JC3504 Robot Technology

Lecture 8: Computer Vision (2)

Dr Xiao Li xiao.li@abdn.ac.uk

Dr Junfeng Gao Junfeng.gao@abdn.ac.uk

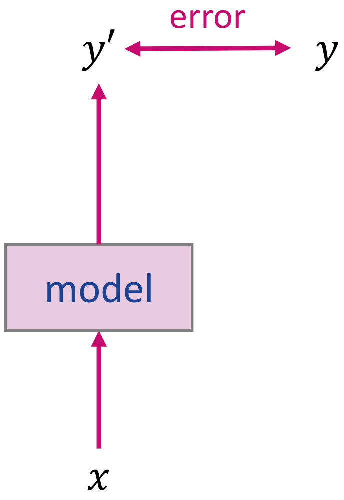
Outline

- Model Training Process (following last lecture)
- Computer Vision Task Review
- Image Classification

Model Training Process

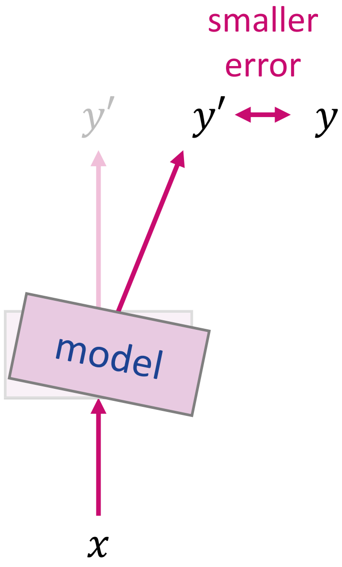
Model Training Process

Given an untrained model, the parameters are initially arbitrary, leading to arbitrary model outputs. We refer to the model's output as the predicted value y' and the desired output as the true value y.



The training process aims to optimize the model's parameters based on the error, making the predicted value closer to the true value.

This process is repeated multiple times until the error no longer decreases.



Model Training

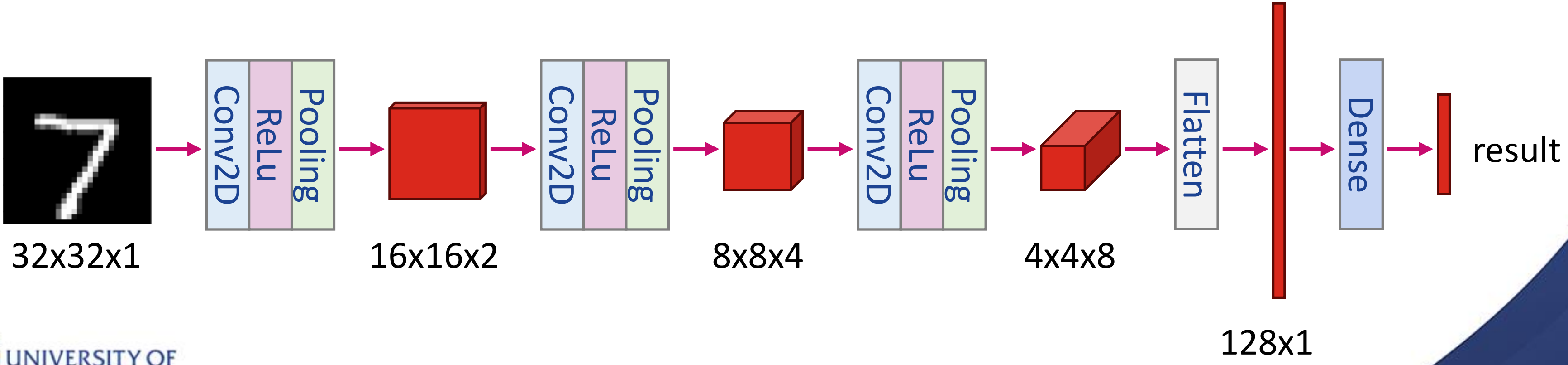
There are two questions for the training process:

- How to measure the error between the predicted value and the true value (use the loss function).
- How to optimise the parameters according to the loss?

Review: Convolutional Neural Network

Recall that we introduced the basic neural network architecture for recognising a handwritten number from an image.

Given an image, it is fed into the first Conv2D layer, and then, the output of each layer becomes the input of the subsequent layer. This process is called **Forward Propagation**.



Loss Function

Loss Function is an indicator that measures the difference between the model's predicted value (y') and the true value (y). It is a non-negative real number. The smaller the value of the loss function, the closer the prediction result of the model is to the actual situation.

When the model predicts continuous output values, we can naturally adopt the Mean Squared Error (MSE):

$$loss = MSE(y', y) = (y' - y)^2$$

When the model predicts the classification (like our task), MSE no longer works, because of the probability interpretation mismatch. In classification tasks, the probability distribution produced (by applying the softmax function to the output layer) is a multinomial distribution, but MSE assumes that the distribution follows a Gaussian distribution.

Cross-Entropy Loss

Cross-Entropy Loss is for classification tasks. It measures the dissimilarity between the true label distribution and the predictions made by the model, effectively penalizing predictions that diverge from the actual labels.

In binary classification, it's known as Binary Cross-Entropy, and for multi-class classification, it's often referred to as Categorical Cross-Entropy.

$$loss(p, q) = -\sum_{i=0}^{C} q_c \log p_c$$

Where p is the prediction, q is the ground true.

NB: Cross-Entropy Loss requires p to be a vector of probabilities, i.e., entries in p must be greater than 0 and the sum of p must equal 1. Therefore, an additional softmax operation is required before computing the loss.

NB: Here, the Cross-Entropy Loss formula is based on the definition of entropy. PyTorch/TensorFlow use a different form, where the true values are label indexes. Please check their documentation when using it.

Cross-Entropy Loss

Suppose the output of the model is y', y' is the vector with each entry corresponding to a class.

