

# EECS 4422/5323 Computer Vision

## Image Understanding 3

Calden Wloka

9 October, 2019


# Announcements

- Note: There is recommended reading posted on the schedule for this lecture
- Next week is Reading Week
- Marks from Assignment 1 will likely be posted over Reading Week
- Feedback on your Proposals will take priority

# For the record: Google knows about Steve Buscemi

Try the API

Faces    Objects    Labels    **Web**    Properties    Safe Search



Buscemi.jpg

Show JSON ▾

RESET    NEW FILE

### Web Entities

Steve Buscemi	14.892
Actor	0.5289
Nucky Thompson	0.40575
Boardwalk Empire	0.32505
Fever in the City	0.2489
Artist	0.237
Portrait	0.2309
Poster	0.2309
Photograph	0.2202
Art	0.2002
Christian Weber Photography	0.1957
Kelly Macdonald	0.05004
Dove Cameron	0.03425

Pages with Matched Images

# Outline

- Adversarial Attacks
- Shape and Texture
- Adversarial Training



# Visual Vulnerabilities

As mentioned at the very start of the course, vision in general is intractable, and we must instead rely on heuristically based solutions which can give us the behaviour we want *most* of the time.

## Visual Vulnerabilities

As mentioned at the very start of the course, vision in general is intractable, and we must instead rely on heuristically based solutions which can give us the behaviour we want *most* of the time.

A *robust* system works across a broad range of input or gives answers close to correct even when it fails, whereas a *brittle* system typically works only over a narrow set of input stimuli or provides highly unpredictable results when it fails.

## Visual Vulnerabilities

As mentioned at the very start of the course, vision in general is intractable, and we must instead rely on heuristically based solutions which can give us the behaviour we want *most* of the time.

A *robust* system works across a broad range of input or gives answers close to correct even when it fails, whereas a *brittle* system typically works only over a narrow set of input stimuli or provides highly unpredictable results when it fails.

Even a robust system still eventually fails and ends up drawing incorrect or misleading conclusions about a given stimulus. For humans, we usually call these types of stimuli *optical illusions*.

# Accidental Optical Illusions

Sometimes an optical illusion can happen by accident, and there can be individual variation in perception of the illusion.



Image source: Cecilia Bleasdale

## Optical Illusions by Design

More often, an optical illusion has been specifically designed to exploit a known aspect of how humans process visual stimuli.

For example, even though the three cars shown in the image on the right are pixel-wise identical, the contextual cues of occlusion and parallel line vanishing points induces us to place them at different depth planes and thereby interpret them as vastly different in size.



Image source: Original source unknown

# Optical Illusions for Computers

We can do the same thing for computer vision models. Intentionally designing input to cause mistakes in a model is referred to as an *adversarial attack*.

- Note: Adversarial is **not** the same thing as difficult

# Optical Illusions for Computers

We can do the same thing for computer vision models. Intentionally designing input to cause mistakes in a model is referred to as an *adversarial attack*.

- Note: Adversarial is **not** the same thing as difficult
- Adversarial attacks are not restricted to deep learning, but are often more dramatic in this domain

# Optical Illusions for Computers

We can do the same thing for computer vision models. Intentionally designing input to cause mistakes in a model is referred to as an *adversarial attack*.

- Note: Adversarial is **not** the same thing as difficult
- Adversarial attacks are not restricted to deep learning, but are often more dramatic in this domain
- Adversarial attacks themselves can vary in generality, from robust attacks which affect many networks to highly tailored attacks which specifically target a specific network



# Small Perturbation Attacks

If we have complete access to a network, we can design specific and highly effective tailored attacks.

Take, for instance, this picture of a panda:



$x$

“panda”

57.7% confidence

Image source: [Goodfellow et al., 2015](#)

# Small Perturbation Attacks

If we have complete access to a network, we can design specific and highly effective tailored attacks.

