Understanding CNN Behaviour through **Neural Manifolds**

Thesis Presentation by **Niranjan Rajesh**
Thesis Advised by **Professor Debayan Gupta**

Table of **contents**

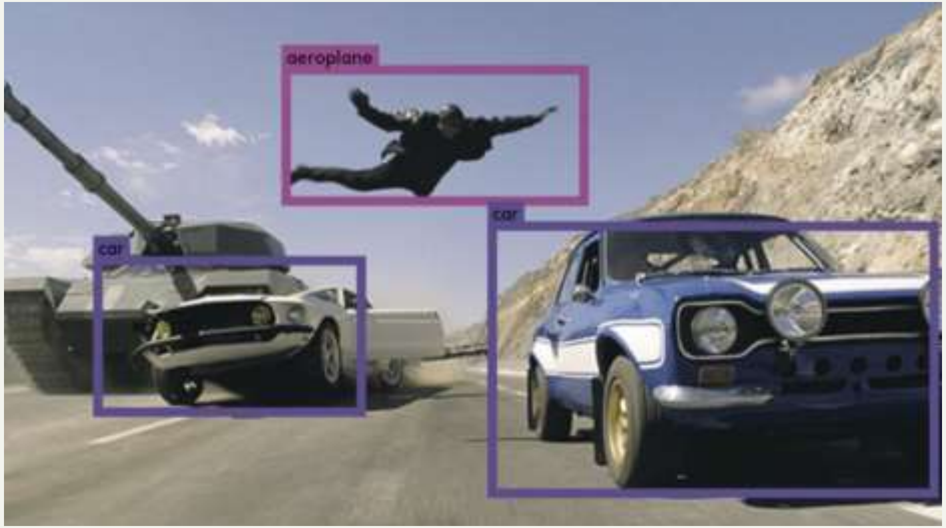
01

Motivation

Solving Visual Intelligence.

State of **Computer Vision** Today

- Object Recognition - Classification, Detection, Segmentation
- Generative Vision - Stable Diffusion



State of **Computer Vision** Today

- Object Recognition - Classification, Detection, Segmentation
- Generative Vision - Stable Diffusion

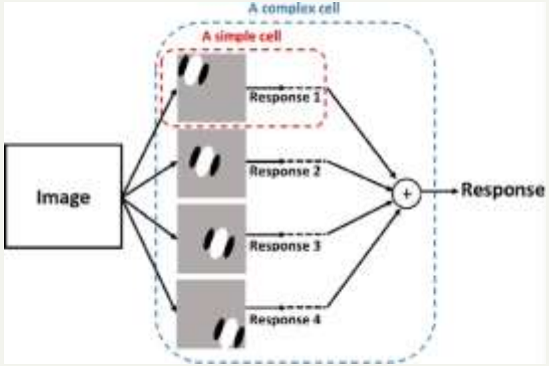
Backbones - Very Deep Neural Networks:

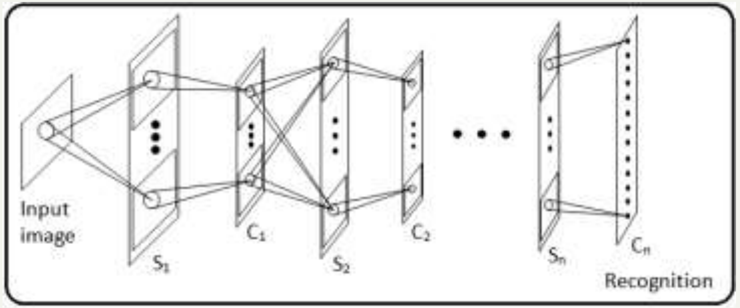
Convolutional Neural Networks (CNNs)
1980

Vision Transformers (ViTs)
2020

Convolutional Neural Networks (CNNs)

- Hubel and Wiesel (1959) - **Simple and Complex cells**
- Fukushima (1980) - **Neocognitron**
- LeCun (1989) - **LeNet**
- AlexNet, ResNets, VGGs, DenseNets ....?

