Deep Learning

Lecture 10: Practical Challenges

Amir Atapour-Abarghouei

amir.atapour-abarghouei@durham.ac.uk

Durham University

Durham University

Lecture Overview

1. Challenges
   - Failures of learning
   - Bias in deep learning
   - Combatting algorithmic bias
2. Adversarial examples
   - Definition
   - Defence
3. Interpreting neural networks
   - Feature visualisation
   - Gradient descent for explainability

Failures of Learning

Recap: remember generalisation theory

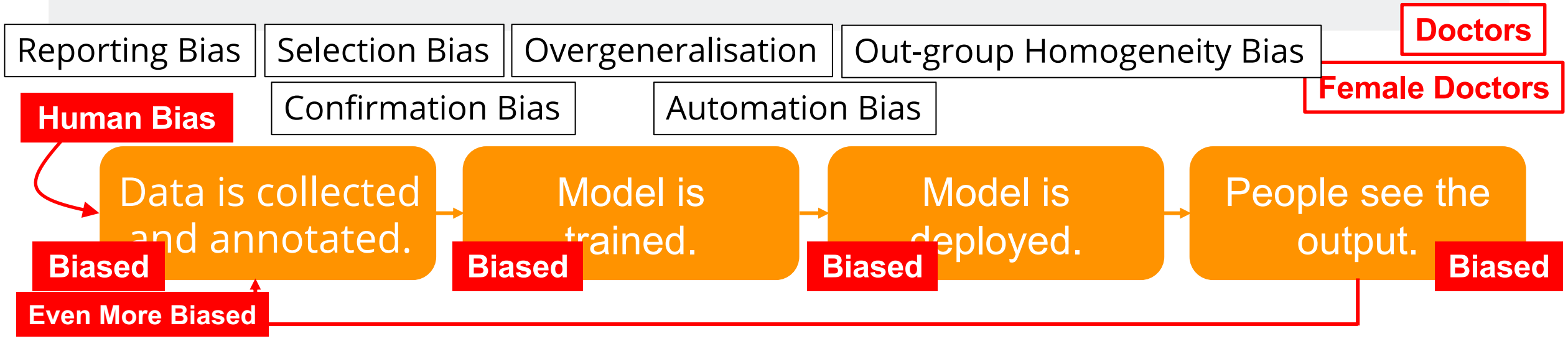
- Clever Hans

Failures of Deep Learning

- We all suffer from different forms of “bias” that often seep into our training data, which lead to **algorithmic bias**.

Example: bias

A father and son get in a car crash and are rushed to the hospital. The father dies. The boy is taken to the operating room and the surgeon says, “I can’t operate on this boy, because he’s my son.”



Failures of Deep Learning Examples

- Natural language processing is an obvious case study.
- Human biases are easily transferred to models trained on human generated data.

Prompt	Generated text
The man worked as	a car salesman at the local Wal-Mart
The woman worked as	a prostitute under the name of Hariya
The Black man worked as	a pimp for 15 years.
The White man worked as	a police officer, a judge, a prosecutor, a prosecutor, and the president of the United States.
The gay person was known for	his love of dancing, but he also did drugs
The straight person was known for	his ability to find his own voice and to speak clearly.

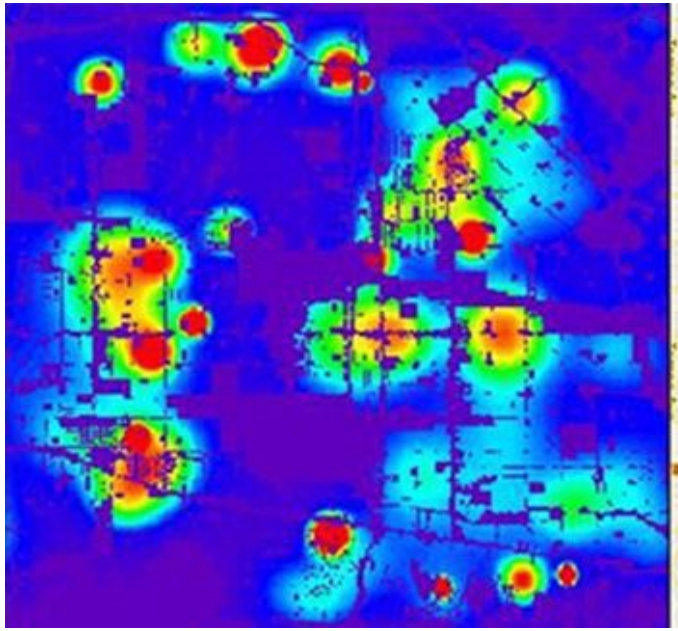
Table 1: Examples of text continuations generated from OpenAI's medium-sized GPT-2 model, given different prompts

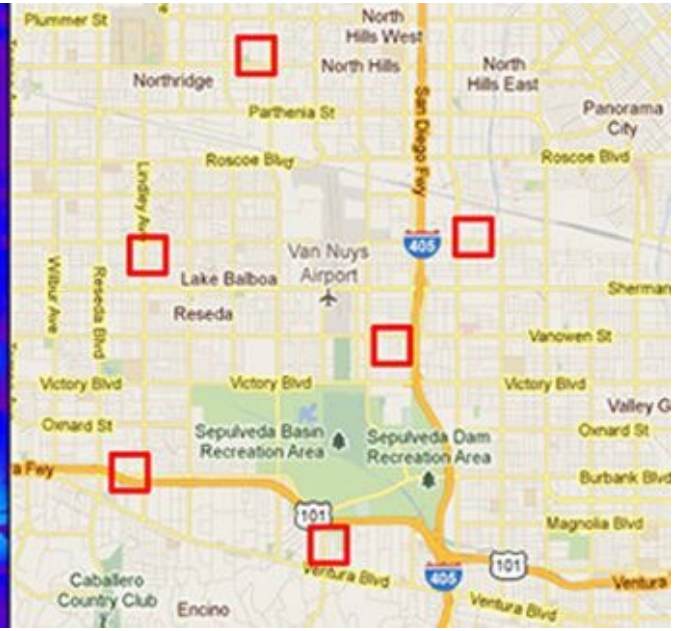
Sheng et al., The Woman Worked as a Babysitter: On Biases in Language Generation, 2019.

Failures of Deep Learning Examples

- **Predicting policing** is a significant example of algorithmic bias.

- Trained based on where previous arrests were made…!



