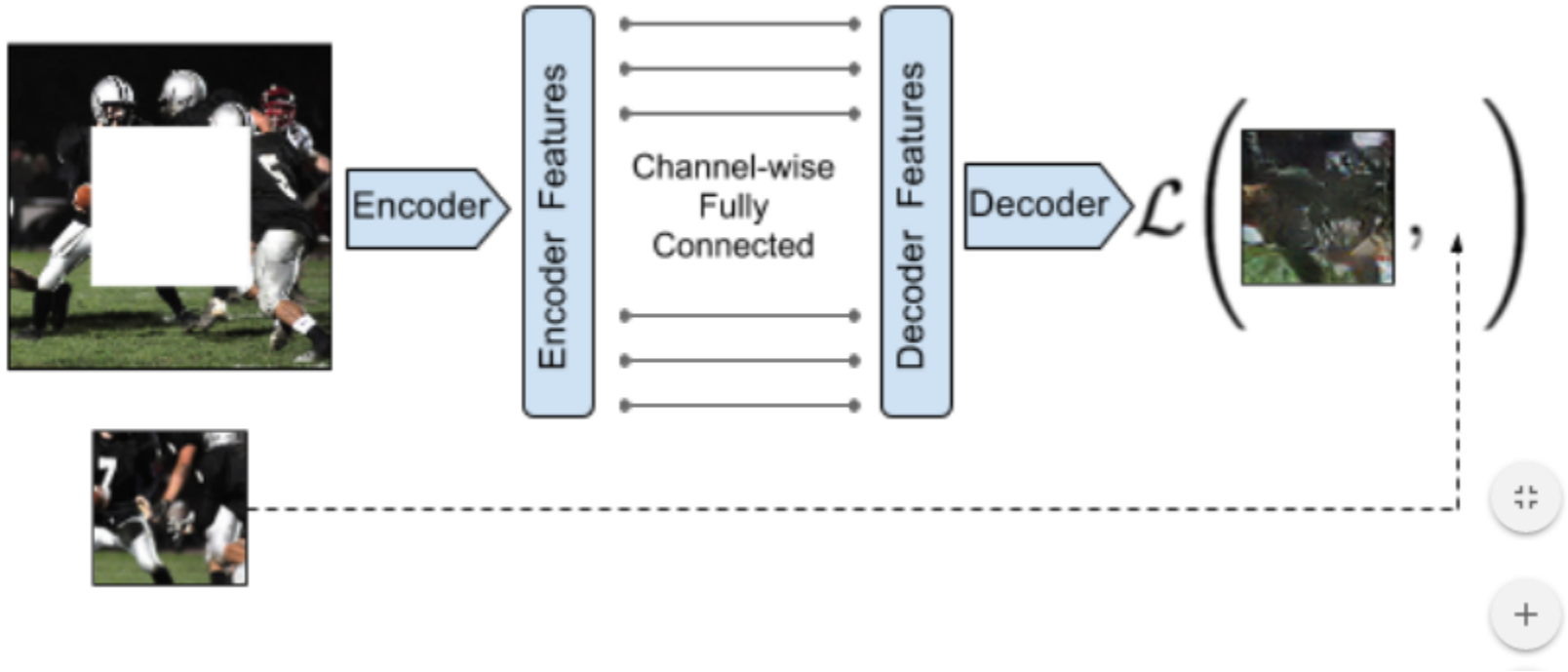
(b) Human artist

(c) Context Encoder ($L2$ loss)

(d) Context Encoder ($L2$ + Adversarial loss)



