Deep Learning for Robotic Vision

Wei Qi Yan

Auckland University of Technology

Table of Contents

1. CNNs or ConvNets
2. YOLO
3. RNNs (LSTM, GRU)
4. Vision Transformer (ViT)

CNNs or ConvNets

Deep Learning

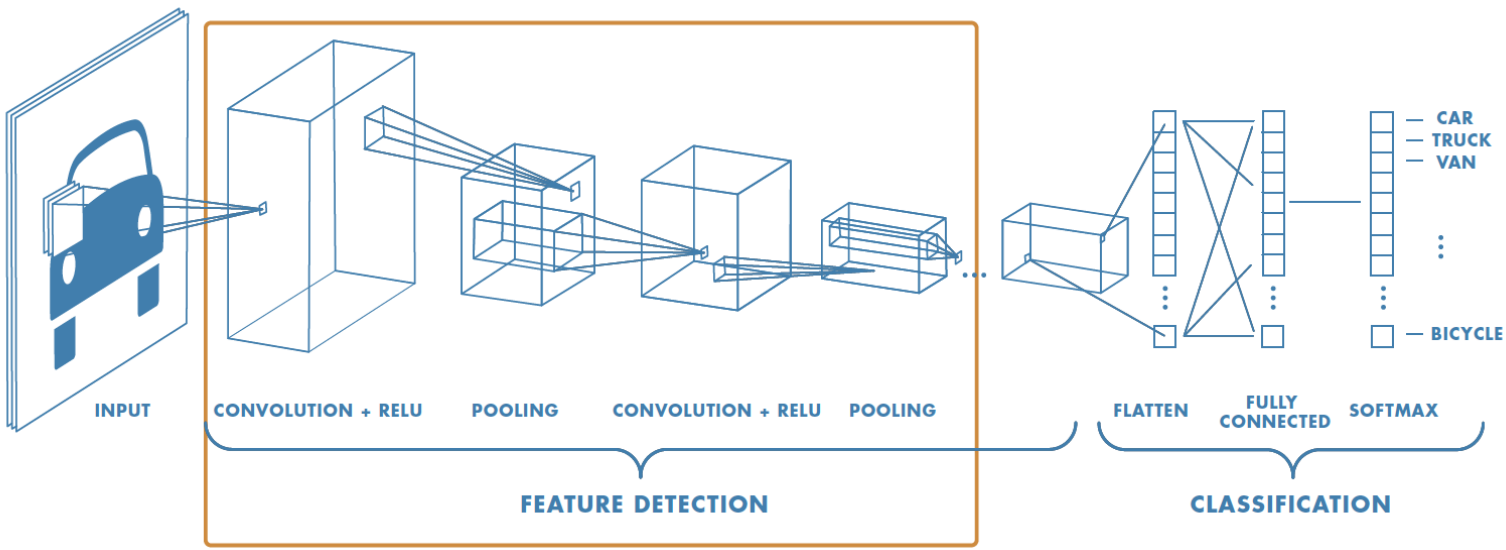
- Deep learning is a type of machine learning methods in which a model is trained to perform classification tasks.
- Deep learning is usually implemented by using a neural network architecture.
- The term "deep" refers to the number of layers in the network.
- Conventional neural networks contain only two or three layers, while deep nets can have more.

The State-of-the-Art Technology

- Easy ways to access massive sets of labeled data
- Increased computing power (e.g., GPU, FPGA, etc.)
- Pretrained models created by experts

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

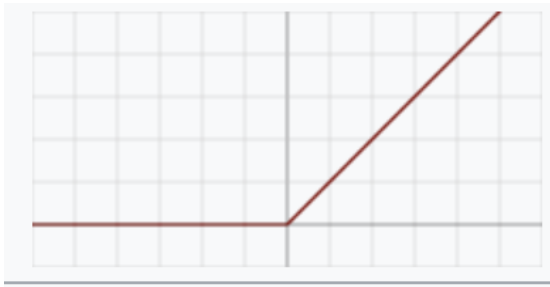
CNNs or ConvNets



Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Three Operations

- *Convolution* puts the input through a set of convolutional filters.
- *Pooling* simplifies the output through nonlinear downsampling.
- *ReLU* (Rectified Linear Unit) allows for fast and effective training by mapping negative values to zero and maintaining positive ones.