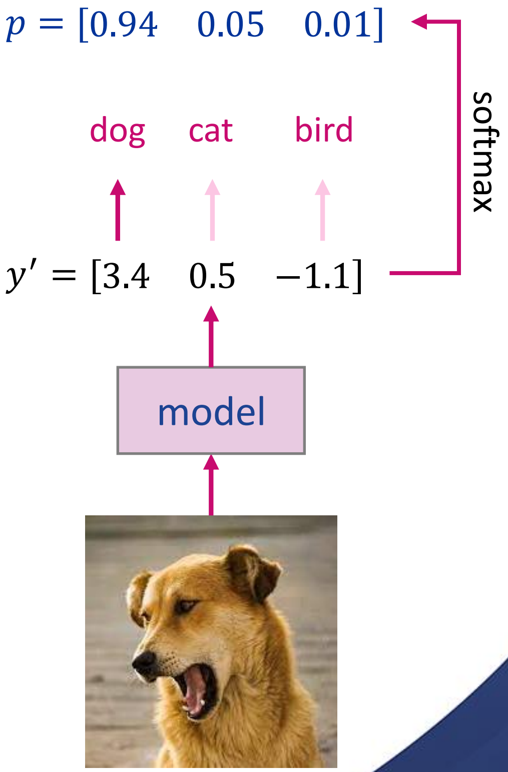
$$p = softmax(y')$$

where

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=0}^{n} e^{z_j}}$$

then

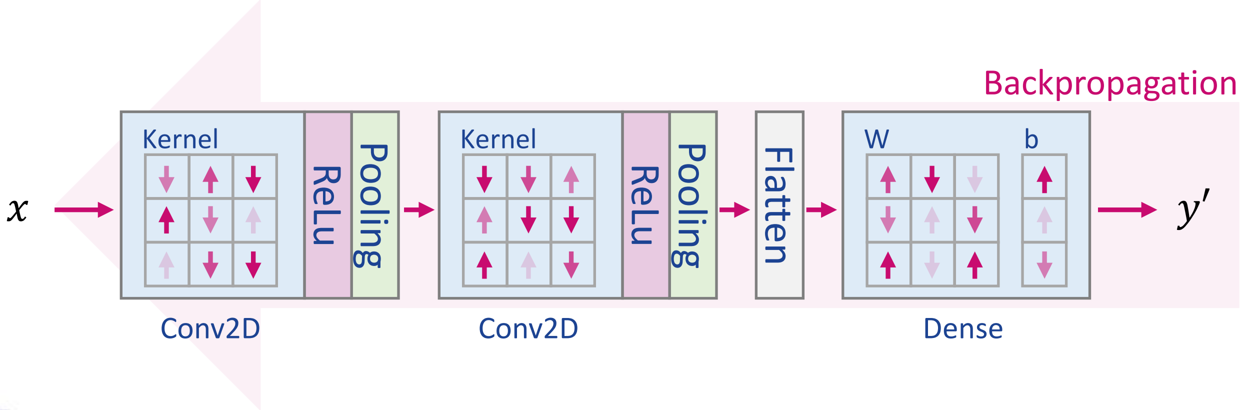
$$loss = \text{CrossEntropy}(p, y) = -\sum_{i=0}^{C} y_c \log p_c$$


$p = [0.94 \quad 0.05 \quad 0.01]$
dog
cat
bird
softmax
$y' = [3.4 \quad 0.5 \quad -1.1]$
model

Backpropagation

Backpropagation is the process of determining how to optimise model parameters based on losses. After backpropagation, each model parameter is assigned a real number (the gradient) that indicates how the parameter should be adjusted.

Most deep learning frameworks can perform backpropagation automatically.



Optimisation

The gradients indicate how parameters should be adjusted; therefore, the optimisation process applies these adjustments by revising the parameters in accordance with their own gradients.

However, each optimisation step only adjusts the parameters slightly, such as 1/1000 of the gradient magnitude. This ratio (i.e., 1/1000 in this example) is referred to as the learning rate.

* what we didn’t introduced

- The basic structure of neural networks: neurons, layers
- Data preparation and pre-processing
- Model initialisation
- Principles of the backpropagation algorithm
- The chain rule
- Gradient descent and its variants
- Stochastic gradient descent
- Selection and tuning of hyperparameters
- Overfitting and underfitting
- Regularisation and dropout

Computer Vision Task Review

1495 UNIVERSITY OF ABERDEEN

Computer Vision Task Review

Computer Vision tasks encompass a wide range of problems that aim to replicate the human visual system's ability to interpret and understand the content of digital images and videos. Here are some key tasks in this field:

- Image Classification
- Object Detection
- Semantic Segmentation
- Instance Segmentation
- Object Tracking
- Pose Estimation
- Depth Estimation
- Facial Recognition
- …

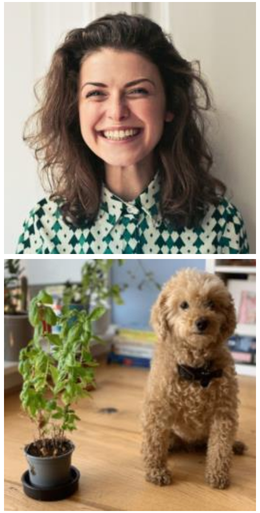
Image Classification

Image classification is the task of assigning a label to an image from a predefined set of categories. It also includes extracting multiple features/objects from a single image.

cat

dog

Classification



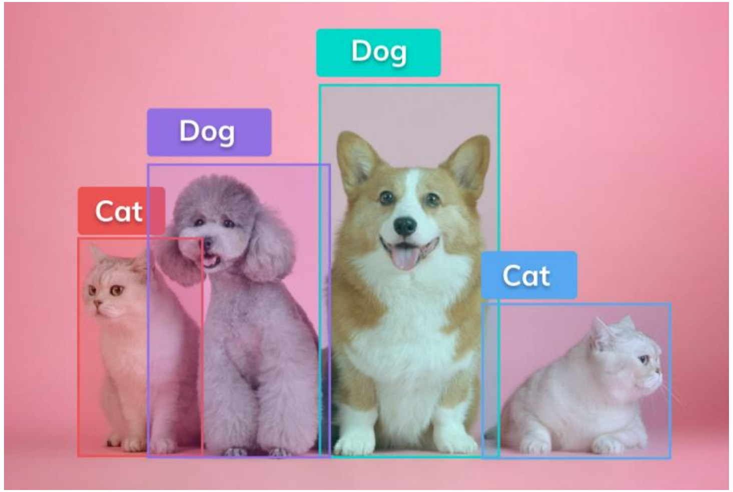
smiling
female
portrait
curly hair
happy
indoor
daylight

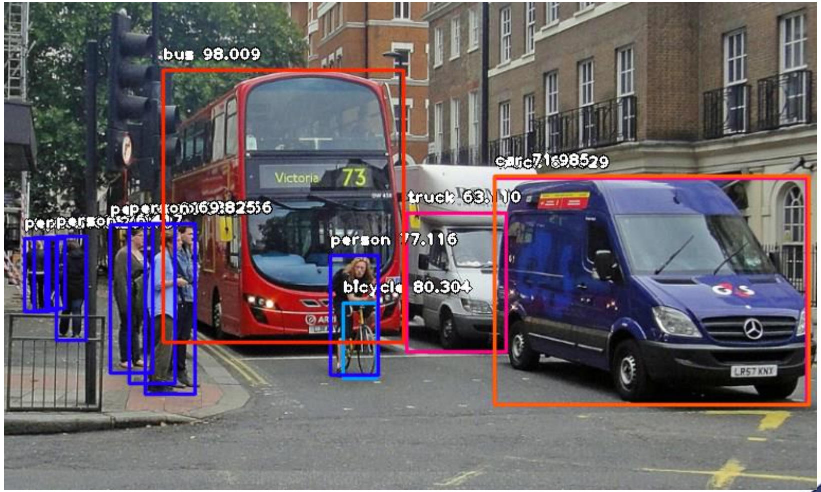
dog
plant

Muti-label Classification

Object Detection

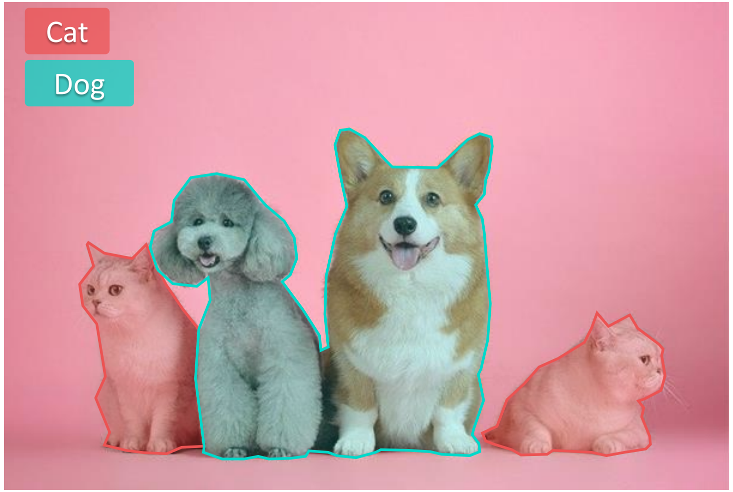
Object detection involves identifying and locating multiple objects within an image or video frame and is typically achieved by outputting a bounding box and a class label for each detected object.

