Deep Nets: What have they ever done for Vision?”

Alan Yuille

Dept. Cognitive Science and Computer Science

Johns Hopkins University

What Have Deep Nets done to Computer Vision?

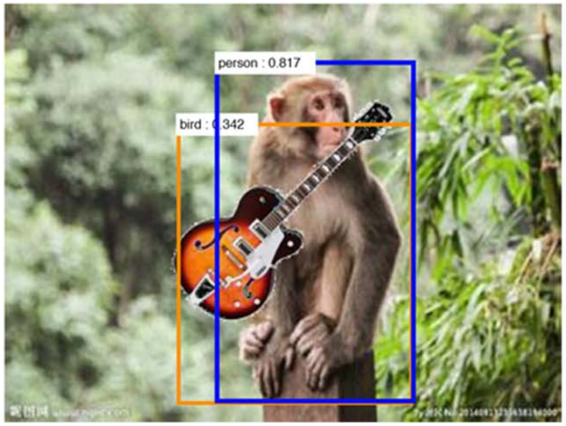
- Compared to human observers, Deep Nets are brittle and rely heavily on large annotated datasets. Unlike humans, Deep Nets have difficulty learning from small numbers of examples, are oversensitive to context, have problems transferring between different domains, and lack interpretability.
- What are the challenges that Deep Nets will need to overcome? What modifications will they need to address these challenges. In particular, how to deal with the combinatorial complexity of real world stimuli.
- *Alan Yuille and Chenxi Liu. "Deep Networks: What have they ever done for Vision?". Arxiv. 2018.*

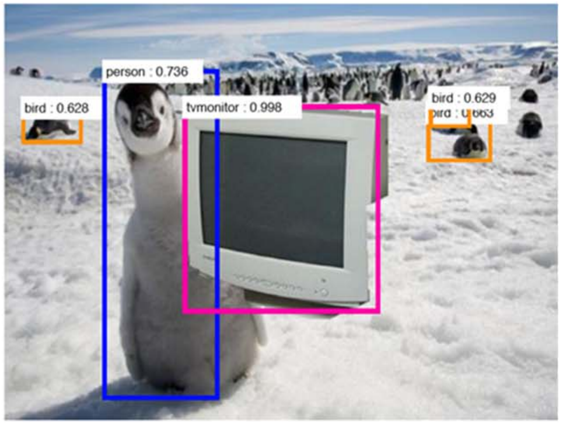
Deep Nets face many challenges

- Deep Nets face many challenges if we want them to develop systems which are robust, effective, flexible, and general-purpose.
- What are their current limitations?
- Dataset Bias, Domain Transfer, Lack of Robustness.
- And perhaps the combinatorial explosion?
- What types of models can deal with these challenges.

Explore the robustness of Deep Nets by photoshopping Ocluders and Context.

- Deep Nets have sensitivity to occlusion and context.

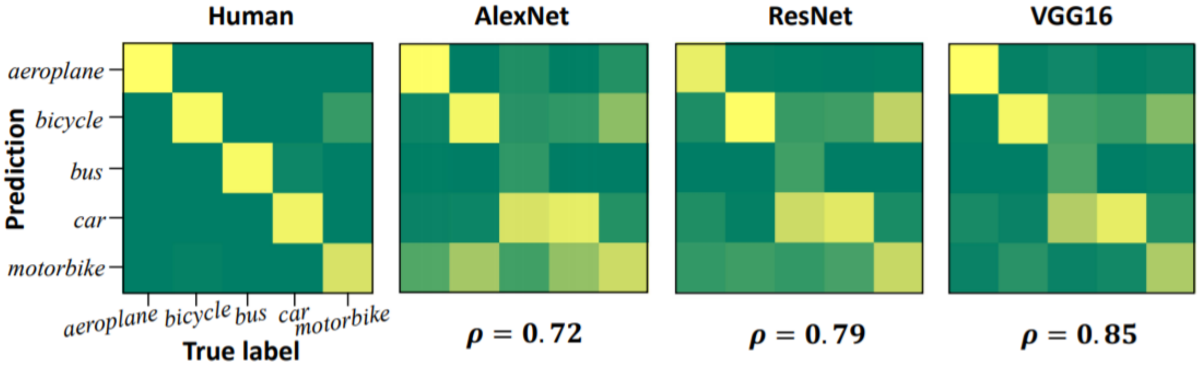




- J. Wang et al. "Visual concepts and compositional voting."In Annals of Mathematical Sciences and Applications.2018,
- See also “The elephant in the room” A. Rosenfeld et al. Arvix. 2018)

Deep Nets have errors to random occlusion.

