ImageNet Classification with Deep Convolutional Neural Networks

Prepared by
Faizaan Naveed
Won Mo (Andy) Jung

Lassonde School of Engineering,
York University
Canada

Objective

- Overview of Convolutional Neural Network
- Training the network
- Techniques to reduce over-fitting
- AlexNet
- Advantages and Disadvantages
- An insight into the blackbox

Objective (cont.)

- Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held.
- In ILSVRC-2012, a CNN model won the challenge, which gave rise to the current popularity of deep learning.
- Convolutional Neural Network - AlexNet
  - 60 million parameters
  - 650,000 neurons
  - 5 convolutional layers
  - 3 fully-connected layers
  - 1000-way softmax
  - Dropout used

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

2

Convolutional Neural Network (CNN)

- CNNs are specific type of feed-forward artificial neural networks that are used for interpreting imagery.
- CNNs use relatively minimal pre-processing compared to other image classification algorithms. The algorithms learns filters that are traditionally hand coded for feature extraction.

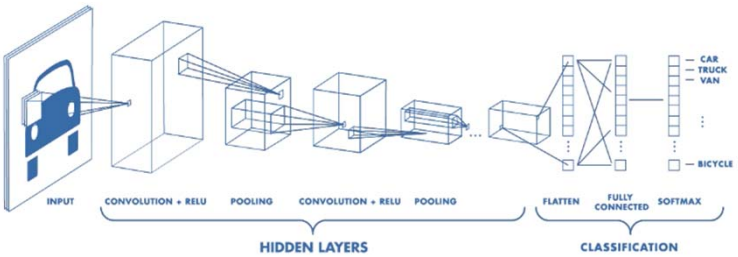


Image Credit: Mathworks

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

3

Convolutional Neural Network (CNN)

- CNNs, unlike regular ANNs, are adaptive to the properties of an image. In a regular ANN, the required number of parameters would lead to overfitting on the dataset.
- The architecture of the CNN exploits image processing techniques to reduce the number of parameters by sharing the weights in local spatially connected region defined by the size of the kernel.

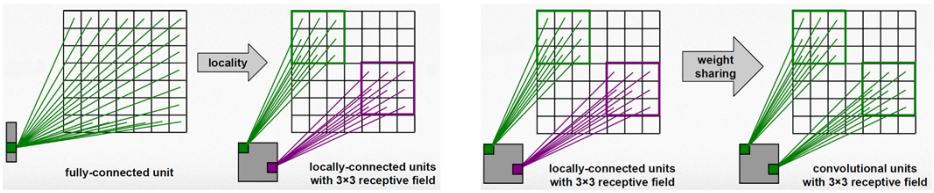
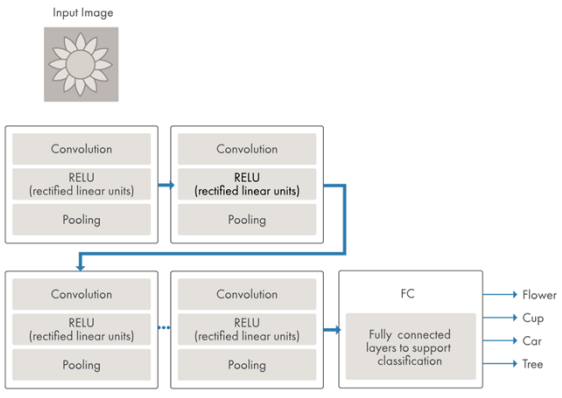


Image Credit: Industrial AI Lab

4

Design of CNN - Overview

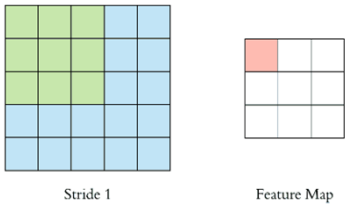
- The hidden layers of the CNN typically consist of Convolution, ReLU, Batch Normalization, Pooling, Fully connected layer and Dropout layers.



5

Design of CNN – Convolution Layers

- Convolution: A set of convolutional filters convolute through the input image to highlight a specific feature.
- There are 4 hyperparameters used to design the convolutional layers:
  - The kernel size: The vertical and horizontal dimensions of the filter.
  - The filter count: The total number of filters to be used for each convolution layer.
  - Stride: The number of pixels filter moves in vertical and horizontal directions.
  - Padding: Appending artificial pixels to the borders of the image to preserve its size.