# Context Encoders: Feature Learning by Inpainting

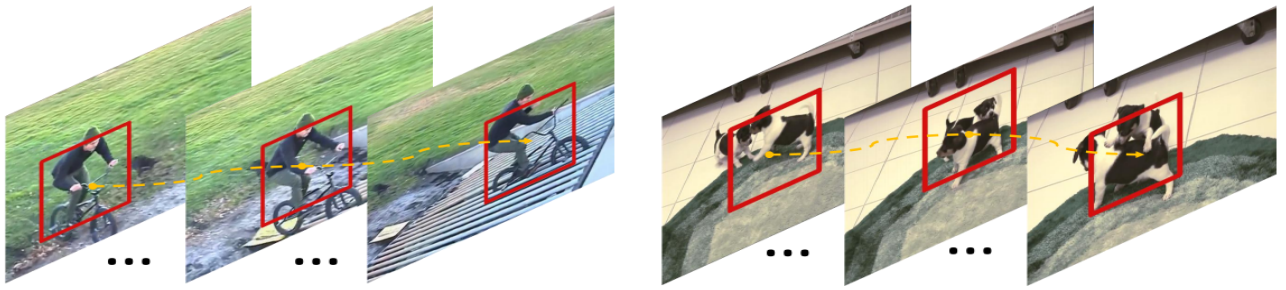


# Unsupervised feature learning by augmenting single images

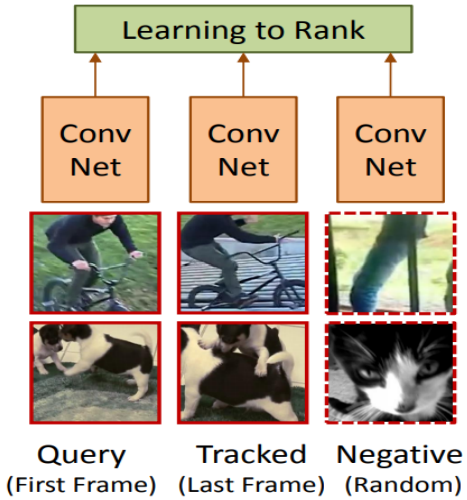
Data augmentation by {Translation,Scale,Contrast,Color} transformations

Training a CNN to classify input data to 4 classes {Translation,Scale,Contrast,Color}

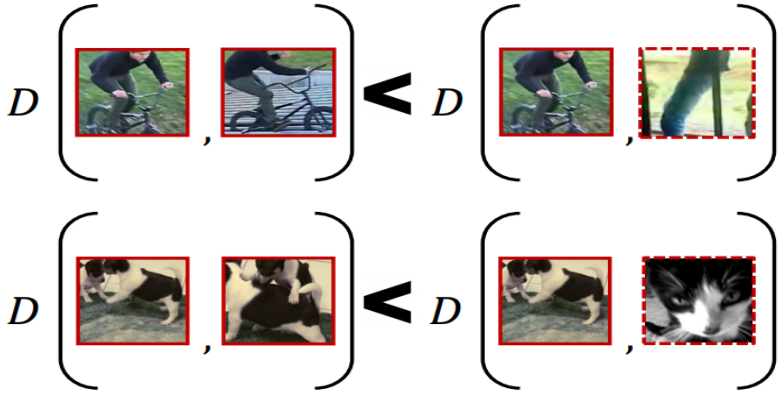
# Unsupervised Learning of Visual Representations using Videos