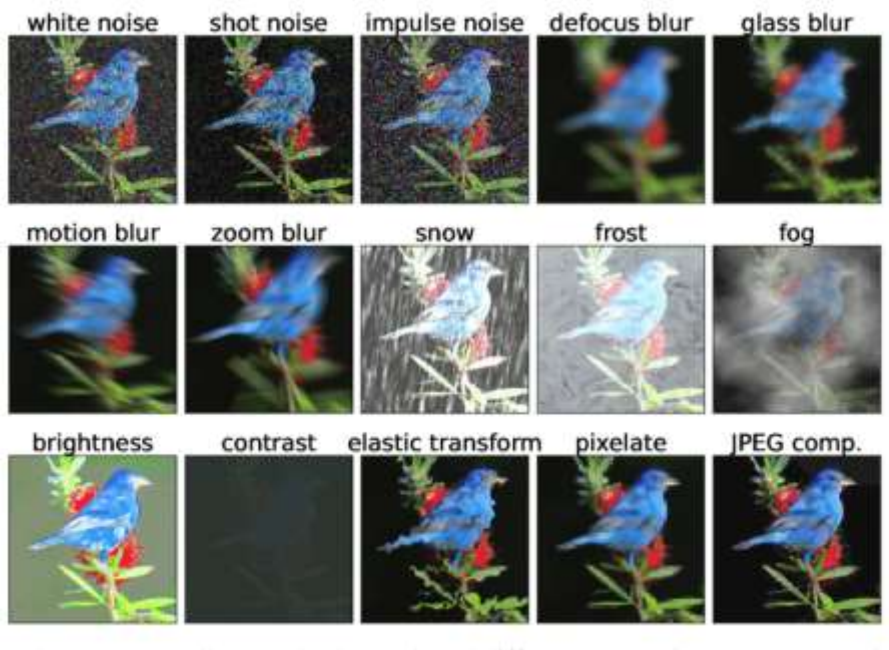




CNNs and **Concerning** Behaviour

Lack of Robustness

- Out-of-Distribution Data
- Corrupted Data
- Adversarially Perturbed Data

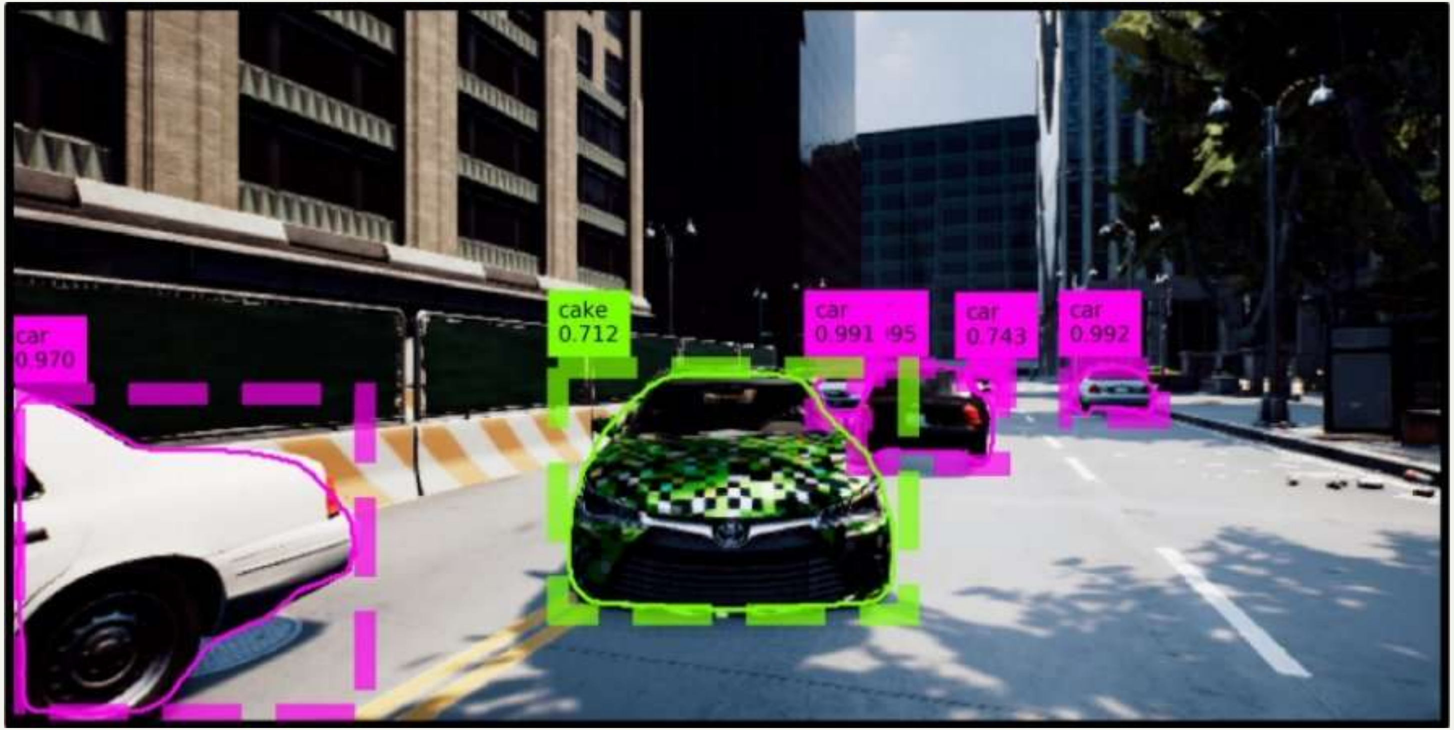


CNNs and **Concerning** Behaviour

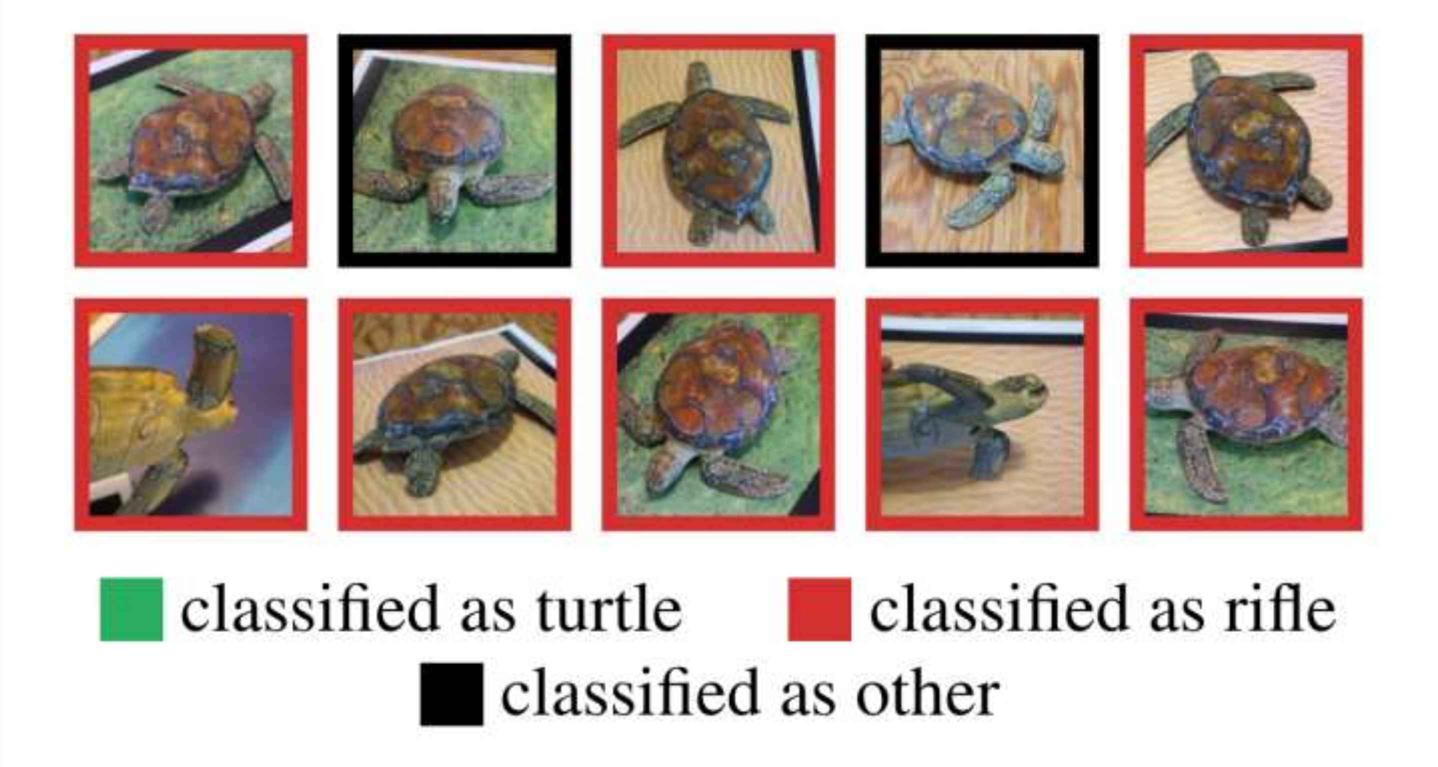
Lack of Robustness

- Out-of-Distribution Data
- Corrupted Data
- **Adversarially Perturbed Data**

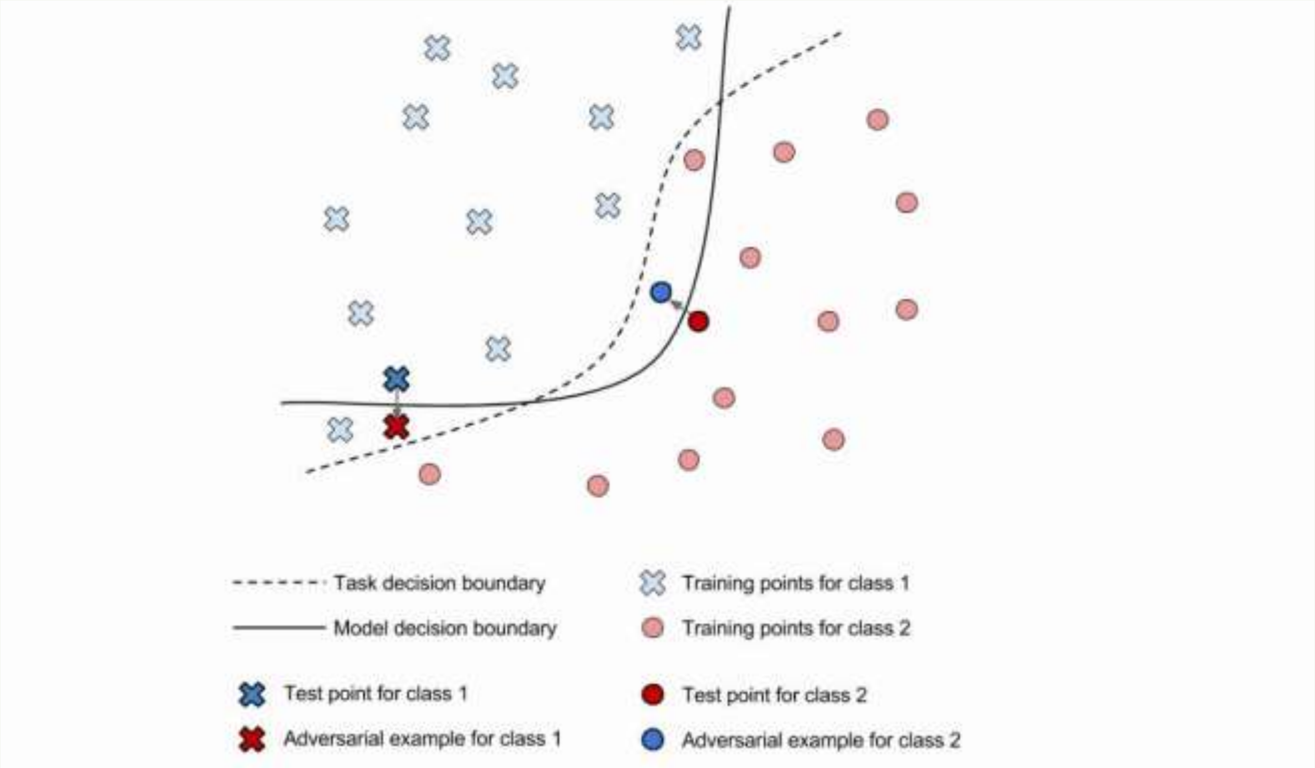
Adversarial **Attacks** - why the concern?



Adversarial **Attacks** – why the concern?



Why do these attacks occur?



Why do these attacks occur?