- Compare Human observers to Deep Nets for classifying objects with random occlusions.
- Deep Net performance is not terrible, but is significantly weaker than humans. Humans occasionally confuse bike with motor-bike, but deep nets have more confusions (e.g., between cars and buses).



- Hongru Zhu et al. Robustness of Object Recognition under Extreme Occlusion in Humans and Computational Models. Proc. Cognitive Science. 2019.

Datasets: Biases. Rare Events, and Transfer

- Deep Net sensitivity to occlusion and context is only one of several challenges.
- Dataset-bias is another challenges. They are a finite set of samples from the enormous domain of real world images. This induces biases, like “rare events”.
- Domain-Transfer is another challenge. Results on one image domain may fail to transfer to images from another image domain (examples later).
- *But, arguably, these are all symptoms of a large problem.*

When are Datasets big enough?

- Deep Nets are learning based methods.
- Like all machine learning methods, they assume that the observed data (X,Y) are random samples from an underlying distribution P(X,Y).
- This is justified by theoretical studies – e.g., Probably Approximately Correct theorems (Vapnik, Valiant, Smale and Poggio) – and, in practice, by using cross-validation to evaluate performance.
- *But these theoretical studies require that the annotated datasets for testing and training Deep Nets are sufficiently large to be representative of the underlying problem domain.*
- When will the datasets be big enough?

Data Set sizes: Examples.

- If the goal is to detect Pancreatic Cancer, then the datasets need to capture the variability of the shapes of the Pancreas and the size and location of tumors. This is a well-defined and constrained domain.
- If the goal is to recognize faces, then the datasets need to be big enough to capture the variability of faces. This is also well-defined and constrained domain.

- In these constrained domains, we need big datasets. But they are finite and it seems possible to obtain them.

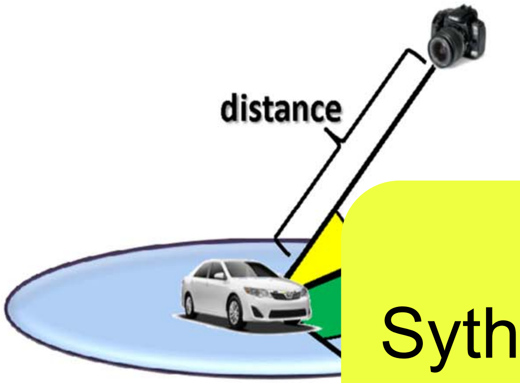
- But for many vision tasks, the domains are much larger.

The Space of Images is Infinite

- The space of images is infinite. There are infinitely many images infinitesimally near every image in the datasets. This is exploited by digital adversarial attacks.
- *This may not be serious because Deep Nets can probably be trained to deal with this problem.* For example, by using the min-max principles (Madry et al. 2017).

- From a computer graphic perspective. A model for rendering a 3D virtual scene into an image will have several parameters: e.g.,. camera pose, lighting, texture, material and scene layout. If we have 13 parameters, see next slide, and they take 1,000 values each then we have a dataset of 10^39 images.
- *Deep Nets may be able to deal with this also.* But they require many examples and might perform worse than an algorithm which could identify and characterize the underlying 13-dimensional manifold *by factorizing geometry, texture, and lighting.*

Images from synthesized computer graphics model.



Sythesized data: INFINITE image space

Camera Pose(4).
azimuth
elevation
tilt(in-plane rotation)
distance

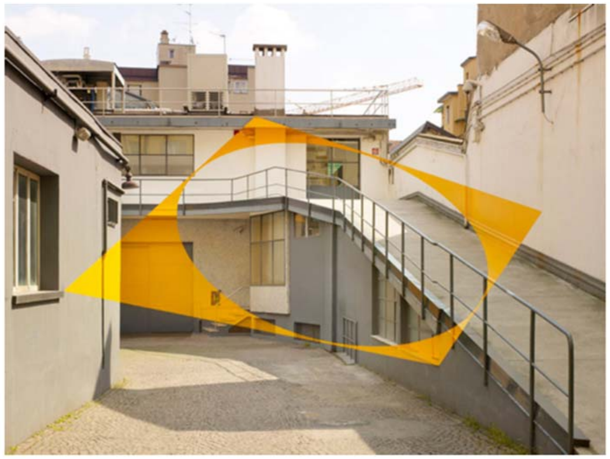
#light source
type(point, dire
omni)
position
color
...