Add a small perturbation which is directed along the gradient of the cost function:


 $x$ 

“panda”

57.7% confidence

+ .007 ×


 $\text{sign}(\nabla_x J(\theta, x, y))$ 

“nematode”

8.2% confidence

Image source: [Goodfellow et al., 2015](#)

# Small Perturbation Attacks

If we have complete access to a network, we can design specific and highly effective tailored attacks.

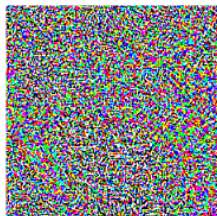
This leads to a highly confident but erroneous new classification.


 $x$ 

“panda”

57.7% confidence

+ .007 ×


 $\text{sign}(\nabla_x J(\theta, x, y))$ 

“nematode”

8.2% confidence

=


 $x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$ 

“gibbon”

99.3 % confidence

Image source: [Goodfellow et al., 2015](#)

# What if we can't access the network error signal?

Even without direct access to the network's inner workings, we can still construct effective adversarial attacks, assuming that we can repeatedly probe the behaviour of the network. The more information we have (e.g. prediction confidence or some other correlate to the gradient), the more subtle we can typically make the attack.

The form of these attacks can take many forms, including:

- Single pixel attacks

# What if we can't access the network error signal?

Even without direct access to the network's inner workings, we can still construct effective adversarial attacks, assuming that we can repeatedly probe the behaviour of the network. The more information we have (e.g. prediction confidence or some other correlate to the gradient), the more subtle we can typically make the attack.

The form of these attacks can take many forms, including:

- Single pixel attacks
- Local area attacks

# What if we can't access the network error signal?

Even without direct access to the network's inner workings, we can still construct effective adversarial attacks, assuming that we can repeatedly probe the behaviour of the network. The more information we have (e.g. prediction confidence or some other correlate to the gradient), the more subtle we can typically make the attack.

The form of these attacks can take many forms, including:

- Single pixel attacks
- Local area attacks
- Noise attacks (e.g. Gaussian blur, salt-and-pepper noise)

## What if we can't access the network error signal?

Even without direct access to the network's inner workings, we can still construct effective adversarial attacks, assuming that we can repeatedly probe the behaviour of the network. The more information we have (e.g. prediction confidence or some other correlate to the gradient), the more subtle we can typically make the attack.

The form of these attacks can take many forms, including:

- Single pixel attacks
- Local area attacks
- Noise attacks (e.g. Gaussian blur, salt-and-pepper noise)
- "Boundary" attack

# Single pixel attack

- Turn a single pixel either to white or black
- Not every pixel works - only “critical” pixels will work
- These pixels can be searched for exhaustively, but a random subset is often sufficient (Naroytska & Kasiviswanathan, 2016)
- The shift in the decision often is more subtle than with other adversarial attacks (e.g. car to truck)



# Single pixel attack examples



(a) original  
Car



(b) perturbed  
Truck



(c) original  
Cat



(d) perturbed  
Dog

Image source: [Naroytska & Kasiviswanathan, 2016](#)

## Local area attacks

Opening up the perturbation to include multiple pixels makes the attack stronger. Once a critical pixel is identified, it is typically more effective to search nearby pixels for adversarial effect. This is easier if we can access the confidence of the network.



Ruffed Grouse

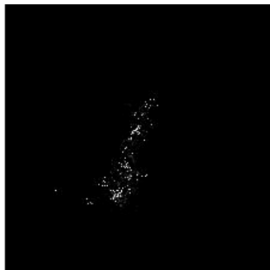


Image Difference



Frill-necked Lizard

Image source: [Naroytska & Kasiviswanathan, 2016](#)

## Boundary attacks

The idea of the boundary attack is to start from an adversarial point, and then randomly explore along the decision boundary to get as close as possible (or desired) to the appearance of the correct class while still being classified as the adversarial class. This can either be a specific desired output (targeted attack), or can be any incorrect output (untargeted attack).

