Object Recognition in Images and Video: The State-of-the-arts

http://www.micc.unifi.it/bagdanov/obrec

Prof. Andrew D. Bagdanov

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze
andrew.bagdanov AT unifi.it

April 11, 2017

Outline

Overview

Overview

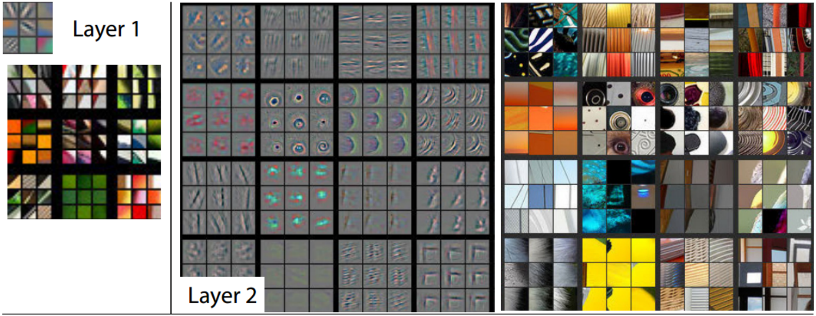
- Today we will wrap up our high-entropy course on object recognition with a look at some state-of-the-art papers.
- These papers are selected from high-impact results from top vision conferences of the last few years.
- With the explosion of interest in CNNs, the community has rapidly discovered new and interesting applications.
- Many of these were thought impossible just a few years ago.
- First, we will look at some early interpretations and re-interpretations of CNNs.

Reflections

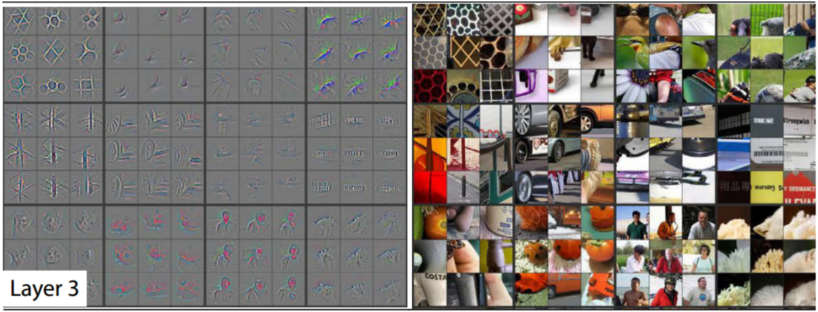
- Remember last week I mentioned that one of the biases against using neural networks was that lack of interpretability.
- As soon as the spectacular results of CNNs on object recognition started coming in, researchers began inventing new ways to interpret the innards of these huge networks.
- This idea was first thoroughly explored in

Visualizing and understanding convolutional networks. MD Zeiler, R Fergus. In: European Conference on Computer Vision, 2014.

- This paper has a ton of interesting analysis of how these networks work.
- I am only going to talk about how visualizations of feature map activations demonstrate what's going on.

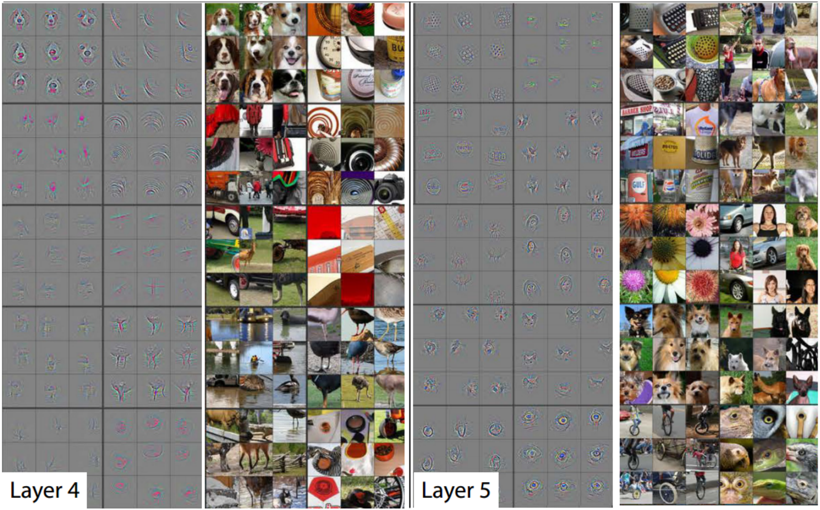


- As we go deeper into the network, feature activations correspond to higher-level semantics.