CNNs must be learning **different visual representations** compared to humans

Defences against attacks

• Adversarial Training	Computationally Expensive + % loss
• Modified Training Process	Computationally Expensive + % loss
• Supplementary Networks	Computationally Expensive
• Tweaking Architecture	Not too effective yet

Reference

Look for a better **solution**?

- Better understanding first - CNNs are a Black Box
- Tool for better understanding?
- Not a solution, but a **diagnosis**

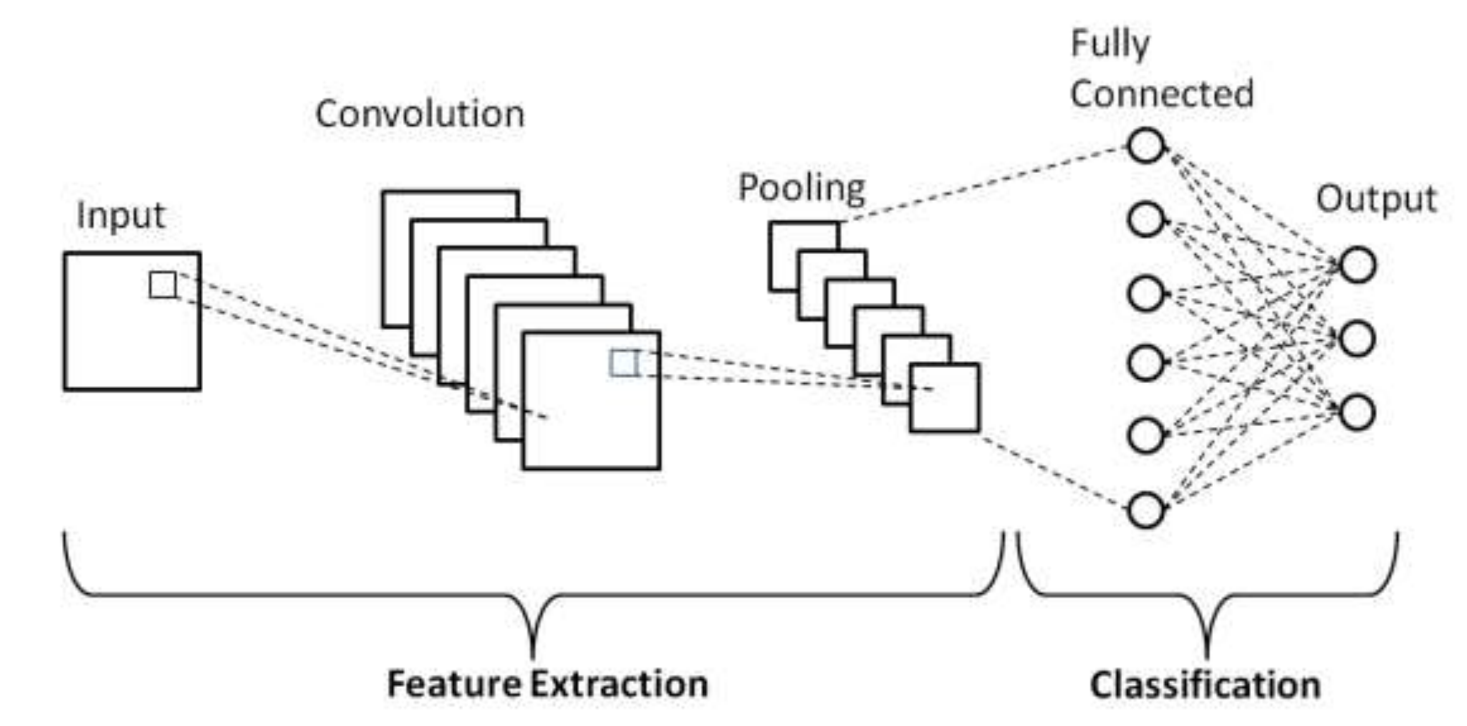
Neural Manifolds – Insights about Neural Dynamics (from Theoretical Neuroscience)