## Boundary attack example

One clever way of initializing the network is to start with an example image from a target class, and then walk toward an example image from the correct class.



Image source: [Brendel et al., 2018](#)

## Visible but well localized

The previously discussed adversarial attacks concentrated on attacks which were minimally different from a correctly classified image (often imperceptibly so to humans), and this typically resulted in attacks concentrated on pixels which are semantically understandable by a human to be important.

## Visible but well localized

The previously discussed adversarial attacks concentrated on attacks which were minimally different from a correctly classified image (often imperceptibly so to humans), and this typically resulted in attacks concentrated on pixels which are semantically understandable by a human to be important.

Karmon *et al.* (2018) instead devised a different style of attack - allow the attack to be visible, but restrict it to a small, localized patch which cannot cover the primary image content (*i.e.* the object). They called this style of attack LaVAN (Localized and Visible Adversarial Noise).

## Visible but well localized

The previously discussed adversarial attacks concentrated on attacks which were minimally different from a correctly classified image (often imperceptibly so to humans), and this typically resulted in attacks concentrated on pixels which are semantically understandable by a human to be important.

Karmon *et al.* (2018) instead devised a different style of attack - allow the attack to be visible, but restrict it to a small, localized patch which cannot cover the primary image content (*i.e.* the object). They called this style of attack LaVAN (Localized and Visible Adversarial Noise).

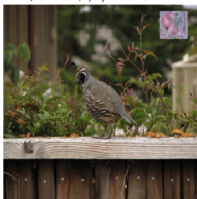
With access to network probability or confidence, these patches can be rather straightforwardly trained via gradient descent.

# LaVAN Examples

Original Image  
Quail: 99.81%, Spiny Lobster: 0.00%



Noised Image  
Quail: 0.63%, Spiny Lobster: 94.64%



Quail (99.8%) → Spiny Lobster (94.6%)

Original Image  
Conch: 99.35%, Go-Kart: 0.00%



Noised Image  
Conch: 0.17%, Go-Kart: 98.09%



Conch (99.4%) → Go-Kart (98.1%)

Original Image  
Lifeboat: 89.20%, Scotch Terrier: 0.00%

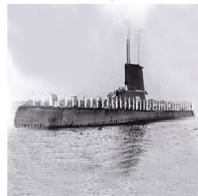


Noised Image  
Lifeboat: 0.03%, Scotch Terrier: 99.77%



Lifeboat (89.2%) → Scotch Terrier (99.8%)

Original Image  
Submarine: 98.87%, Bonnet: 0.00%



Noised Image  
Submarine: 0.24%, Bonnet: 99.05%



Submarine (98.9%) → Bonnet (99.1%)

Image source: [Karmon et al., 2018](#)





# Transferable LaVAN

The LaVAN examples shown in the previous slide are highly brittle; they can be generated approximately 75% of the time for a given image, patch location, and target class, but placing the same patch in another image or even shifting it slightly within the original image will typically result in a collapse of the adversarial effect.

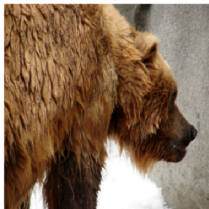
# Transferable LaVAN

The LaVAN examples shown in the previous slide are highly brittle; they can be generated approximately 75% of the time for a given image, patch location, and target class, but placing the same patch in another image or even shifting it slightly within the original image will typically result in a collapse of the adversarial effect.

By modifying the generation procedure to apply the patch to a different randomly selected base image at each training step, the patches can be made much more general and robust.

# Transferable LaVAN Examples

Original Image  
Brown Bear: 92.47%, Baseball: 0.00%



Noised Image  
Brown Bear: 0.16%, Baseball: 96.40%



Brown Bear (92.5%) → **Baseball** (96.4%)

Image source: [Karmon et al., 2018](#)

Original Image  
Gong: 89.30%, Jaguar: 0.00%



Noised Image  
Gong: 0.53%, Jaguar: 96.36%



Gong (89.3%) → **Jaguar** (96.4%)

Image source: [Karmon et al., 2018](#)

# Adversarial Objects

Athalye *et al.* (2018) generalize the concept of adversarial generation across a given set of transformations  $T$ , and demonstrate that they can generate surprisingly robust adversarial examples over a wide range of transformations, including the ability to print 3D objects which are frequently misclassified.