What's Going On

- Until the network is really indicating the presence of "eyes" and "cat faces", etc.



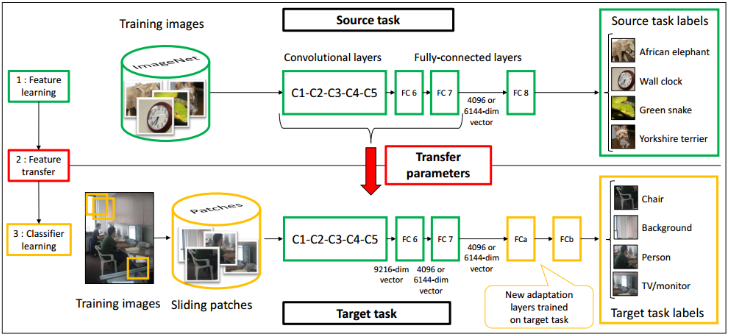
An Astounding Baseline

- Very soon after the AlexNet results become public, the community began asking the natural question: What if I don't have 1.3M images (and unlimited GPU cycles)? What then?
- Well, it turns out that CNNs trained on large-scale datasets (e.g. ImageNet) are also pretty damn good feature extractors.
- The idea: use a trained CNN to extract the activations of the first fully connected layer. Use that (usually a 4K-dimensional descriptor) as a feature representation for standard technique (e.g. SVM).
- This technique was first explored in:

CNN Features off-the-shelf: an Astounding Baseline for Recognition. AS Razavian, H Azizpour, J Sullivan, S Carlsson. In: Proceedings of CVPR, 2014.

Transfer Learning

- We immediately saw that CNNs worg great as feature extractors.
- A 4K-dimensional CNN feature (with linear SVM) works about as well as a 250K-dimensional Fisher Vector.
- But, they can even achieve state-of-the-art results via transfer learning.



- So, if you want to use CNNs, but don't have millions of images, the standard procedure has become:
  1. Take a state-of-the-art CNN pre-trained on ImageNet.
  2. Decapitate the pre-trained network (i.e. remove the FC and classification layers).
  3. Fine-tune new FC and classification layers (randomly initialized) on the new problem.
- My point: think hard about your problem before training a Deep CNN from scratch.

SOA: The YOLO Detector

YOLO: The Idea

- Recall the Fast-RCNN detector from last week.
- Advantages: fast, state-of-the-art performance.
- Disadvantages: relies on external, slow method for object region proposal; not fully end-to-end trainable.
- Our first paper today will look at a current state-of-the-art approach that incorporates region proposal right in the network:

You only look once: Unified, real-time object detection. J Redmon, S Divvala, R Girshick, A Farhadi. In: Proceedings of CVPR, 2016.

YOLO: The Details

- [SWITCH PRESENTATION]

SOA: Fully Convolutional Networks

FCNs: Dense Object Recognition

- Now we will take a look at an approach to semantic image segmentation.
- This problem could be considered a type of extreme object localization.
- The goal: label all pixels in an image with an object category.
- A state-of-the-art approach to this is:

Fully convolutional networks for semantic segmentation. E Shelhamer, J Long, T Darrell. In: IEEE Transactions of PAMI, 2017.

- This technique takes the idea of fully convolutional networks to the limit.

FCNs: The Details

- [SWITCH PRESENTATION]

SOA: Dense Image Captioning

DENSECAP: A Sexy Application

- These days it seems like everyone is talking about image captioning.
- This is another application of object recognition that seemed impossible just a few years ago.
- We will now look at a recent work on dense image captioning from CVPR 2016.