cene Layout(3):
Background
Foreground
Position(Occlusion)

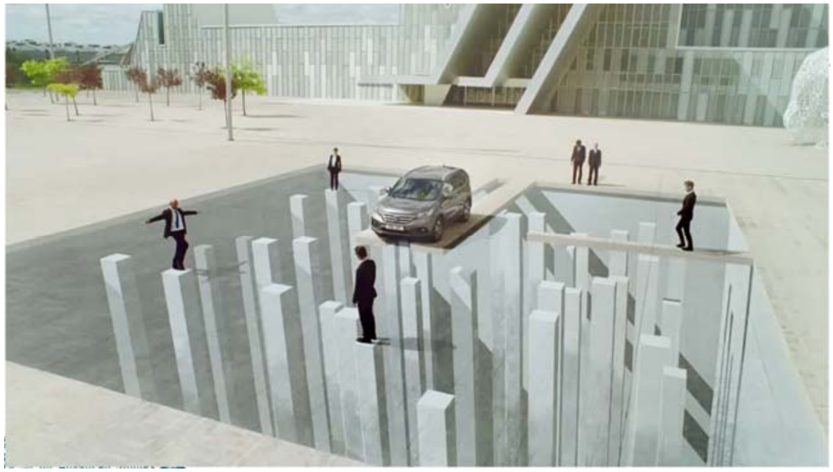
Suppose we simply sample 10^3 possibilities of each parameter listed...

Factorize geometry, texture and lighting.

- *Humans can usually factorize geometry, texture, and lighting.*
- But occasionally they make mistakes: from C. von der Malsburg.
- Right: what is this image? Left: are the men safe?



Felice Varini



The Big Challenge: Combinatorial Complexity

- More seriously:
- *Combinatorial possibilities arise when we start placing objects together in visual scenes. M objects can be placed in N possible locations in the image.*
- *Combinatorial possibilities even arise if we consider a single rigid object which is occluded. E.g., The object can be occluded by M possible occluding patches in N possible positions.*

- Perhaps most of these combinatorial possibilities rarely happen – they are all “rare events”.
- But in the real world, rare events can kill people (e.g., failing to find a Pancreatic tumor, an automatic car failing to detect a pedestrian at night, or a baby sitting in the road).

The Combinatorial Complexity Challenge

- What happens if we have combinatorial complexity? There are two big questions:
- (I). *How can we train algorithms from finite amounts of data, but which generalize to combinatorial amounts.* Can Deep Nets generalize in this manner?
- Their sensitivity to Context and Occluders is worrying.
- (II). *How can we test algorithms on finite amounts of data and ensure that they will work on combinatorial amounts of data.* The performance of Deep Nets when tested with random occlusions and patches is worrying.

Deep Nets and combinatorial complexity: Learning.

- Like all Machine Learning methods, Deep Nets are trained on finite datasets. It is impractical to train them on combinatorially large datasets (which may be available using Computer Graphics, see later).
- What to do?
- (I) We may be able to develop strategies where the Deep Net actively searches a combinatorially large space to find good training data (e.g., an active robot).
- (II) Can we develop Deep Nets, or other visual architectures, which can learn from finite amounts of data but generalize to combinatorially large datasets?

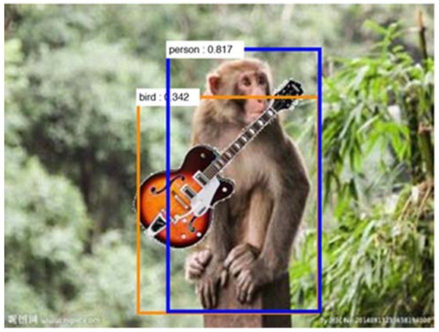
Deep Nets and Combinatorial Complexity: Testing

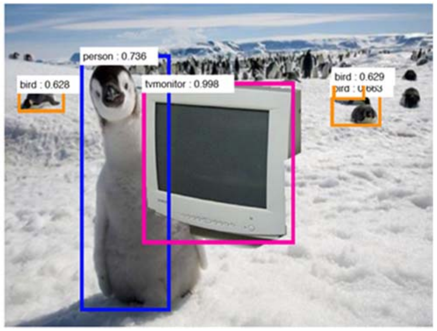
- How to test algorithms – like Deep Nets – if the datasets are combinatorially large?
- Average case performance may be very misleading. Worst case performance may be necessary.
- *To test on combinatorially complex datasets would require actively searching over the dataset to find the most difficult examples.* These requires generalizing the idea of an adversarial attack from differentiable digital attacks to more advanced non-local and non-differentiable attacks – like occluding parts of objects.
- *“Let your worst enemy test your algorithm”.*

Can Deep Nets deal with Combinatorial Complexity?