02

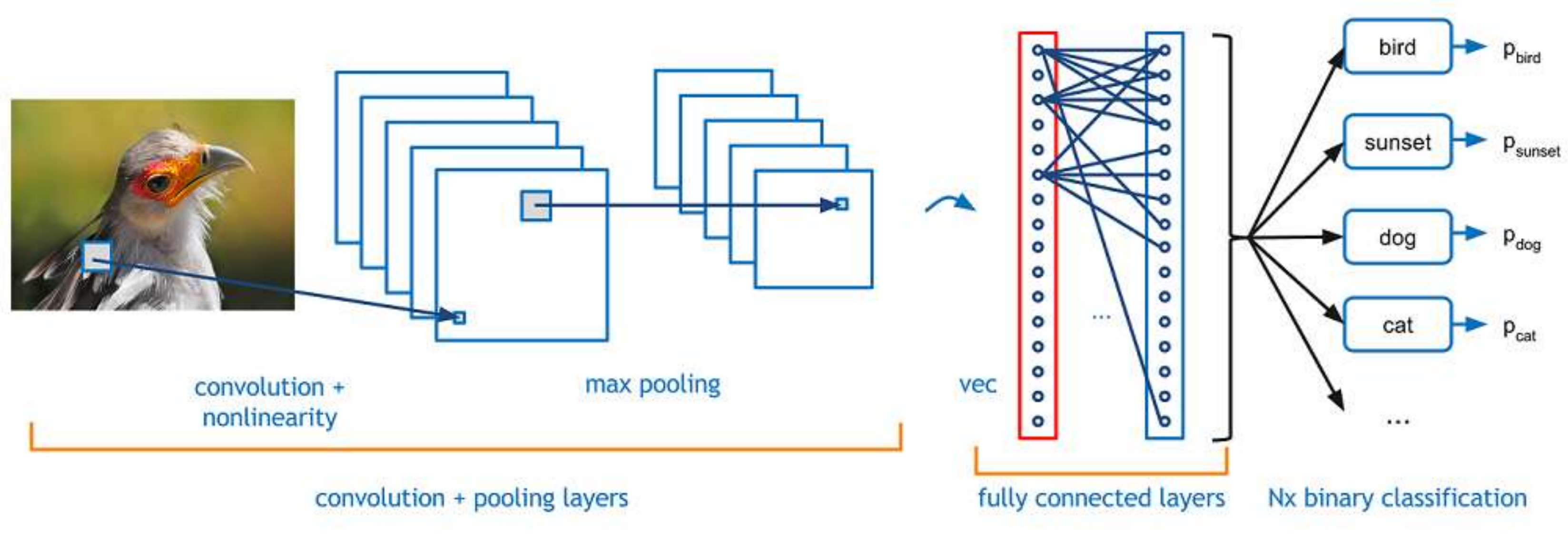
Background

CNNs? Adversarial Attacks?? Neural Manifolds???

How does a **CNN work**?



How does a **CNN work**?



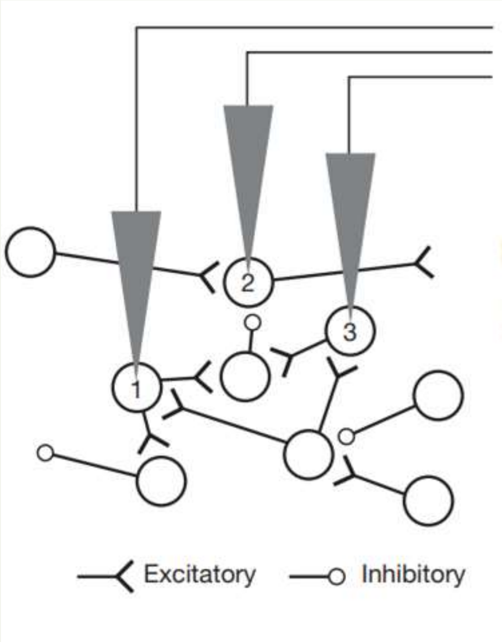
Adversarial Attacks

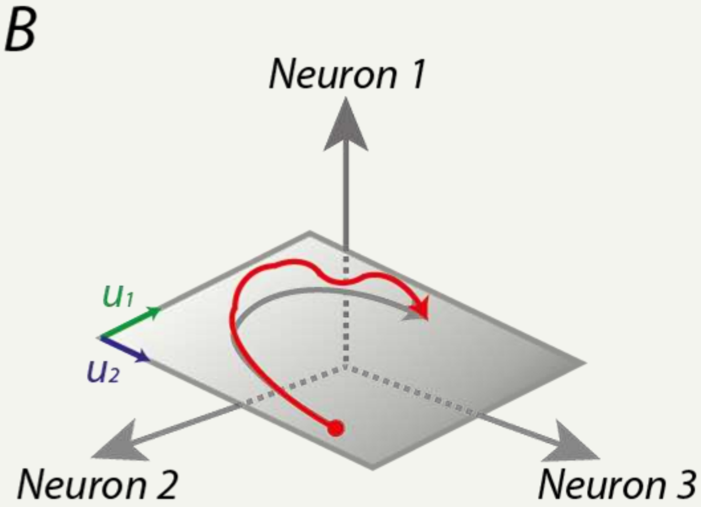
- Addition of imperceptible perturbations
- White Box vs Black Box Attacks
- Imperceptibility adhered to with perturbation budget

$$x^{adv} = \underset{\hat{x}:\|\hat{x}-x\|_p \leq \epsilon}{\operatorname{argmax}} L(\hat{x}, y)$$

Neural **Manifolds**

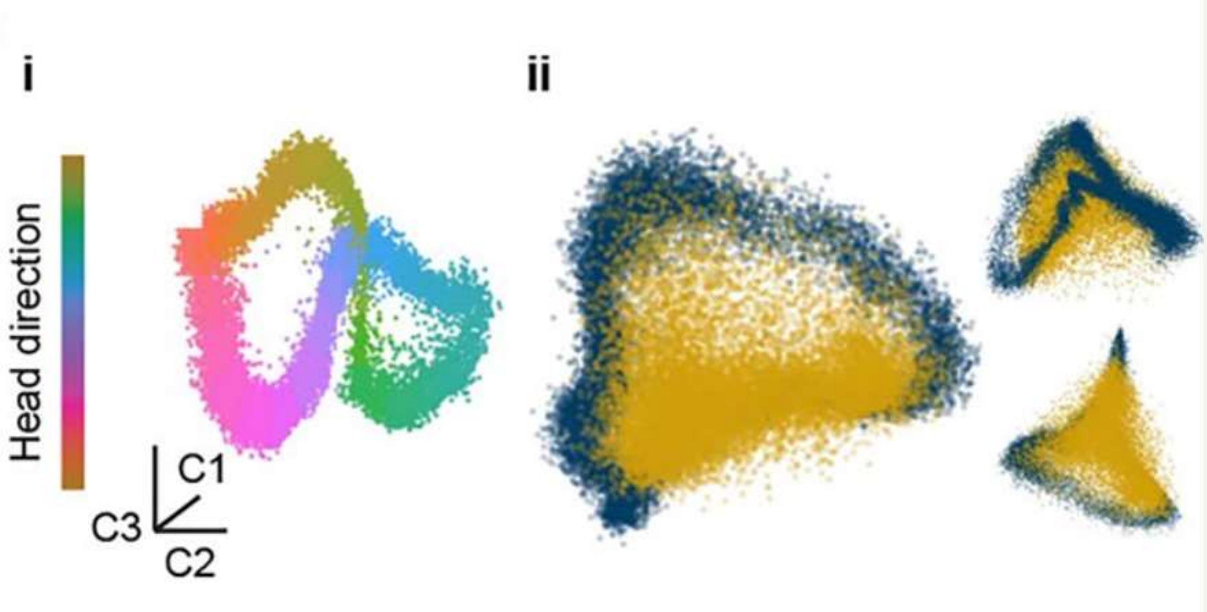
- Helps interpret neural activity at the population level





