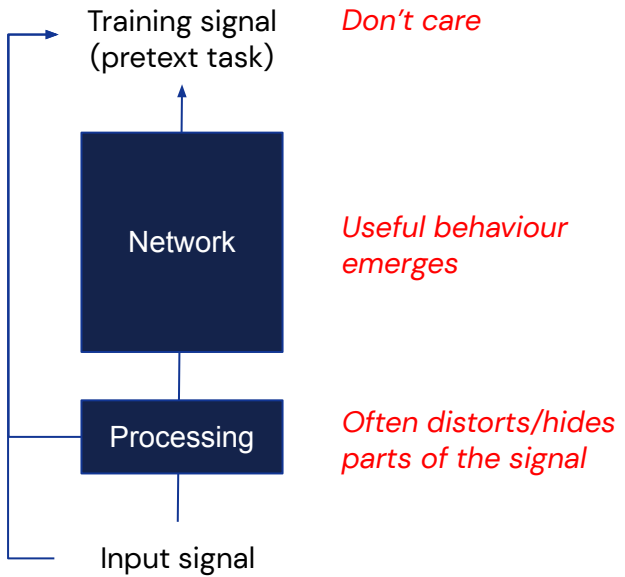
2 A tour of pretext tasks

Relja Arandjelović

Self-supervised learning in a nutshell

Self-supervised learning

- **Goal:** Learn good representations
- **Means:** Construct a pretext task
  - Don't care about the pretext task itself
  - Only important it enables learning

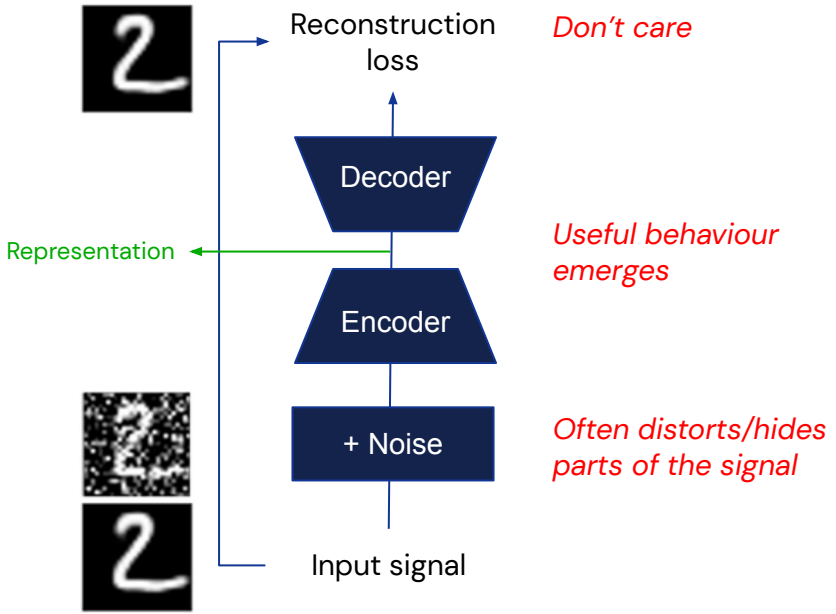


["Extracting and composing robust features with denoising autoencoders", Vincent et al. 08]

Self-supervised learning in a nutshell

Self-supervised learning

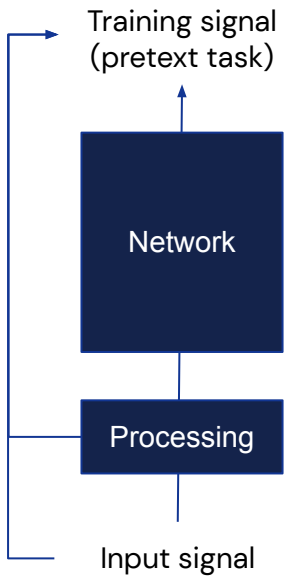
- **Goal:** Learn good representations
- **Means:** Construct a pretext task
  - Don't care about the pretext task itself
  - Only important it enables learning



A tour of pretext tasks

Self-supervised learning

- **Goal:** Learn good representations
- **Means:** Construct a pretext task
  - Don't care about the pretext task itself
  - Only important it enables learning



Don't care

Useful behaviour emerges

Often distorts/hides parts of the signal

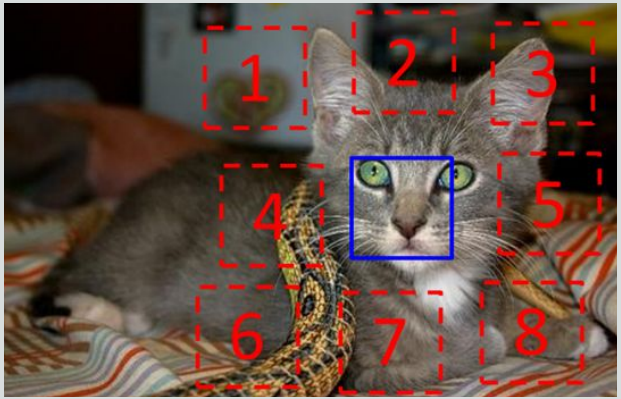
Rough pretext task classification

- Inferring structure
- Transformation prediction
- Reconstruction
- Exploiting time
- Multimodal
- Instance classification

Disclaimer

- Rough classification of tasks, some fit multiple categories
- Trying to cover many but inevitably missing many works
- Often have to pick one of multiple concurrent similar methods
- If A comes before B in this presentation, it doesn't mean A did it first

Inferring structure



Context prediction

Can you guess the spatial configuration for the two pairs of patches?

Question 1:

Question 2:

Context prediction

Can you guess the spatial configuration for the two pairs of patches? Much easier if you recognize the object!

Question 1:

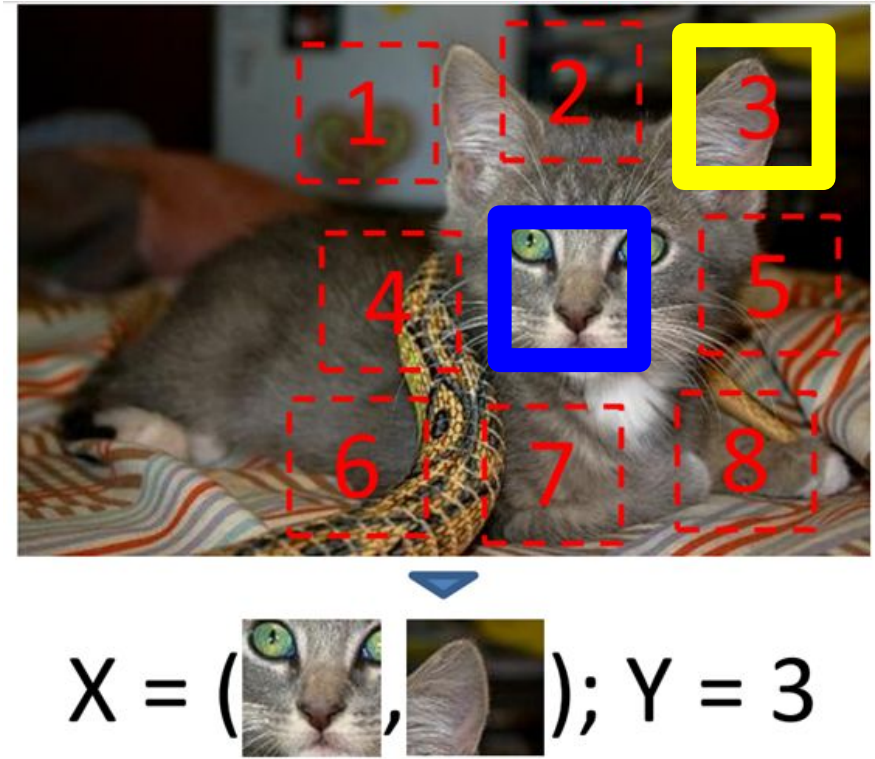
Question 2:

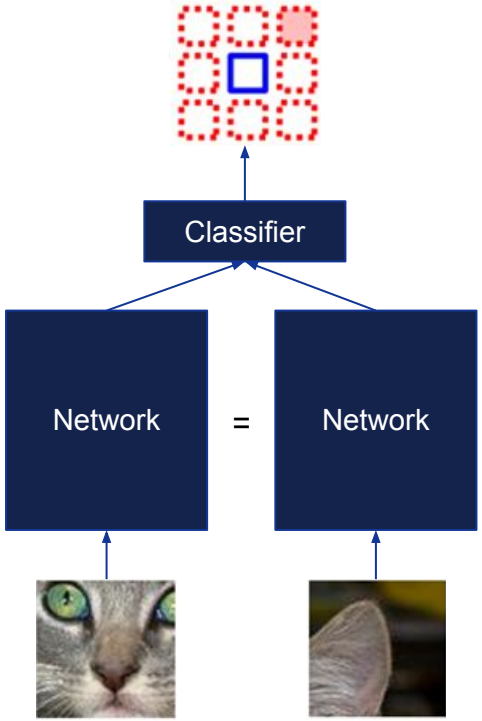
Intuition

- The network should learn to recognize object parts and their spatial relations

[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]

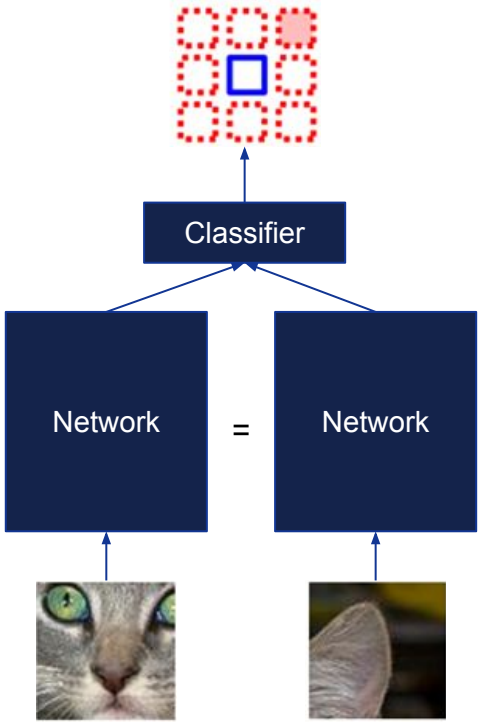
Context prediction





[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]