- *Objects can be occluded in a combinatorial number of ways.* It is not practical to train Deep Nets of all of these. Instead, we can train on some occluders and hope they will be robust to the others.

- Recall that Deep Nets have difficulty with occlusion and unusual context.





- Recall that Deep Nets perform worse than human at recognizing objects under occlusion. (Hongru Zhu et al. 2019).

Can Deep Nets deal with Combinatorial Complexity?

- This is an open issue.
- My opinion is that they will need to be augmented in at least three ways:
- (I) Compositional – explicit semantic representations of object parts and subparts. (Not “compositional functions).
- (II) 3D Geometry – representing objects in terms of 3D geometry, enables generalization across viewpoints (and useful for robotics).
- (III) Factorize appearance into geometry, material/texture, and lighting – as done in Computer Graphics models.
- I will give a few slides about (I) and (II).

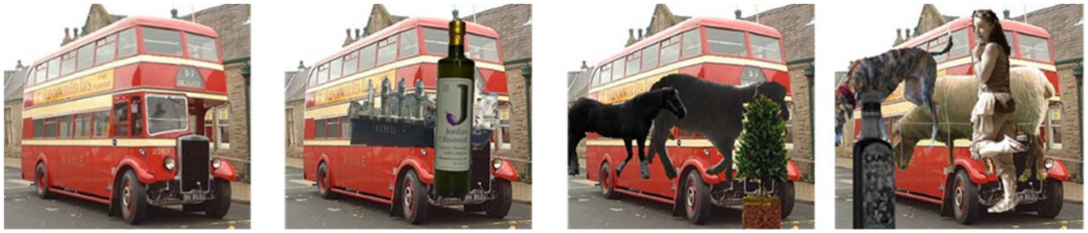
Contrast Deep Nets with Compositional Nets

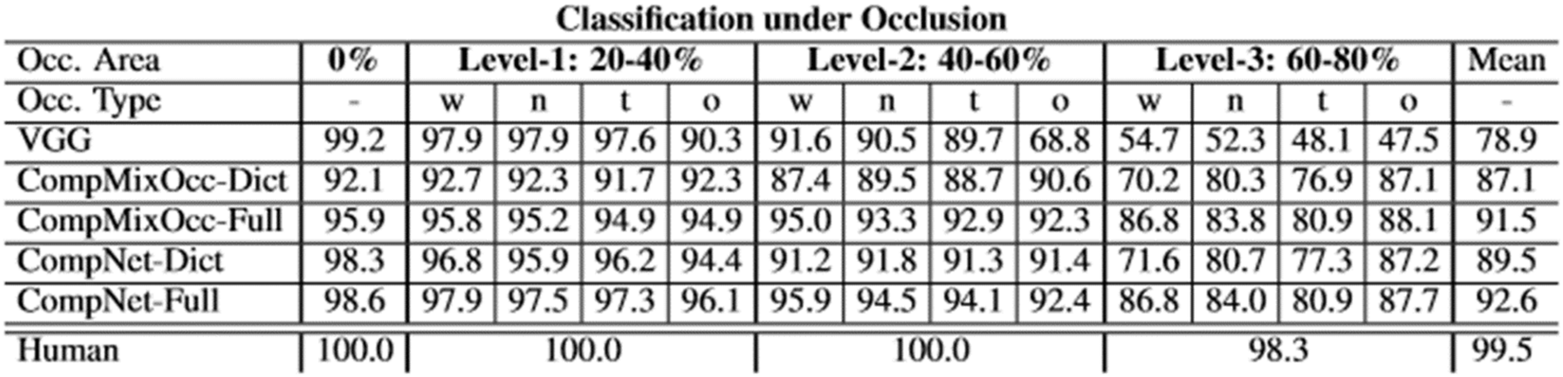
- *Compositional Deep Nets are an alternative architecture which contain explicit representations of parts. Deep Nets have internal representations of parts, but these are implicit and often hard to interpret.*
- The explicit nature of parts in Compositional Deep Nets means that they are more robust to occluders (without training) because they can automatically switch off subregions of the image which are occluded.
- See poster A. Kortylewski et al. Neural Architecture Workshop. 28/Oct. Talk by A. Yuille in Interpreting Machine Learning. Tutorial 27/Oct.