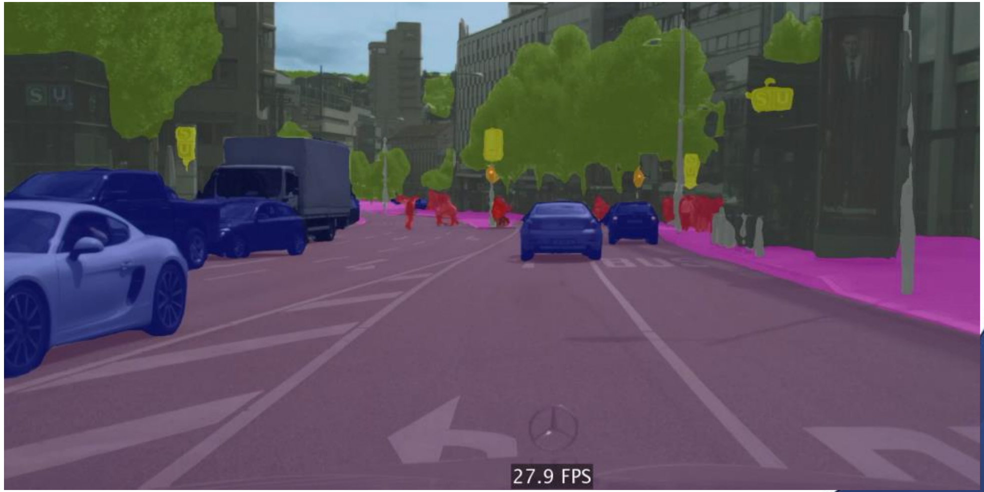




Semantic Segmentation

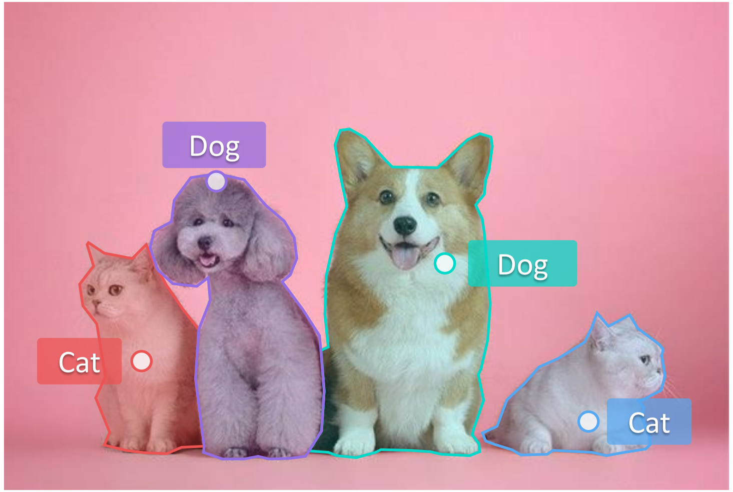
Semantic segmentation is the process of partitioning an image into segments, where each pixel is classified into a predefined category, effectively enabling the understanding of the image at the pixel level.

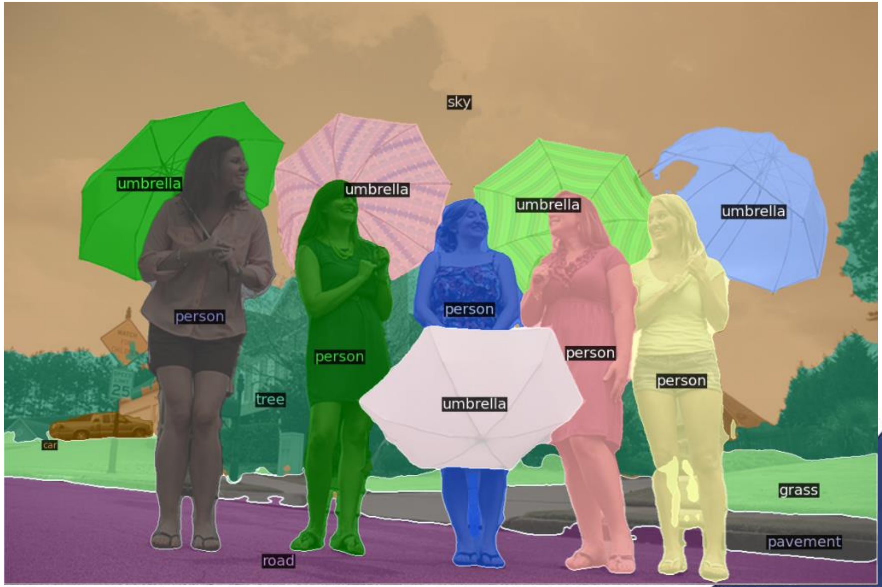




Instance Segmentation

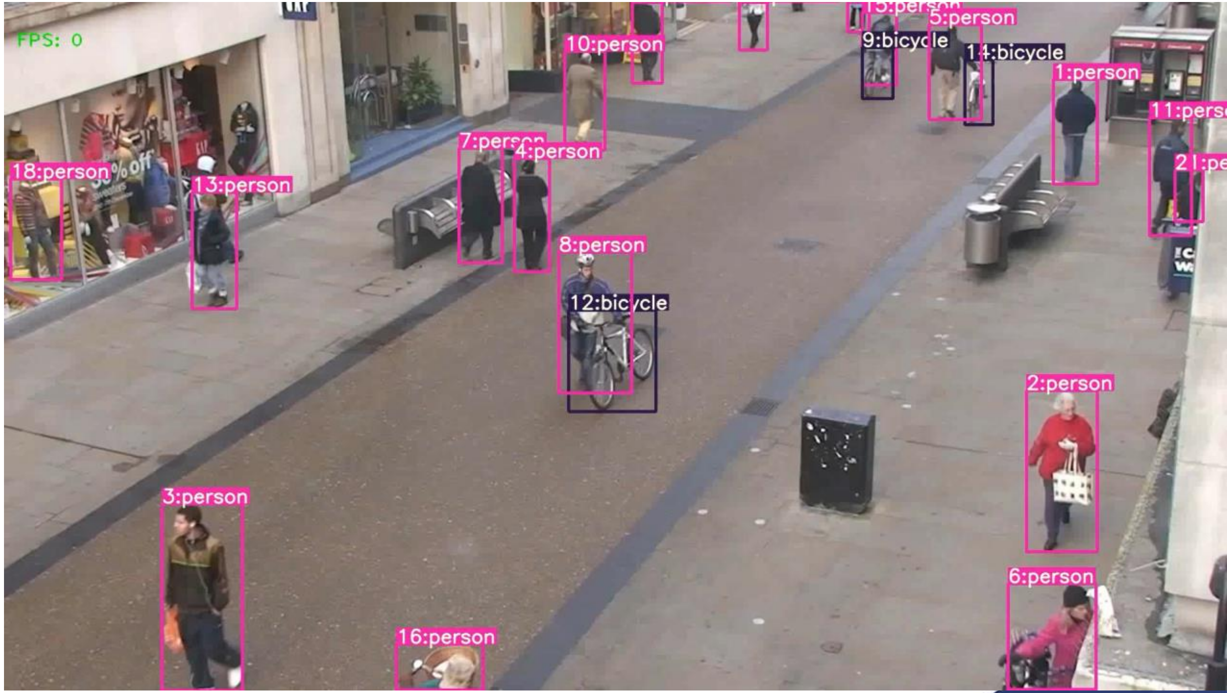
Instance Segmentation is like semantic segmentation but separating different instances of the same class.