Densecap: Fully convolutional localization networks for dense captioning.
J Johnson, A Karpathy, L Fei-Fei. In: Proceedings of CVPR, 2016.

DENSECAP: What is Image Captioning?

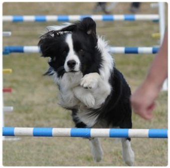
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."



"black and white dog jumps over bar."

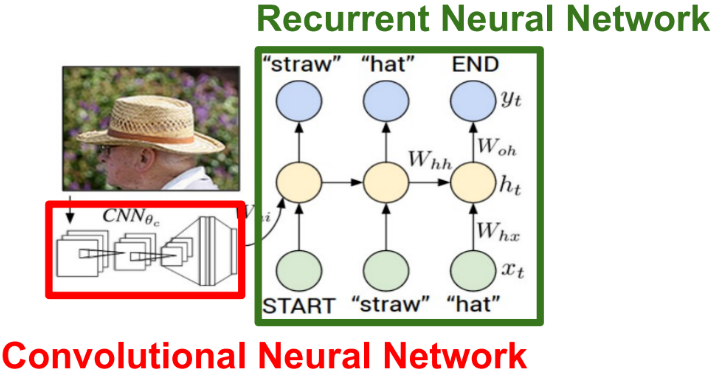
"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

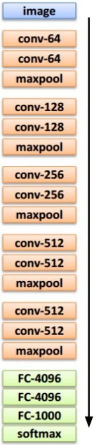
DENSECAP: What is Image Captioning?

- This is clearly an extremely hard problem:

- So, how do we do it?
- By combining a CNN (which we already know how to build), with a recurrent neural network:

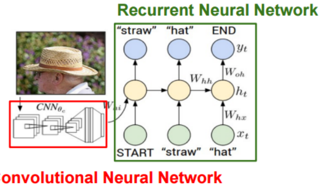


DENSECAP: What is Image Captioning?

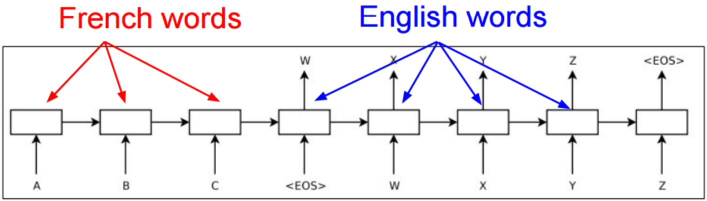


Summary so far:

Convolutional Networks express a single differentiable function from raw image pixel values to class probabilities.



- The other half of the equation is a recurrent network.
- Recurrent networks are good at modeling sequential data.
- An excellent example is machine translation.
- If you train the network on a huge number of sentence translation pairs, you can learn a network that translates text sequentially.



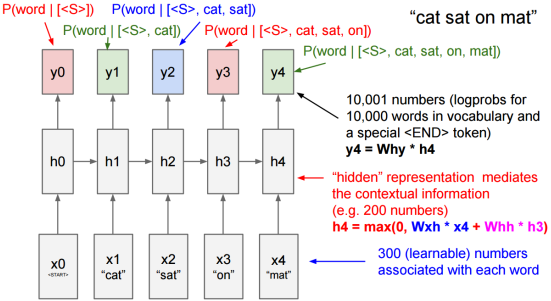
DENSECAP: What is Image Captioning?

- We aren't interested in translating sentences, however.
- We want to "translate" images.
- Captioning usually uses a sequential model of sentence generation:

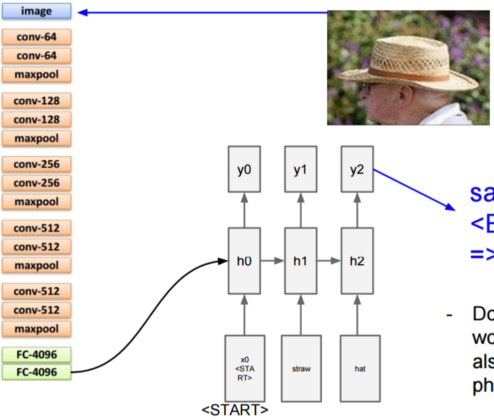
We want to train a **language model**:
P(next word | previous words)

i.e. want these to be high:
P(cat | [<S>])
P(sat | [<S>, cat])
P(on | [<S>, cat, sat])
P(mat | [<S>, cat, sat, on])

- The standard recurrent network for this type of task is the Long Short-Term Memory (LSTM) Network.
- This network has a hidden representation (a memory) that is sequentially updated to model context during generation.
- At each step, you can remember the top, say, 100 candidate words.
- Then beam search can be used to find the best output sentence.