- Note: compositional means "semantic composition". It does not mean "functional composition", which Deep Nets already have.

Contrast Deep Nets with Compositional Nets

- Evaluation: train on unoccluded data, test on occluded data. CompNets outperform Deep Nets as occlusion increases.



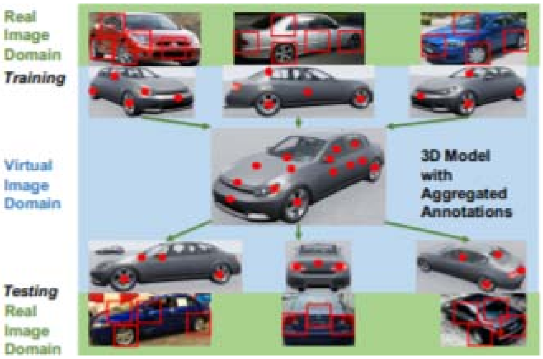


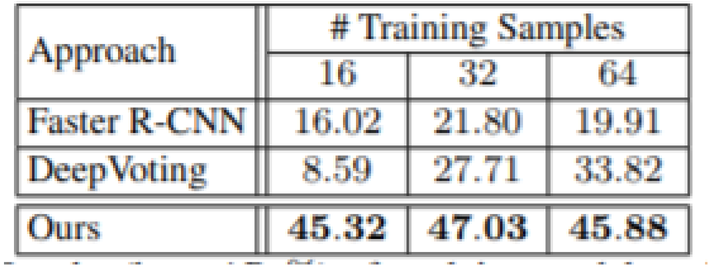
Classification under Occlusion

Occ. Area	0%	Level-1: 20-40%				Level-2: 40-60%				Level-3: 60-80%				Mean
Occ. Type	-	w	n	t	o	w	n	t	o	w	n	t	o	-
VGG	99.2	97.9	97.9	97.6	90.3	91.6	90.5	89.7	68.8	54.7	52.3	48.1	47.5	78.9
CompMixOcc-Dict	92.1	92.7	92.3	91.7	92.3	87.4	89.5	88.7	90.6	70.2	80.3	76.9	87.1	87.1
CompMixOcc-Full	95.9	95.8	95.2	94.9	94.9	95.0	93.3	92.9	92.3	86.8	83.8	80.9	88.1	91.5
CompNet-Dict	98.3	96.8	95.9	96.2	94.4	91.2	91.8	91.3	91.4	71.6	80.7	77.3	87.2	89.5
CompNet-Full	98.6	97.9	97.5	97.3	96.1	95.9	94.5	94.1	92.4	86.8	84.0	80.9	87.7	92.6
Human	100.0	100.0				100.0				98.3				99.5

3D Geometry:

- Representing objects as 3-dimensional models enable us to better recognize them from unusual viewpoints.



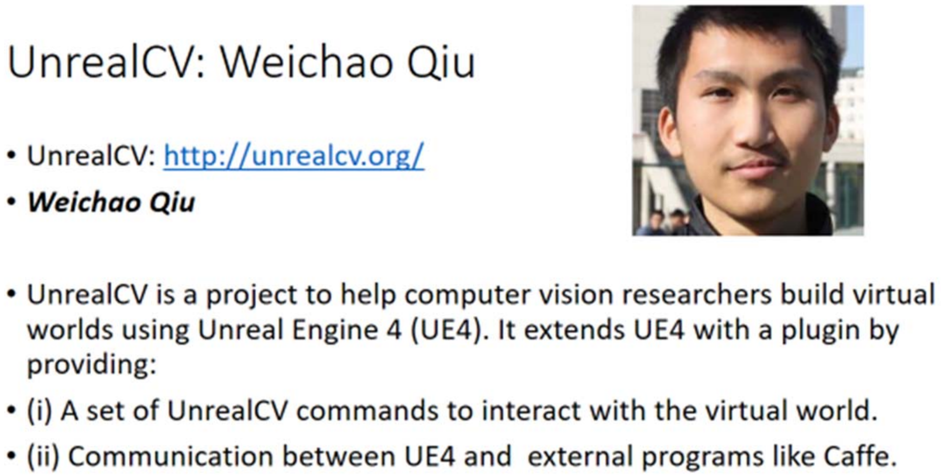


Approach	# Training Samples		
	16	32	64
Faster R-CNN	16.02	21.80	19.91
DeepVoting	8.59	27.71	33.82
Ours	**45.32**	**47.03**	**45.88**

- Yutong Bai et al. Semantic Part Detection via Matching: Learning to Generalize to Novel Viewpoints from Limited Training Data. ICCV. 2019.