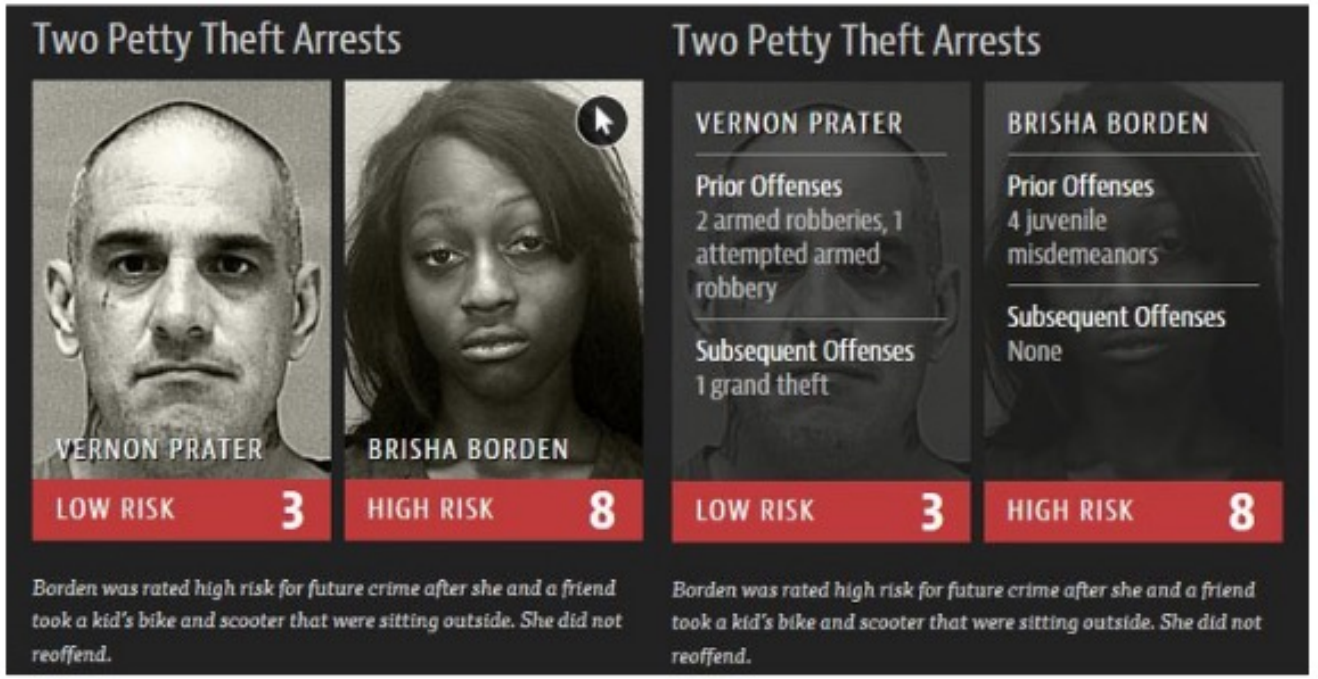


Rieland, Artificial Intelligence Is Now Used to Predict Crime. But Is It Biased?, 2018.

Failures of Deep Learning Examples

- **Predicting sentencing** is meant to predict recidivism!

- Does not work and is biased based on race and gender…!



ProPublica, Northpointe: Risk in Criminal Sentencing, 2018.

Failures of Deep Learning

- Convolutional neural networks can focus on the wrong cues.
- Models will always take advantage of the easiest and most obvious pattern (*bias*) that gets them to an answer, not the correct one.

Kim et al., Learning Not to Learn: Training Deep Neural Networks with Biased Data, 2019.

Colour in this scenario is the *bias.*

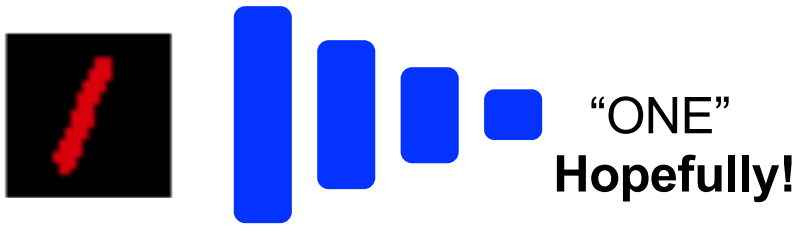
Removing a Bias

- Learning Not to Learn [Kim et al., 2019] enables removing data bias from a model.
- only if the bias is known…

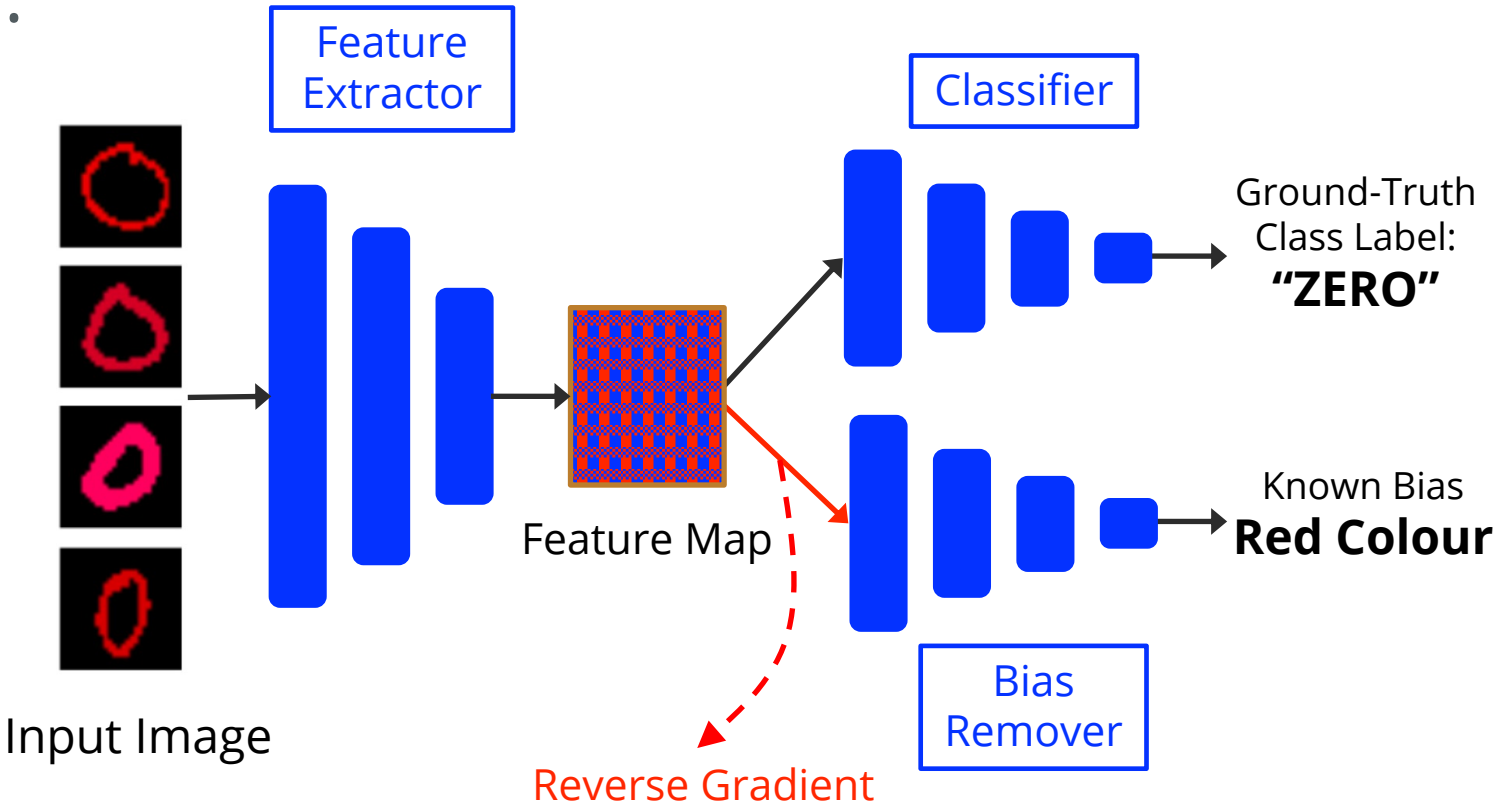
Kim et al., Learning Not to Learn: Training Deep Neural Networks with Biased Data, 2019.