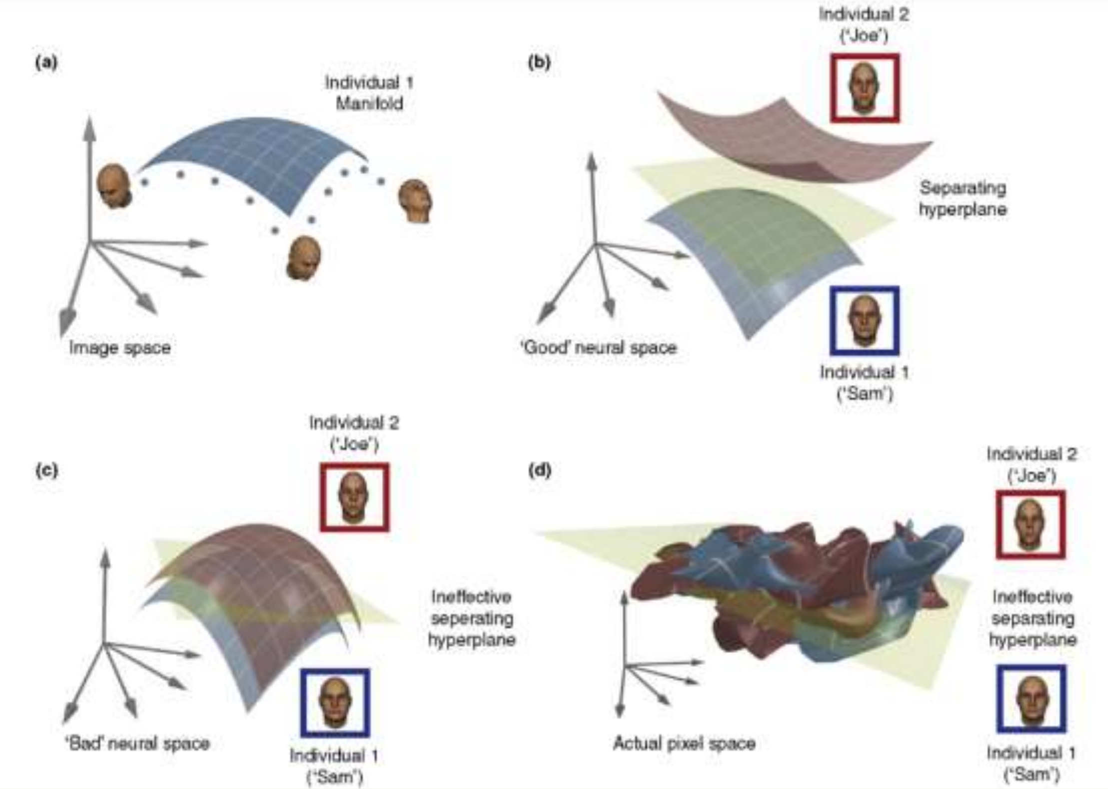
Neural Manifolds

- Helps interpret neural activity at the population level



Mouse head-direction circuit
(Chaudhuri et al., 2019)

Object Manifolds in **Vision**



DiCarlo, J. J., & Cox, D. D. (2007)

03

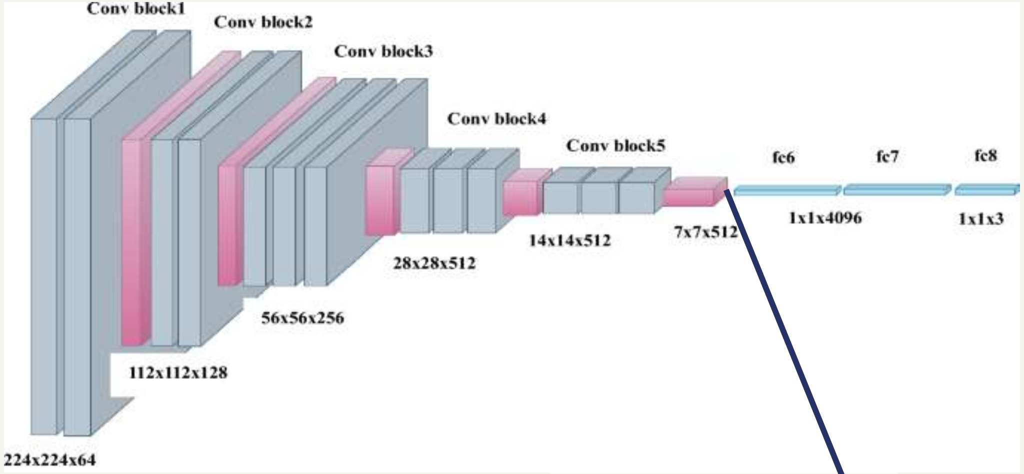
Problem Statement

Manifolds and Robustness ?

Class Activation Manifolds

- CNN Analog and Generalisation of object manifolds
- Capture neural activations in a CNN for a class
- Treat each activation as a point on a manifold