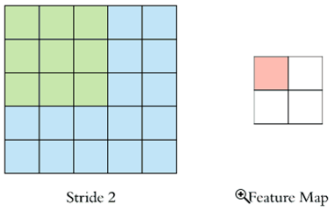


6

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Design of CNN – Convolution Layers

- Convolution: A set of convolutional filters convolute through the input image to highlight a specific feature.
- There are 4 hyperparameters used to design the convolutional layers:
  - The kernel size: The vertical and horizontal dimensions of the filter.
  - The filter count: The total number of filters to be used for each convolution layer.
  - Stride: The number of pixels filter moves in vertical and horizontal directions.
  - Padding: Appending artificial pixels to the borders of the image to preserve its size.

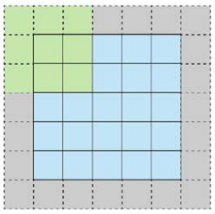


6

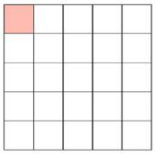
LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Design of CNN – Convolution Layers

- Convolution: A set of convolutional filters convolute through the input image to highlight a specific feature.
- There are 4 hyperparameters used to design the convolutional layers:
  - The kernel size: The vertical and horizontal dimensions of the filter.
  - The filter count: The total number of filters to be used for each convolution layer.
  - Stride: The number of pixels filter moves in vertical and horizontal directions.
  - Padding: Appending artificial pixels to the borders of the image to preserve its size.



Stride 1 with Padding



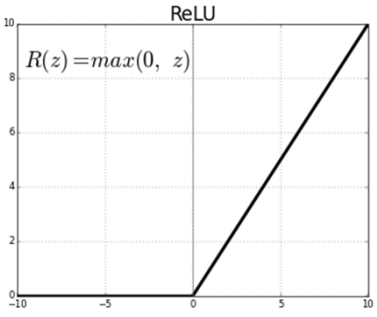
Feature Map

6

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Design of CNN – ReLU Activation

- Convolutional layers are followed by activation functions to introduce non-linearity in the model.
- Rectified Linear Unit (ReLU): In terms of training time with gradient descent, other saturating nonlinearities are much slower than the non-saturating nonlinearity ReLU (Krizhevsky, 2014).
- f(z) = max(0, z).

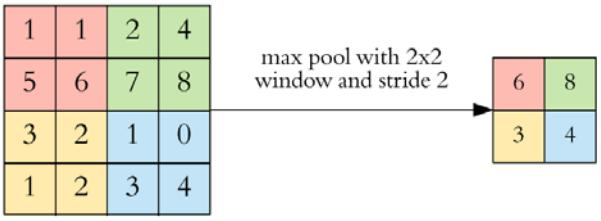


7

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Design of CNN – Pooling layers

- Pooling layers of the CNN are used to progressively downsample the information present in the image. This helps reduce the number of parameters that the network needs to learn.
- Max-pooling is a commonly used operation where the maximum value in a *n* x *n* kernel is used to downsample the image.



8

Design of CNN – Fully Connected Layers

- Fully Connected layer: The neurons in FC layer are connected to all the activations from the previous layers (as is the case with regular NN).
- The difference between convolutional layers and FC layers is that in convolutional layers the neurons are only connected to a local region in the input and the parameters are shared between neurons.

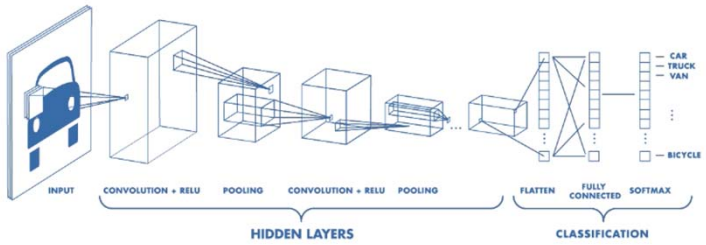
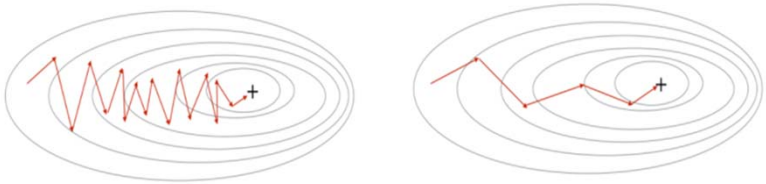


Image Credit: Mathworks

9

Training the CNN

- Since Stochastic Gradient Descent (SGD), several optimization techniques have been developed to accelerate training.
- SGD has trouble navigating areas around local optima.
  - Slower convergence
  - Large oscillations in irrelevant directions
- Momentum update dampens the oscillations and helps accelerate gradients vectors in the right directions.