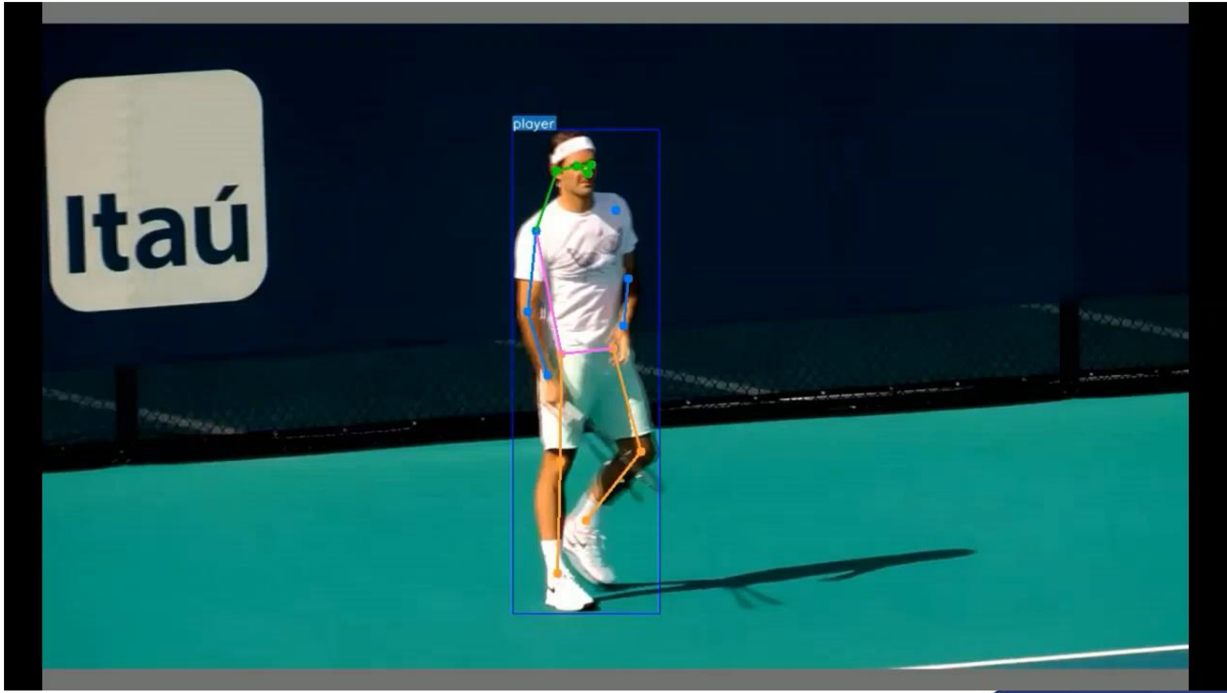
Object Tracking

Object tracking is the computational task of locating a moving object (or multiple objects) over time in a video sequence, often using the object's position and appearance information.



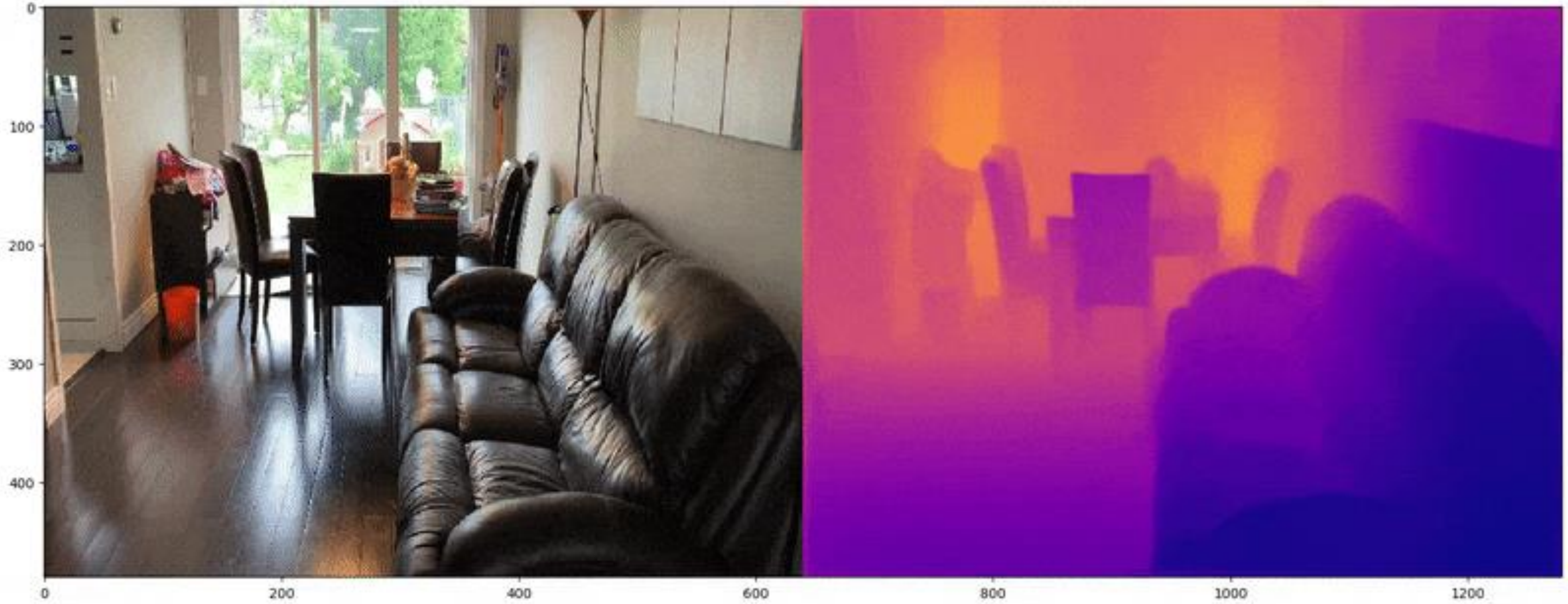
Pose Estimation

Pose estimation is the process of determining the position and orientation of objects or beings within images or videos, often focusing on identifying the arrangement of body parts in the case of human figures.



Depth Estimation

Depth estimation involves predicting the distance of each point in an image from the camera, creating a depth map that reflects the 3D structure of the scene.



Facial Recognition

Facial recognition is a technology that identifies or verifies a person's identity by analysing and comparing patterns based on facial features from images or video frames.

Integration of Multiple CV Technologies

Robots often require the integration of multiple computer vision technologies to effectively navigate, interact with, and understand their environments.

This multi-faceted approach can include object detection for identifying and locating items of interest, semantic segmentation for understanding the layout and categorization of space, depth estimation for gauging distances and navigating around obstacles, and SLAM for building and updating a map of their surroundings while tracking their own position.