Class Activation Manifolds

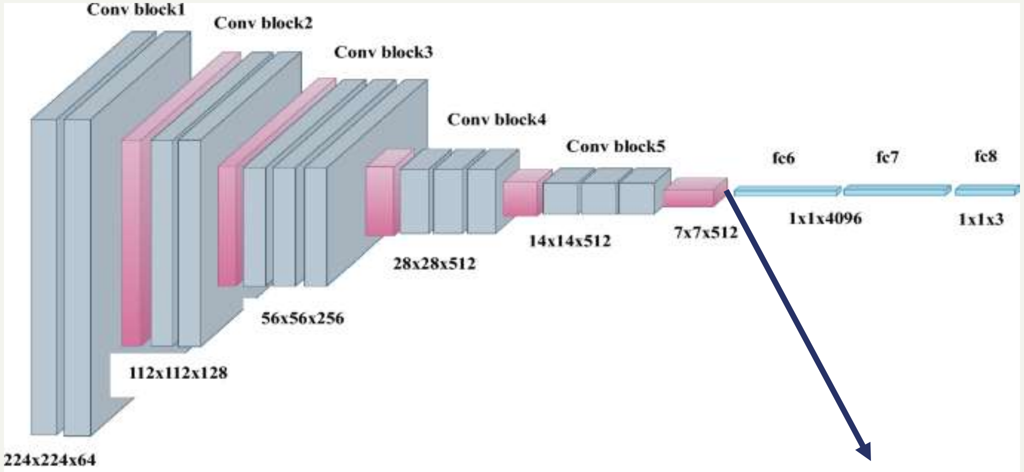


Dog

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

n=25,088

Class Activation Manifolds



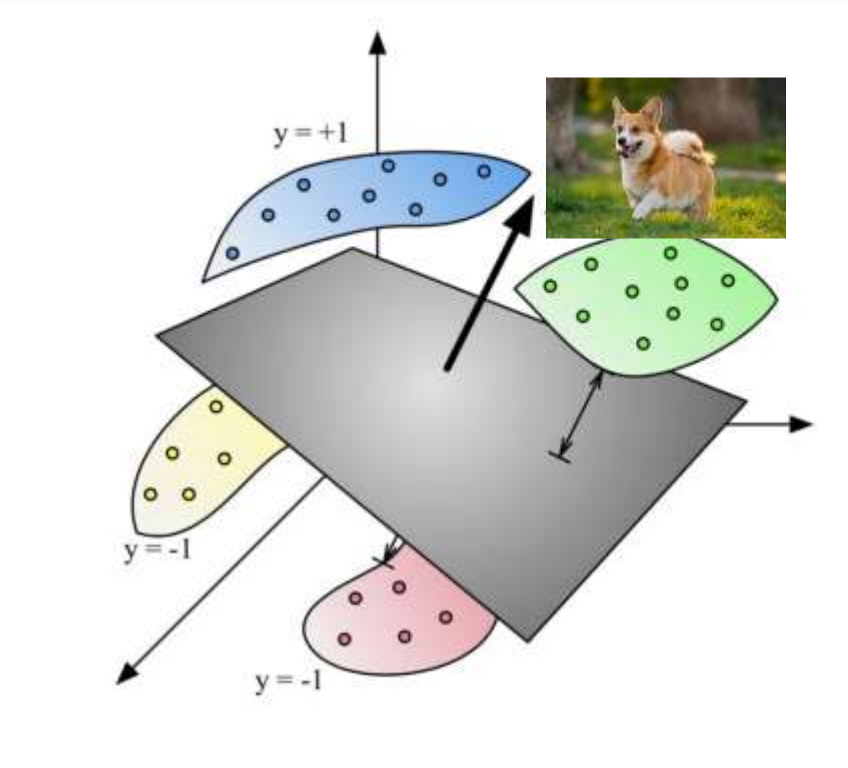
Dog

Repeat for all images in a class!

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

n=25,088

Class Activation Manifolds



My Research Question

Does the dimensionality of CAMs play a role in how adversarially robust the CNN is?

Cohen et al. (2019)

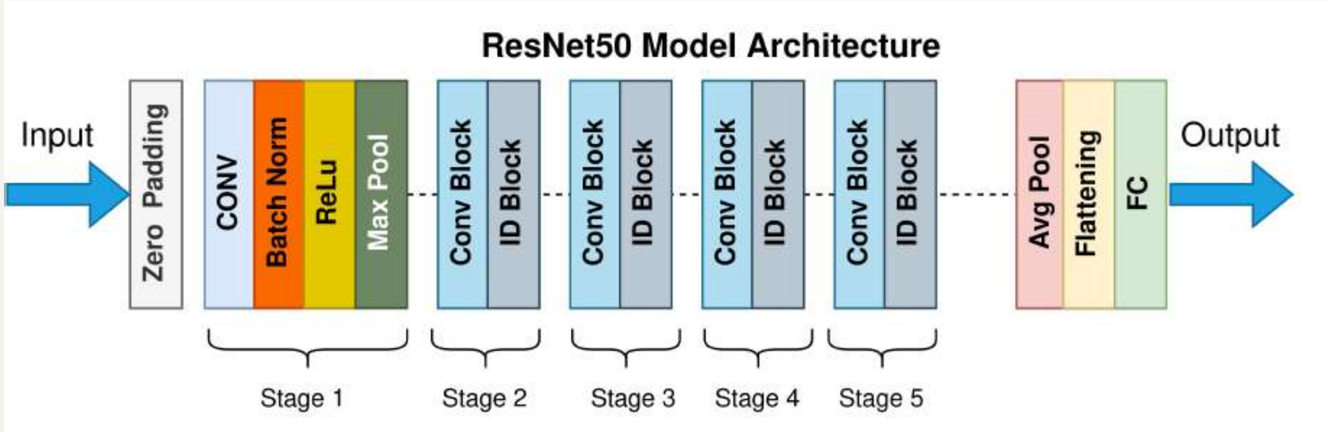
04

Methodology

Putting it all together

CNN **Architecture**

- ResNet50 – Most prevalent CNN
- 50 layers
- Residual Skip Connections



Dataset

- ImageNet-1K
- 1000 classes of naturally images
- Most widely-used dataset
- ResNet50 pre-trained on ImageNet

Adversarial Attack - PGD

- Projected Gradient Descent
- White Box Attack
- (Loss) Gradient-based Attack
- Iteratively takes a step to maximise loss

Algorithm 1: Projected Gradient Descent (PGD) Adversarial Attack (l_∞)

Input: Original image x, Target class y, Loss function $J(\theta, x, y)$, Perturbation size ϵ, Step size α, Number of iterations K

Output: Adversarial example x_{adv}

Sample random noise n from Uniform distribution in range $(-\epsilon, \epsilon)$;

Initialize $x_{\text{adv}} = x + n$;

for $k = 1$ **to** K **do**

  Compute the gradient of the loss function w.r.t. the input:

  $\text{grad} := \nabla_x J(\theta, x_{\text{adv}}, y)$;

  Compute the step necessary for the adversarial attack:

  $\text{step} := \text{sign}(\text{grad})$;

  Compute the adversarial input:

  $x_{\text{adv}} := x_{\text{adv}} + \alpha \cdot \text{step}$

  Clip the step to ensure it lies within $[x - \epsilon, x + \epsilon]$:

  $x_{\text{adv}} := \text{clip}(\text{step}, x - \epsilon, x + \epsilon)$;

end

CAM **Dimension Estimation**

- Principal Component Analysis
- Components needed for 95% explained variance = dimension

Algorithm 2: Estimation of Class Activation Manifold Dimensionality

Input: Number of classes n, Number of images per class m, Threshold for variance explained γ=0.95

Output: List of dimensions for each class d_{classes}

Randomly sample n ImageNet classes;

for *each class* **do**

    Sample m images from the class;

    for *each image* **do**

        Extract activations of the final non-classification layer with D neurons;

        Record activations as D-dimensional list, a;

    end

    Concatenate recorded activations into a single $m \times D$ matrix A;

    Perform PCA on matrix A;

    Compute cumulative explained variance ratio;

    $d :=$ Number of principal components required to explain 95% of variance;

end

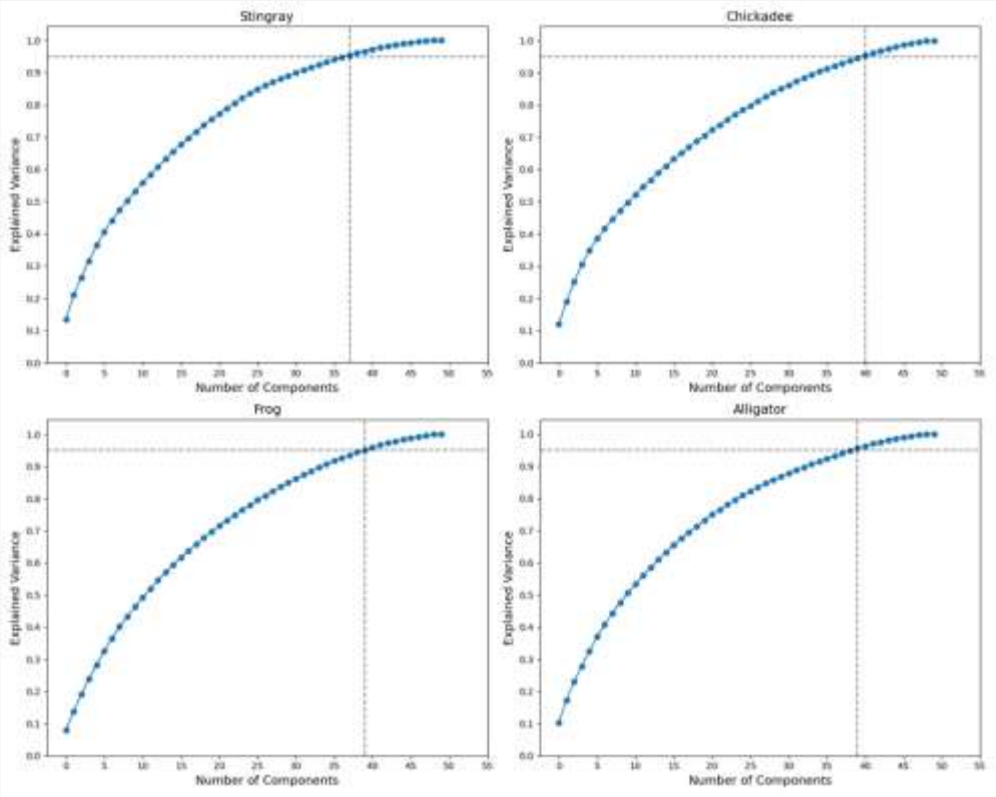
Concatenate all d's into a list d_{classes}