(a) Unsupervised Tracking in Videos




(b) Siamese-triplet Network

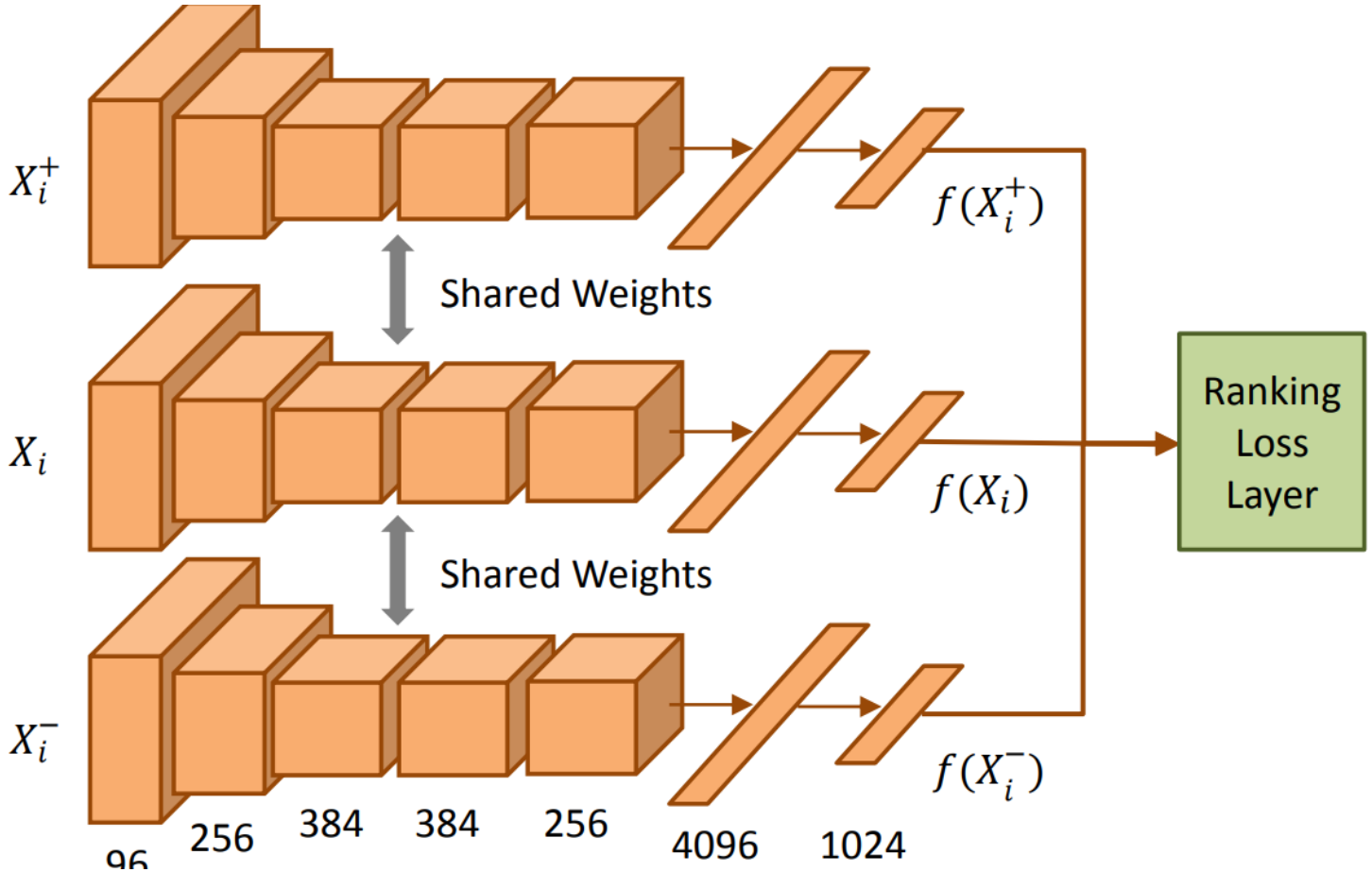



$D$: Distance in deep feature space

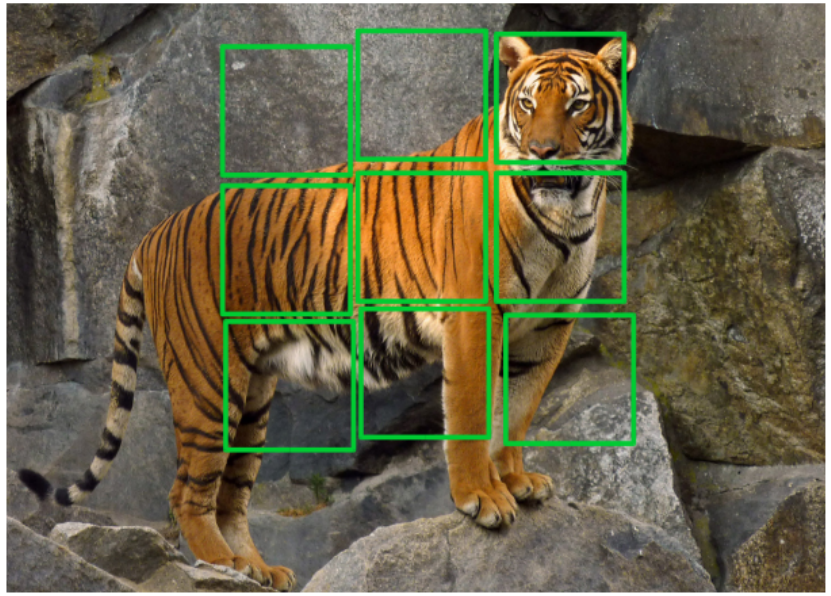
(c) Ranking Objective

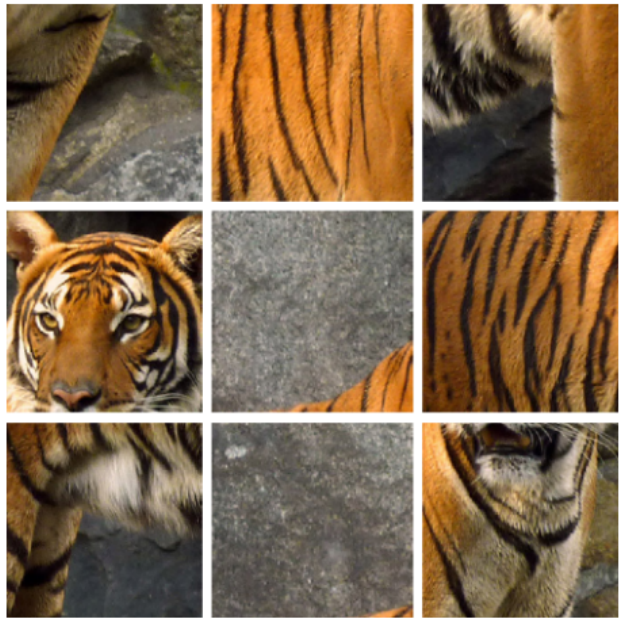
key idea is that visual tracking provides the supervision. That is, two patches connected by a track should have similar visual representation in deep feature space since they probably belong to the same object or object part