Virtual Data: Making Controlled Datasets

- Tools like UnrealCV enable us to generate datasets which have many annotations and which test algorithms systematically.
- This enables us to stress test algorithms in challenging conditions.



UnrealCV: Weichao Qiu

- UnrealCV: http://unrealcv.org/
- ***Weichao Qiu***

- UnrealCV is a project to help computer vision researchers build virtual worlds using Unreal Engine 4 (UE4). It extends UE4 with a plugin by providing:
- (i) A set of UnrealCV commands to interact with the virtual world.
- (ii) Communication between UE4 and external programs like Caffe.

Using Virtual Stimuli to Stress-Test Algorithms.

- Object detection algorithms (W. Qiu & A.L. Yuille. ECCV workshop 2016).
- E.g., Sofa detectors trained on ImageNet may not work on other data.

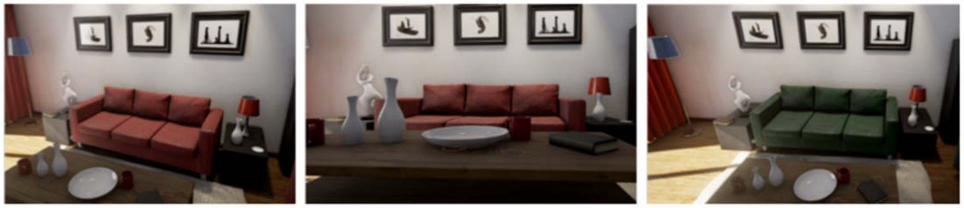
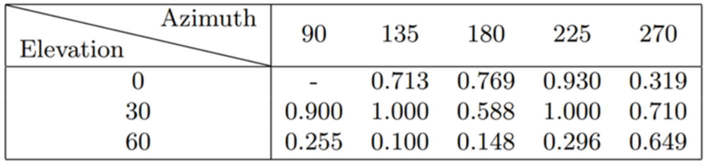


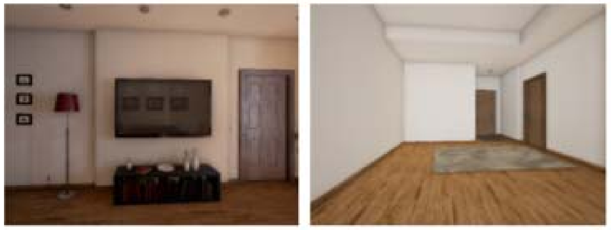
Fig. 4. Images with different camera height and different sofa color.



Elevation \ Azimuth	90	135	180	225	270
0	-	0.713	0.769	0.930	0.319
30	0.900	1.000	0.588	1.000	0.710
60	0.255	0.100	0.148	0.296	0.649

Table 1. The Average Precision (AP) when viewing the sofa from different viewpoints. Observe the AP varies from 0.1 to 1.0 showing the sensitivity to viewpoint. This is perhaps because the biases in the training cause Faster-RCNN to favor specific viewpoints.

- Stress-test binocular stereo. Yi Zhang et al. UnrealStereo. 3DV. 2018.



(a) Specularity

(b) No texture

(c) Disparity jumps

(d) Transparency

Synthetic Data: Activity Recognition

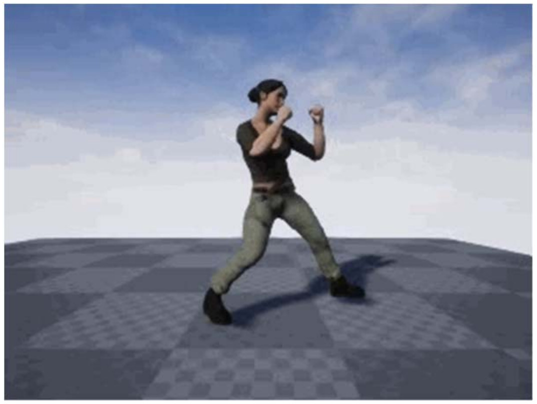
- Activity Recognition is a visual task which is at big risk for combinatorial complexity. Synthetic Data can be used to explore this.
- We render some synthetic videos of humans punching. Train state-of-the art activity recognition methods (TSN and I3D) on these tasks using the USC101 activity dataset.