05

Results

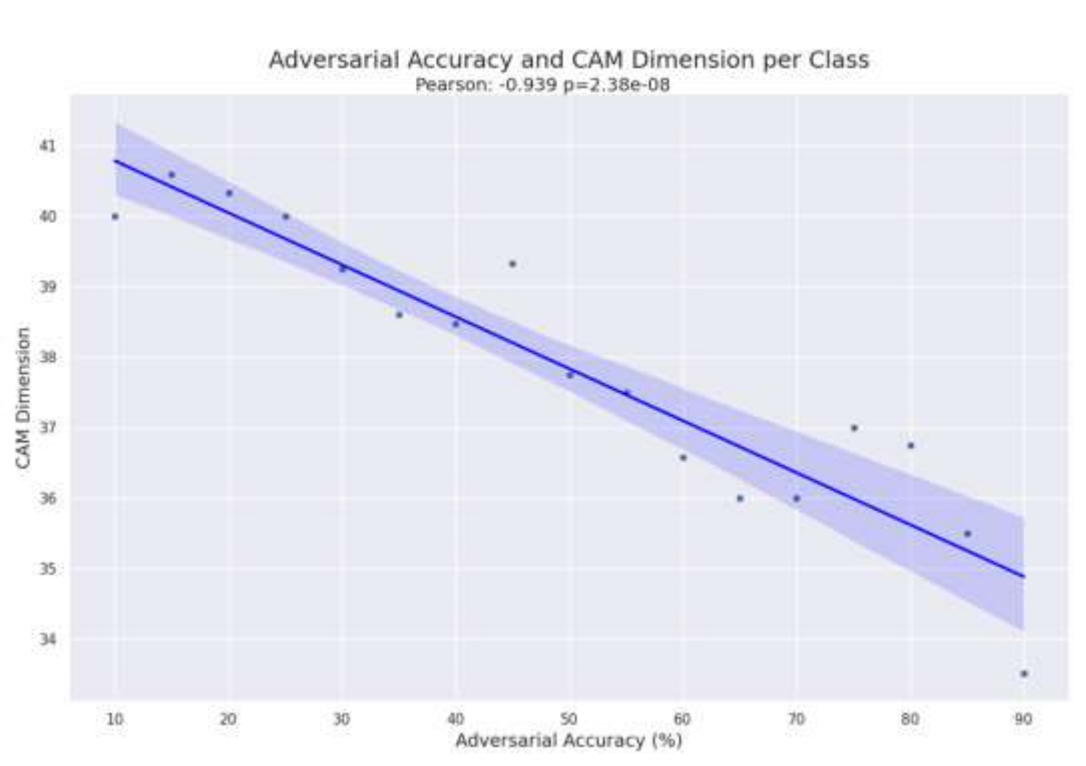
Finally

Dimensionality of ResNet50 CAMs



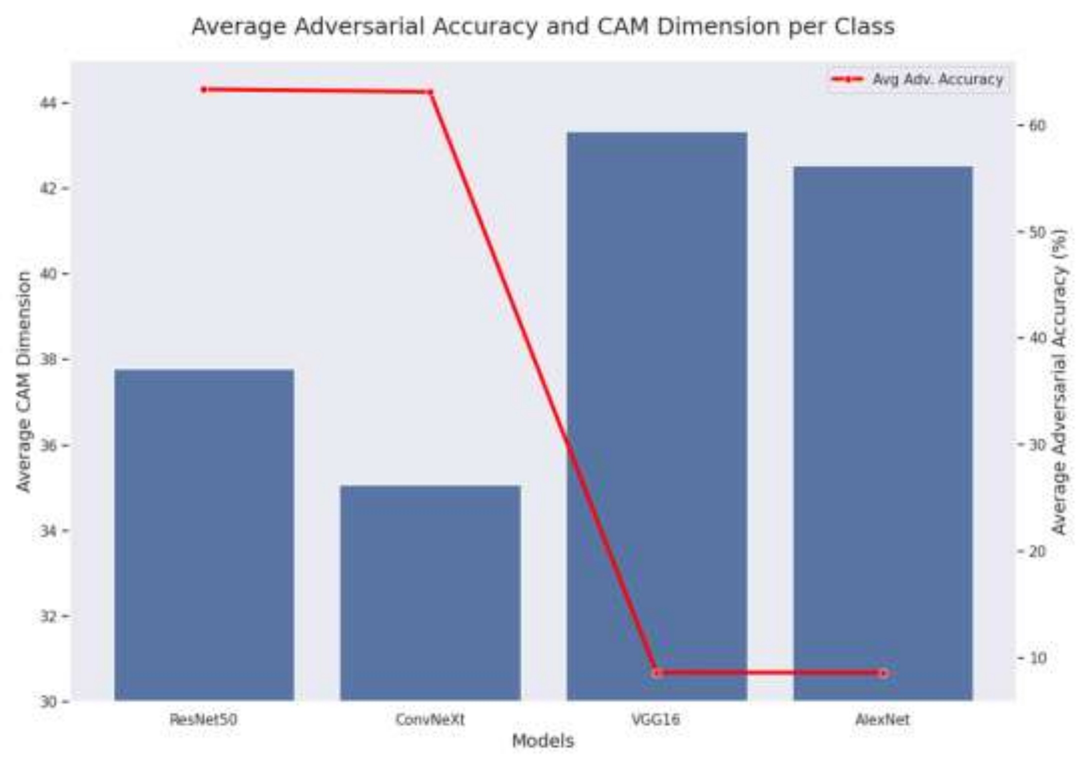
- 2048 dimension activations
- CAM ~40 dimensions
- Heavy correlation for each class

Adversarial Accuracy **and** CAM Dims



- 100 randomly sampled classes
- More robust ➡ Lower CAM dimensions
- Strong negative correlation