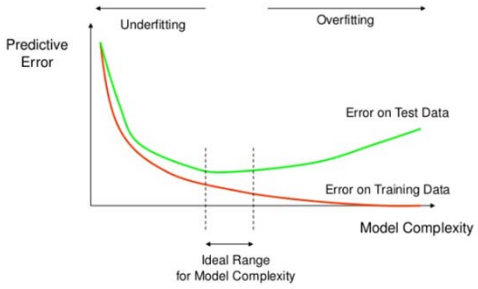


10

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Reducing Overfitting - Techniques

- Deeper networks easily overfit on the training dataset, due to the small number of examples and large number of parameters.
- Techniques to reduce overfitting:
  - Dataset augmentation
  - Early stopping
  - Weight penalty (L1 and L2)
  - Dropout



11

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

AlexNet Architecture

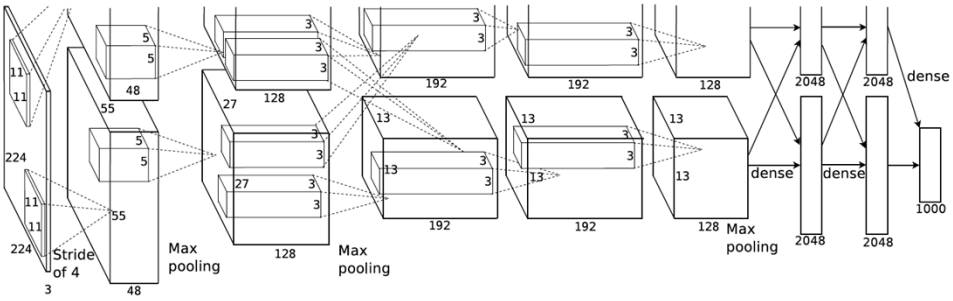


Image Credit: Krizhevsky, 2014

Local Response Normalization

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

- $a^i_{x,y}$ the activity of a neuron computed by applying kernel *i* at position *(x, y).*
- $b^i_{x,y}$ *Response-normalized activity.*
- ReLU neurons have unbounded activations and we need LRN to normalize that. This scheme bears some resemblance to the local contrast normalization scheme.
- More correctly termed “brightness normalization”. The idea is to enhance the peaks and dampen the flat responses.

Reducing Overfitting: Data Augmentation

- 1st form of data augmentation: Generate image translations and reflections
  - Extract random 224 × 224 patches from the 256×256 images.
  - Generate image translations and horizontal reflections.
  - This increases the size of training set by a factor of 2048.
  - The network is tested on five 224x224 patches extracted from the original image and their horizontal reflections.
- 2nd form of data augmentation
  - Alter the intensities of the RGB channels in training images.
  - Perform PCA on the set of RGB pixel values throughout the ImageNet training set.
  - The RGB values are then added to the principal components.

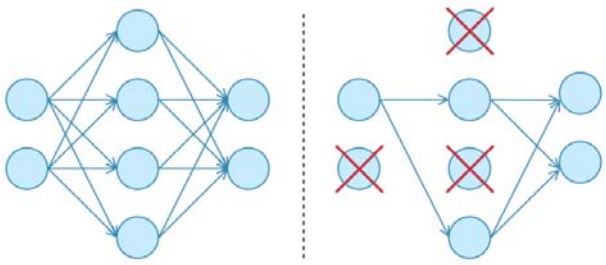
$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

14

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Reducing Overfitting: Dropout

- During training time, at each iteration, a neuron is temporarily "dropped" or disabled with probability *p*.
- Dropout prevents the network to be too dependent on a small number of neurons and forces every neuron to be able to operate independently.



15

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Details of Training

- Stochastic Gradient Descent (SGD)
- Batch size 128
- Momentum of 0.9
- Weight decay of 0.0005
- Initial Learning rate: 0.01

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

- *i: iteration index*
- *v: momentum variable*
- ϵ *: learning rate*

$\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$ • *Average over ith batch Di* of the derivative of the objective with respect to w, evaluated at wi.