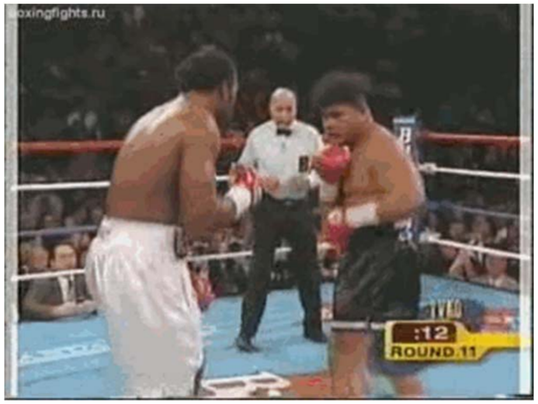
Model	Class Name	Top-1 accuracy	Top-5 accuracy
TSN	Punching	0.00	0.00
I3D	Punching bag	6.25	41.67
I3D	Punching person	6.25	31.25

- Why are the Deep Nets (TSN and I3D) so bad at generalizing to the synthetic data?
- (There are problems for algorithms trained on real to generalize to synthetic, but they are not usually as bad as this).

Why TSN fail to recognize synthetic punching ?

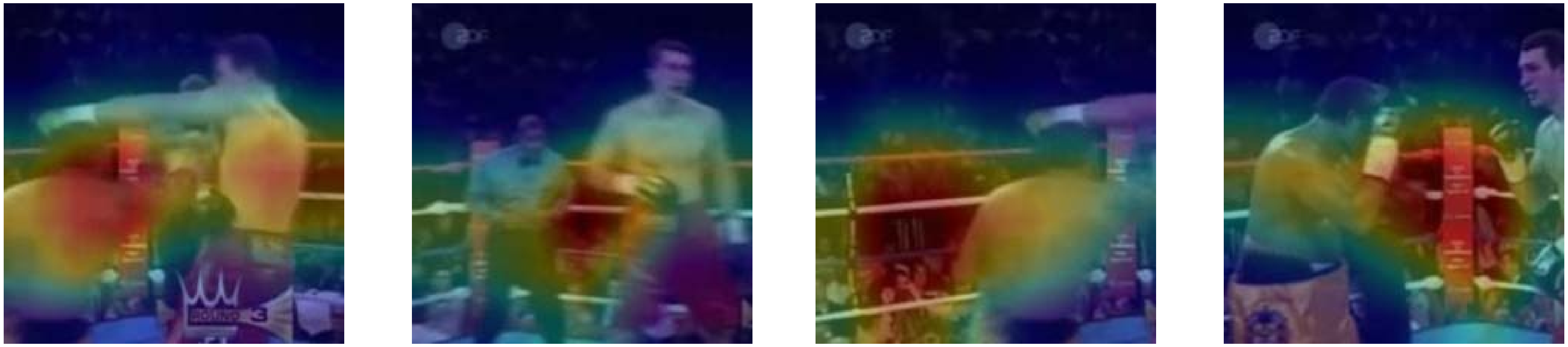
- Conjecture: TSN model trained on UCF101 (right) may have overfit to background and are unable to localize punching action. Synthetic data consists of a single boxer (left).
- Videos from this class in UCF101 are mostly boxing games and punching sandbags.





Can the TSN correctly localize the punching action ?

- Class Activation Maps (CAM) are a standard technique to detect the discriminative image regions used by a CNN to identify a specific activity class.
- CAMs of punching videos from UCF101 test set – detecting ropes.



Summary

- This talk has discussed some of the challenges that Deep Nets faces when dealing with the enormous complexity of the real world.
- We argue that the key challenges arise because the set of all images is infinite and that for some visual tasks the space of images will need to be combinatorially large to be representative of the real world.
- Combinatorial complexity raises challenges for both training and testing algorithms. It is unclear that Deep Nets will be able to overcome them without significant modifications.
- Modifications may include compositionality, 3D geometry, and factorizability.
- Computer Graphics – virtual worlds – can be very helpful for generating controlled challenging adversarial examples for testing algorithms.