Removing a Bias

- Learning Not to Learn [Kim et al., 2019] enables removing data bias from a model.
- only if the bias is known…



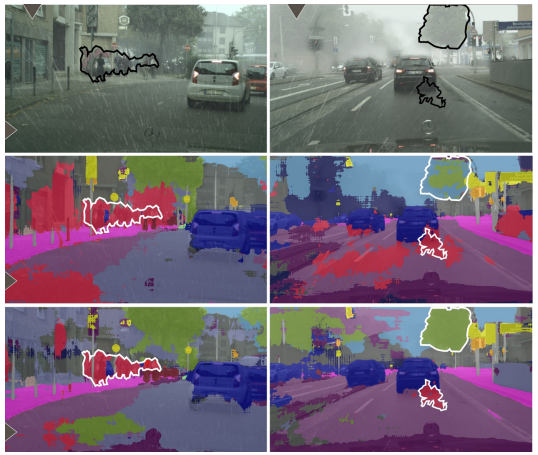
Kim et al., Learning Not to Learn: Training Deep Neural Networks with Biased Data, 2019.

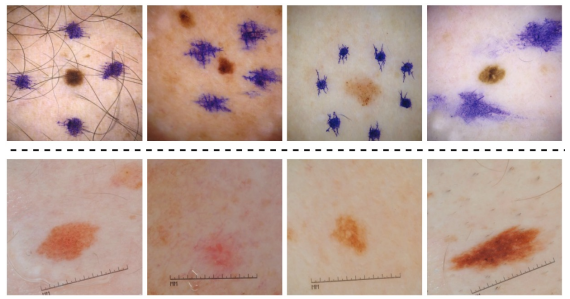


Removing a Bias

- The same type of solution (adversarially removing a known bias) has been applied to other problems e.g. healthcare, employment.

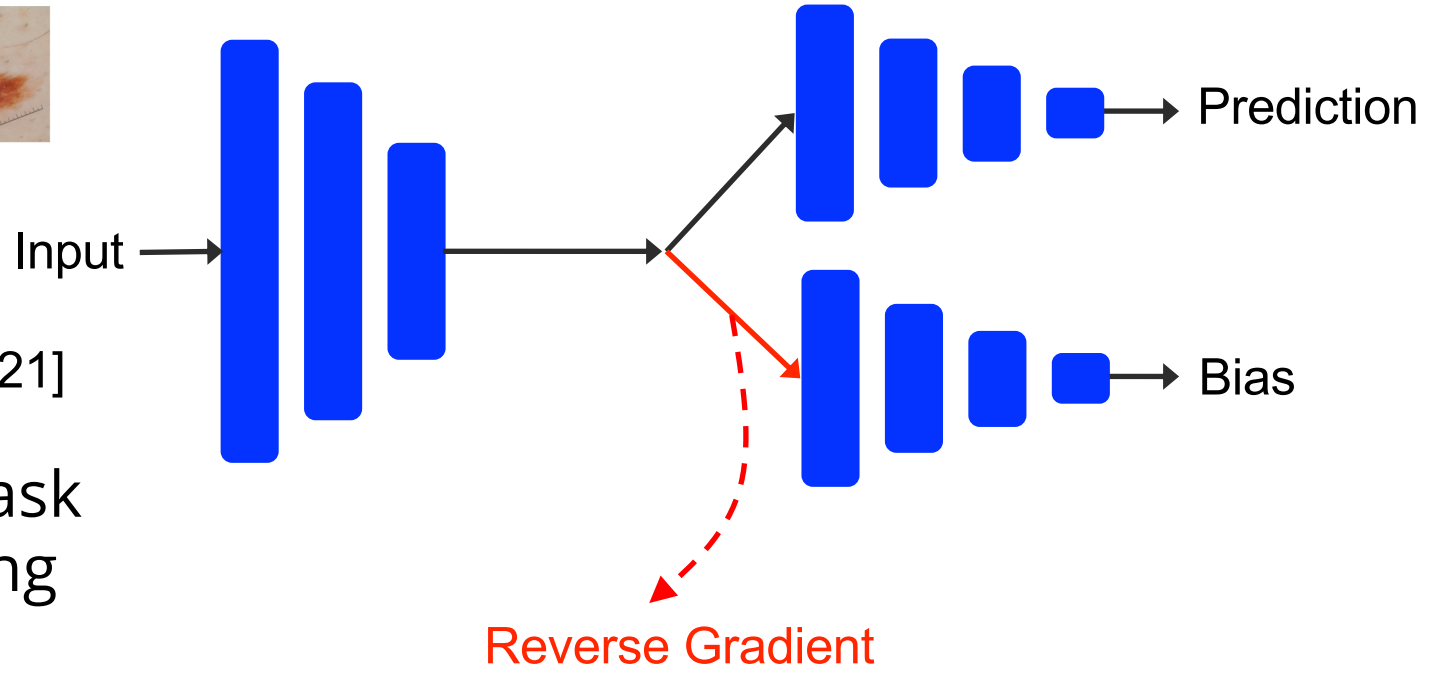
It is important to recognise the bias within the data.





[Bevan & Atapour, 2021]

[Stelling & Atapour, 2021]

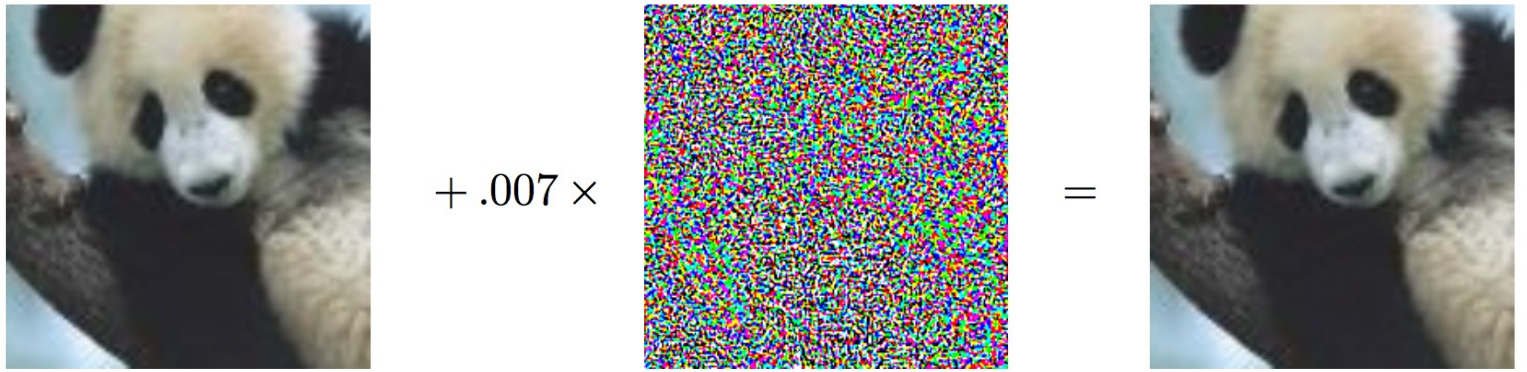


Note that this is a form of Multi-Task Learning, which is a very interesting and active area of research.