16

Details of Training

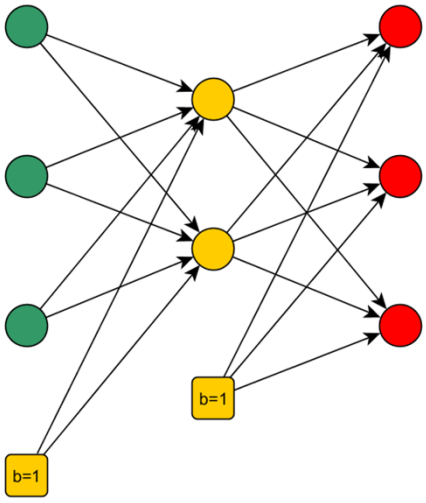


Image Credit: Stack Exchange

- Initialized the neuron biases in the second, fourth, and fifth convolutional layers, as well as in the fully-connected hidden layers, with the constant 1.
- Started with equal learning rate for all layers, then adjusted manually throughout training.
- The learning rate was initialized at 0.01 (ended up reducing it 3 times prior to termination).
- Trained the network for roughly 90 cycles through the training set.

17

AlexNet Results

Model	Top-1	Top-5
Sparse coding	47.1%	28.2%
SIFT + FVs	45.7%	25.7%
CNN	37.5%	17.0%

Comparison of results on ILSVRC-2010 test set

Model	Top-1 (Val)	Top-5 (Val)	Top –5 (Test)
SIFT + FVs	--	--	26.2%
1 CNN	40.7%	18.2%	--
5 CNNs	38.1%	16.4%	16.4%
Pre-trained 1 CNN	39.0%	16.6%	--
Pre-trained 7 CNNs	36.7%	15.4%	15.3%

Comparison of results on ILSVRC-2012 test set

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

18

Problems with typical CNN

- CNN does not encode the position and orientation of the object into their predictions.

Image Credit: Saama Technologies Inc.

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

19

Problems with typical CNN

Image Credit: Saama Technologies Inc.

- Australia or Africa?

20

Problems with typical CNN

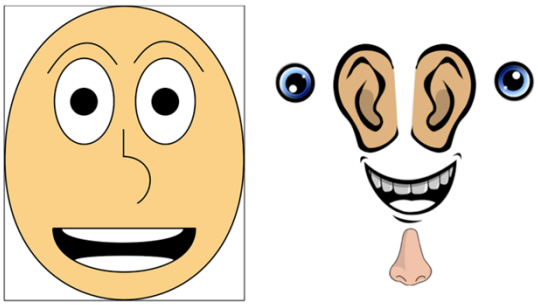


Image Credit: Saama Technologies Inc.

- Does CNN consider both images as "face"?

21

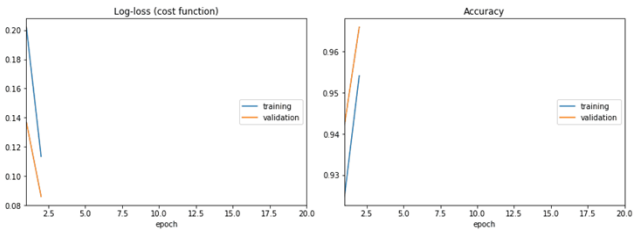
Limitation with AlexNet Structure

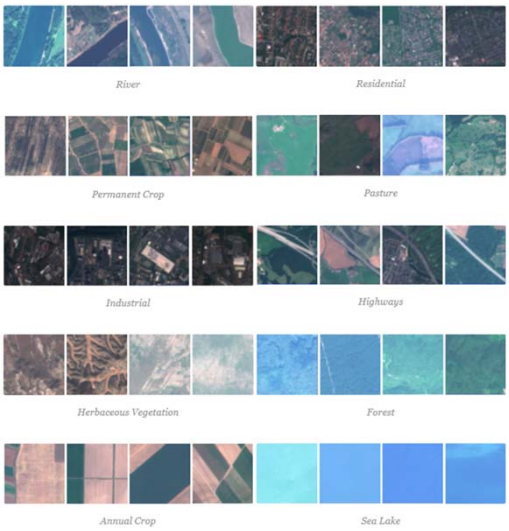
- Limited to image base inputs.
- Require significant GPU memory to train the model.
- Cost expensive (took 5~6 days to train the network at that time).

22

EuroSat Dataset

- EuroSat dataset contains 27,000 images divided into 10 classes.
- Each class contained 2,000-3,000 64 x 64 images with 13 spectral bands.
- 4:1 Train-Test split.