Context prediction



Pros

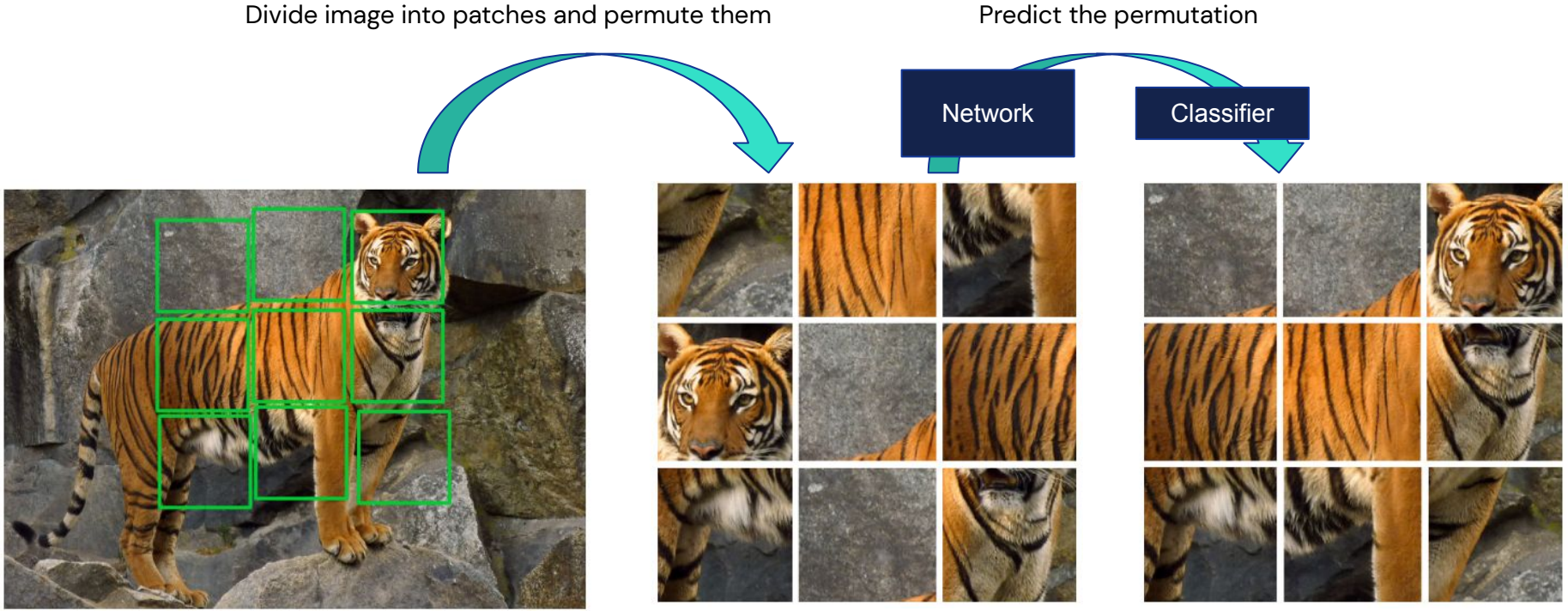
- (arguably) The first self-supervised method
- Intuitive task that should enable learning about object parts

Cons

- Assumes training images are photographed with canonical orientations (and canonical orientations exist)
- Training on patches, but trying to learn image representations
- Networks can “cheat” so special care is needed [discussed later]
  - Further gap between train and eval
- Not fine-grained enough due to no negatives from other images
  - e.g. no reason to distinguish cat from dog eyes
- Small output space - 8 cases (positions) to distinguish?

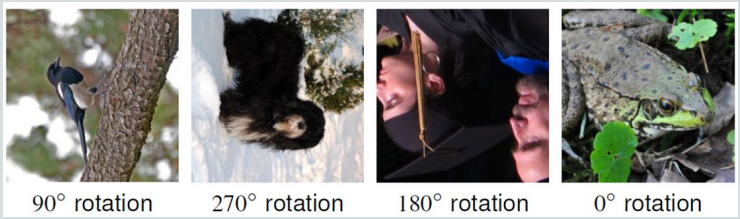
[“Unsupervised learning of visual representations by solving jigsaw puzzles”, Noroozi et al. 17]

Jigsaw puzzles



Pros & Cons: Same as for context prediction apart from being harder

Transformation prediction

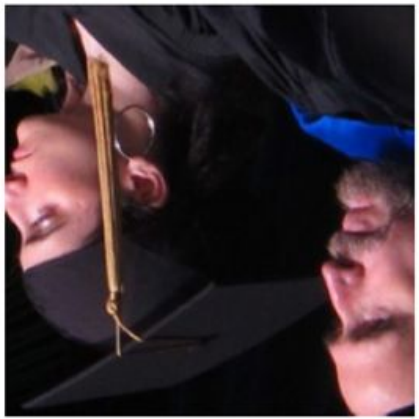


["Unsupervised representation learning by predicting image rotations", Gidaris et al. 18]

Rotation prediction

Can you guess how much rotated is applied?



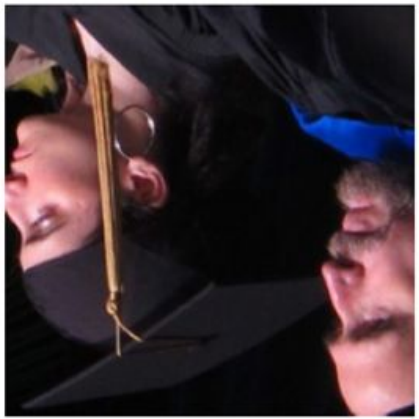



Rotation prediction

Can you guess how much rotated is applied? Much easier if you recognize the content!

90° rotation

270° rotation

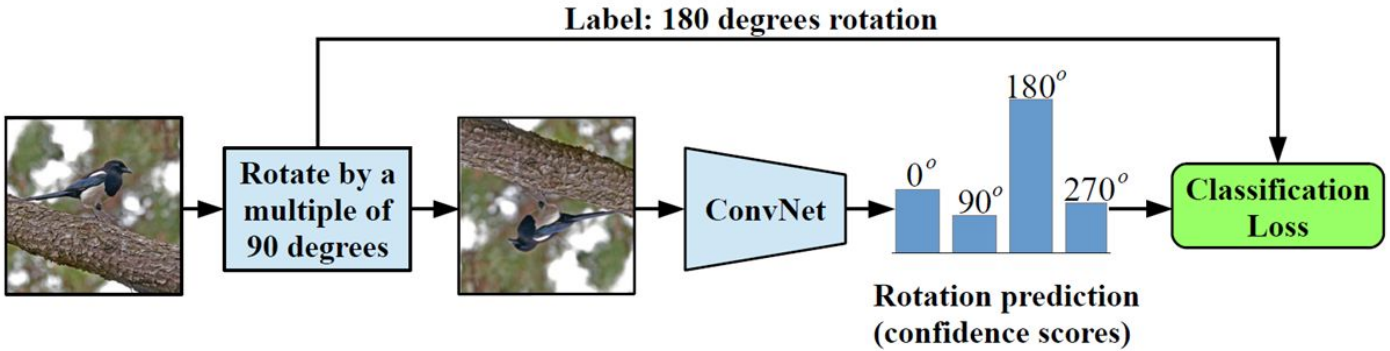


180° rotation

0° rotation

[“Unsupervised representation learning by predicting image rotations”, Gidaris et al. 18]

Rotation prediction



Pros

- Very simple to implement and use, while being quite effective

Cons

- Assumes training images are photographed with canonical orientations (and canonical orientations exist)
- Train-eval gap: no rotated images at eval
- Not fine-grained enough due to no negatives from other images
  - e.g. no reason to distinguish cat from dog
- Small output space - 4 cases (rotations) to distinguish [not trivial to increase; see later]
- Some domains are trivial e.g. StreetView ⇒ just recognize sky

[“AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data”, Zhang et al. 19]

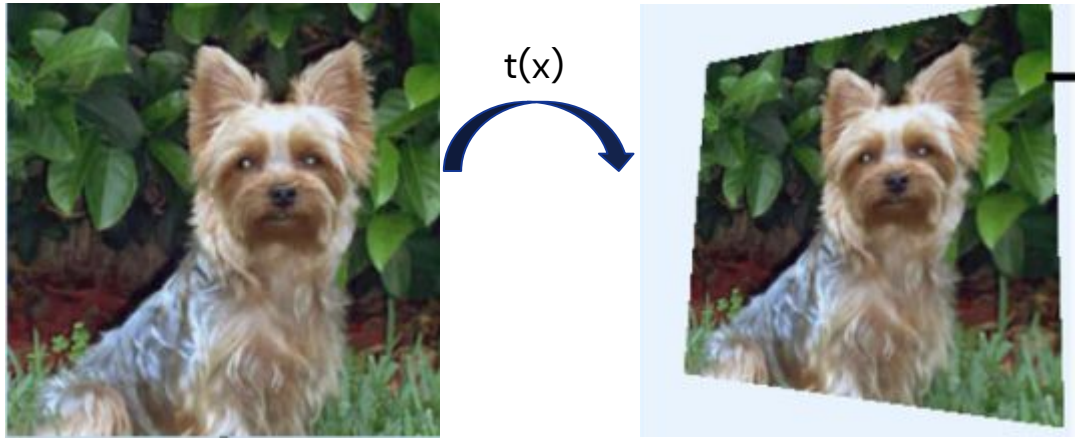
Relative transformation prediction

Estimate the transformation between two images.