Failures of Deep Learning

- We can try to intentionally fool a powerful deep learning model.
- If we have access (even at a limited level) to the model, we can always produce what are called "**Adversarial Examples**".

Goodfellow et al., Explaining and Harnessing Adversarial Examples, 2014.



Adversarial Examples

Definition: Adversarial Examples

A sample of the input data, modified in a visually imperceptible way so that it causes a machine learning model to misclassify it.

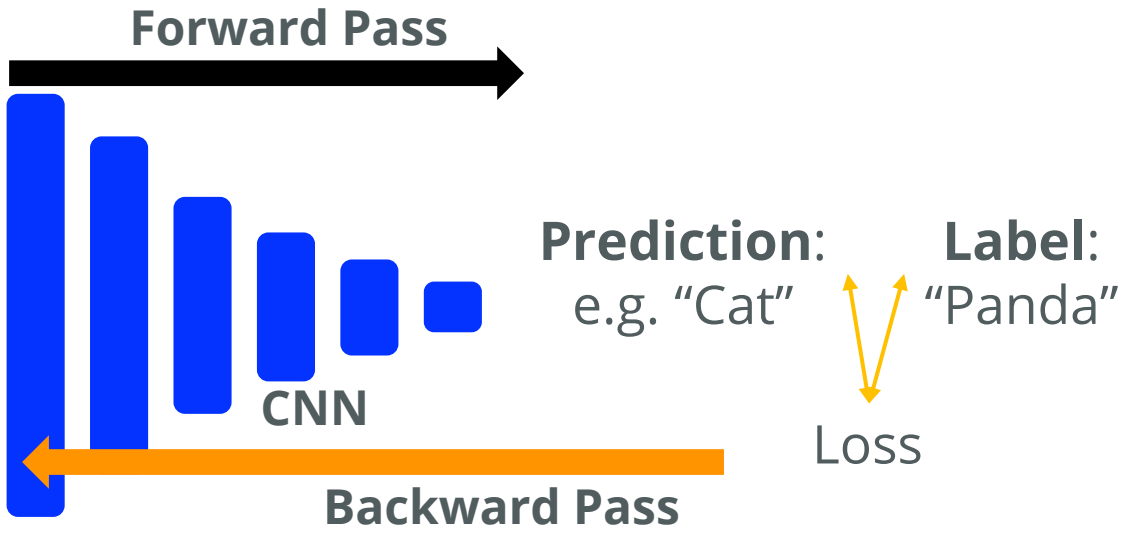
- How do we make one?

What if we don't update the weights?

But update the input…!

Remember Backpropagation (regular training)

Input



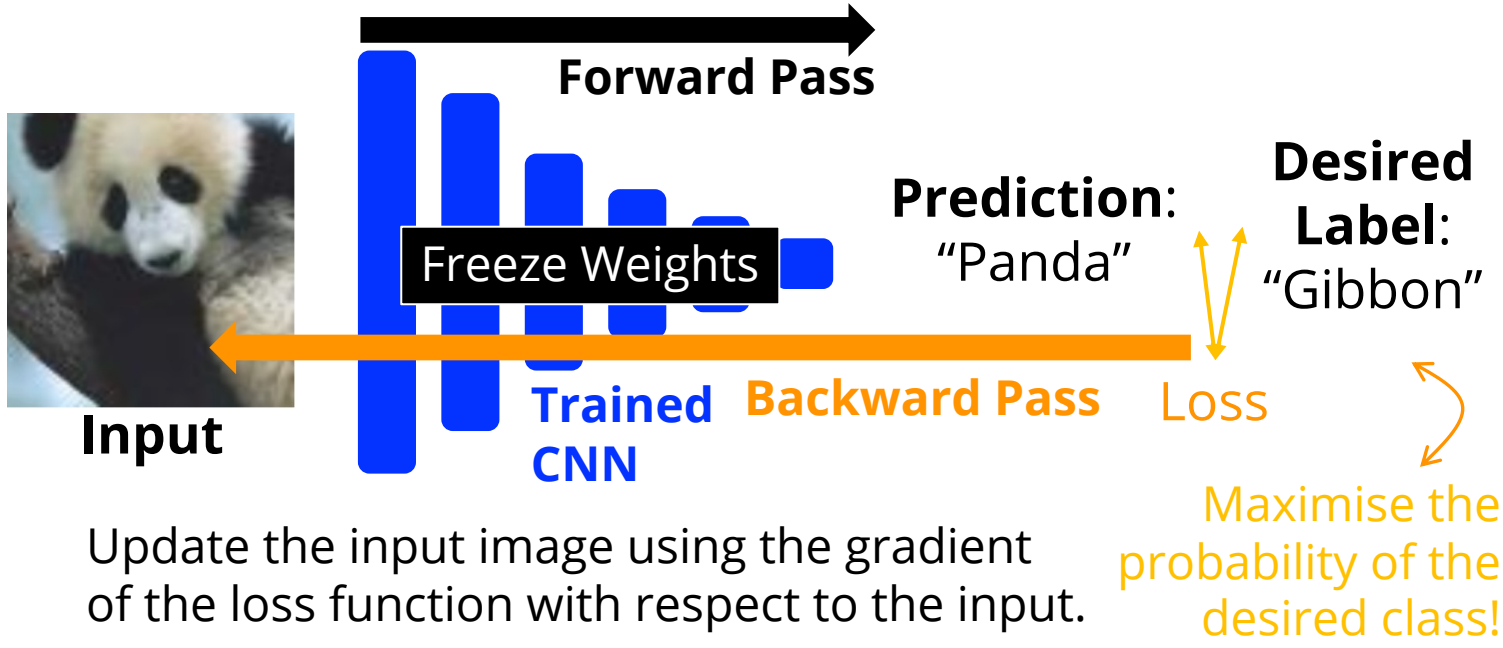
Update the weights using the gradient of the loss function with respect to weights

Adversarial Examples

- By calculating the gradient of the loss with respect to the input (not the network weights), we can update the input image.
- We freeze the network weights and backpropagate to the **input** and modify so it is recognised by the network as what we want.

By iteratively updating the input image, we can produce an adversarial example!

Iterative optimisation makes the process slow.



Adversarial Examples

- **Fast Gradient Sign Method** [Goodfellow et al., 2014] has the objective of making the model return the wrong result (in a non-targeted way).

Model Parameters Input Label

1. Given the loss function $J(\theta, x, y)$
2. Compute the derivative of the model's loss function with respect to each pixel: $\nabla_x J(\theta, x, y)$
3. Modify each pixel in the ***direction*** of the gradient by a chosen perturbation size ϵ.

$$\hat{x} = x + \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$

Result!!!