23

EuroNet Structure

- Network Structure:
  - 6 convolutional layers
  - 2 fully-connected layers
  - ~6 million parameters
  - 10-way sigmoid
  - Dropout used
- The network was trained with a batch size of 50 images for 20 epochs.
  - Validation Accuracy: 98.3%
  - Training Accuracy: 99.6%
  - Training Loss: 0.007
  - Validation Loss: 0.06

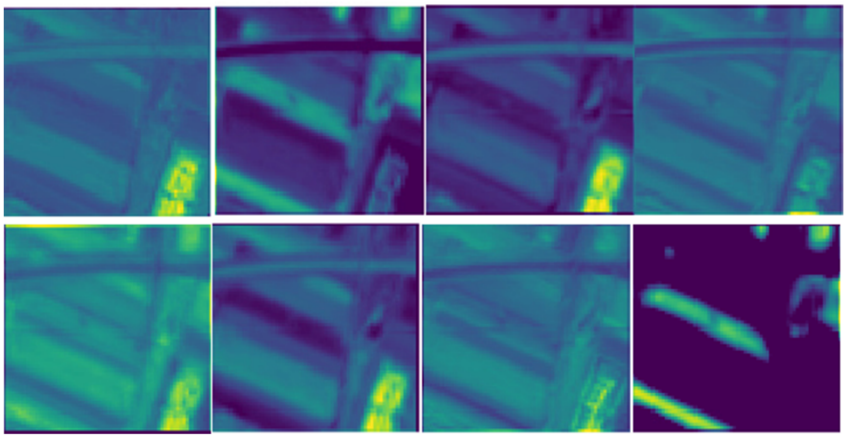
Layer (type)	Output Shape	Param #
conv2d (Conv2D) + ReLU	(None, 28, 64, 64)	3304
conv2d_1 (Conv2D) + ReLU	(None, 28, 64, 64)	7084
max_pooling2d (MaxPooling2D)	(None, 28, 32, 32)	0
conv2d_2 (Conv2D) + ReLU	(None, 56, 32, 32)	14168
conv2d_3 (Conv2D) + ReLU	(None, 56, 32, 32)	28280
max_pooling2d_1 (MaxPooling2	(None, 56, 16, 16)	0
conv2d_4 (Conv2D) + ReLU	(None, 112, 16, 16)	56560
conv2d_5 (Conv2D) + ReLU	(None, 112, 16, 16)	113008
max_pooling2d_2 (MaxPooling2	(None, 112, 8, 8)	0
flatten (Flatten)	(None, 7168)	0
dense (Dense) + ReLU	(None, 784)	5620496
dense_1 (Dense) + Sigmoid	(None, 10)	7850