Image Classification

Image Classification

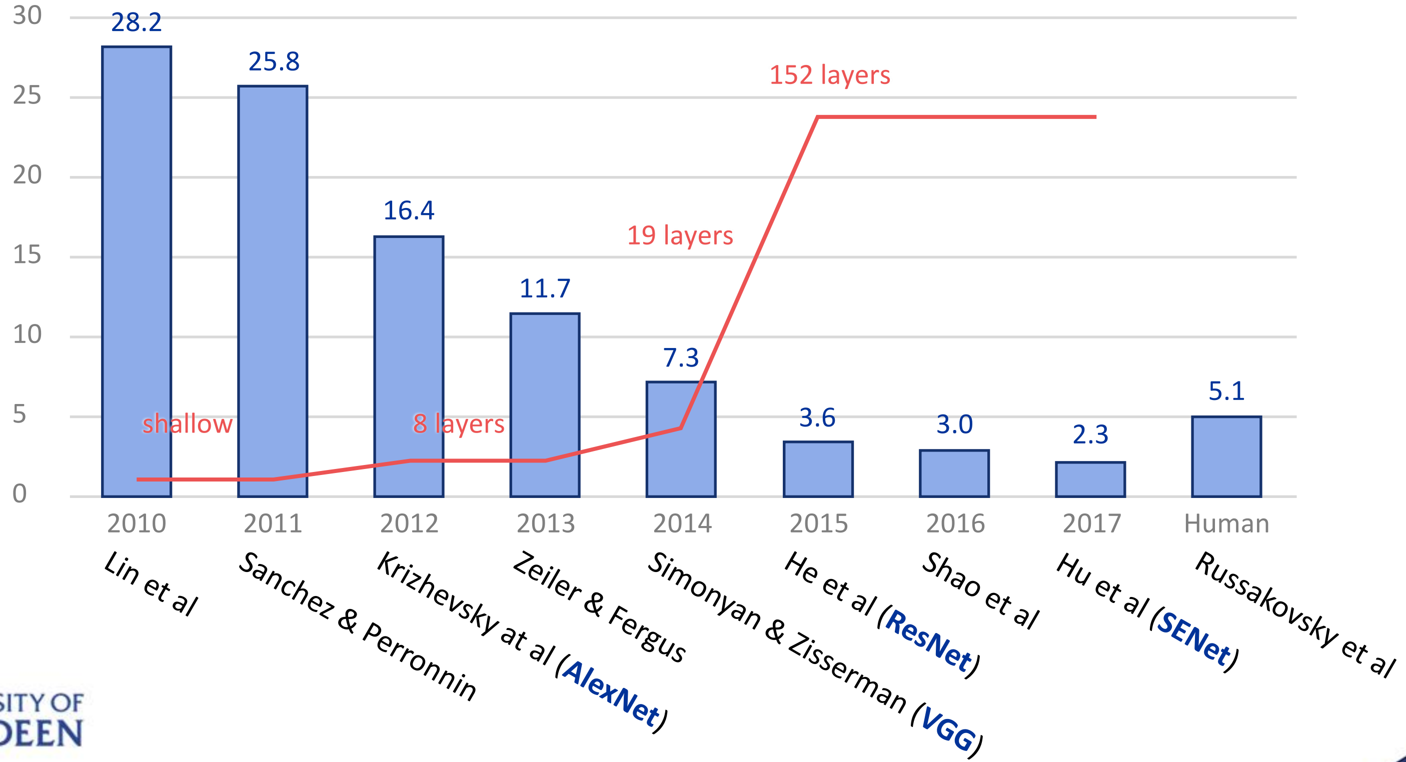
Image classification is the most basic task among all CV tasks.

Although we showed a very simple CNN for writing number recognition in the last session, the generalised image classification task is complex.

There are many approaches are proposed to achieve the task.

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual competition where researchers and algorithms compete to accurately classify and detect objects and scenes in ImageNet (a large dataset of images, containing millions of labelled images across thousands of categories).

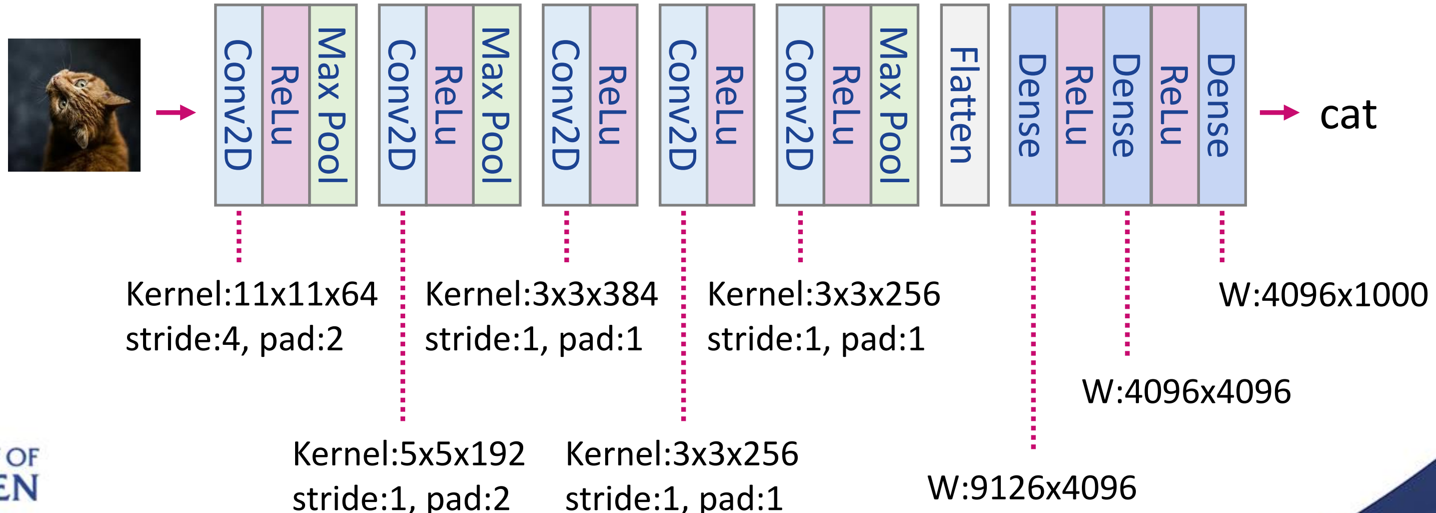
ILSVRC Winners



1495 UNIVERSITY OF ABERDEEN

AlexNet (Krizhevsky et al. 2012)

AlixNet is the first CNN-based model appeared in ILSVRC. Its architecture is quite similar to we showed in last lecture, but uses much larger kernels.



Conv2D ReLu Max Pool → Conv2D ReLu Max Pool → Conv2D ReLu → Conv2D ReLu → Conv2D ReLu Max Pool → Flatten → Dense ReLu Dense ReLu Dense → cat