- How do we get the whole thing started?
- We pass a learned representation of the image to the LSTM:



test image

sample!
<END> token
=> finish.

- Don't have to do greedy word-by-word sampling, can also search over longer phrases with **beam search**

DENSECAP: What is Image Captioning?

- Train the network on a huge set of image/text pairs.
- And see what happens:

a group of people standing around a room with remotes
logprob: -9.17

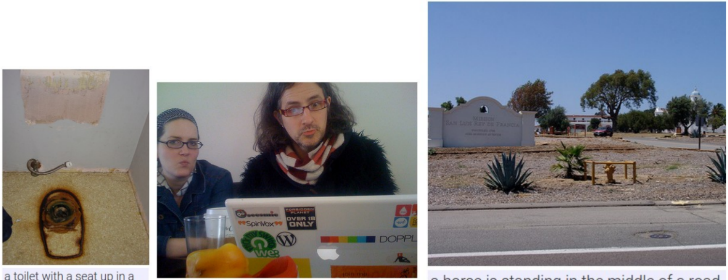
a young boy is holding a baseball bat
logprob: -7.61

a cow is standing in the middle of a street
logprob: -8.84

DENSECAP: What is Image Captioning?

UNIVERSITÀ DEGLI STUDI FIRENZE

- When it fails, it's not really clear why. . .



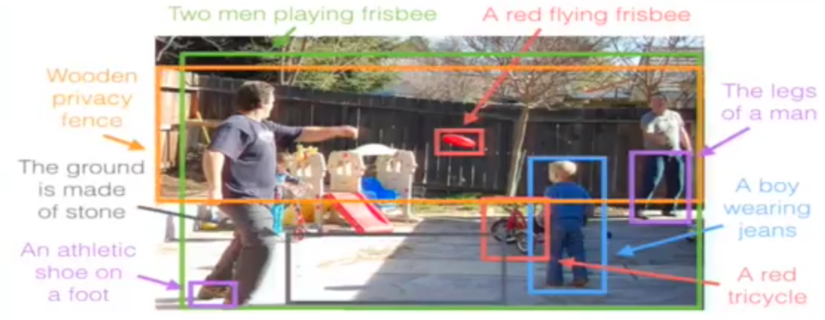
a toilet with a seat up in a bathroom
logprob: -13.44

a woman holding a teddy bear in front of a mirror
logprob: -9.65

a horse is standing in the middle of a road
logprob: -10.34

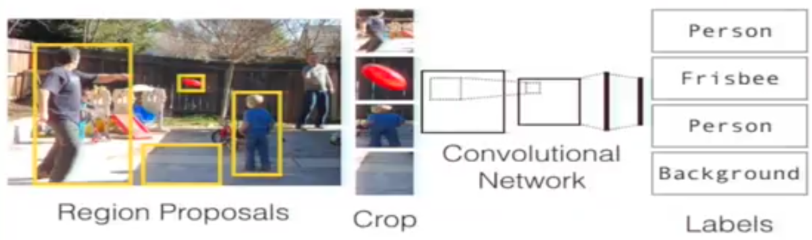
DENSECAP: Going Dense

- Main observation: when you ask people to annotate images with textual descriptions, you get lots of interesting and dense information:



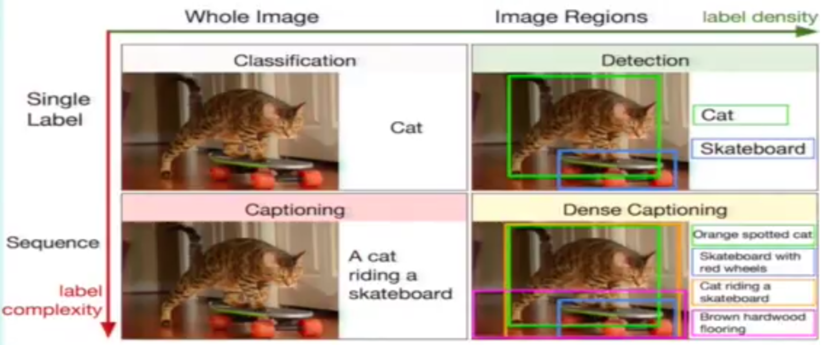
DENSECAP: Going Dense

- Main innovation: use region proposals to generate candidate regions for captioning.
- This leverages the ideas behind Fast-RCNN and YOLO.
- BUT: the captioning system doesn't do detection.

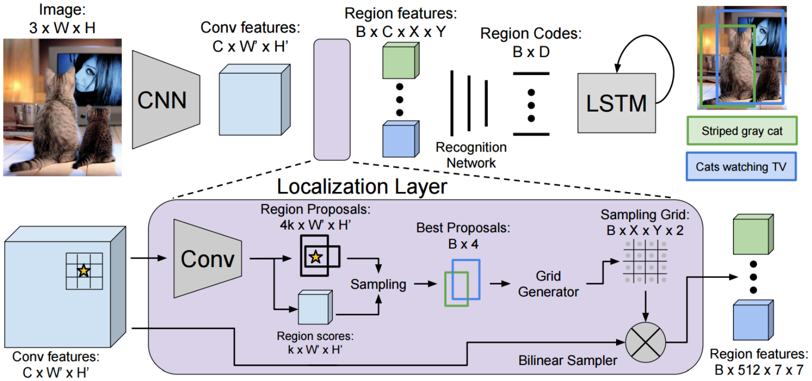


DENSECAP: Going Dense

- This is a good overview of the current panorama of object recognition.

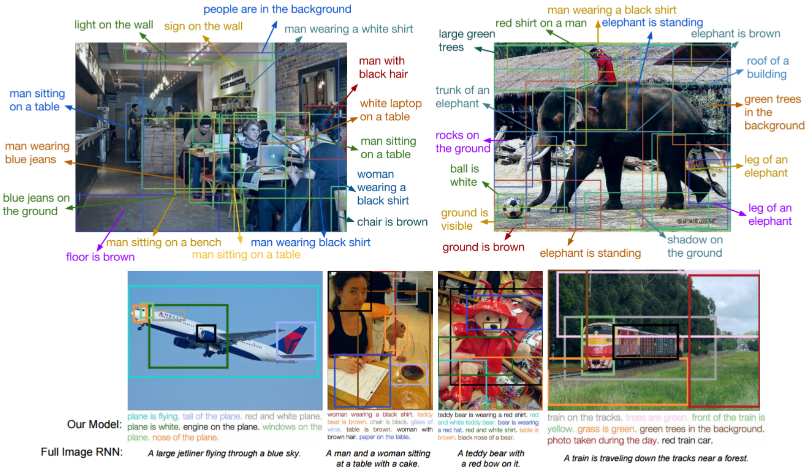


- Use a localization network that localizes semantically relevant objects (i.e. captioned scene elements).
- Feed CNN features from these regions into an LSTM for captioning.
- Train end-to-end on a large, densely captioned image dataset (Visual Genome).