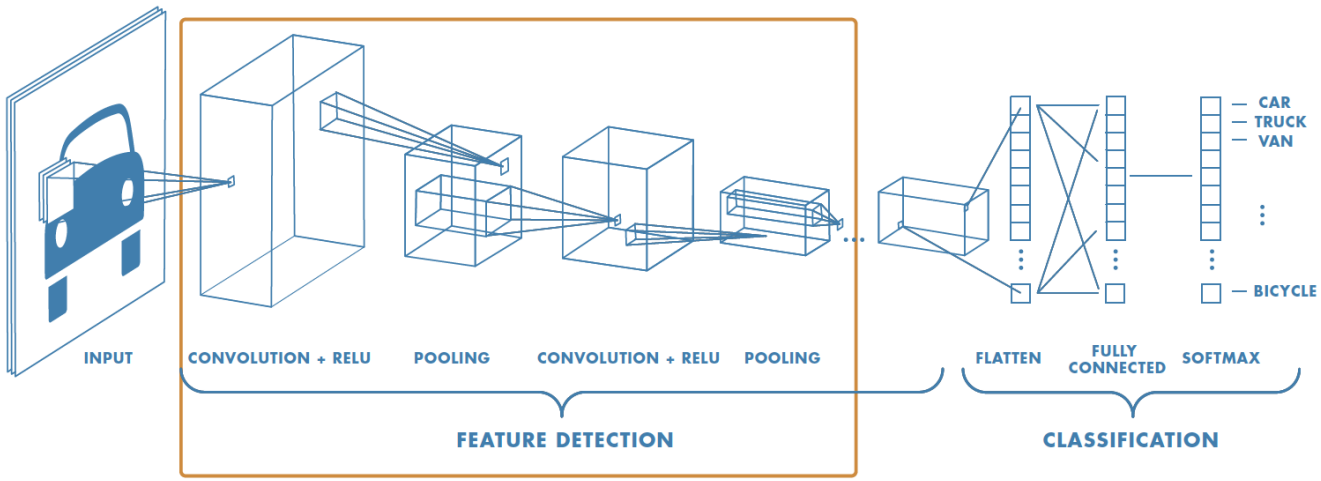


$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

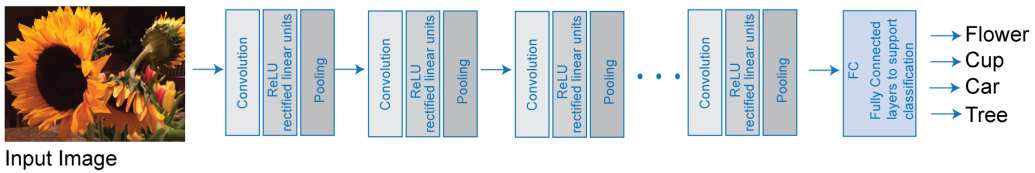
Web: https://en.wikipedia.org/wiki/Activation_function

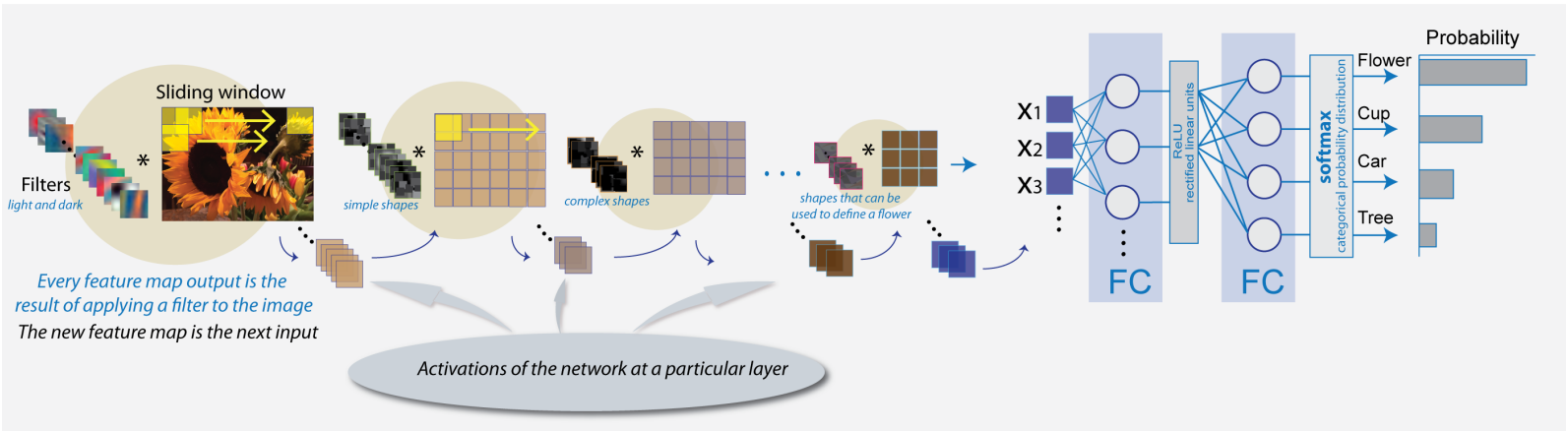
The Classification

- A fully connected layer (FC) outputs a vector of k dimensions where k is the number of classes that the net is able to predict.
- The vector contains the probabilities for each class of any images being classified.
- The final layer of the CNN architecture uses a softmax function to provide the classification output.