[“AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data”, Zhang et al. 19]

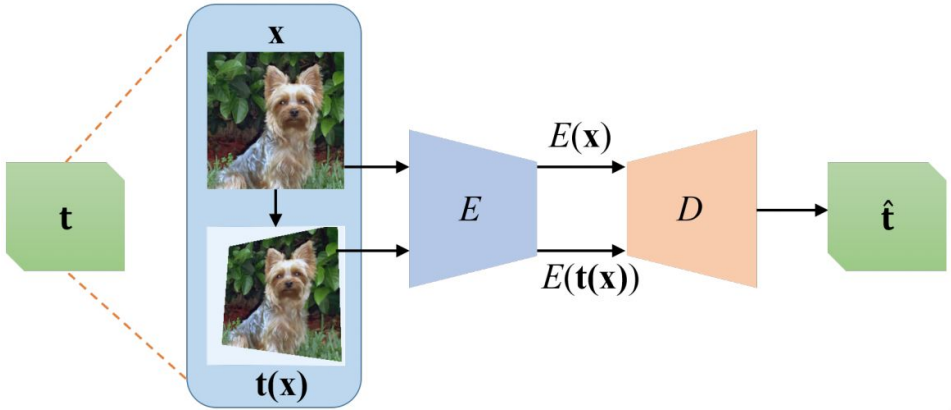
Relative transformation prediction

Estimate the transformation between two images. Requires good features



["AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data", Zhang et al. 19]

Relative transformation prediction



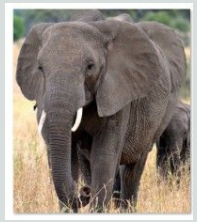
Pros

- In line with classical computer vision, e.g. SIFT was developed for matching

Cons

- Train-eval gap: no transformed images at eval
- Not fine-grained enough due to no negatives from other images
  - e.g. no reason to distinguish cat from dog
- Questionable importance of semantics vs low-level features (assuming we want semantics)
  - Features are potentially not invariant to transformations

Reconstruction

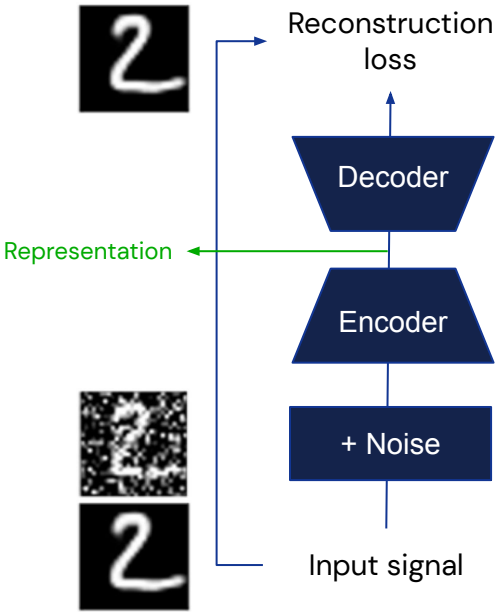


[“Extracting and composing robust features with denoising autoencoders”, Vincent et al. 08]

Denoising autoencoders

What is the noise and what is the signal?

Recognizing the digit helps!



Pros

- Simple classical method
- Apart from representations, we get a denoiser for free

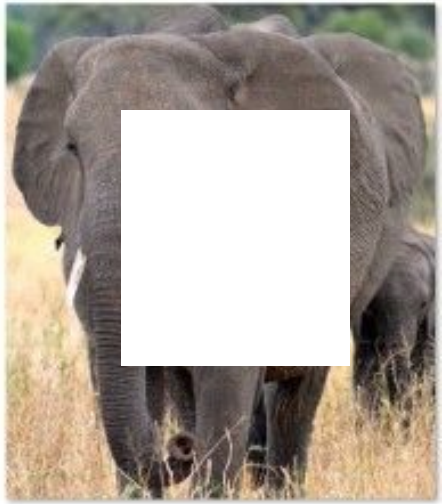
Cons

- Train-eval gap: training on noisy data
- Too easy, no need for semantics - low level cues are sufficient

[“Context encoders: Feature learning by inpainting”, Pathak et al. 16]

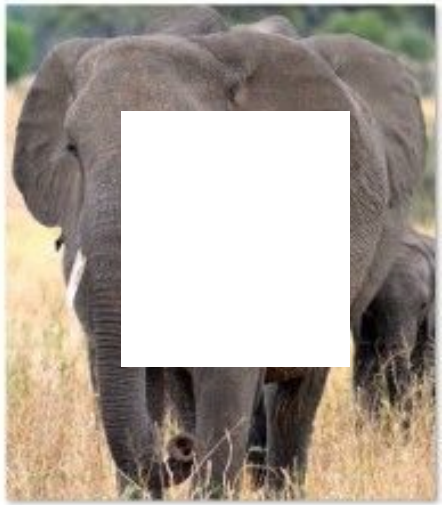
Context encoders

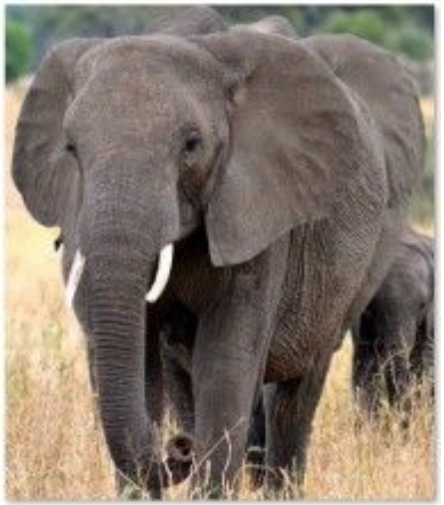
What goes in the middle?



Context encoders

What goes in the middle? Much easier if you recognize the objects!





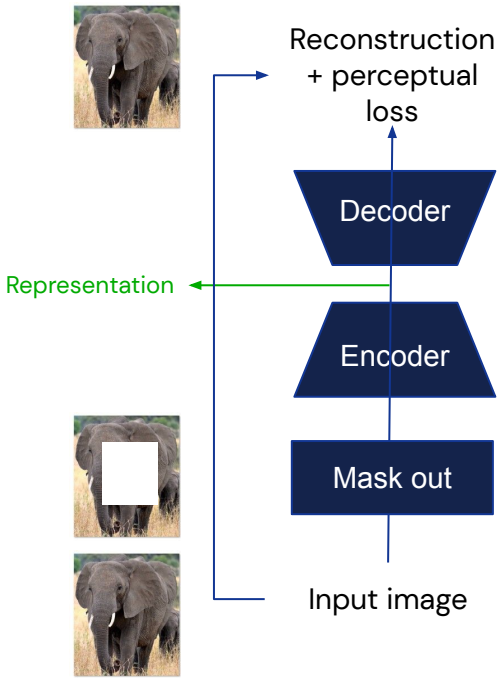
Natural language processing (e.g. word2vec, BERT)

`All [MASK] have tusks. ⇒ All elephants have tusks.`

["Distributed representations of words and phrases and their compositionality", Mikolov et al. 13]
["BERT: Pre-training of deep bidirectional transformers for language understanding", Devlin et al. 18]

[“Context encoders: Feature learning by inpainting”, Pathak et al. 16]

Context encoders



Pros

- Requires preservation of fine-grained information

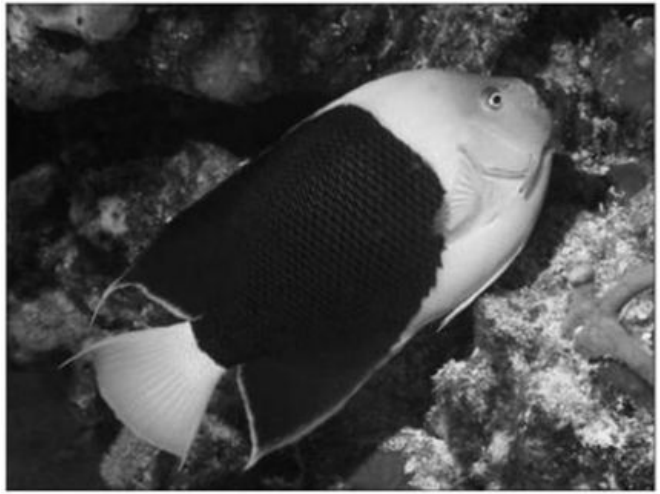
Cons

- Train-eval gap: no masking at eval
- Reconstruction is too hard and ambiguous
- Lots of effort spent on “useless” details: exact colour, good boundary, etc

[“Colorful image colorization”, Zhang et al. 16]

Colorization

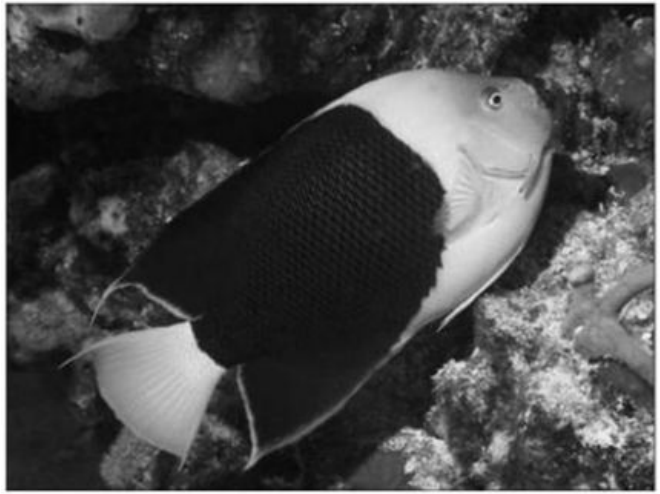
What is the colour of every pixel?



[“Colorful image colorization”, Zhang et al. 16]

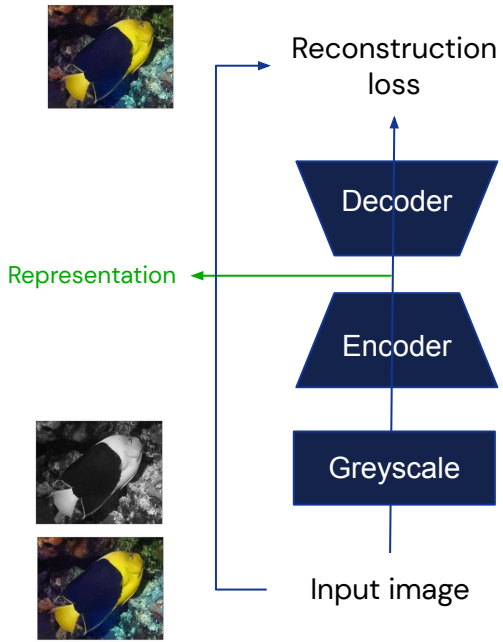
Colorization

What is the colour of every pixel? Hard if you don’t recognize the object!