Total params: 5,850,750
Trainable params: 5,850,750
Non-trainable params: 0

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Feature Maps

Output from First convolutional layer

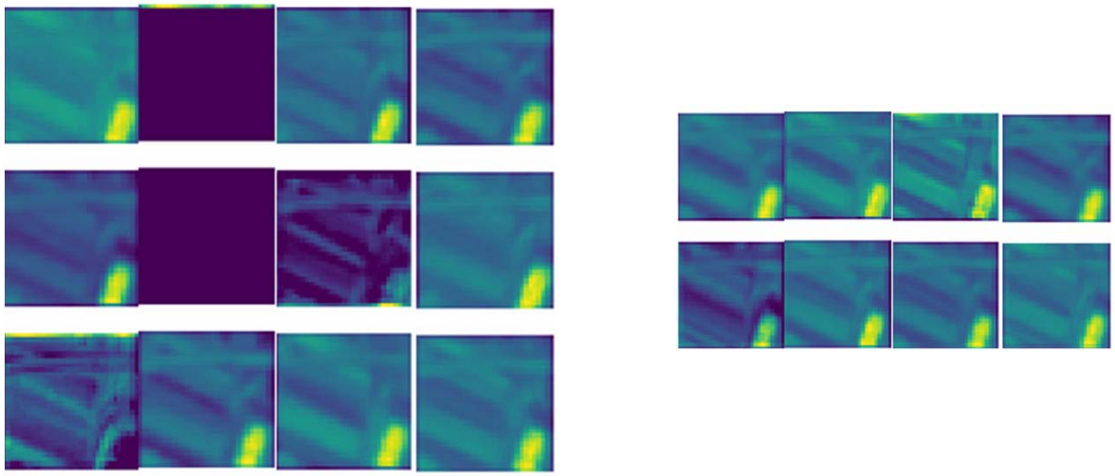


LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

24

Feature Maps

Output from Second convolutional layer

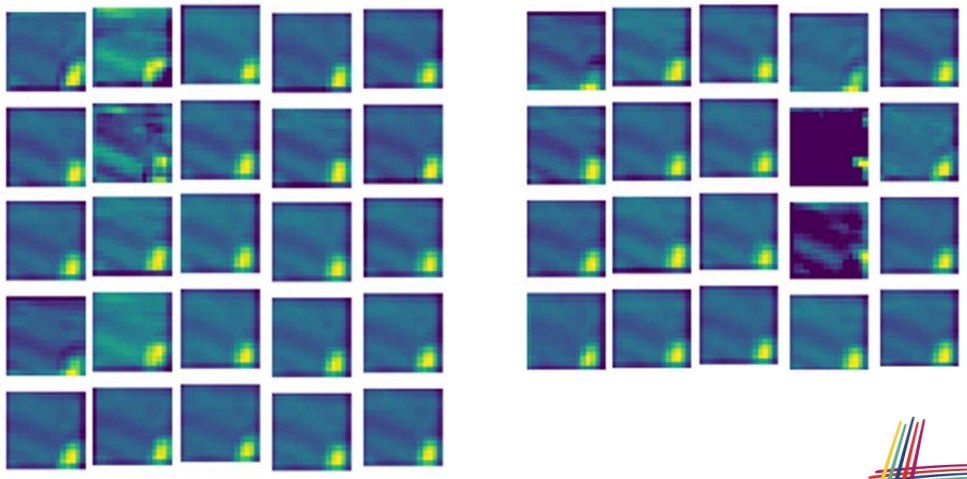


24

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Feature Maps

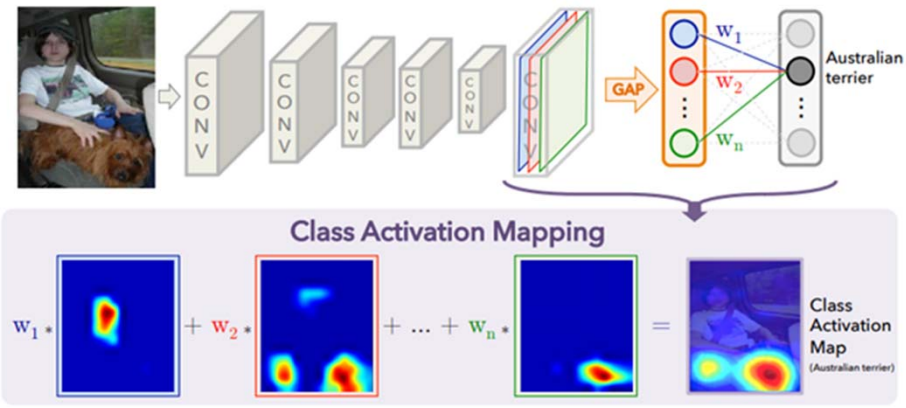
Output from Third convolutional layer



24

LASSONDE SCHOOL OF ENGINEERING | YORK UNIVERSITÉ UNIVERSITY

Class Activation Map

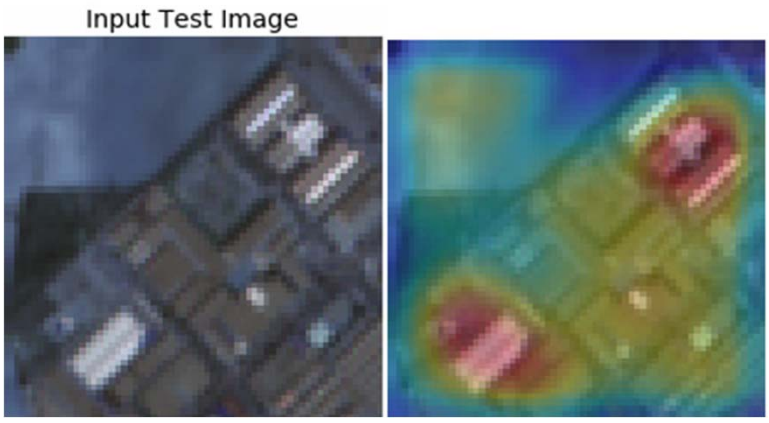


25

Class Activation Map

26

Class Activation Map



27

Thank You For Listening

Questions?