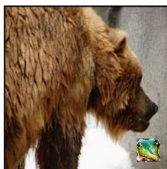


■ classified as turtle      ■ classified as rifle  
■ classified as other

Image source: [Athalye et al., 2018](#)

# Human Perceptions of Robust Examples

Zhou & Firestone (2019) examined human perceptions of these more robust adversarial examples (transferable LaVAN and the method of Athalye *et al.*), and found that even if humans predominantly picked the original object class as the correct label for an image, if forced to predict what they thought the network would pick, they could reliably predict the adversarial class.



**Tree frog**



**Jaguar**



**Power drill**



**Jigsaw puzzle**

Example stimuli used by Zhou & Firestone.

Image source: [Zhou & Firestone, 2019](#)

# Why do these robust adversarial attacks work?

The reason for the efficacy of both LaVAN and Athalye *et al.*'s method appears to be due to the heavy weight given to local texture over more global views of object shape.



(a) Texture image

81.4% **Indian elephant**  
 10.3% indri  
 8.2% black swan



(b) Content image

71.1% **tabby cat**  
 17.3% grey fox  
 3.3% Siamese cat



(c) Texture-shape cue conflict

63.9% **Indian elephant**  
 26.4% indri  
 9.6% black swan

Image source: [Geirhos \*et al.\*, 2018](#)

# How does this compare to human vision?

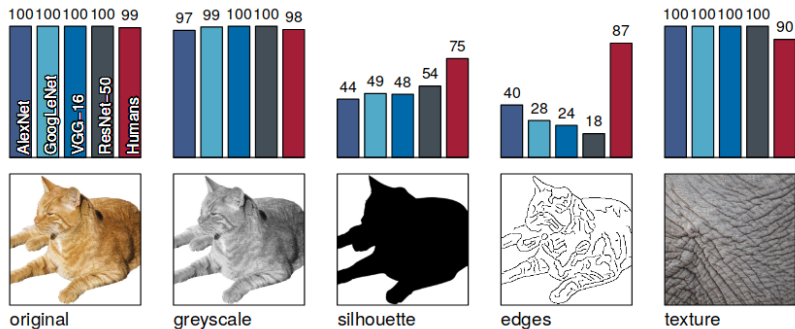


Image source: [Geirhos et al., 2018](#)

Humans appear to give much greater weight to overall shape than the networks (note: humans were shown the stimulus for only 300ms; performance would likely be higher given more viewing time).

# How does this compare to human vision?

We can see these differences are extremely pronounced when performance is averaged over shape-texture conflicting images.

- Red circles indicate average human response for a given category
- Blue/grey shapes indicate the behaviour of neural networks

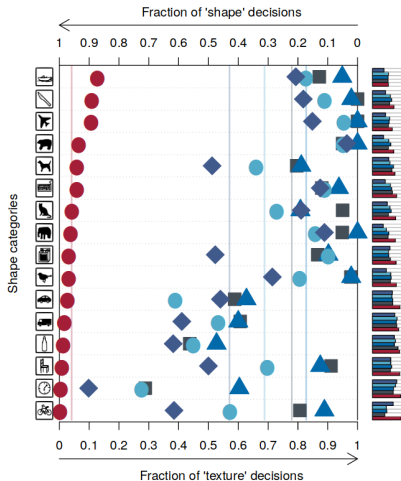
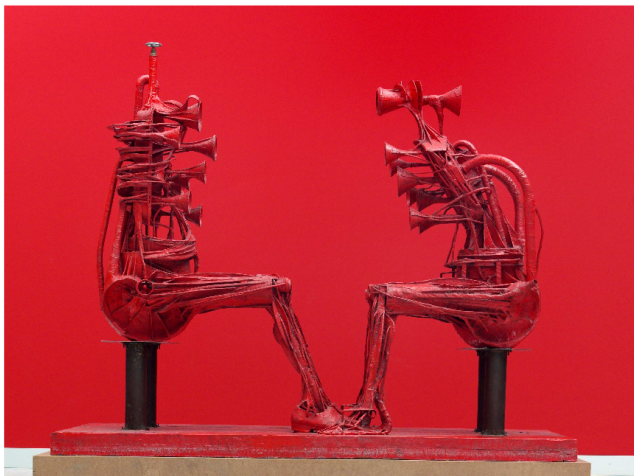