[“Colorful image colorization”, Zhang et al. 16]

Context encoders



Pros

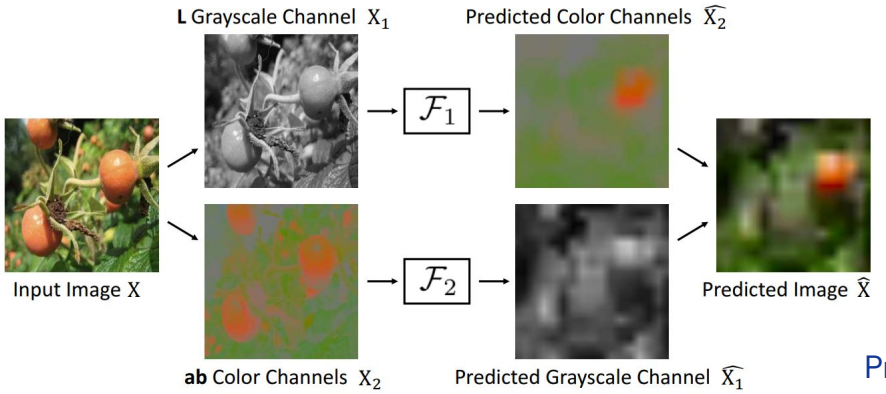
- Requires preservation of fine-grained information

Cons

- Reconstruction is too hard and ambiguous
- Lots of effort spent on “useless” details: exact colour, good boundary, etc
- Forced to evaluate on greyscale images, losing information

["Split-brain autoencoders: Unsupervised learning by cross-channel prediction", Zhang et al. 17]

Context encoders ⇒ Split-brain encoders



Pros

- Requires preservation of fine-grained information

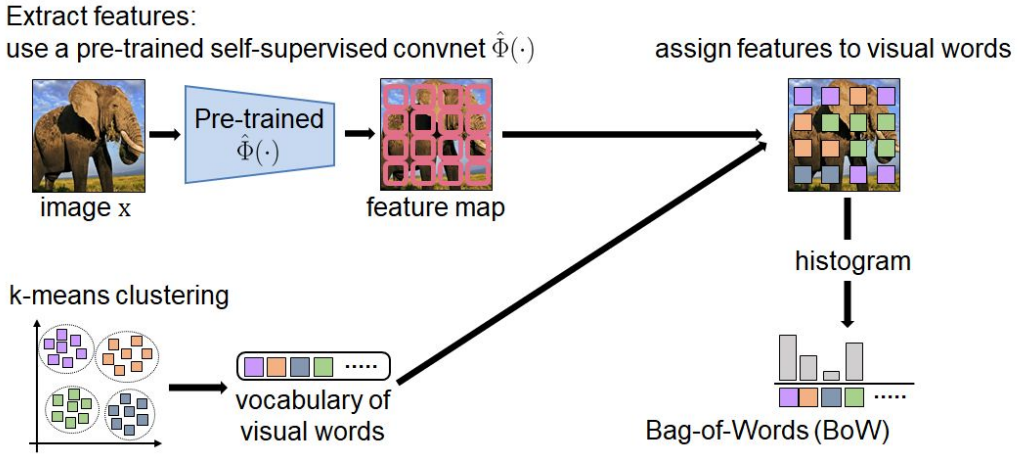
Cons

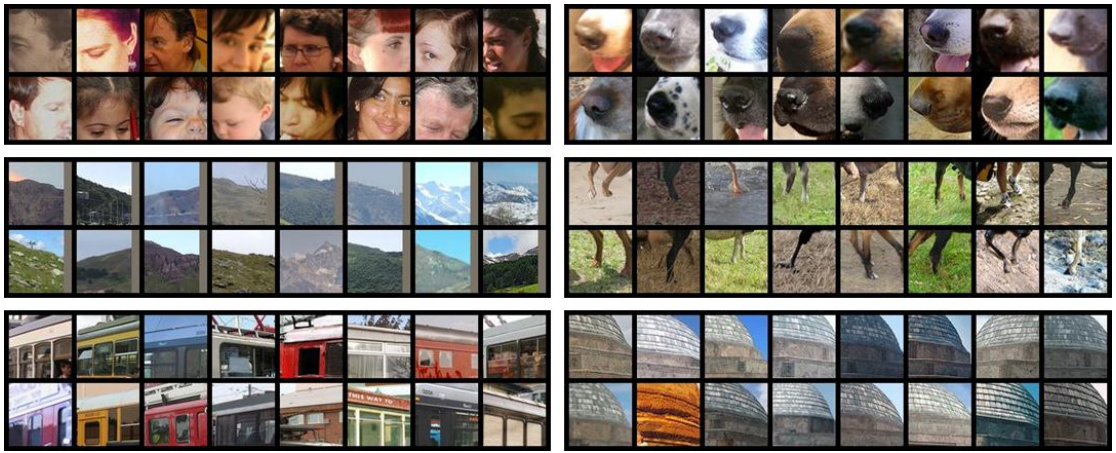
- Reconstruction is too hard and ambiguous
- Lots of effort spent on "useless" details: exact colour, good boundary, etc
- ~~Forced to evaluate on greyscale images, losing information~~
- Processes different chunks of the input independently

["Learning representations by predicting bags of visual words", Gidaris et al. 20]

Predicting bag-of-words

Bag-of-words reminder

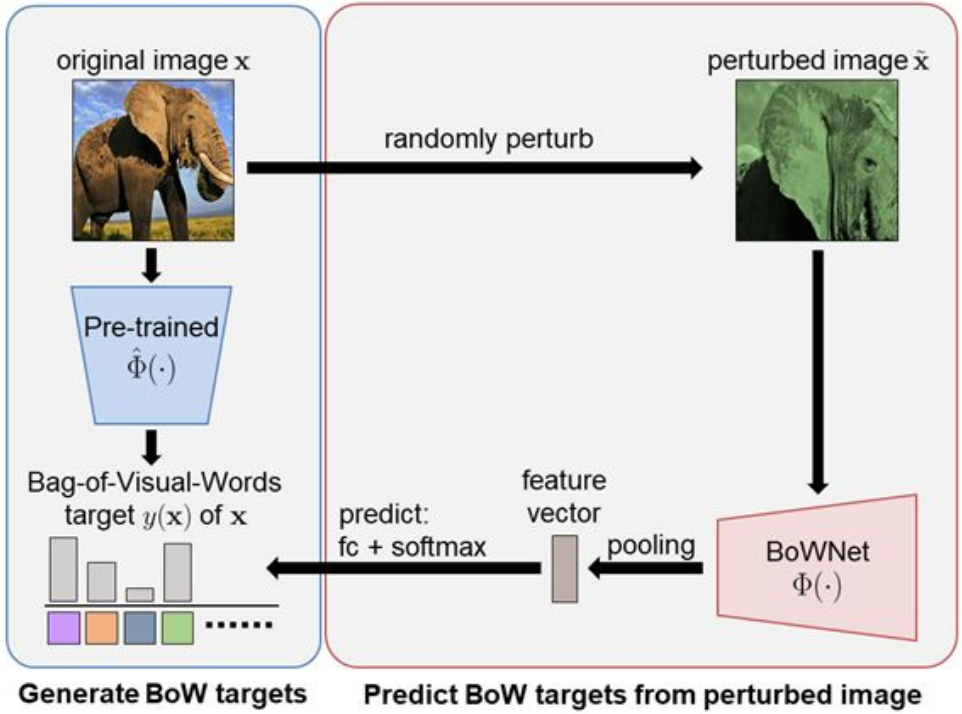




- Loses low-level details
- Encodes mid/high-level concepts

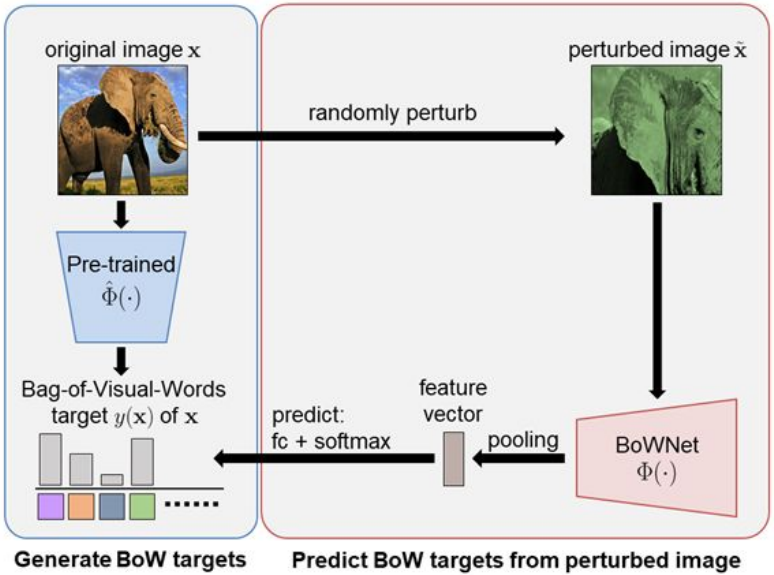
["Learning representations by predicting bags of visual words", Gidaris et al. 20]

Predicting bag-of-words



Inspired by NLP: targets = discrete concepts (words)

Predicting bag-of-words



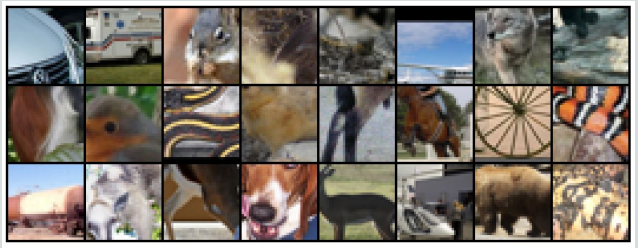
Pros

- Representations are invariant to desired transformations
- Learn contextual reasoning skills
  - Infer words of missing image regions

Cons

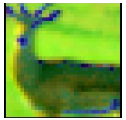
- Requires bootstrapping from another network
  - e.g. hard to learn more fine-grained features
- Pitfalls of BoW
  - (partial) loss of spatial information
  - SpatialBoW not improving

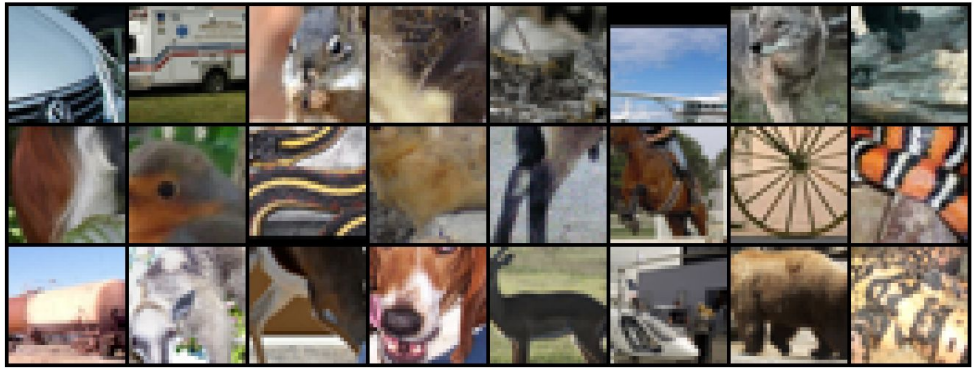
Instance classification



[“Discriminative unsupervised feature learning with exemplar convolutional neural networks”, Dosovitskiy et al. 14]