Convolutional Neural Networks





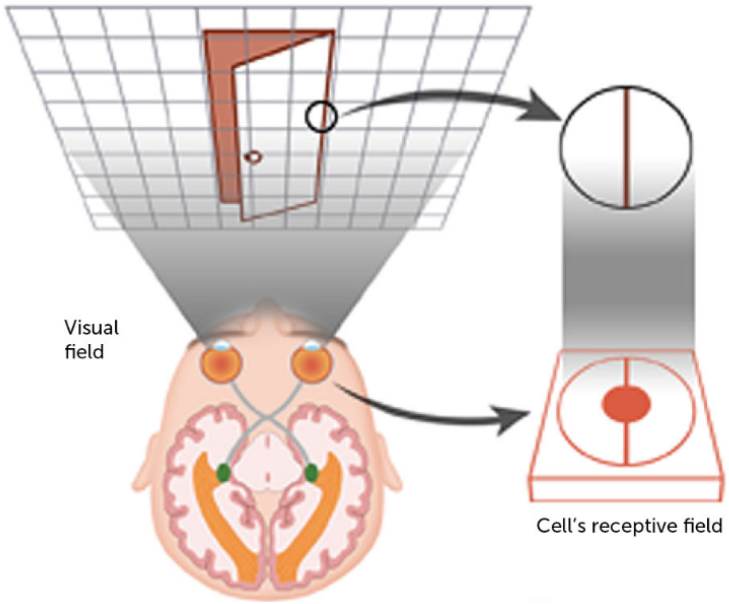
CNNs or ConvNets

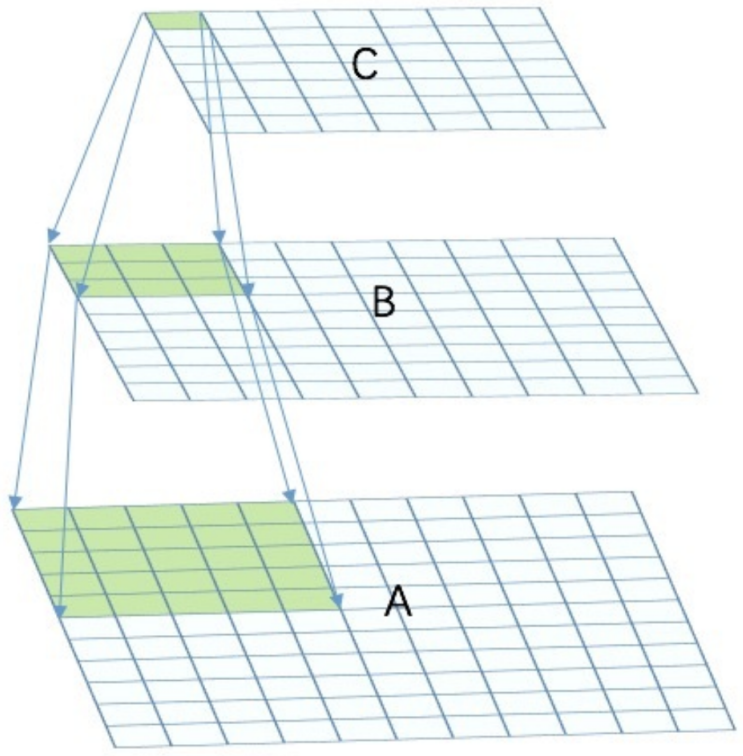
Convolutional Neural Networks

- ConvNets are inspired from the biological structure of a visual cortex, which contains arrangements of simple and complex cells.
- These cells are activated based on the subregions of a visual field, i.e., **receptive field**.
- A ConvNet reduces the number of parameters with the number of connections, shared weights, and downsampling.
- A ConvNet consists of multiple layers, such as convolutional layers, max pooling or average pooling layers, and fully connected layers.

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Receptive Field





Web: https://brainconnection.brainhq.com/2004/03/06/overview-of-receptive-fields/