Kernel:11x11x64
stride:4, pad:2

Kernel:5x5x192
stride:1, pad:2

Kernel:3x3x384
stride:1, pad:1

Kernel:3x3x256
stride:1, pad:1

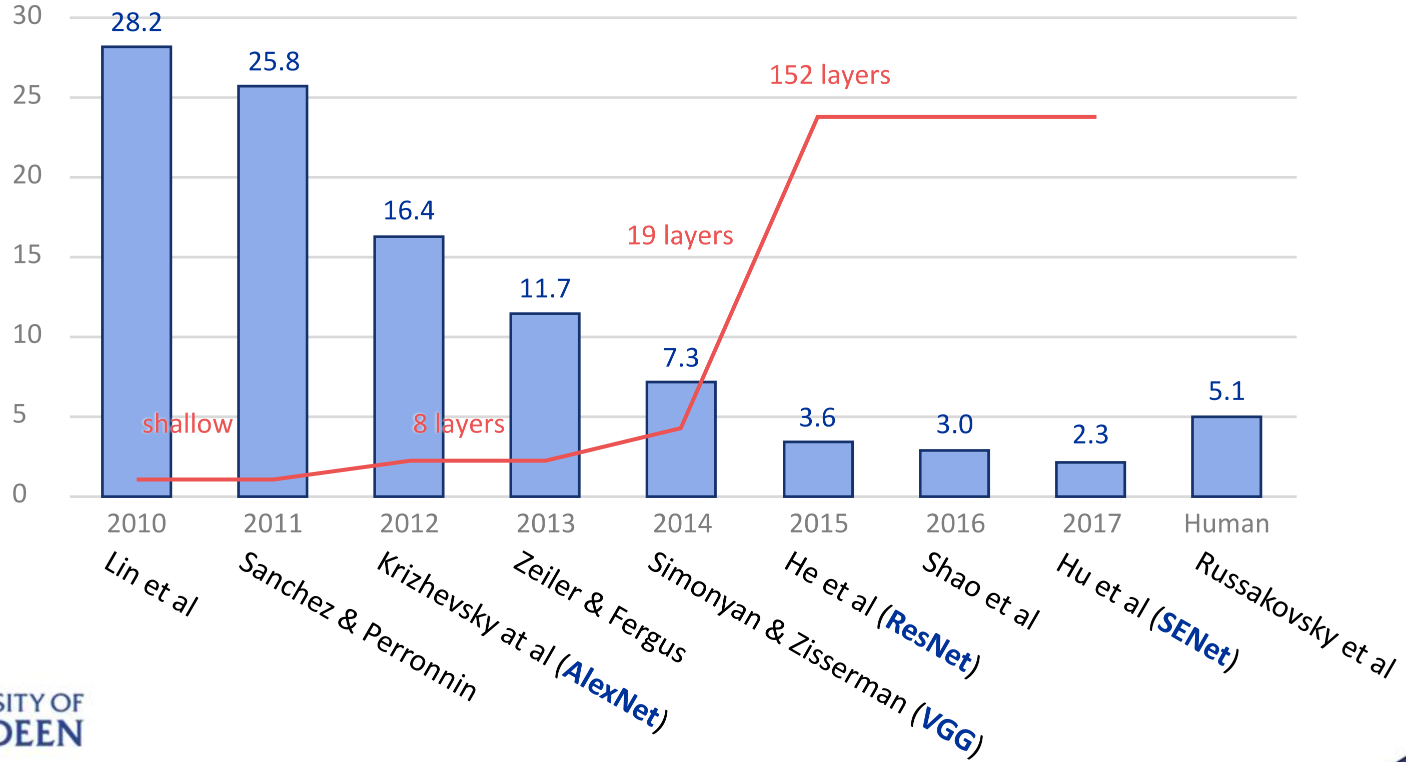
Kernel:3x3x256
stride:1, pad:1

W:9126x4096

W:4096x4096

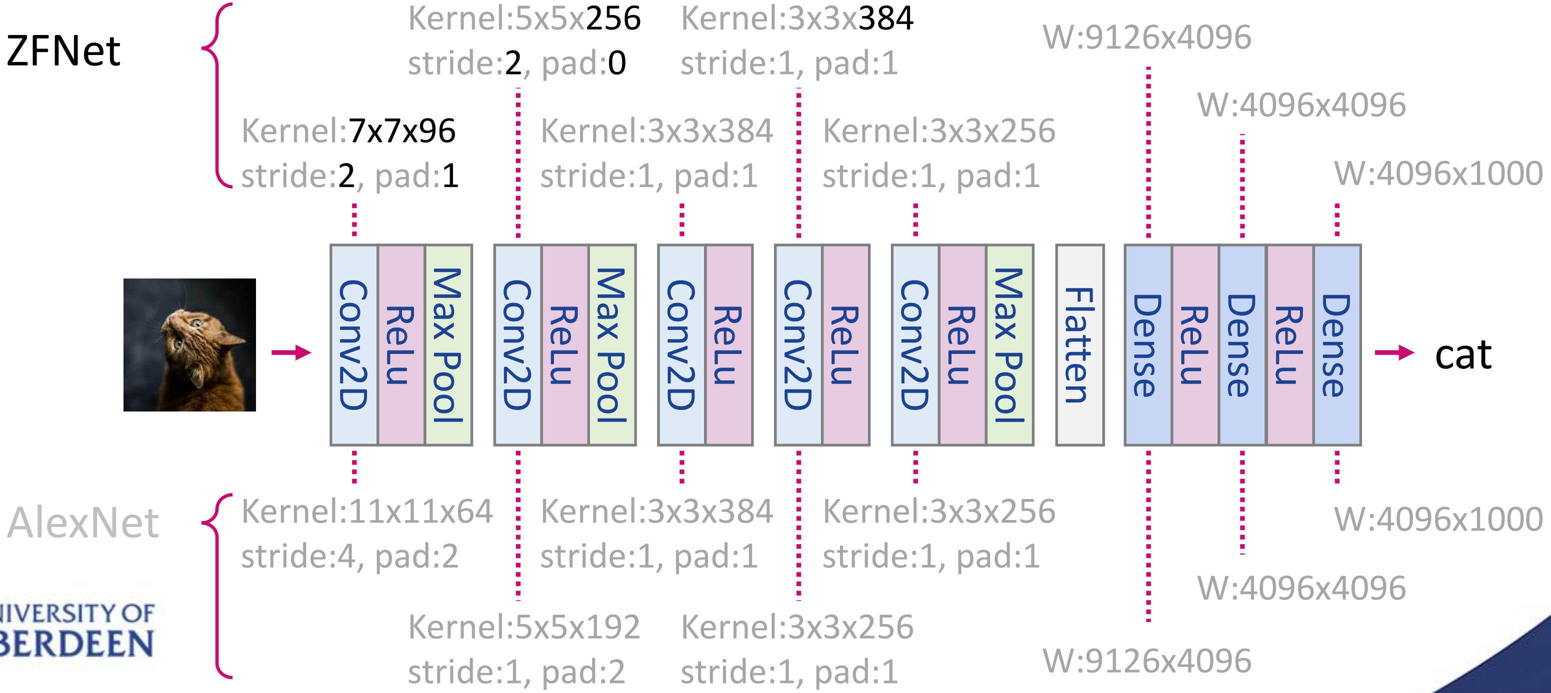
W:4096x1000

ILSVRC Winners



ZFNet (Zeiler and Fergus, 2013)

ZFNet has the same architecture with AlexNet, only with different kernel sizes.



ZFNet

Kernel:7x7x96
stride:2, pad:1

Kernel:5x5x256
stride:2, pad:0

Kernel:3x3x384
stride:1, pad:1

Kernel:3x3x384
stride:1, pad:1

Kernel:3x3x256
stride:1, pad:1

W:9126x4096

W:4096x4096

W:4096x1000

Conv2D ReLu Max Pool → Conv2D ReLu Max Pool → Conv2D ReLu → Conv2D ReLu → Conv2D ReLu Max Pool → Flatten → Dense ReLu Dense ReLu Dense → cat