Image source: [Geirhos et al., 2018](#)



# Inspiration from Art



Source: Image from Dickinson, 2009; Sculpture by Nepraš

# Abstract Interpretation

We can see that this sculpture does not literally contain a person, but we can nevertheless see that it evokes the form of a seated interlocutor.

This is particularly impressive given that even local contours do not match particularly well with the necessary shape abstraction.

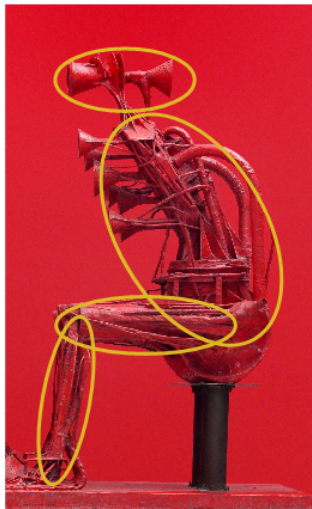


Image Source: Dickinson, 2009

## A Quick Aside: Style Transfer

Gatys *et al.* (2015) introduced a technique for blending images using a CNN in what they dubbed “style transfer”. The intent is to combine the *content* of one image with the *style* of another.



# Leveraging Art



Image source: [Geirhos et al., 2018](#)

Geirhos *et al.* sought to shift network reliance from texture to shape by leveraging style transfer during training. By rendering training images with different artistic styles, the normal local texture cues present in an object would be disrupted.

# Learning with style transfer successfully shifts network behaviour

- Red circles indicate average human response for a given category
- Orange squares indicate ResNet-50 trained on style transfer
- Grey squares indicate standard ResNet-50

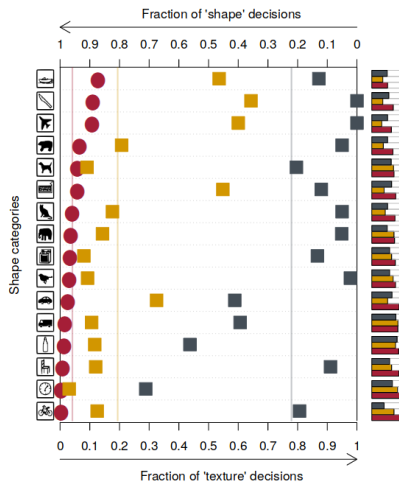
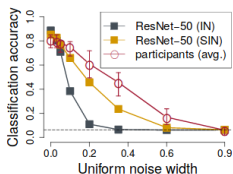


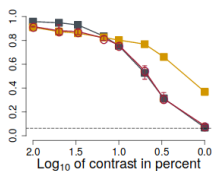
Image source: [Geirhos et al., 2018](#)

# Reliance on Shape Increases Robustness to Noise

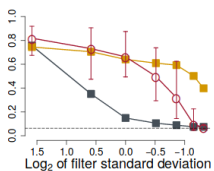
An interesting additional result found by Geirhos *et al.* is that a network forced to be more responsive to shape becomes more robust to different types of noise attacks.