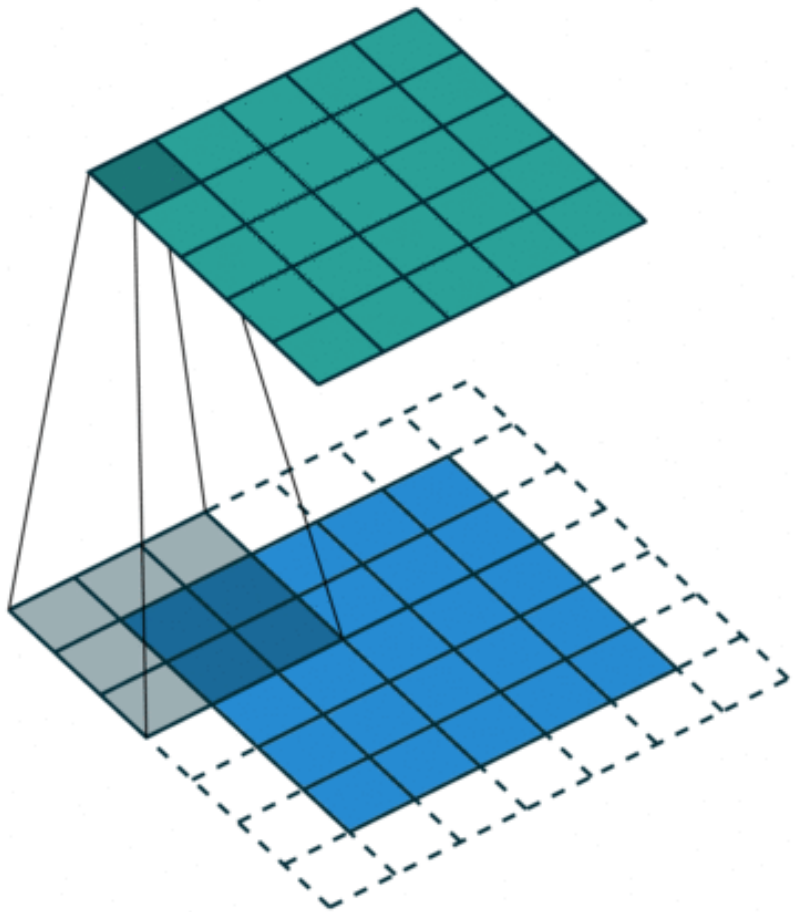
CNNs or ConvNets

Concepts

- **Receptive field**: A region of the original image corresponding to a pixel of the feature map of a filter or kernel
- **Feature map**: The output of convolution operations
- **Stride**: The step length of convolution operations
- **Fsize**: The size of convolution kernels or filters
- **Padding**: The filled region of an image boundary
- **Top to down**: From a deep layer to its next layer

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Padding



Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

CNNs or ConvNets

Convolutional Neural Networks

- The input layer defines the size of the input of a convolutional neural network and contains the raw values of the input.
- A convolutional layer consists of neurons that connect to subregions of the input or the outputs of the layer, which learns the features localized by these regions.
- A set of weights is called a filter, which moves along the input image vertically and horizontally, repeating the same computation.
- Padding is basically adding rows or columns of zeros to the borders of an image input, which helps to control the output size of the layer.

Web: https://au.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Convolutional Neural Networks

- Batch normalization layers normalize the activations and gradients propagating through a neural network, making network training an easier optimization problem.
- A ReLU layer performs a threshold operation to each element $y = \max(x, 0)$.
- A leaky ReLU layer multiplies input values less than zero by a fixed scalar, allowing negative inputs to "leak" into the output.
- A max pooling layer returns the maximum values of rectangular regions of its input.
- The average pooling layer outputs the average values of rectangular regions of its input.

Convolutional Neural Networks

- A dropout layer randomly sets the layer's input elements to zero with a given probability.
- All neurons in a fully connected layer connect to all the neurons in the previous layer. This layer combines all of the features learned by the previous layers across the image to identify the larger patterns
- The softmax function, i.e., normalized exponential function, is the output unit activation function after the last fully connected layer.
- A regression output layer must follow the final fully connected layer. The default loss function for a regression layer is the mean squared error.

Evaluations

- A full pass through the whole dataset is called an *epoch*.
- Iteration in deep learning is the number of batches needed to complete one epoch.
- What a larger *learning rate* is gradually reduced during optimization enables smaller steps towards to optimum value.
- Performing *validation* at regular intervals during training can determine whether the network is *overfitting* over the training data.
- To check whether a network is overfitting, compare *training loss* and *accuracy* corresponding to validation metrics.

Questions?

Questions?

In deep learning, the term “deep” refers to,

1. the number of layers in the network.
2. the number of neurons in the network.
3. the number of activation functions in the network.
4. the number of iterations in the network.

The right answer is:___

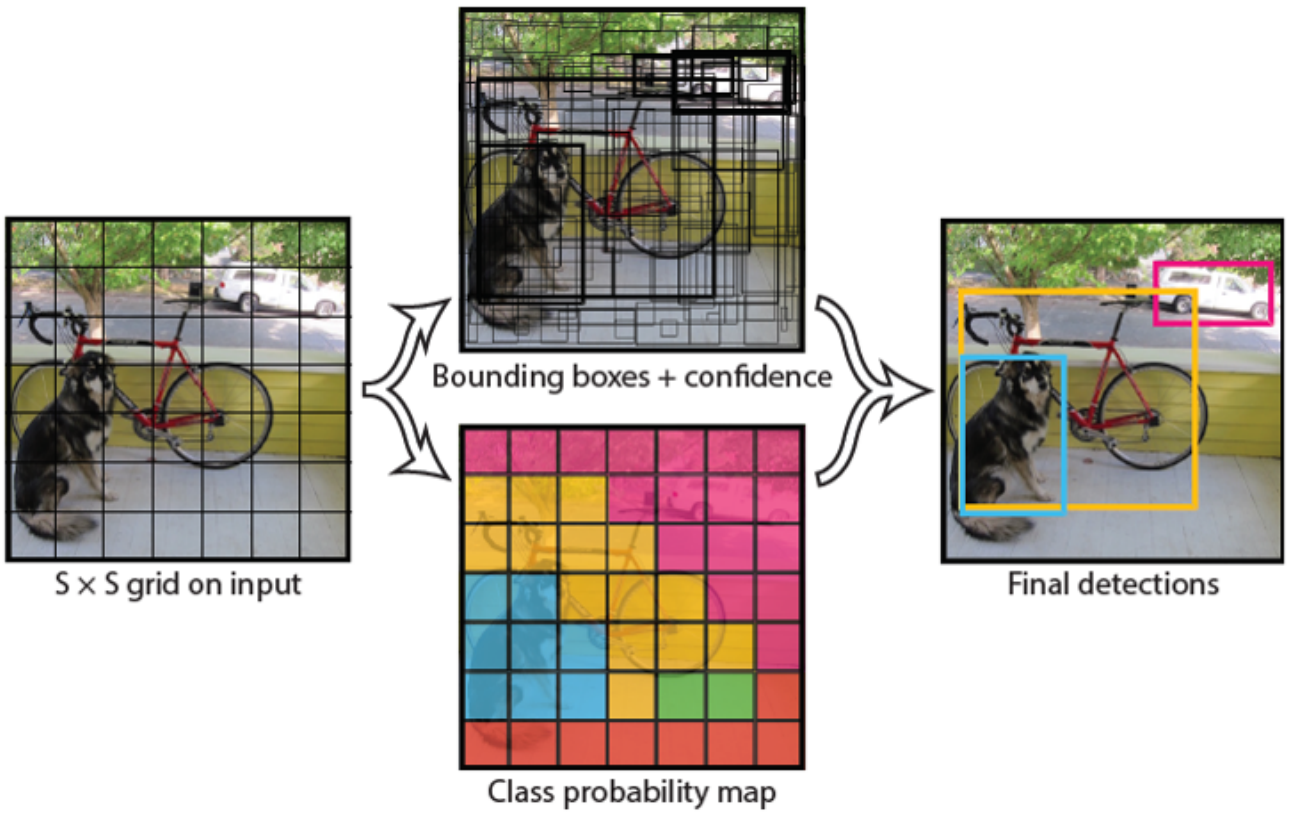
Questions?