AlexNet

Kernel:11x11x64
stride:4, pad:2

Kernel:5x5x192
stride:1, pad:2

Kernel:3x3x384
stride:1, pad:1

Kernel:3x3x256
stride:1, pad:1

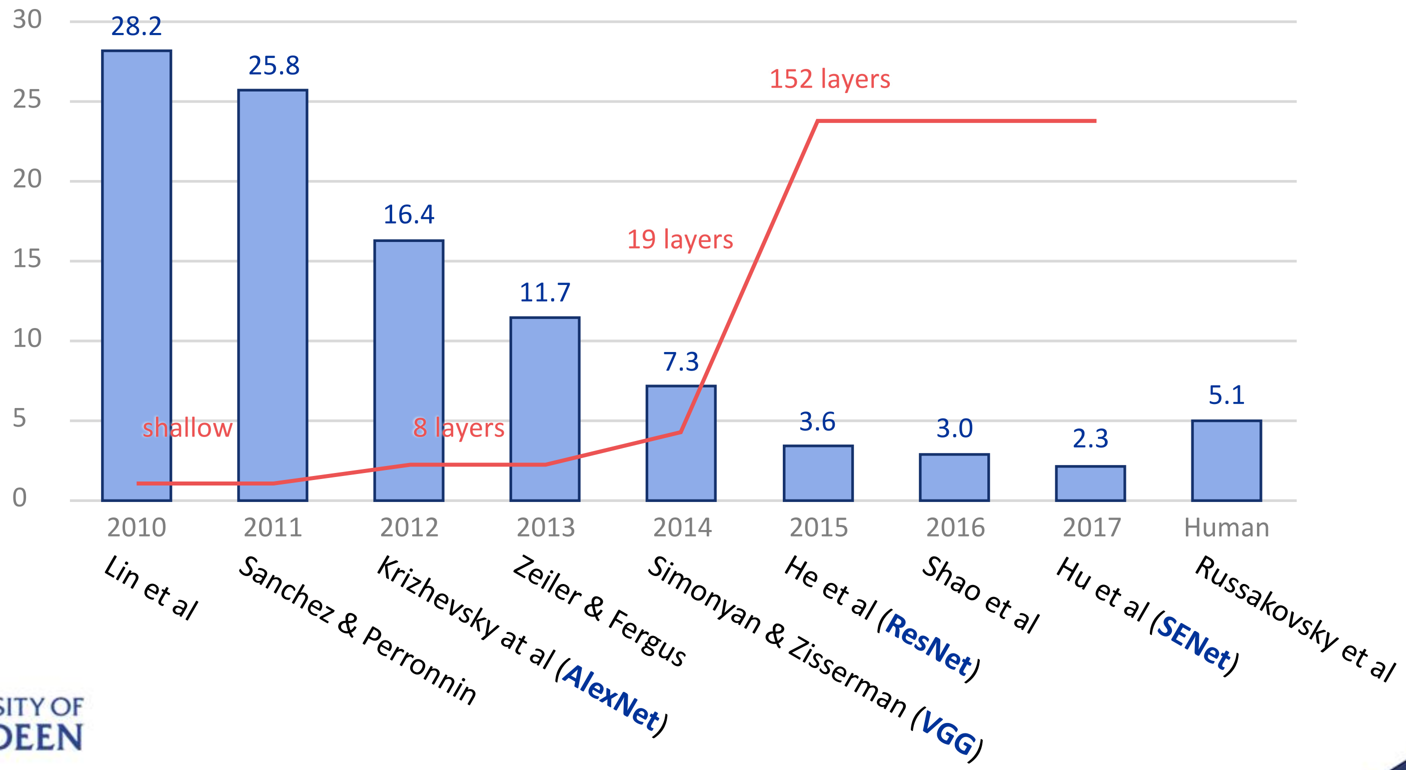
Kernel:3x3x256
stride:1, pad:1

W:9126x4096

W:4096x4096

W:4096x1000

ILSVRC Winners



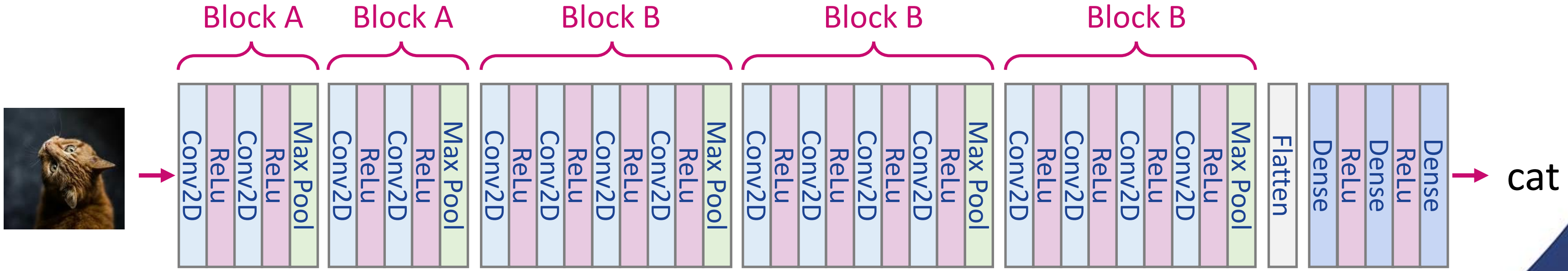
1495 UNIVERSITY OF ABERDEEN

VGG (Simonyan and Zisserman, 2014)

VGG is **deep** neural network. It uses block like architecture, and blocks of the same type use the same architecture.

All Conv2D uses 3x3 kernels (stride=1, pad=0), and all maxpool use 2x2 kernels.

VGG showed that the deep architecture can significantly improve model performance.

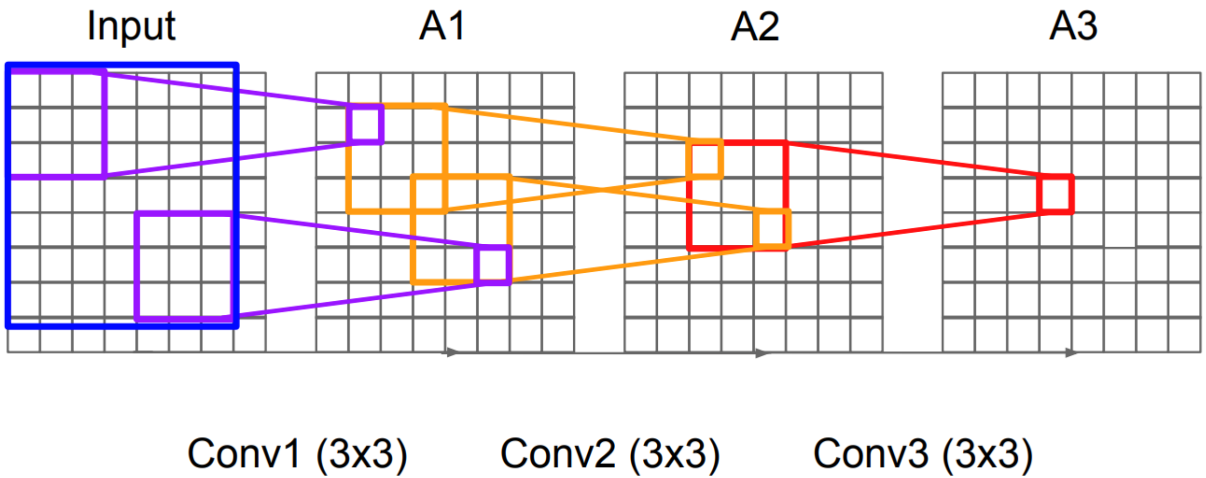


VGG (Simonyan and Zisserman, 2014)

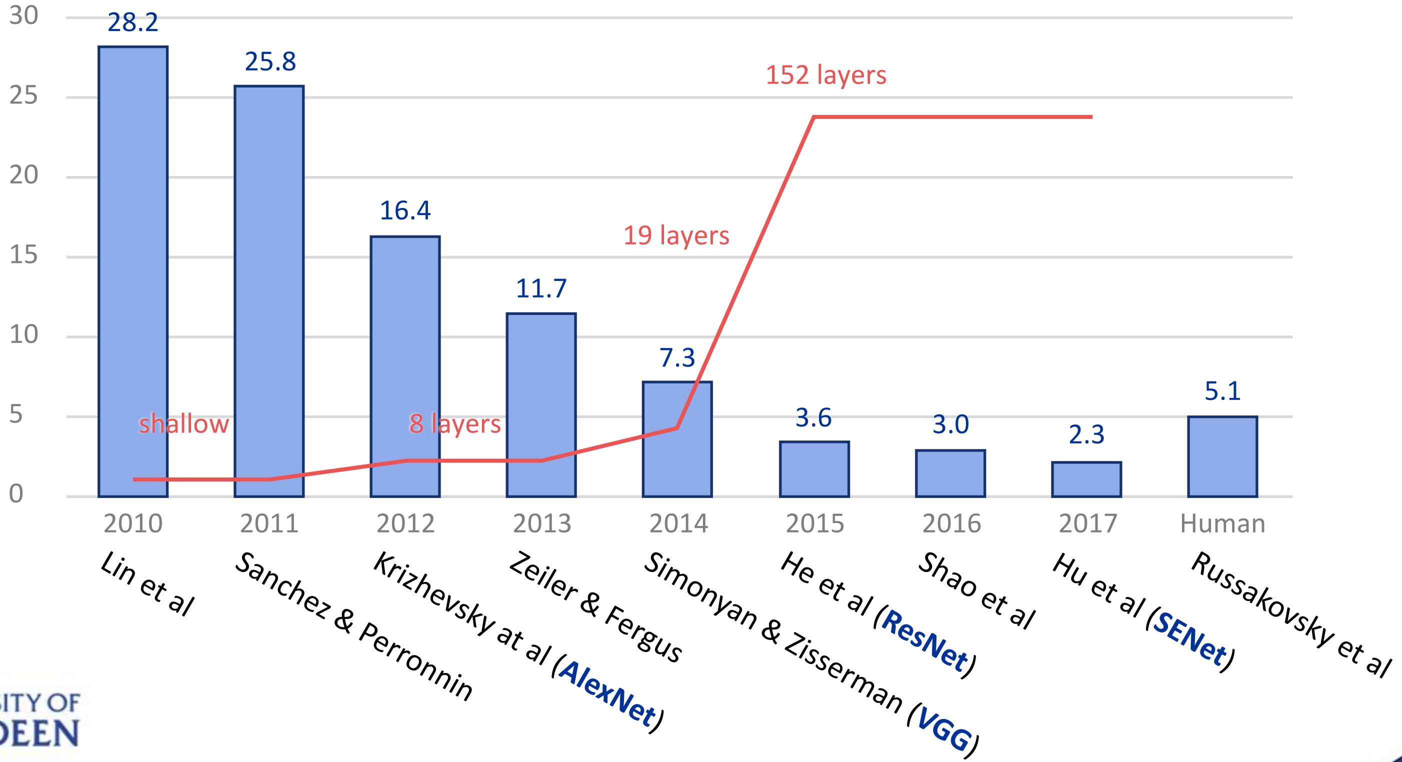
VGG utilizes smaller filters (i.e. 3x3 convolutional kernels) to enhance the network's depth.