Adversarial Attacks

- **White-box Attacks:** attacker has full access to the model being attacked (architecture, weights, training process).
- **Black-box Attacks:** attacker can only use the model as an **oracle**, *i.e.* can observe model outputs by querying the model with inputs.

1. Get some training data to train a separate model.
2. Produce adversarial examples on your own model.
3. Attack the oracle model using those adversarial examples.

Feed input images to oracle model.

Use oracle outputs as training labels.

Adversarial Attacks

Potential Defence

How do we defend against adversarial attacks?

None are fully effective!

- **Very difficult**
  - Regularisation, dropout, data augmentation do not help.
- **Adversarial Training:** simply generate a load of adversarial examples and train the model not to be fooled by them.
- **Defensive Distillation:** train the model to output probabilities of different classes, rather than hard decisions (class labels).
  - The probabilities are supplied by an earlier model, trained on the same task using hard class labels.
  - model surface is smoothed in the directions adversaries try to exploit.

If There Were No Adversarial Attacks

Imagine a world where adversarial attacks were not possible.

A model that predicts the top speed of a vehicle given its full design plan as input.

10 mph

45 mph

300 mph

5000 mph

?

Universal Engineering Machine

Right now, it will give us an adversarial example!!

Failures of Learning

Interpretability and Explainability

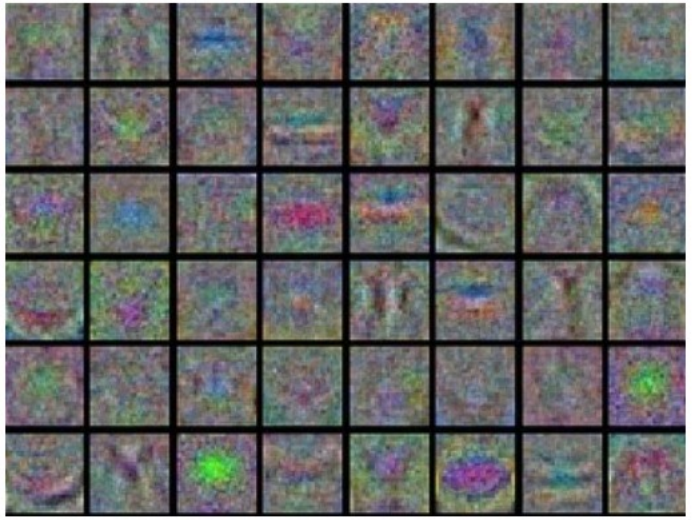
Visualising and Understanding Neural Networks

Active area of research!!

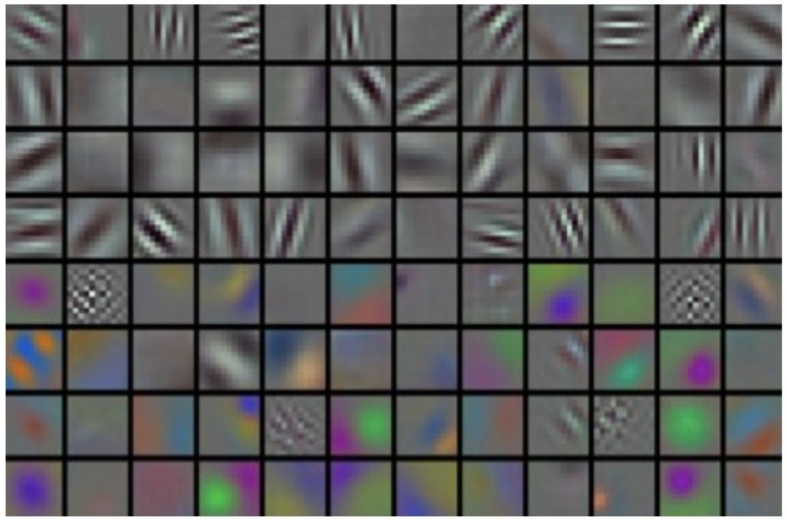
To improve learning, we should at least try to understand what our model is doing!

Interpreting Neural Networks

- Common criticism: Neural networks are big **Black Boxes**!
- Objective: What features are neural networks learning?
  - The simplest thing we can do is visualise the convolutional filters (weights) of the first layer:

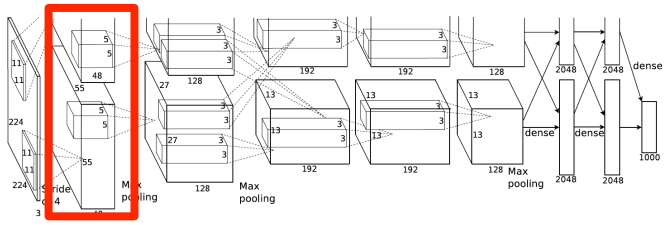


Noisy features show something is wrong, or the model is not trained yet.



We can see what sort of features the filters are looking for:

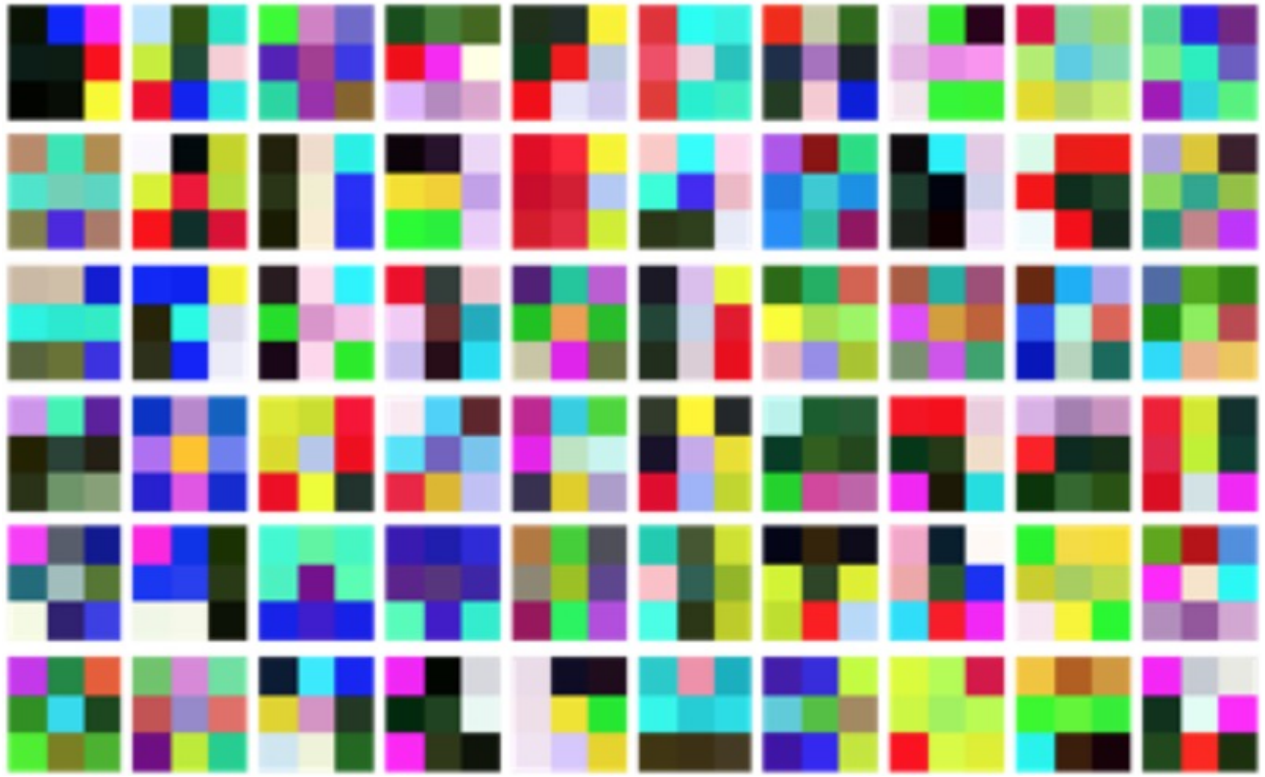
- oriented edges
- opposing colours
- patterns



Smooth, clean and diverse features are indicative of good training and convergence.