YOLO: You Only Look Once

- YOLO: A single neural network predicts bounding boxes and class probabilities directly from full images.
- YOLO is trained based on full images that directly optimizes detection performance.
- YOLO achieves more than twice of the mean average precision of other real-time methods.
- YOLO sees the entire image during training and testing time so that it encodes **contextual information** of all classes.

Redmon, J. et al. (2016) You Only Look Once: Unified, real-time object detection. IEEE CVPR.

YOLO: You Only Look Once



Redmon, J. et al. (2016) You Only Look Once: Unified, real-time object detection. IEEE CVPR.

YOLO Models

YOLO: You Only Look Once

- YOLO segments the input image into $S \times S$ grids.
- Each grid cell predicts bounding boxes and confidence scores for those boxes.
- Each bounding box consists of 5 predictions (x, y, w, h) and confidence.
- Each grid cell also predicts conditional class probabilities
- YOLO multiplies the conditional class probabilities and the individual box confidence predictions as the class-specific confidence scores.
- The class-specific confidence scores encode both the probability of that class appearing in the box and the predicted box fits the object.

J., Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR'16.

YOLO Models

YOLO: You Only Look Once

- YOLO **predicts** what objects present and where they are.
- A single convolutional network simultaneously **predicts** multiple bounding boxes and class probabilities for those boxes.
- YOLO is trained based on full images that directly optimizes detection performance.
- YOLO runs a neural network on a new image at testing time to predict detections.
- YOLO is highly generalizable and is less likely to break down when applied to new domains or unexpected inputs.

Redmon, J. et al. (2016) You Only Look Once: Unified, real-time object detection. IEEE CVPR.

YOLO Models

YOLO: You Only Look Once

- YOLO uses regression.
- 45 frames per second, mAP 57.9%
- A picture is divided into 7×7 blocks; objects with confidence and coordinates are detected in each block.
- Small object could not be detected.

J. Redmon, et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. CVPR'16.

YOLO

MATLAB YOLOv2

- YOLOv2 uses a single stage object detection network.
- YOLOv2 is faster than other two-stage deep learning object detectors, such as Faster R-CNN.
- YOLOv2 runs a CNN in deep learning based on an input image to produce network prediction.
- YOLOv2 uses *anchor boxes* to detect classes of objects in an image.

YOLOv2 Predictions

- Intersection over Union (IoU): Predict the objectiveness score of each anchor box.
- Anchor box offset: Refine the anchor box position.
- Class probability: Predict the class label assigned to each anchor box.

Web: https://au.mathworks.com/help/vision/ug/getting-started-with-yolo-v2.html

Anchor Box

- Anchor boxes are a set of predefined bounding boxes.
- Each anchor box is tiled across the image.
- The use of anchor boxes enables a network to detect multiple objects, objects of different scales, and overlapping objects.

Web: https://au.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html

YOLO Models

Anchor Box

- Anchor boxes eliminate the need to scan an image with a sliding window that computes a separate prediction at every potential position.
- Using anchor boxes can process an entire image at once, making real-time object detection possible.
- The use of anchor boxes replaces and drastically reduces the cost of the sliding window approach.
- Using anchor boxes, we design efficient visual object detectors to encompass all three stages (detection, feature encoding, and classification).

Web: https://au.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html

MATLAB YOLOv2

Web: https://au.mathworks.com/help/vision/ug/getting-started-with-yolo-v2.html

YOLO Models

MATLAB YOLOv3

- YOLOv3 improves upon YOLOv2 by adding detection at multiple scales to detect smaller objects.
- The loss function of YOLOv3 for training is separated into mean squared error for bounding box regression and binary cross entropy for object classification to improve detection accuracy.
- YOLOv3 detector utilizes anchor boxes estimated through training data to have better initial priors and predict the boxes accurately.