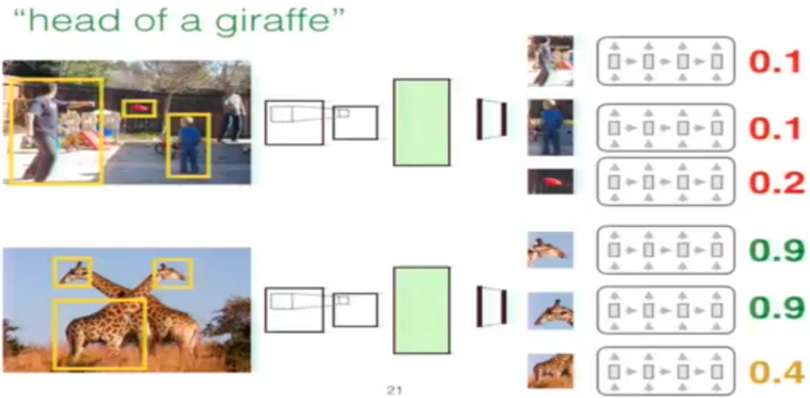


DENSECAP: Going Dense

- Main observation: when you ask people to annotate images with textual descriptions, you get lots of interesting and dense information:

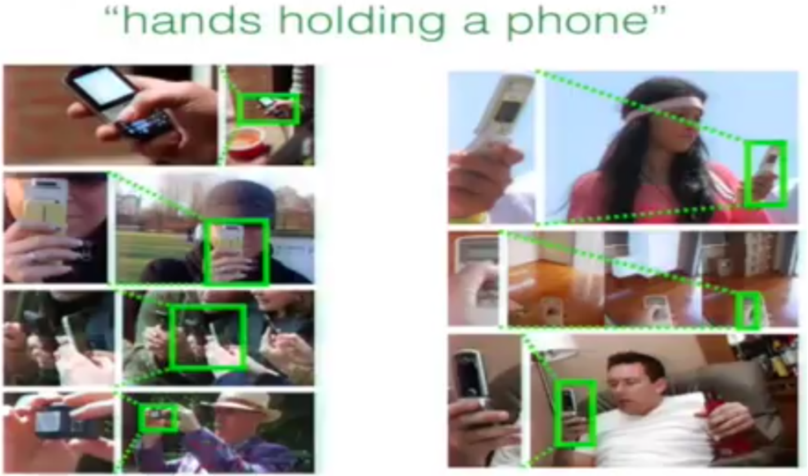


- You can also run the system in reverse to access image content using natural language queries.
- Propagate all proposal regions through the network, compute likelihood of query caption from the beam.



DENSECAP: Going Dense

- This approach is able to localize high-level semantic concepts in images.



DENSECAP: Going Dense

- Results takeaway: captioning is hard and subjective; retrieval might be "easy".

Region source	Language (METEOR) EB	RPN	GT	Dense captioning (AP) EB	RPN	GT	Test runtime (ms) Proposals	CNN+Recog	RNN	Total
Full image RNN [21]	0.173	0.197	0.209	2.42	4.27	*14.11*	210ms	2950ms	**10ms**	3170ms
Region RNN [21]	0.221	0.244	0.272	1.07	4.26	*21.90*	210ms	2950ms	**10ms**	3170ms
FCLN on EB [13]	**0.264**	**0.296**	0.293	4.88	3.21	*26.84*	210ms	**140ms**	**10ms**	360ms
Our model (FCLN)	**0.264**	0.273	**0.305**	**5.24**	**5.39**	***27.03***	**90ms**	**140ms**	**10ms**	**240ms**

	Ranking R@1	R@5	R@10	Med. rank	Localization IoU@0.1	IoU@0.3	IoU@0.5	Med. IoU
Full Image RNN [21]	0.10	0.30	0.43	13	-	-	-	-
EB + Full Image RNN [21]	0.11	0.40	0.55	9	0.348	0.156	0.053	0.020
Region RNN [21]	0.18	0.43	0.59	7	0.460	0.273	0.108	0.077
Our model (FCLN)	**0.27**	**0.53**	**0.67**	**5**	**0.560**	**0.345**	**0.153**	**0.137**

DENSECAP: Summary

- The DENSECAP system is able to generate rich annotations of images.
- For someone that has been in the field for 20 years, the results are amazing.
- Note that the system is built from well-known components: CNNs and LSTMs.
- This is becoming a common trend in object recognition: piece together known building blocks, make sure gradient pathways exist, and train end-to-end.
- Note, however, that this technique requires a massive amount of manual annotation upfront.