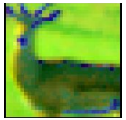
Exemplar ConvNets

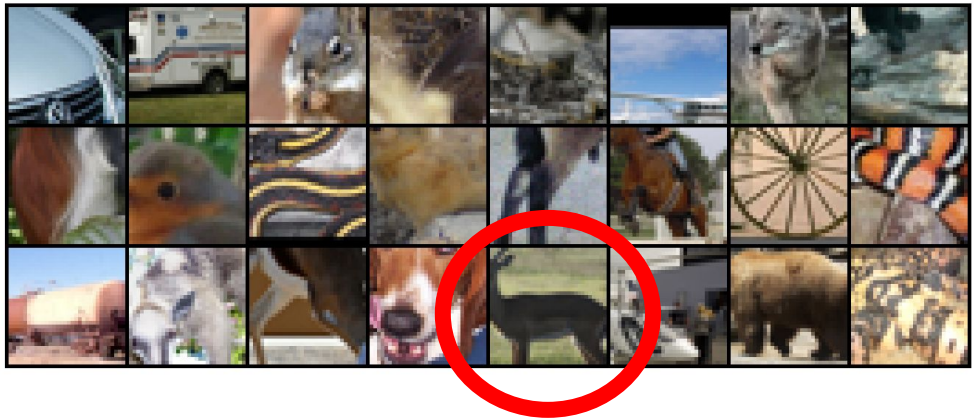
This  is a distorted crop extracted from an image, which of these crops has the same source image?



[“Discriminative unsupervised feature learning with exemplar convolutional neural networks”, Dosovitskiy et al. 14]

Exemplar ConvNets

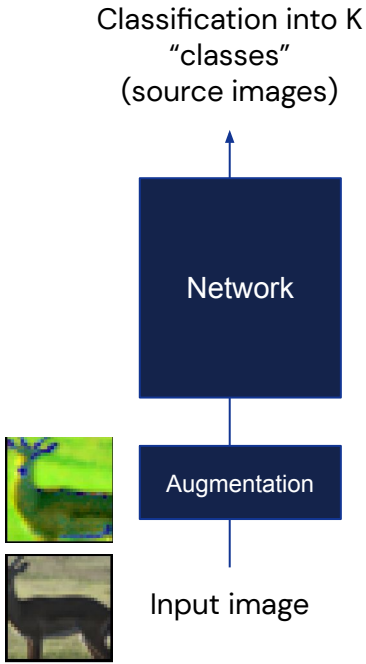
This  is a distorted crop extracted from an image, which of these crops has the same source image?



Easy if robust to the desired transformations (geometry and colour)

[“Discriminative unsupervised feature learning with exemplar convolutional neural networks”, Dosovitskiy et al. 14]

Exemplar ConvNets



Pros

- Representations are invariant to desired transformations
- Requires preservation of fine-grained information

Cons

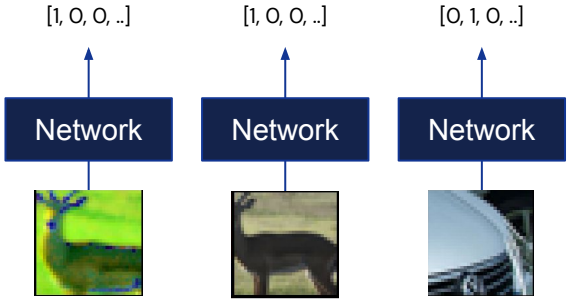
- Choosing the augmentations is important
- Exemplar based: images of the same class or instance are negatives
  - Nothing prevents it from focusing on the background
- Original formulation is not scalable (number of “classes” = dataset size)

Exemplar ConvNets via metric learning

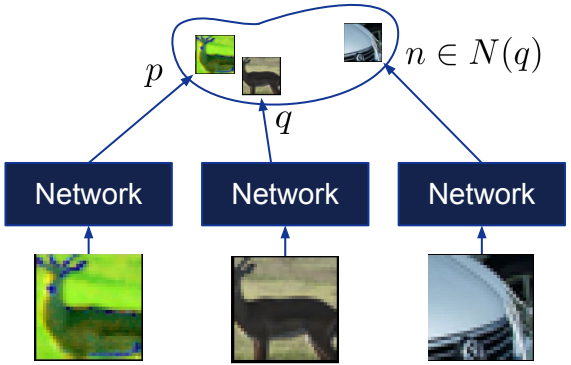
Exemplar ConvNets are not scalable (number of "classes" = number of training images)

- Reformulate in terms of metric learning
- Traditional losses such as contrastive or triplet ["Multi-task self-supervised visual learning", Doersch and Zisserman 17], ["HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips", Miech et al. 19]
- Recently popular: InfoNCE ["Representation Learning with Contrastive Predictive Coding", van den Oord et al. 18]
  - Used by many recent methods: CPC, AMDIM, SimCLR, MoCo, ..

Classification



Metric learning



Noise Contrastive Estimation

InfoNCE loss (a specific popular version)

- For **q**uery, **p**ositive and **n**egative:

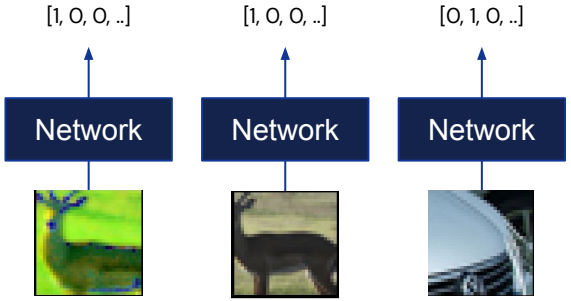
$$-\log \frac{\exp(q^T p)}{\exp(q^T p) + \sum_{n \in N(q)} \exp(q^T n)}$$

- Like a ranking loss: (q,p) should be close, (q,n) should be far
- An implementation

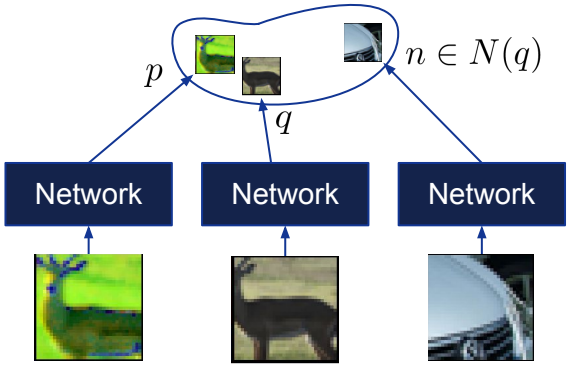
$$logits = [q^T p, q^T n_1, q^T n_2, ..] = q^T [p, n_1, n_2, ..]$$
$$labels = [1, 0, 0, ..]$$
$$InfoNCE = cross\_entropy(softmax(logits), labels)$$

- Squint and see classification loss
  - Replace $[p, n_1, n_2, ..]$ with $[w_p, w_{n_1}, w_{n_2}, ..]$
  - Like classification with weight=exemplars
- More details and perspectives in the next part

Classification



Metric learning

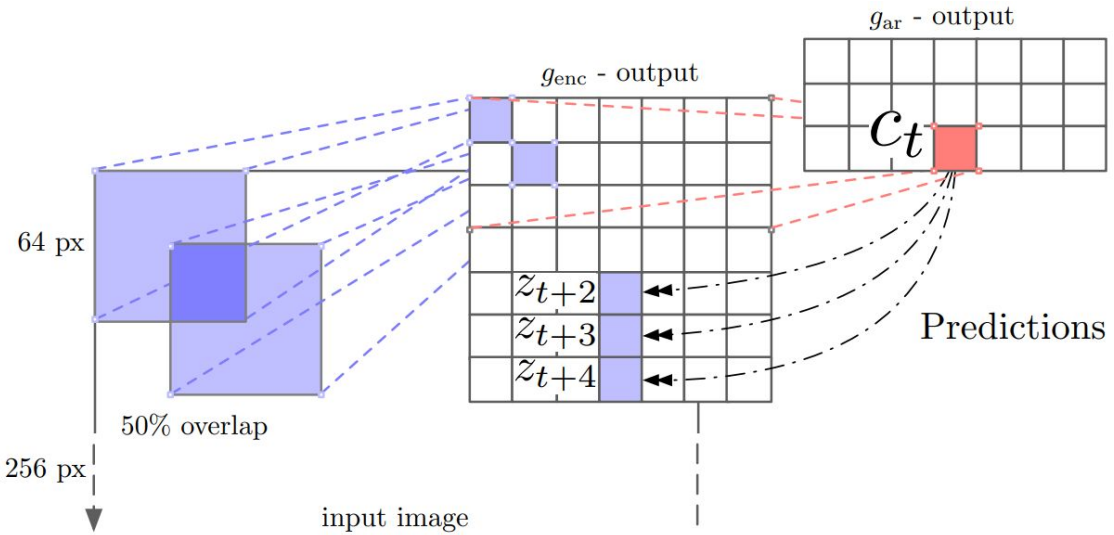


[“Representation Learning with Contrastive Predictive Coding”, van den Oord et al. 18]
[“Data-efficient image recognition with contrastive prediction coding”, Hénaff et al. 19]

Contrastive predictive coding (CPC)

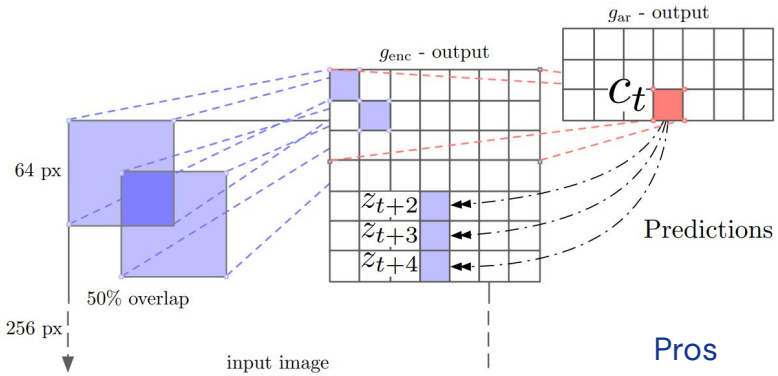
Roughly: Context Prediction + Exemplar ConvNets