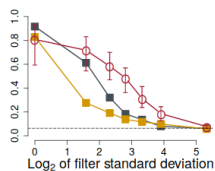
(a) Uniform noise



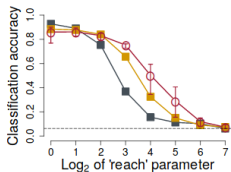
(b) Contrast



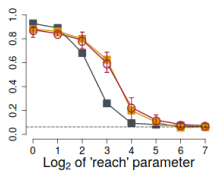
(c) High-pass



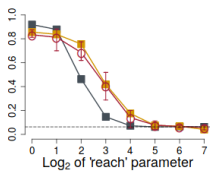
(d) Low-pass



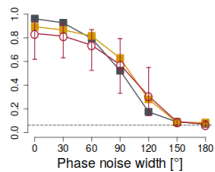
(e) Eidolon I



(f) Eidolon II



(g) Eidolon III



(h) Phase noise

## Does this solve the issue?

Unfortunately, Geirhos *et al.* didn't test on robust adversarial attacks, but one can imagine their method would improve performance over these styles of attack.

Nevertheless, there are a number of open problems which remain:

- Are these networks learning the link between 2D shape and 3D structure?

## Does this solve the issue?

Unfortunately, Geirhos *et al.* didn't test on robust adversarial attacks, but one can imagine their method would improve performance over these styles of attack.

Nevertheless, there are a number of open problems which remain:

- Are these networks learning the link between 2D shape and 3D structure?
- Can we design a system which efficiently extracts shape cues, rather than requiring a large increase in augmented training data?



## Does this solve the issue?

Unfortunately, Geirhos *et al.* didn't test on robust adversarial attacks, but one can imagine their method would improve performance over these styles of attack.

Nevertheless, there are a number of open problems which remain:

- Are these networks learning the link between 2D shape and 3D structure?
- Can we design a system which efficiently extracts shape cues, rather than requiring a large increase in augmented training data?
- How can we more flexibly allow for multiple simultaneous percepts to be integrated (e.g. recognizing the sculpture as representative of two people, but also as a painted collection of objects stuck together)?

# Leveraging Adversarial Examples: GANs

Goodfellow *et al.*, 2014 first proposed the innovative idea of *Generative Adversarial Networks* (GANs).

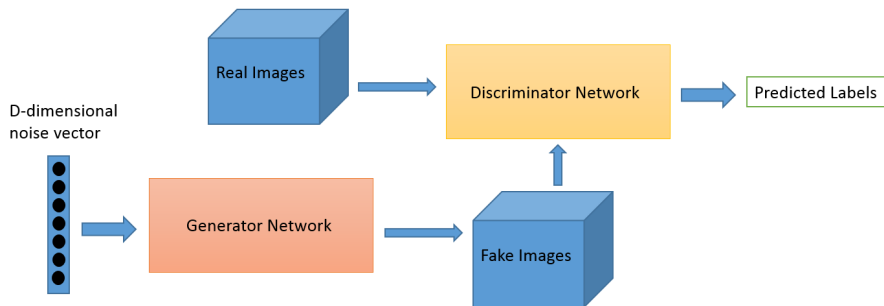


Image source: Skymind AI

# Benefits of GANs

- You are generating training data, reducing (not eliminating!) the burden of supervision

# Benefits of GANs

- You are generating training data, reducing (not eliminating!) the burden of supervision
- You get a generator at the end, and modern generators *can be really good*

# Drawbacks of GANs

- Can be difficult to train correctly - networks might mutually wander into highly non-optimal territory

# Drawbacks of GANs

- Can be difficult to train correctly - networks might mutually wander into highly non-optimal territory
- Even more prone to overfitting and learning biases than standard deep learning approaches

# Foolbox: An Adversarial Toolbox

If you are interested in adversarial attacks, there is an interesting project from the Bethge lab called [Foolbox](#) which provides an API implementation for applying many different kinds of adversarial attacks.

Original documentation is in an [arXiv paper](#), and a continually maintained set of documentation can be found [here](#).