SOA: DCGANs for Representation Learning

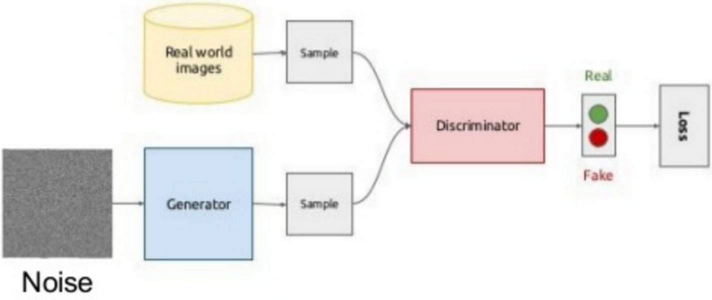
DCGANs: A New Idea

- We will now talk about an extremely hot topic in computer vision and machine learning.
- The Generative Adversarial Network (GAN) is a model that simultaneously learns to generate and discriminate.
- The most recent incarnation of this idea showed how we can use a GAN to perform unsupervised training of feature extractors:

Unsupervised representation learning with deep convolutional generative adversarial networks. A Radford, L Metz, S Chintala. In: arXiv preprint arXiv:1511.06434, 2015.

- The idea: leverage the huge amount of unlabeled image data available to learn representations good for (later) discrimination.

- According to Yann LeCun: "The most important and interesting new idea in training neural networks in recent years."
- The basic idea is really simple:



The GAN Idea

- And the idea doesn't lose its elegance even in the details.
- How can we optimize such a model?

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

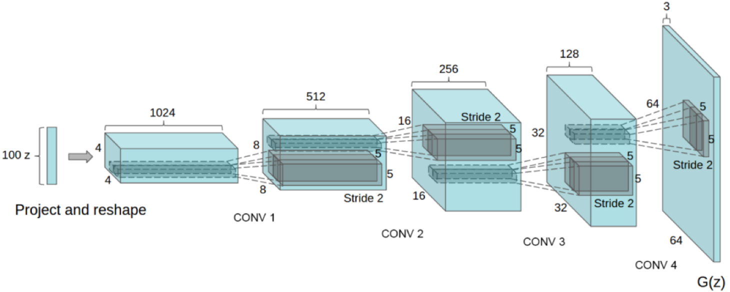
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

The GAN Idea

- Simple:
  - The generator is trained to generate good counterfeit images.
  - While the discriminator is trained to be good at discriminating real images from fake ones.
- Not so simple:
  - What should the architecture of the generator be?
  - What should the architecture of the discriminator be?
  - Are GANs useful for anything?

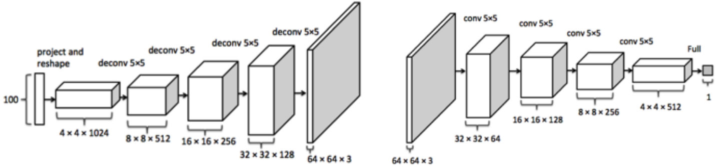
The DCGAN Architecture

- The Deep Convolutional GAN (DCGAN) is a GAN that uses a Deep Convolutional Network (duh).
- The generator looks like this:



- We are skipping over some details, like how we can use convolutions to scale up instead of down.
- See the paper (or the presentation on the website) for the details.

- What should the discriminator look like?
- Well, why not have it be exactly like the generator, but in reverse.
- The discriminator is a CNN that takes an image of size 64x64 and crunches it down to a single output: real or fake.