- From a patch, predict representations of other patches below it
- Use InfoNCE loss to contrast the (predictions, correct, negatives)
    - Negatives: other patches from the same image and other images



[“Representation Learning with Contrastive Predictive Coding”, van den Oord et al. 18]
[“Data-efficient image recognition with contrastive prediction coding”, Hénaff et al. 19]

Contrastive predictive coding (CPC)



Pros

- Generic framework easily applied to images, video, audio, NLP, ..
- Exemplar: Requires preservation of fine-grained information
- Context prediction: Should enable learning about object parts

Cons

- Exemplar based: images of the same class or instance are negatives
- Train-eval gap: training on patches, evaluating on images
- Assumes training images are photographed with canonical orientations (and canonical orientations exist)
- Somewhat slow training due to dividing into patches

Exploiting time

[“Learning features by watching objects move”, Pathak et al. 16]

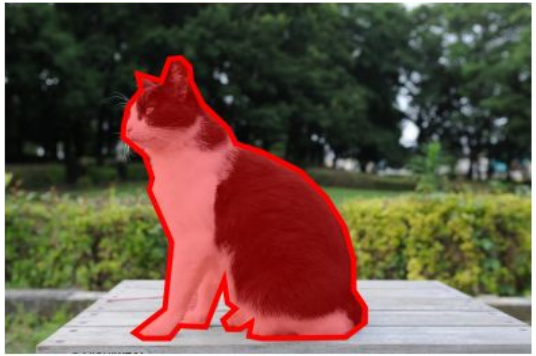
Watching objects move

Which pixels will move?

[“Learning features by watching objects move”, Pathak et al. 16]

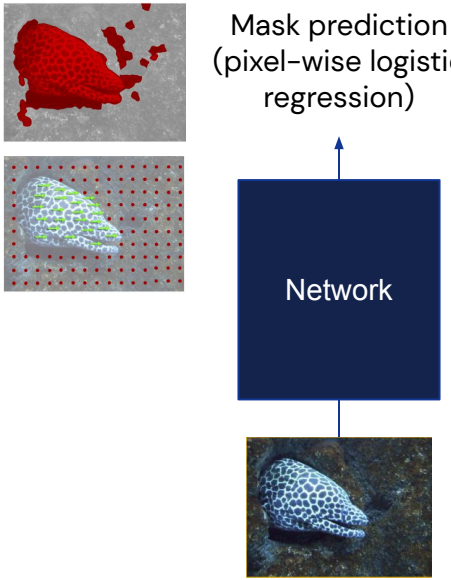
Watching objects move

Which pixels will move? Easy if we can segment objects!



[“Learning features by watching objects move”, Pathak et al. 16]

Watching objects move



Pros

- Emerging behaviour: segmentation
- No train-eval gap

Cons

- “Blind spots”: stationary objects
- Potential focus on large salient objects
- Depends on an external motion segmentation algorithm
- Cannot be extended to temporal nets (pretext task would be trivial)

["Tracking emerges by colorizing videos", Vondrick et al. 18]

Tracking by colorization

Given an earlier frame, colourize the new one.

[“Tracking emerges by colorizing videos”, Vondrick et al. 18]

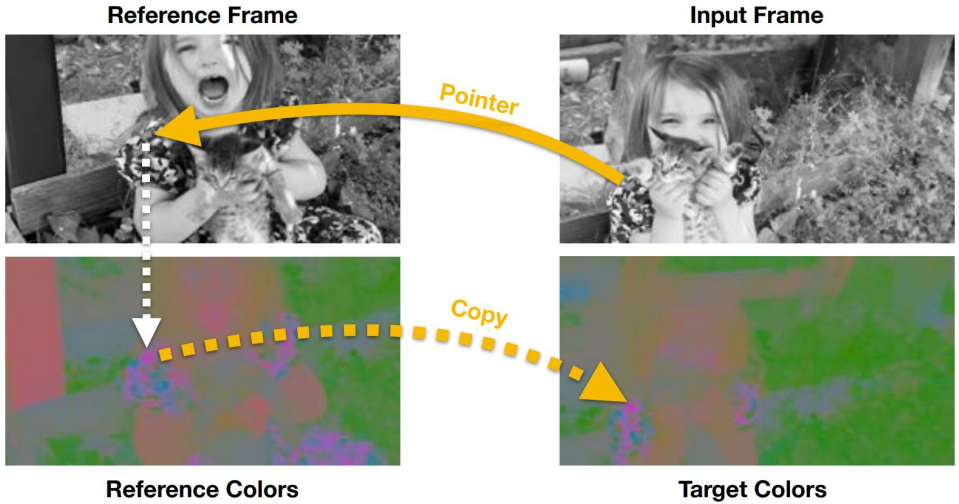
Tracking by colorization

Given an earlier frame, colourize the new one. Easy if everything can be tracked!




[“Tracking emerges by colorizing videos”, Vondrick et al. 18]

Tracking by colorization



Pros

- Emerging behaviour: tracking, matching, optical flow, segmentation

Cons

- Low level cues are effective - less emphasis on semantics
- Forced to evaluate on greyscale frames, losing information

[“Shuffle and learn: Unsupervised learning using temporal order verification”, Misra et al. 16]

Temporal ordering

Is this sequence of frames correctly ordered?

[“Shuffle and learn: Unsupervised learning using temporal order verification”, Misra et al. 16]

Temporal ordering

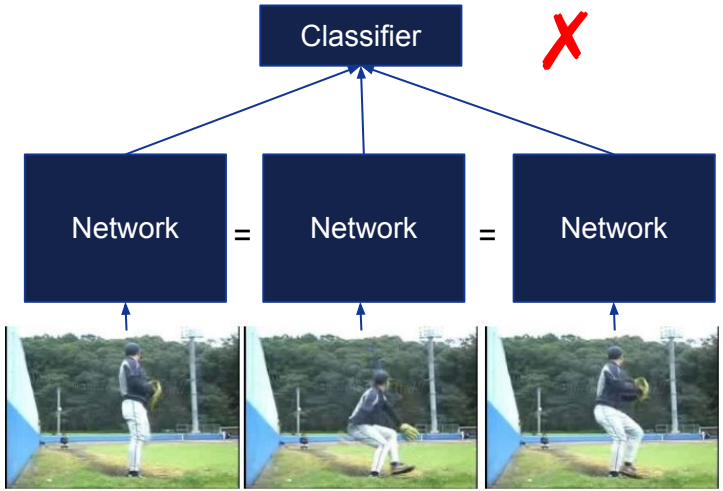
Is this sequence of frames correctly ordered? Easy if we recognize the action and human pose!

✓

✗

[“Shuffle and learn: Unsupervised learning using temporal order verification”, Misra et al. 16]

Temporal ordering



Pros

- No train-eval gap
- Learns to recognize human pose

Cons

- Mostly focuses on human pose - not always sufficient
- Questionable if it can be extended to temporal nets (potentially task becomes too easy)

Extensions

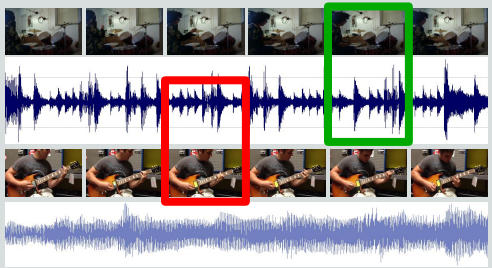
- N frames with one randomly placed - find it

  [“Self-supervised video representation learning with odd-one-out networks”, Fernando et al. 16]

- Ranking loss: embeddings should be similar for frames close in time and dissimilar for far away frames

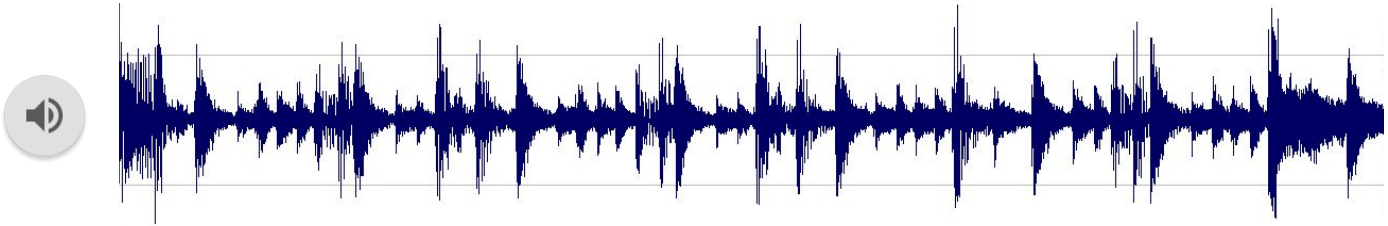
  [“Time-contrastive networks: Self-supervised learning from video”, Sermanet et al. 17]

Multimodal



Audio-visual correspondence

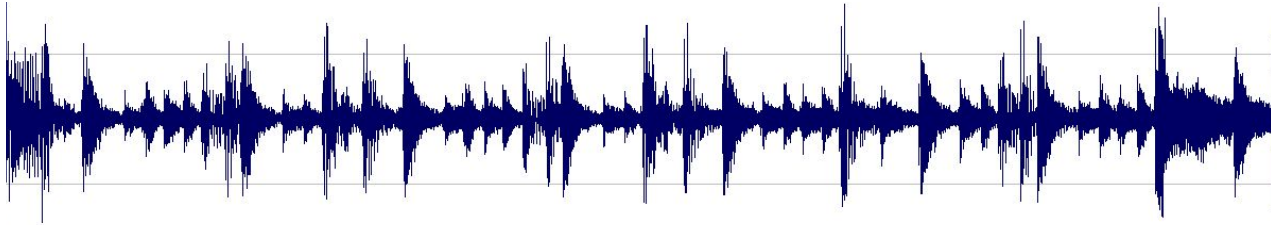
Does the sound go with the image?