Web:
https://au.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html

YOLO Models

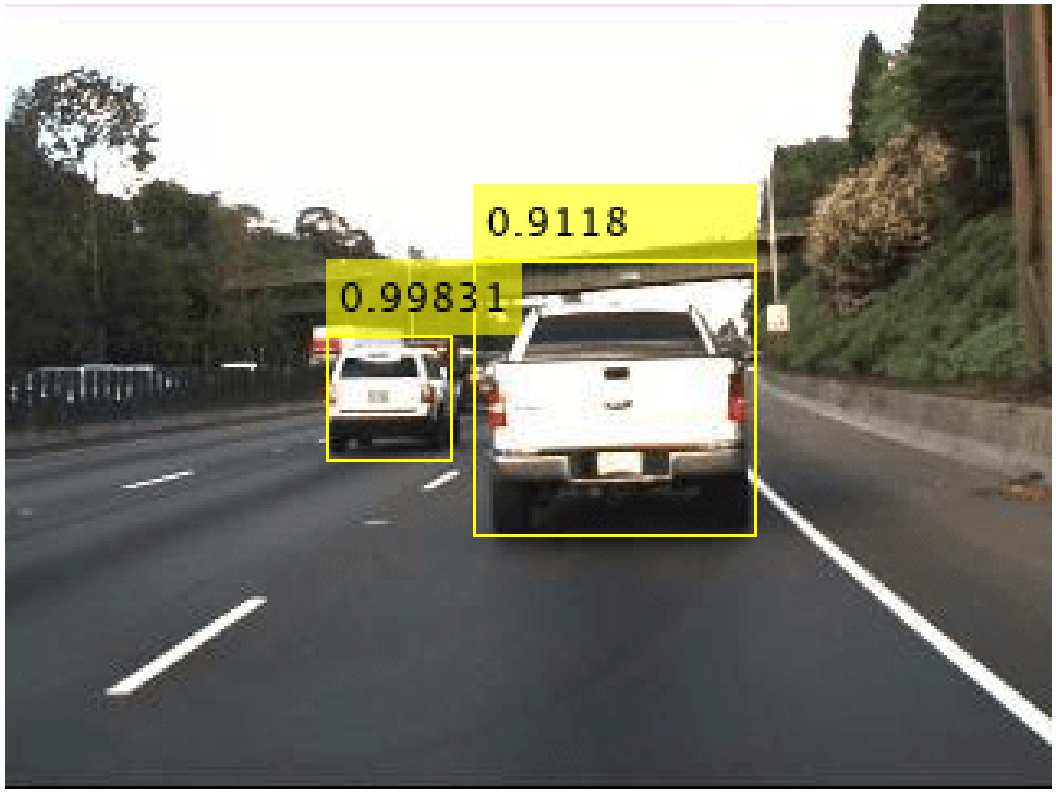
MATLAB YOLOv4

- YOLOv4 is a one-stage object detection network that is composed of three parts: Backbone, neck, and head.
- The backbone of the YOLOv4 network acts as the feature extraction network that computes feature maps from the input images.
- The neck connects the backbone and the head, which is composed of a spatial pyramid pooling (SPP) module and a path aggregation network (PAN).
- The head processes the aggregated features and predicts the bounding boxes, objectness scores, and classification scores.

Web: https://au.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html

YOLO Models

MATLAB YOLOv4



Web: https://au.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html

Questions?

YOLO Models

Questions?

In deep learning, YOLO predicts:

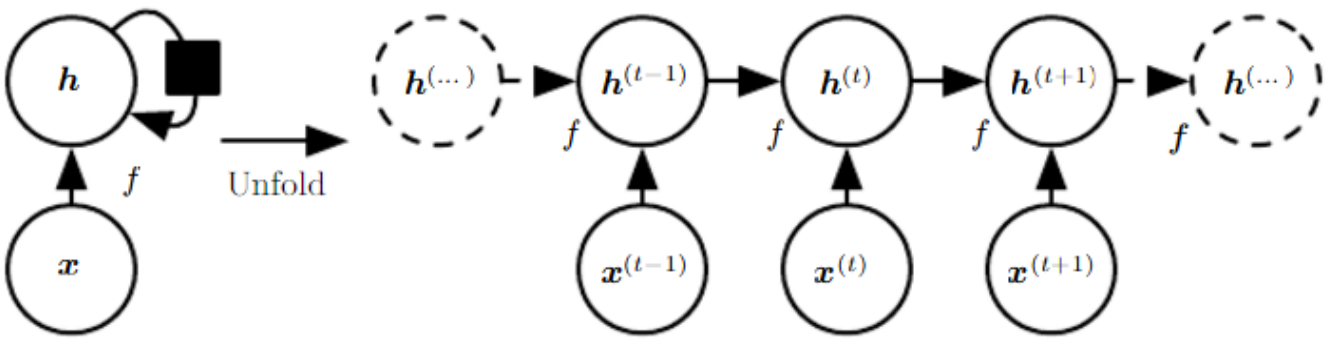
1. what objects present and where they are.
2. what objects present only.
3. where objects are only.
4. none of the given options.