DCGAN Results

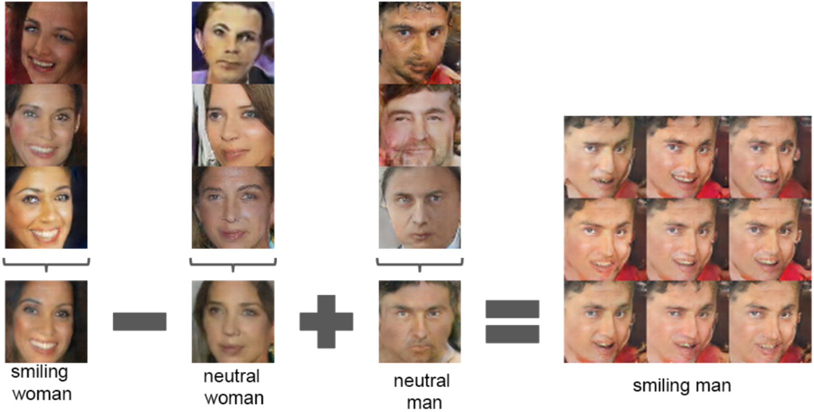
- This idea, at first glance seems almost stupidly simple.
- One thinks (at least I did): there's no way in hell this could work.
- Results after only a single epoch over the LSUN Bedrooms dataset:

DCGAN Results

- After five epochs, the results are positively convincing:

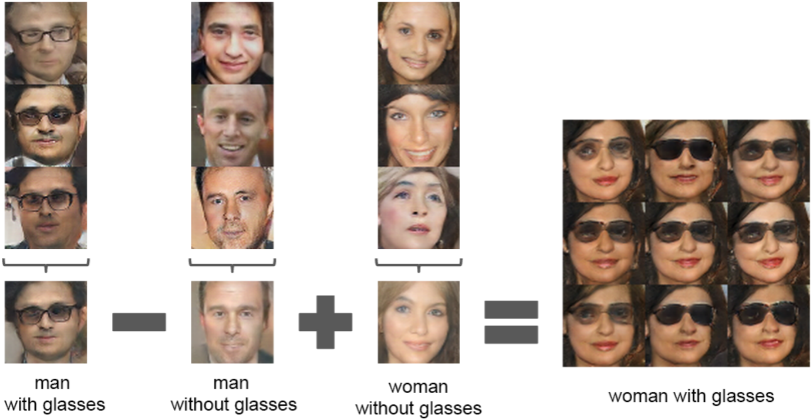
DCGAN: Cool Party Tricks

- In the DCGAN paper they demonstrate how the latent space can be used (specifically, the vector space properties of the latent space) to do interesting things.



DCGAN: Cool Party Tricks

- Another example that is a little less terrifying:



DCGAN: Cool Party Tricks

- Of course, this is particularly interesting because trying to do the same thing in pixel space is hopeless.
- Note: these examples are obviously cherry picked; it is entirely unclear how to derive the $\mathbf{z}$ of "smiling woman", for example. . .

Results of doing the same arithmetic in pixel space

DCGAN: Wait, there's more!

- Of course, as the title of the paper indicates, the objective of this work isn't just to generate cool images from noise (although that's pretty awesome).
- The authors observe that we can take the learned features in the discriminator and use them to solve new problems.
- They train a GAN on ImageNet without labels – that is, they just use ImageNet images as real images without using class labels.
- Then, they extract and concatenate all of the convolutional feature maps for an image, which results in a 30K-dimensional vector as a feature representation for an image.
- Finally, they train a linear SVM to classify each class in CIFAR-10 (a image recognition dataset with 10 classes).

DCGAN: Wait, there's more!

- The results are competitive with the state-of-the-art.
- The really impressive aspect of this technique, is that DCGAN is generating training examples out of thin air.

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

Discussion

- For the exam, remember that everyone should select a paper from a recent top conference.
- You should prepare a brief presentation explaining this work to me (reading group style).
- Some tips:
  - Try to select and interpret the paper you present through the lens of your own work and interests.
  - Use all resources available (e.g. the paper itself, publicly available code, other presentations on the paper, etc).
- As soon as you select your papers, send me an email to let me know.
- I would like to finish all exams before 16 June, if possible.

Discussion

- In this short course on object recognition, I hope I have communicated how active and vibrant object recognition has become.
- Much of this is due to the renaissance of Convolutional Neural Networks.
- There are hundreds of interesting techniques applications being explored today:
  - Captioning: really hot topic.
  - Style transfer: really cool party tricks.
  - Tracking: real-time recognition and localization.
  - Generative models: GANs are only one approach.
  - ...
- Object recognition is the motor application driving innovation in Computer Vision and Machine Learning today.