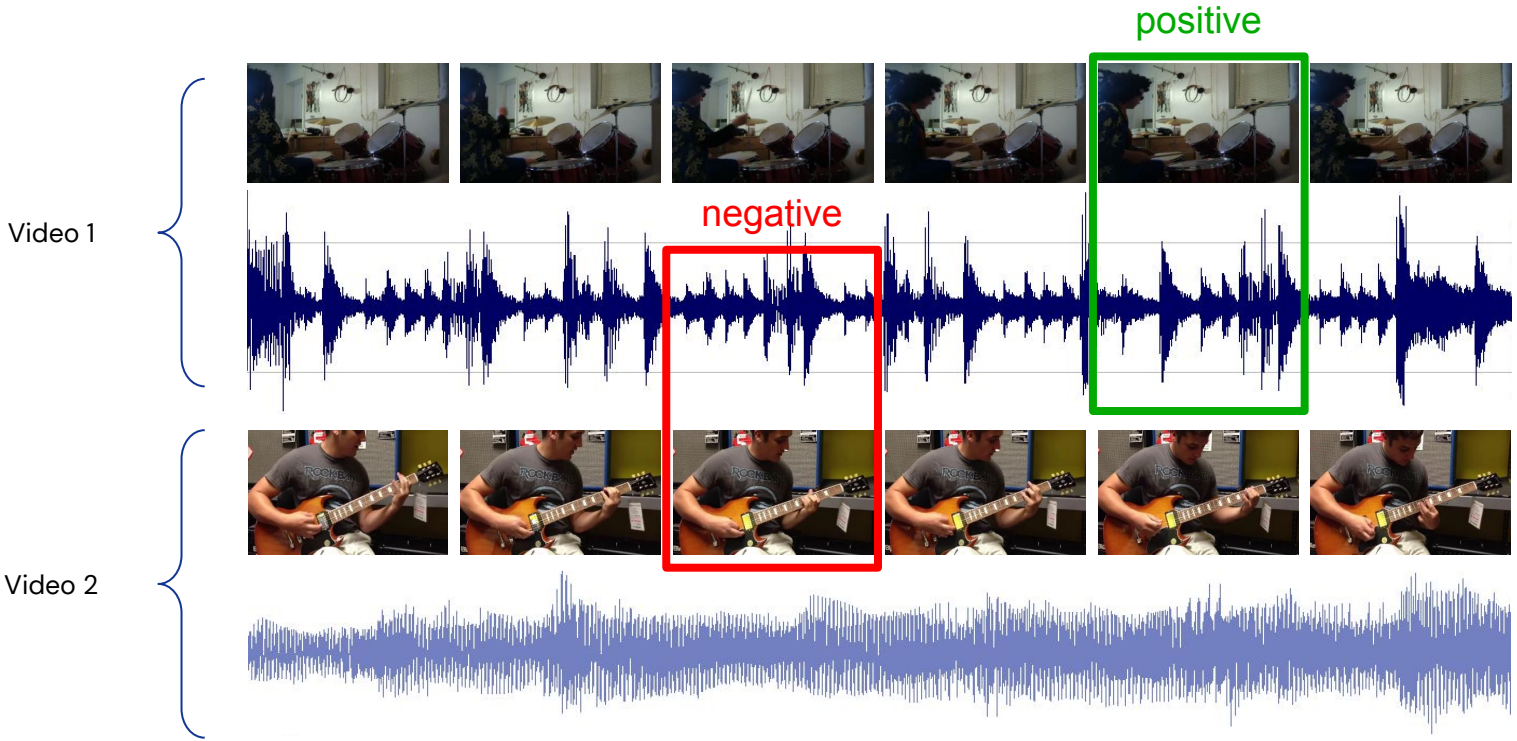
[“Look, Listen and Learn”, Arandjelović et al. 17]

Audio-visual correspondence

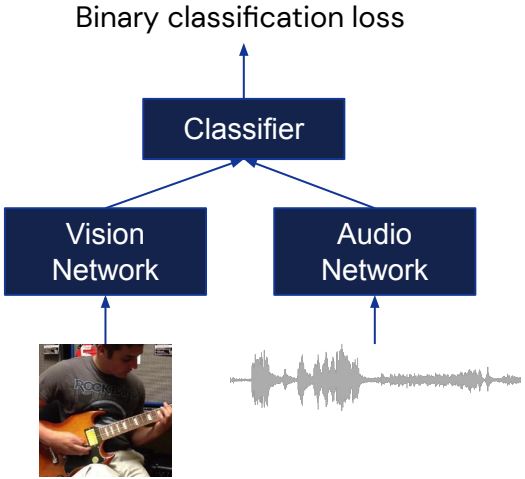
Does the sound go with the image? Easy if we recognize what is happening in both the frame and the audio



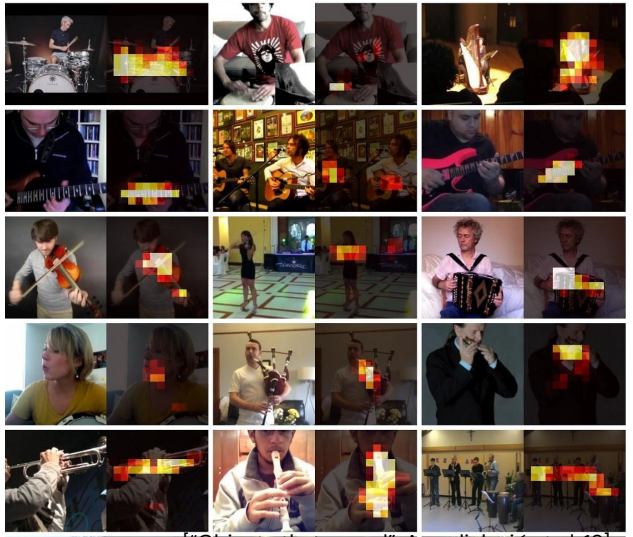
Audio-visual correspondence



Audio-visual correspondence



["Look, Listen and Learn", Arandjelović et al. 17]



["Objects that sound", Arandjelović et al. 18]

Pros

- Natural different views of the training data, no need for augmentations
- No train-eval gap
- Representations in both modalities for free

Cons

- "Blind spots": not everything makes a sound
- Exemplar based: videos of the same class or instance are negatives
- Small output space - two cases (corresponds or not)
  - Can be improved by contrastive approaches

[“End-to-end learning of visual representations from uncurated instructional videos”, Miech et al. 19]

Leveraging narration

Does the narration go with the video?

(Text obtained from automatic speech recognition)

[“End-to-end learning of visual representations from uncurated instructional videos”, Miech et al. 19]

Leveraging narration

Does the narration go with the video? Easy if we recognize what is happening in the video and narrations

(Text obtained from automatic speech recognition)

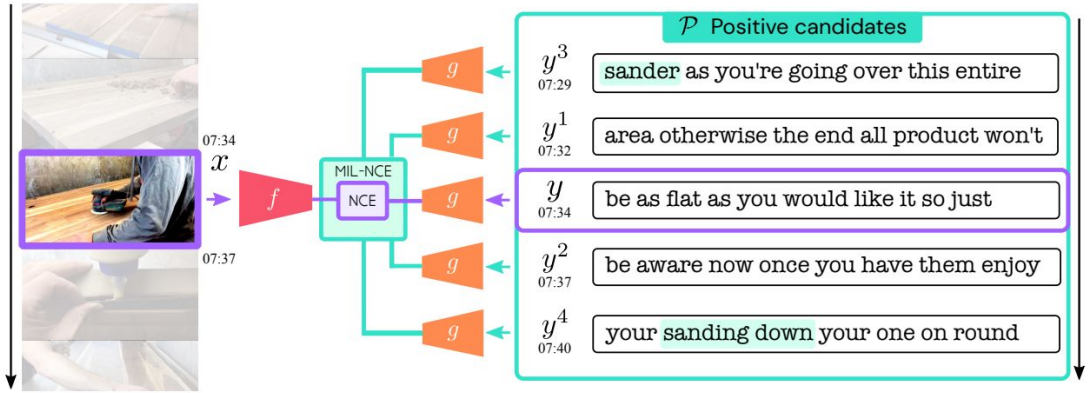
Complication compared to the audio-visual case:

- Narration and visual content are less aligned

[“End-to-end learning of visual representations from uncurated instructional videos”, Miech et al. 19]

Leveraging narration

Multiple instance learning extension of the NCE loss



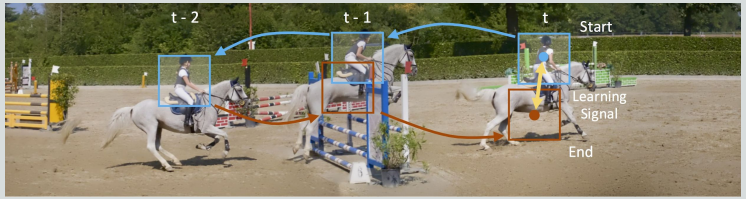
Pros

- Natural different views of the training data, no need for augmentations
- No train-eval gap
- Representations in both modalities for free

Cons

- “Blind spots”: not everything is mentioned in narrations
- Exemplar based: videos of the same class or instance are negatives
- Assumes a single language, potentially non-trivial to extend to more

Further reading



Further reading
(yet another non-exhaustive list)

"Unsupervised learning of visual representations using videos", Wang and Gupta 15

"Out of time: Automated lip sync in the wild", Chung and Zisserman 16

"Learning visual groups from co-occurrences in space and time", Isola et al. 16

"Multi-task self-supervised visual learning", Doersch and Zisserman 17

"Learning and using the arrow of time", Wei et al. 17

"Unsupervised learning by predicting noise", Bojanowski and Joulin 17

"Deep clustering for unsupervised learning of visual features", Caron et al. 18

"Cooperative learning of audio and video models from self-supervised synchronization", Korbar et al. 18

"Audio-visual scene analysis with self-supervised multisensory features", Owens and Efros 18

"Playing hard exploration games by watching YouTube", Aytar et al. 18

"Time-contrastive networks: Self-supervised learning from multi-view observation", Sermanet et al. 18

"Self-supervised learning by cross-modal audio-video clustering", Alwassel et al. 19

"Video representation learning by dense predictive coding", Han et al. 19

"Self-supervised learning for video correspondence flow", Lai et al. 19

"Temporal cycle-consistency learning", Dwibedi et al. 19

"Learning correspondence from the cycle-consistency of time", Wang et al. 19

"Learning representations by maximizing mutual information across views", Bachman et al. 19

"Momentum contrast for unsupervised visual representation learning", He et al. 19

"A simple framework for contrastive learning of visual representations", Chen et al. 20