The right answer is:___

Questions?

Recurrent Neural Networks

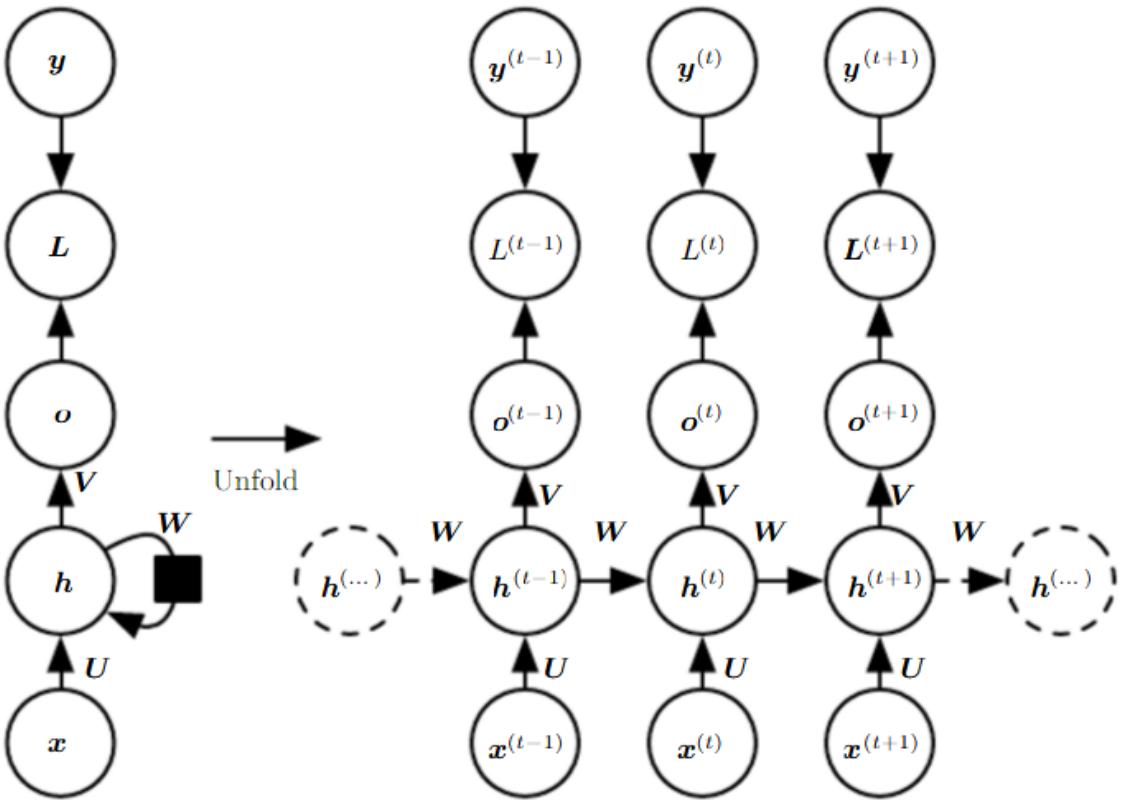
RNNs are a family of neural networks for processing sequential data, which is a dynamical system. It is possible to use the same transition function with the same parameters at every time step.



Goodfellow, I. (2016) Deep Learning. MIT Press.

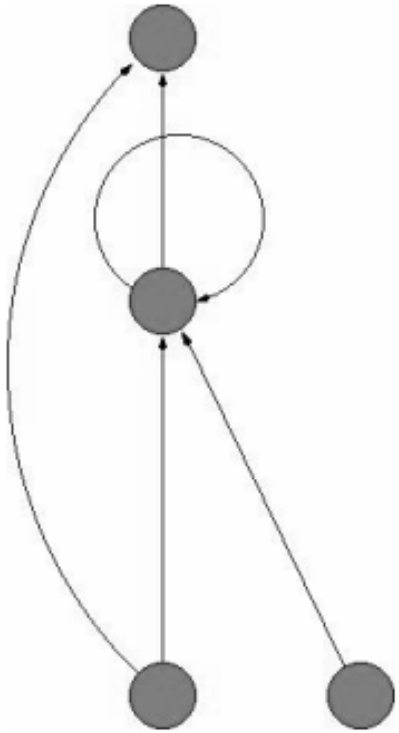
Unfolded RNNs

The notations are: Input x, state h, output o, loss function L, training target y, weights U, V, and W.