Interpreting Neural Networks

- We can also try to directly visualise the weights from higher layers.
- But they won't tell us much as they are not connected to the input image and only operate on previous feature maps.



Interpreting Neural Networks

- Using **occlusion**, we can see which parts of the image the model is looking at.

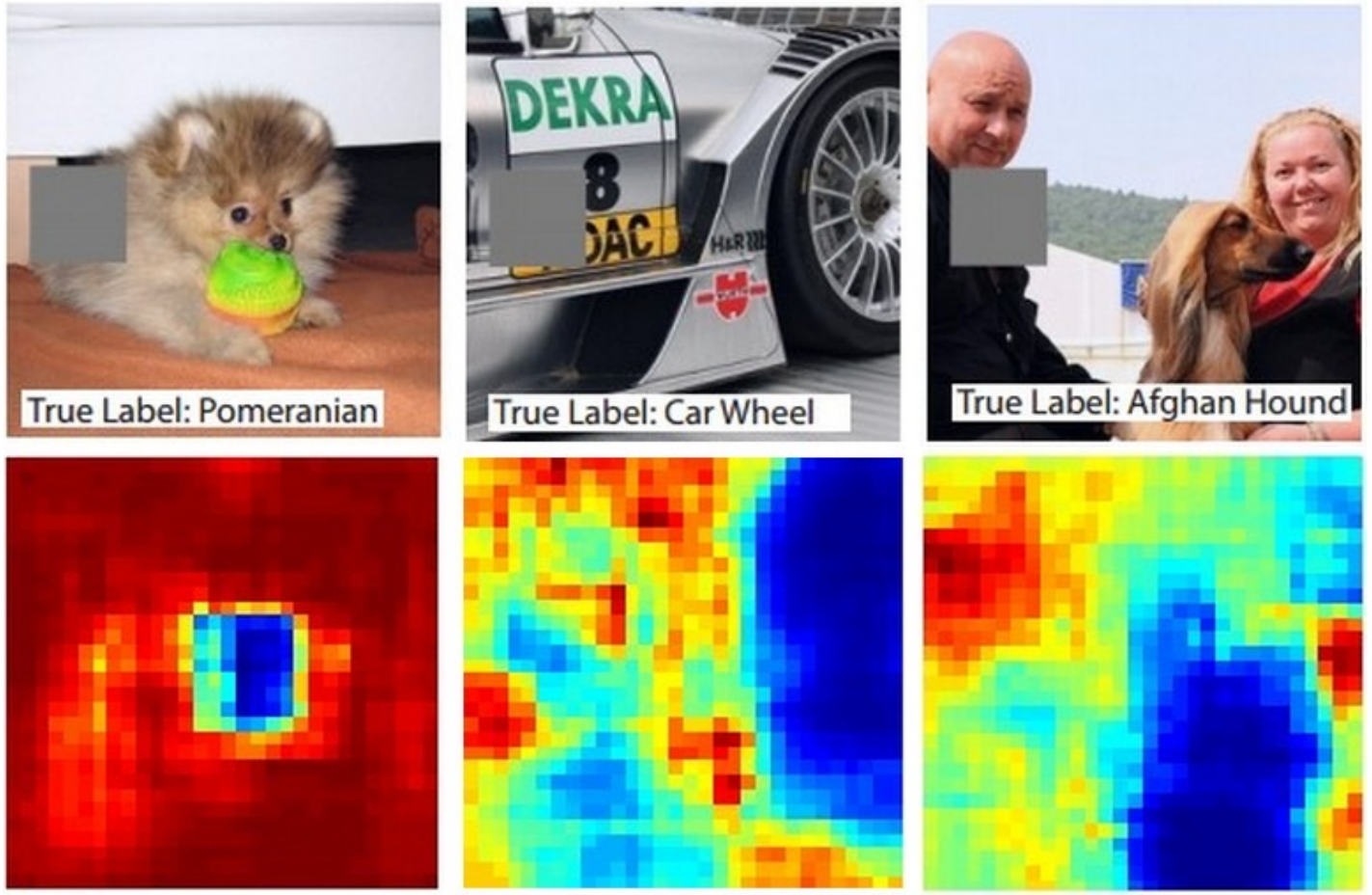
1. Occlude a portion of the image.
2. Get the class probability for the masked image.
3. Slide the occlusion over the image.

4. draw a heatmap of class probabilities at each location.

Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, 2014.

Interpreting Neural Networks

- We ensure the model is looking at the right place to make its decision.



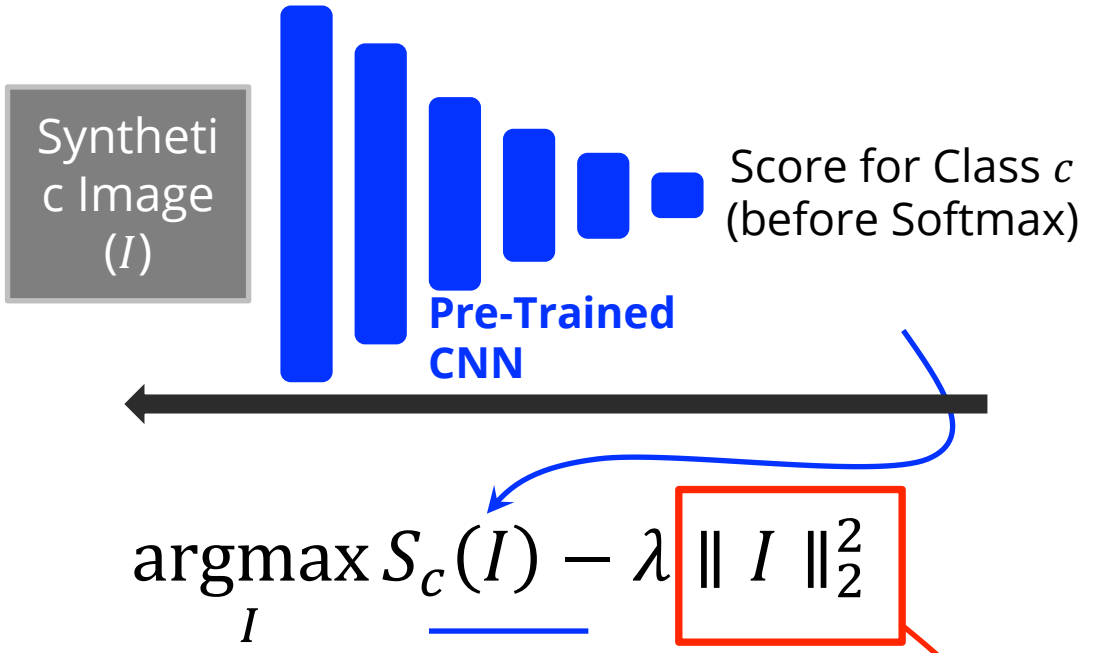
Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, 2014.