"Improved baselines with momentum contrastive learning", Chen et al. 20

"Self-supervised learning of pretext-Invariant representations", Misra and van der Maaten 20

"Self-supervised learning of interpretable keypoints from unlabelled videos", Jakab et al. 20

"Evolving losses for unsupervised video representation learning", Piergiovanni et al. 20

"Bootstrap your own latent a new approach to self-supervised learning", Grill et al, 20

3 Practical considerations

Relja Arandjelović & Andrei Bursuc

Practical considerations

In theory

- Set up pretext task, train ⇒ done

In practice

- Networks often “cheat”, find “shortcuts” to solve the pretext task
  - Shortcut prevention is essential
- “Details” (augmentation, network architecture, etc) make a big difference
  - ~~Details~~ Essential components of the method

Next

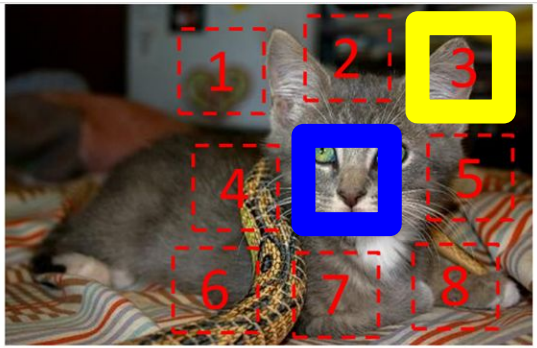
- Shortcut prevention (Relja Arandjelović)
- Further practical considerations (Andrei Bursuc)

Shortcut prevention

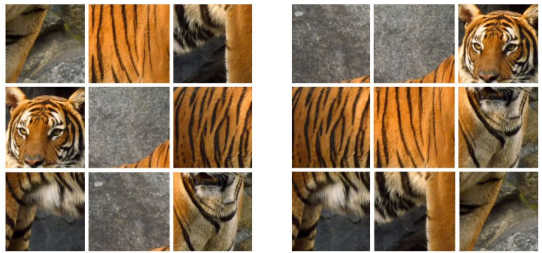
[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]
[“Unsupervised learning of visual representations by solving jigsaw puzzles”, Noroozi et al. 17]

Exploiting the capturing process

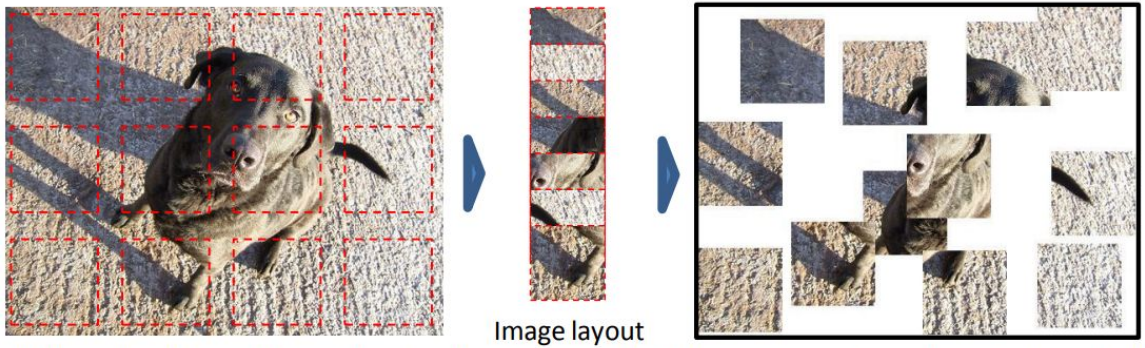
Recall: Context prediction



X = (,); Y = 3



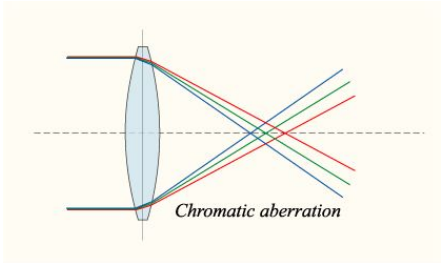
Networks can learn to predict the absolute patch position!



Initial layout, with sampled patches in red

Image layout is discarded

We can recover image layout automatically



source: Wikipedia

Prevent by keeping only 1 channel

- Increases the train-eval gap

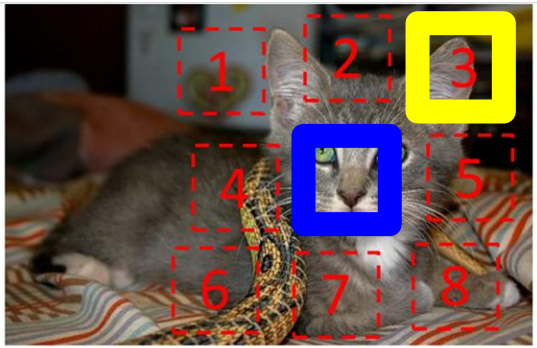
Alternatives

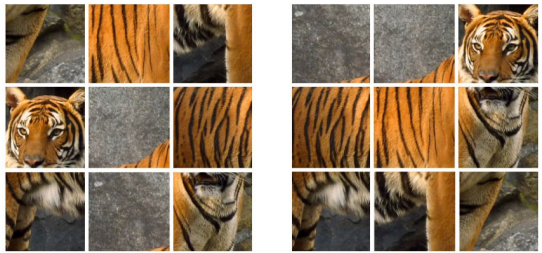
- Spatially jitter the channels [Jigsaw]

[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]
[“Unsupervised learning of visual representations by solving jigsaw puzzles”, Noroozi et al. 17]

Exploiting local content

Recall: Context prediction





Edge continuity and shared boundary patterns

- Leave a gap between patches [Context prediction, Jigsaw]
- Jitter patch locations [Context prediction, Jigsaw]

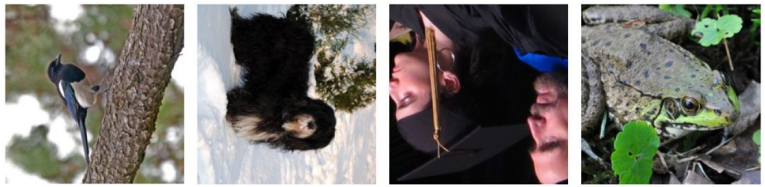
Similar low level statistics

- Normalize by mean and std of each patch independently [Jigsaw]

[“Unsupervised representation learning by predicting image rotations”, Gidaris et al. 18]

Exploiting low-level artefacts: Images

Recall: Rotation prediction



For more complicated transformations (more rotation angles, scales)

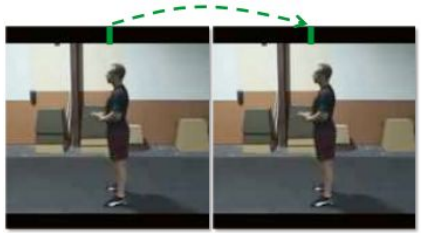
- Network can detect low-level artefacts of the transformations
- Forced to do only 4 rotations as they are implemented purely with flip and transpose artefact-less operations

[“Learning and using the arrow of time”, Wei et al. 18]
[“Video representation learning by dense predictive coding”, Han et al. 19]

Exploiting videos

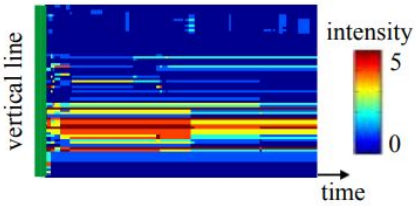
Learning to do optical flow

- Per-frame independent colour jittering



Compression artefacts

- Black frames are not black
  - Remove black framing



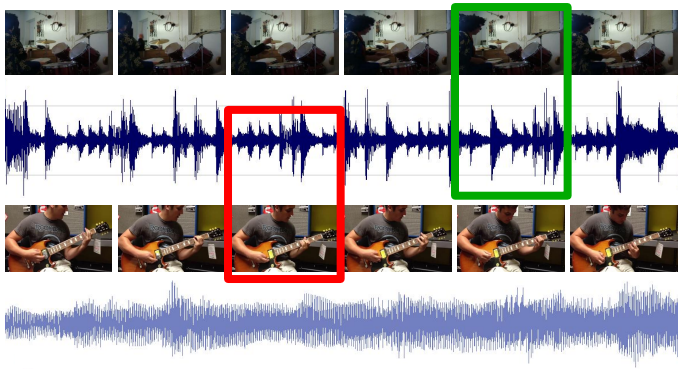
Camera motion

- Some camera motions more likely than others, e.g. zooming in vs zooming out vs randomly changing zoom
  - Partial solution: stabilize the video

[“Look, Listen and Learn”, Arandjelović et al. 17]
[“Objects that sound”, Arandjelović et al. 18]

Exploiting low-level artefacts: Audio

Recall: Audio-visual correspondence



Naive: for negatives, sample a random audio clip from a different video

- Audio is higher fps than video
- Network can detect low-level artefacts (sampling, compression) to detect if audio is aligned with a video frame

Solutions

- Randomly misalign even the positive audio
- Sample negatives to be aligned with video frames

Communicating batch normalization

For NCE Exemplar-based methods

- Typical efficient implementation: form negatives from the rest of the batch
- Sample independence conditions violated
  - Batch normalization suffers, handwavey explanation ["Data-efficient image recognition with contrastive prediction coding", Hénaff et al. 19]
    - Depends on statistics (mean, std) extracted from the batch
    - There is a "communication channel" across the batch
    - Does artificially well during training, so the useful training signal is reduced
- Few ad hoc fixes
  - Remove batch normalization and use e.g. layer normalization ["Data-efficient image recognition with contrastive prediction coding", Hénaff et al. 19]
  - Use large batches so the "communication" via mean and std is harder ["A simple framework for contrastive learning of visual representations", Chen et al. 20]
  - Shuffle batch statistics across replicas in a distributed setup ["Momentum Contrast for Unsupervised Visual Representation Learning", He et al. 19]

Other potential shortcuts (unreported)

["Learning features by watching objects move", Pathak et al. 16]

- Motion can be deduced from artefacts arising from video compression, interlacing, rolling shutter, ..

Exemplar-based methods

- Possible to detect the capture device or compression parameters ["Fighting Fake News: Image Splice Detection via Learned Self-Consistency", Huh et al. 18]

And probably many more

Automatic shortcut removal? ["Automatic shortcut removal for self-supervised representation learning", Minderer et al. 20]

Further practical considerations

Andrei Bursuc