# Unsupervised Learning of Visual Representations using Videos

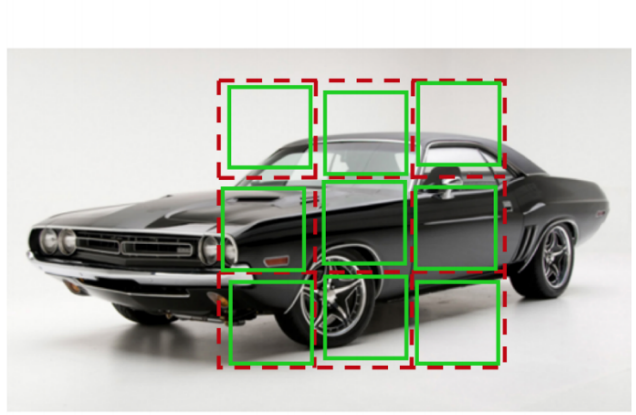


# by Solving Jigsaw Puzzles





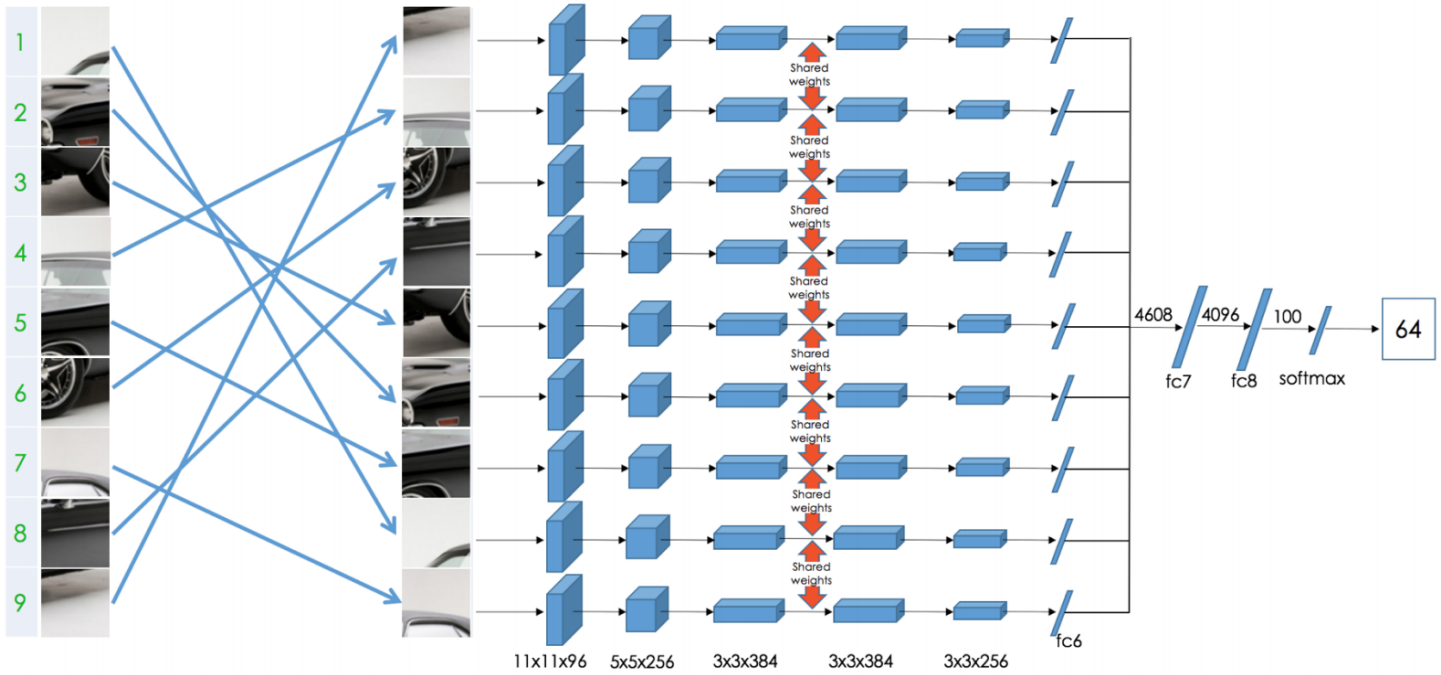
# by Solving Jigsaw Puzzles



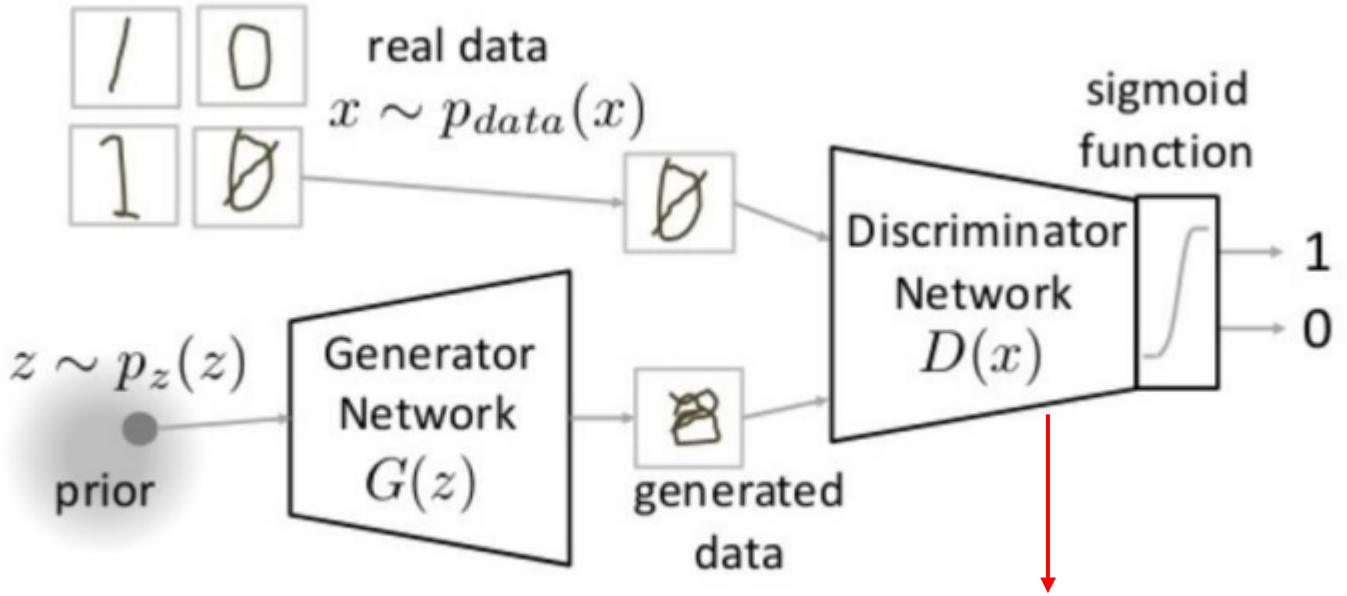
Permutation Set

| index | permutation |
|---|---|
| | |
| 64 | 9,4,6,8,3,2,5,1,7 |
| | |
| | |

Reorder patches according to the selected permutation



# Probability Estimation




**Can be used a pre-trained network?**

# Adversarial Discriminative Domain Adaptation