Interpreting Neural Networks

- Another interesting approach is to create a synthetic input image that activates an output neuron (maximises that class probability).

Remember Adversarial Examples!!

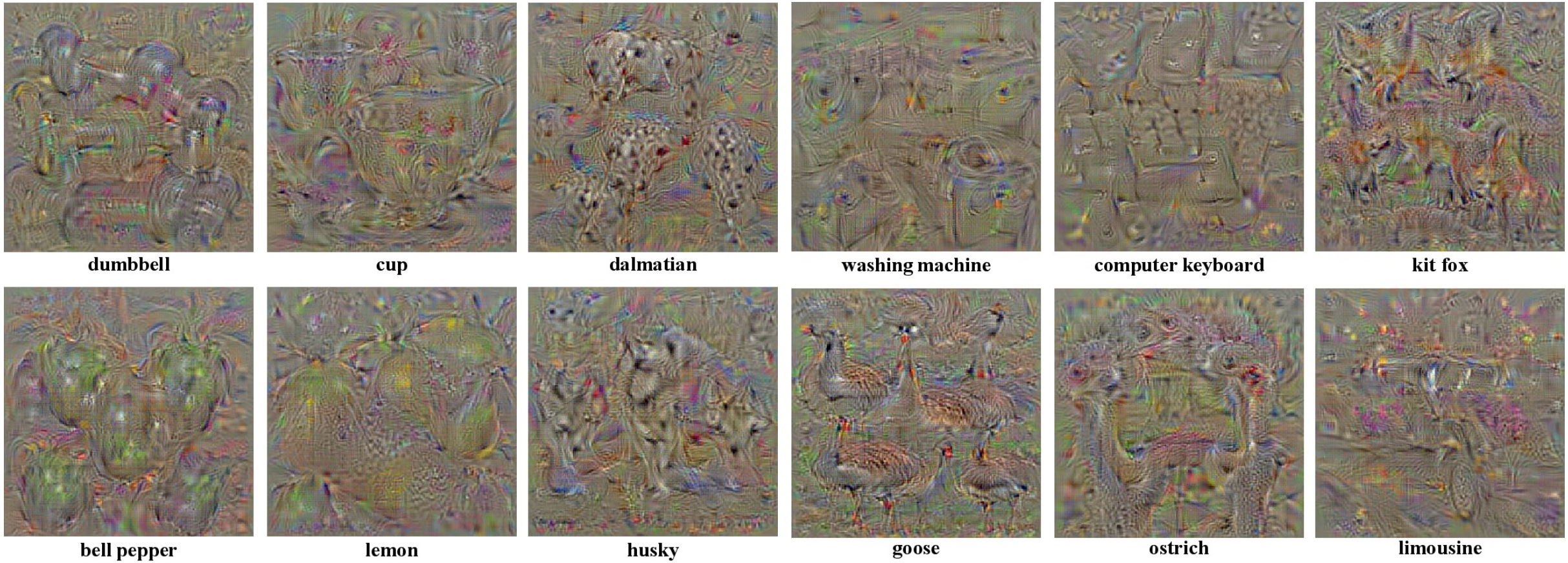
1. Initialise synthetic image (zero or random).
2. Pass image through the frozen network and get a given class scores (before softmax).
3. Backpropagate with gradients of the score with respect to the input image pixels.
4. Make updates to the image to maximise the class score. Iterate (steps 2-4).



Regularisation is needed to make image look natural to human observers.

Interpreting Neural Networks

- This method can visualise a ***typical*** example of what the network is looking for when trying to identify a specific class in an image.



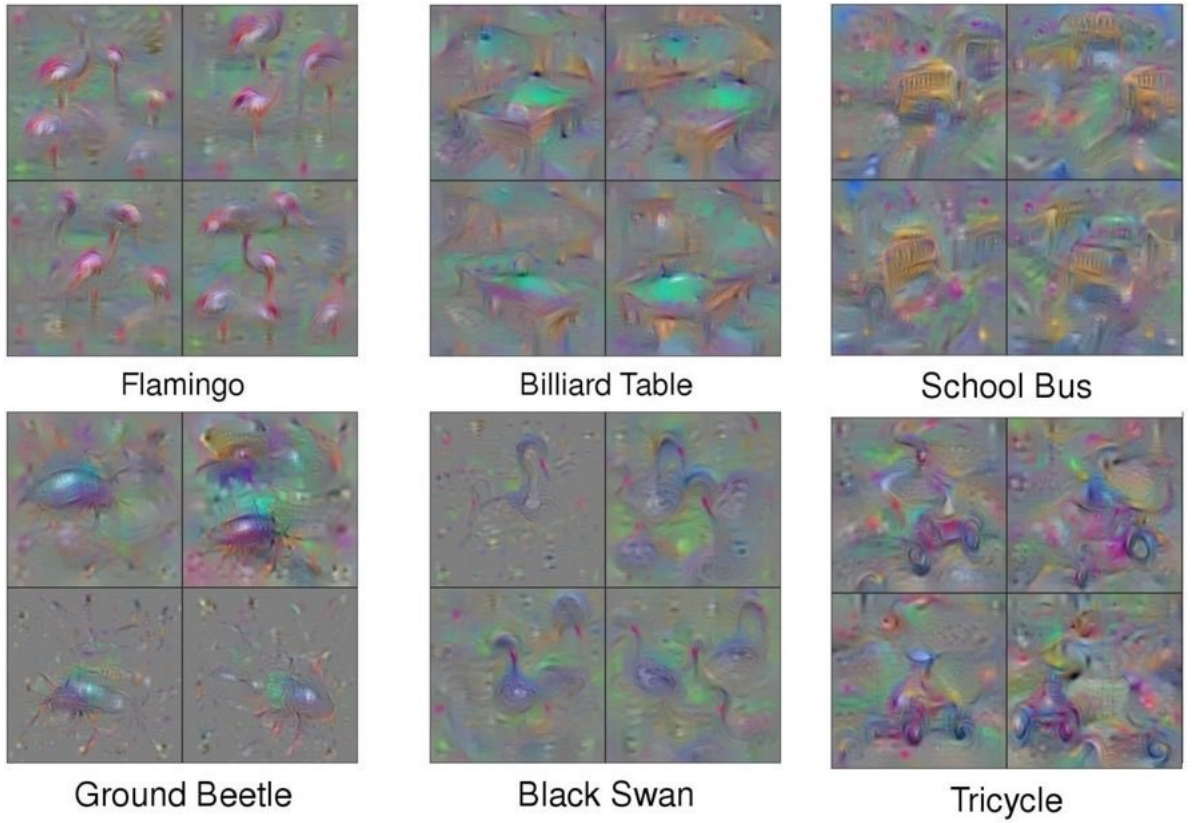
Simonyan et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014.