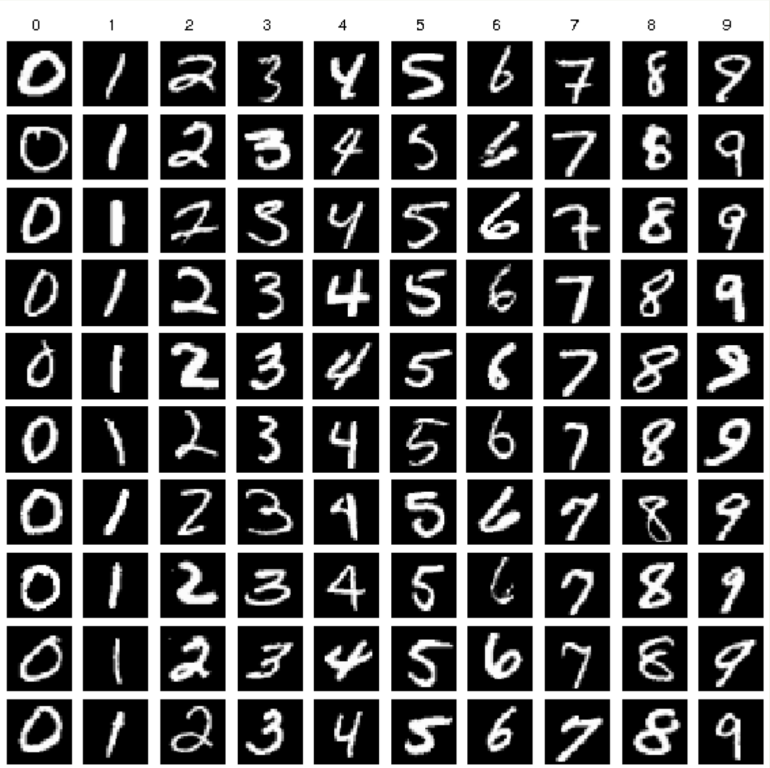
Relationship in **Multiple CNNs**



- Experiment Repeated with 4 CNNs (averages here)
- Relationship Preserved
- More Sensitive to Greater changes in robustness

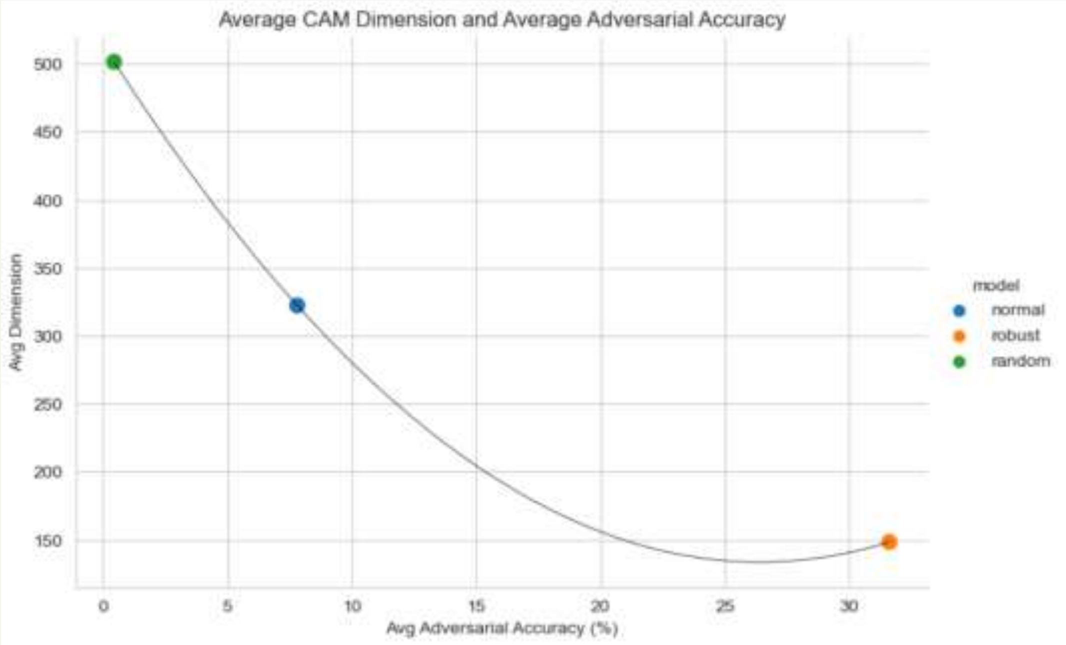
Effect of **Adversarial Training**

- MNIST Dataset
- PDG Adversarial Training (Madry, 2019)
- 3 Models:
  - No training - **random** weights
  - No AT - **normal** training
  - AT - **robust** model



Effect of **Adversarial Training**

- MNIST Dataset
- PDG Adversarial Training (Madry, 2019)
- 3 Models:
  - No training - **random** weights
  - No AT - **normal** training
  - AT - **robust** model
- Supporting evidence
- Lower Avg Dimension - n_classes ?



06

Conclusion

Making sense of it all

Experimental **Results Analysis**

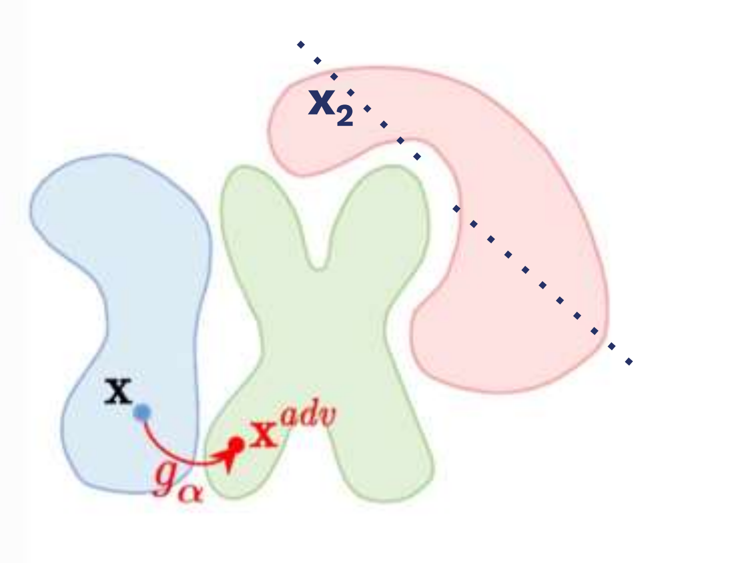
- Average dimension from PCA << output shape of activations layer
  ➡ Class Activations do live in lower dimensions

- Strong negative correlation between ResNet50’s class-wise adversarial robustness and CAM Dimensions

- Relationship verified across models and through AT

Lower dimensional CAMs

- CAMs show ‘where’ processed image before point of classification
- Lower dimensional CAMs ➡ less sensitive to perturbations
- More compact and efficient representations



Implications

- Intentionally train CNNs to align with manifold properties
- Class Activation Manifolds as a diagnostic tool for CNN behaviour
- Attempt at mechanistically understanding the CNN

Future Work

- Further verify relationship between Adversarial Robustness and CAM dims
- Test hypothesis across more CNNs, datasets, modalities
- Other types of robustness
- Other properties of CAMs
- Multi-objective networks to align manifolds

Thank You!

- Prof Debayan
- Prof Venkat (BITS Pilani)
- Prof Raghavendra
- Prof Subhashis
- Friends and Family