By stacking multiple layers, VGG can **increase non-linearity**, while maintaining a reasonable number of parameters (fewer parameters than a 7x7 Conv2D layer i.e. 3x3x3<7x7).

Finally, stack of three 3x3 conv (stride 1) layers has same effective receptive field as one 7x7 conv layer.



ILSVRC Winners

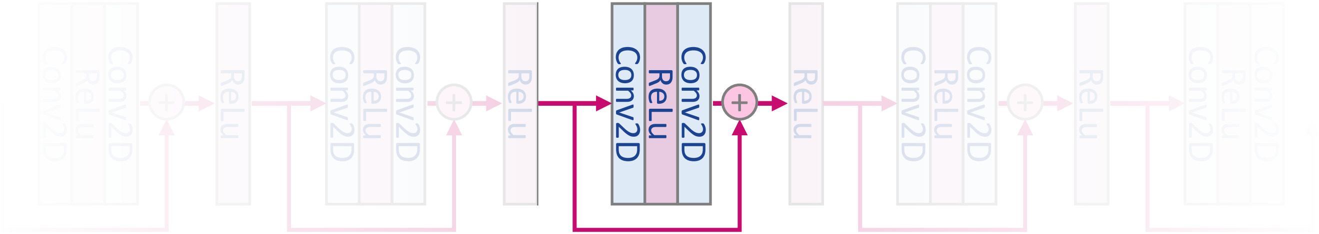


ResNet (He et al., 2015)

ResNet addresses the vanishing gradient problem and allows for the training of networks that are much much deeper than was previously feasible.

Its key innovation is the introduction of **residual blocks** with skip connections that pass the input of a block across a few layers and add it back to the output, facilitating the training process by allowing gradients to flow through the network more effectively.

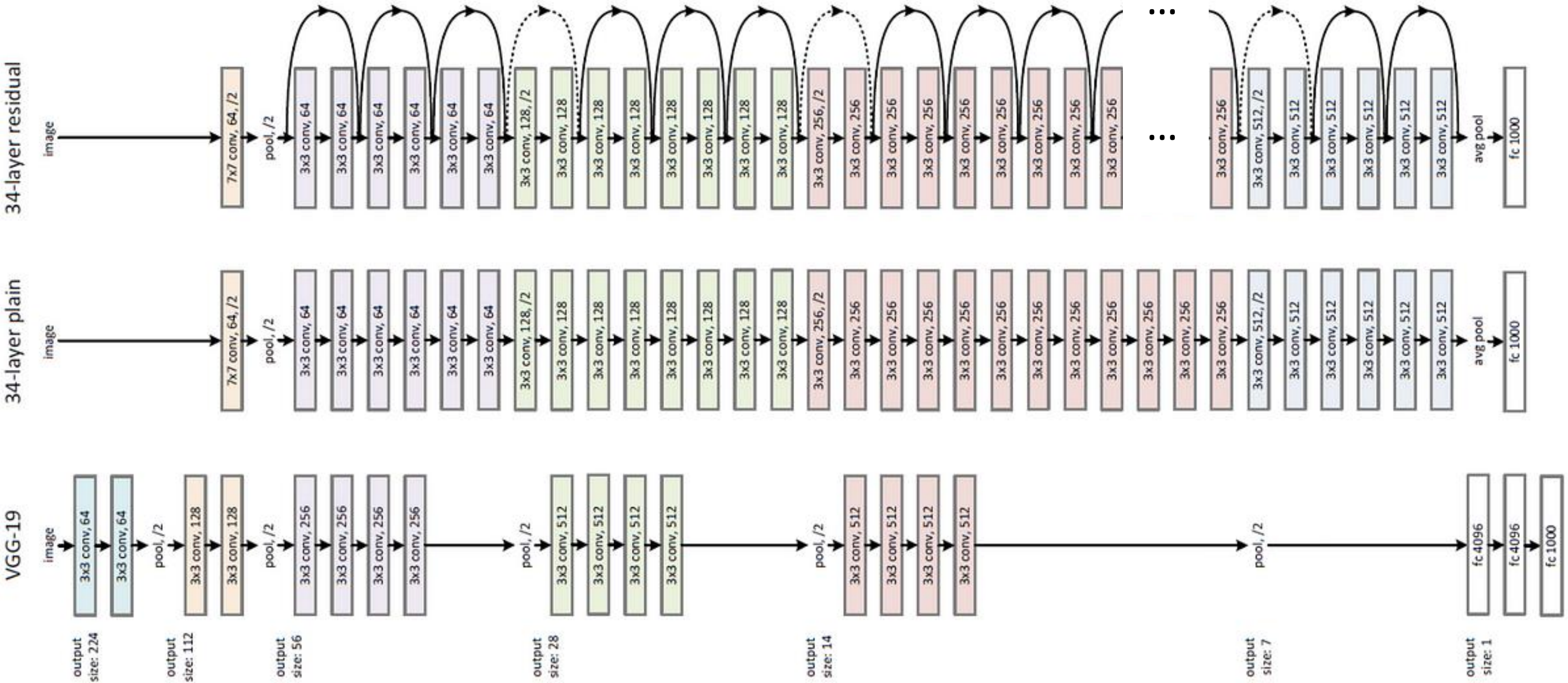
It revolutionised the way deep neural networks are constructed.



Residual Block

ResNet (He et al., 2015)

Architecture of ResNet has 307 layers (including 152 Conv2D layers).



SENet (Hu et al. 2017)

SENet, short for Squeeze-and-Excitation Networks, is an architectural enhancement to convolutional neural networks (CNNs) introduced by Hu et al. in 2017.

SENet allows the network to focus more on informative features and suppress less useful ones, leading to significant improvements in performance across a variety of tasks, including image classification, detection, and segmentation, without significantly increasing computational complexity or model size.

Conclusion

- The purpose of the model training process is to iteratively adjust the model's parameters to minimize the difference between its predicted outputs and the actual data, thereby improving its accuracy and performance on a given task.
- The main computer vision tasks include: Image Classification, Object Detection, Semantic Segmentation, Instance Segmentation, Object Tracking, Pose Estimation, Depth Estimation, Facial Recognition

Conclusion

For the image classification models:

AlexNet showed that you can use CNNs to train Computer Vision models.

ZFNet shows the importance of parameter adjustment.

VGG shows that bigger networks work better.

ResNet showed us how to train extremely deep networks.

- Limited only by GPU & memory!
- Showed diminishing returns as networks got bigger.
- After ResNet: CNNs were better than the human metric.