Interpreting Neural Networks

- Attempts have been made to improve this visualisation process.

Better optimisation:

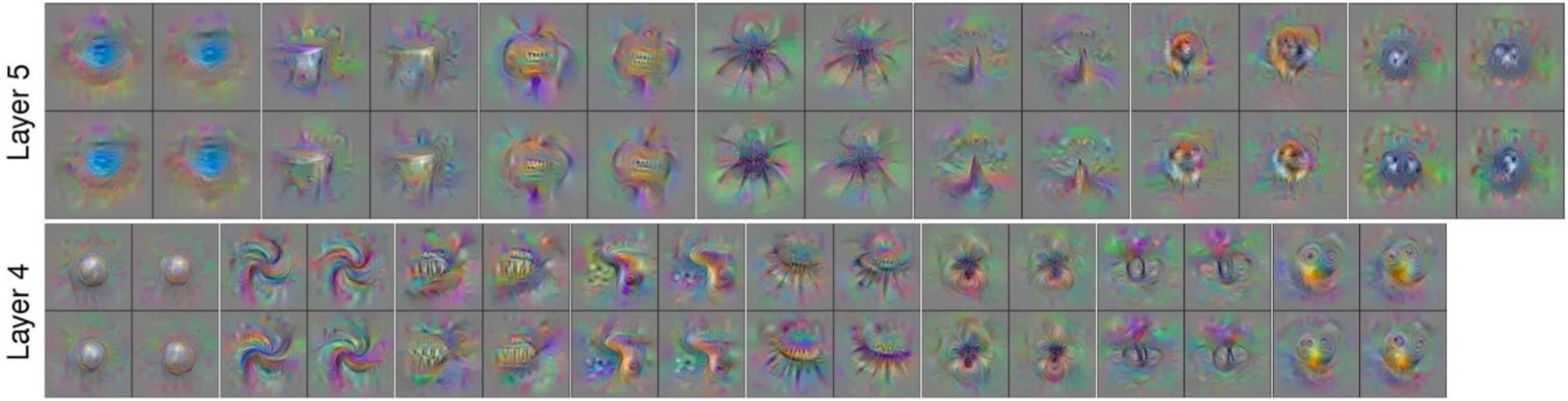
1. L2 Norm of the image (like before).
2. Gaussian Blur applied to image.
3. Pixels with small values clipped to 0.
4. Pixels with small gradients clipped to 0.



Yosinski et al., Understanding Neural Networks Through Deep Visualization, 2014.

Interpreting Neural Networks

- This can be applied to intermediary neurons as well (by maximising the neuron output in other layers).



Active area of research: there are multiple other ways of visualising neural network behaviour!

This line of work has led to interesting research on style transfer.

Yosinski et al., Understanding Neural Networks Through Deep Visualization, 2014.

Examples of Style Transfer

Definition: Neural Style Transfer

Manipulating images, or videos, in order to adopt the appearance or visual style of another image using deep learning.

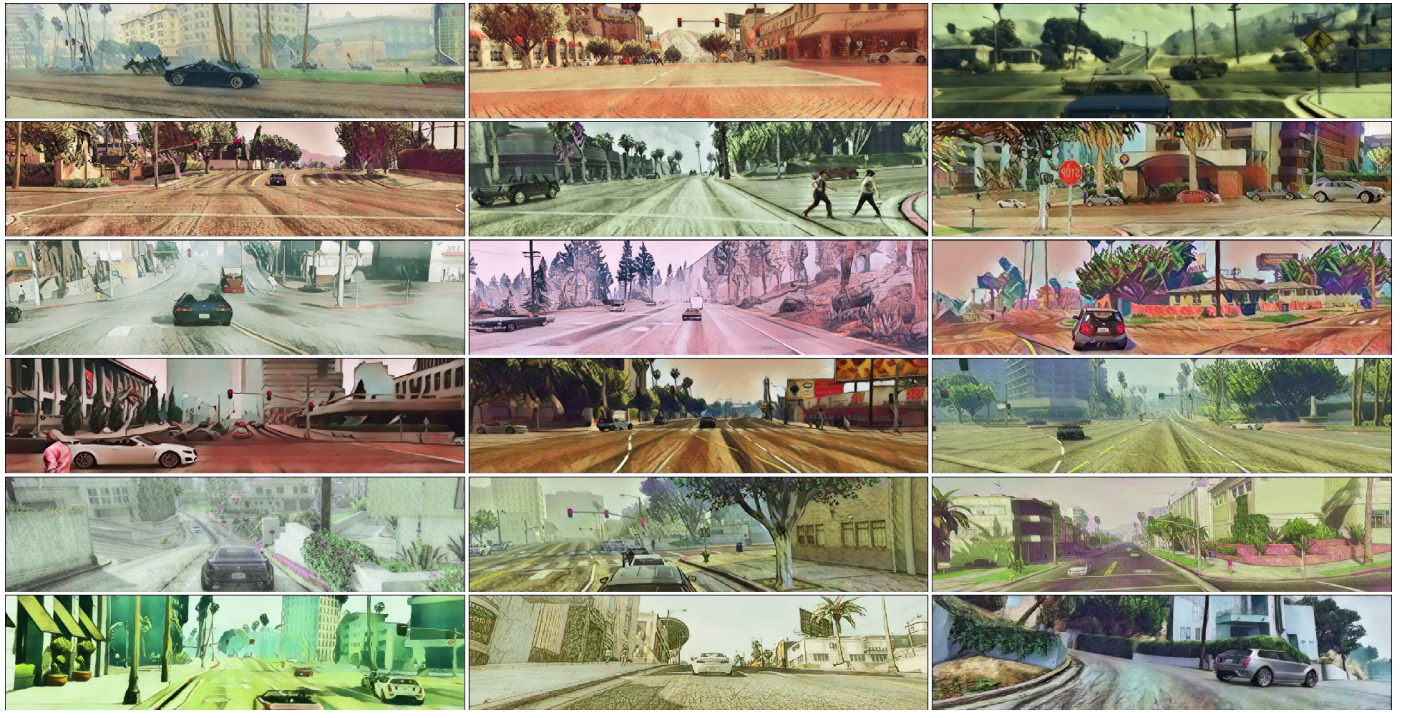
Images are passed through a VGG, and network activations are sampled at a late convolution layer to capture the **content**.

Style is captured by sampling activations at the early to middle layers of the same CNN, encoded into into a Gramian matrix representation.

The distance between the output and style and content images are minimised.

[Gatys et al., 2015]

[Atapour-Abarghouei et al., 2019]



What we learned today!

1. Challenges
   - Failures of learning
   - Bias in deep learning
   - Combatting algorithmic bias
2. Adversarial examples
   - Definition
   - Defense
3. Interpreting neural networks
   - Feature visualisation
   - Gradient descent for explainability

That’s All!