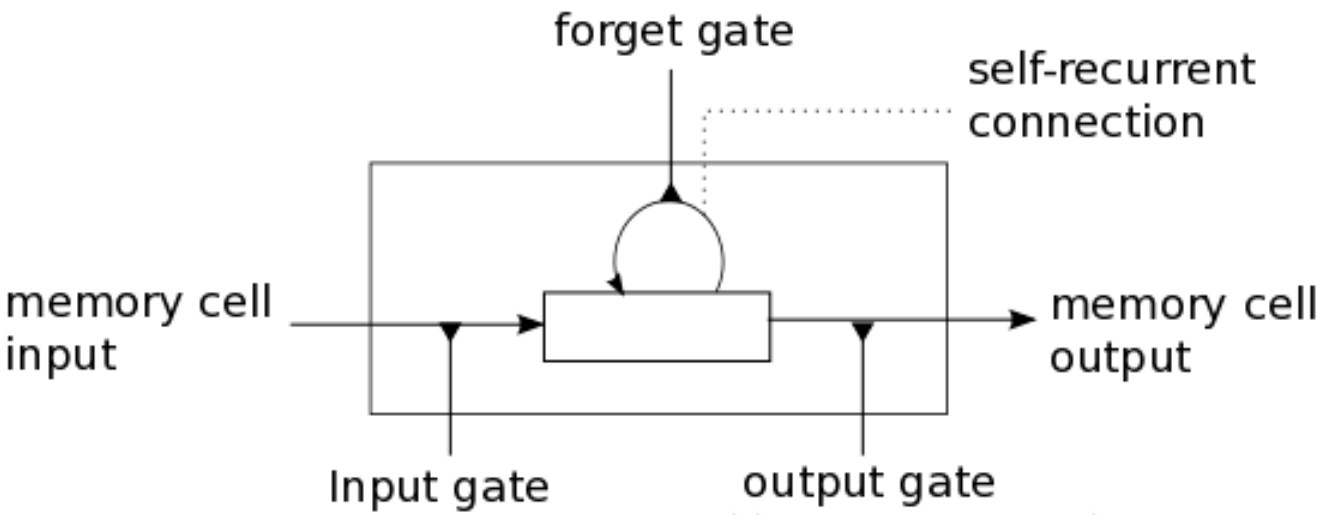
Goodfellow, I. (2016) Deep Learning. MIT Press.

LSTM: Long Short-Term Memory



Hochreiter, S. et al. (1997) Long short-term memory, Neural Computation, 9(8):1735-1780.

LSTM: Long Short-Term Memory



LSTM: Long Short-Term Memory

- LSTM is a model for short-term memory which can last for a long period of time.
- An LSTM unit consists of four gates: Input gate, cell, forget gate, and output gate.
- LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events.
- LSTM (memory) cell stores a value (or state), for either long or short time periods.
- LSTM gates compute an activation, often using the logistic function.
- LSTM was developed to deal with the exploding and vanishing gradient problems.

Web: https://en.wikipedia.org/wiki/Long_short-term_memory.

MATLAB LSTM for Classification

- An LSTM network is a type of RNNs that can learn long-term dependencies between time steps of sequence data.
- A sequence input layer inputs sequence or time series data into the network.
- To predict class labels, the network ends with a fully connected layer, a softmax layer, and a classification output layer.

https://au.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html#mw_05665f1b-4343-422c-874d-2f550eb87f01

Questions?

Introduction

Transformers

- Transformer is a deep learning model that adopts the mechanism of self-attention, deferentially weighting the significance of each part of the input data.
- Like RNNs, transformers were designed to handle sequential input data, such as natural languages, for tasks such as *translation* and text *summarisation*.
- Unlike RNNs, transformers do not necessarily process the data in order. The attention mechanism provides context for any position in the input sequence.
- Transformer allows for more parallelisation than RNNs, therefore reduces training times.

https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

Introduction

Transformers

- Transformer is based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.
- Transformers are the state-of-the-art type of model for dealing with sequences, e.g., in text processing, machine translation, etc.
- Transformers were introduced in 2017 by Google Brain for NLP problems, replacing RNN models (LSTM).
- Transformer models are trained with large datasets.
- Transformer models can be fine-tuned for specific tasks.

https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

Transformers

- Transformers are built on attention mechanisms which can match the performance of RNNs with attention.
- **BERT (Google)**: Bidirectional Encoder Representations from Transformers (BERT) was pre-trained based on two tasks: (1) Language modelling; (2) The next sentence prediction.
- **GPT (OpenAI)**: Generative Pre-trained Transformer (GPT) shows how a generative model of language is able to acquire world knowledge and process long-range dependencies by pre-training on a diverse corpus with long stretches of contiguous text.

https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

GPT

Generative Pre-trained Transformer (GPT):

- **2018**: GPT-1 model is able to acquire world knowledge and process long-range dependencies by pre-training on a diverse corpus with long stretches of contiguous text.
- **2019**: GPT-2 is a general-purpose learner and achieves the state-of-the-art accuracy.
- **2020**: GPT-3 succeeds at meta-learning tasks, which can generalize the purpose of a single input-output pair.
- **2022**: ChatGPT is built on OpenAI's GPT-3.5 language models which has been fine-tuned (an approach to transfer learning) using both supervised and reinforcement learning.
- **2023**: GPT-4 is capable of accepting text or image inputs, which can also read, analyze or generate up to 25,000 words of text, and write code in all major programming languages.

ViT: Vision Transformer

Vision Transformer

- In a standard Transformer directly to images, an image is split into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer.
- Image patches are treated the same way as tokens (words).
- The transformer model is trained for image classification in supervised fashion.
- Transformers could not generalize well when trained on insufficient amounts of data.

A. Dosovitskiy, et al. (2021) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR

ViT: Vision Transformer

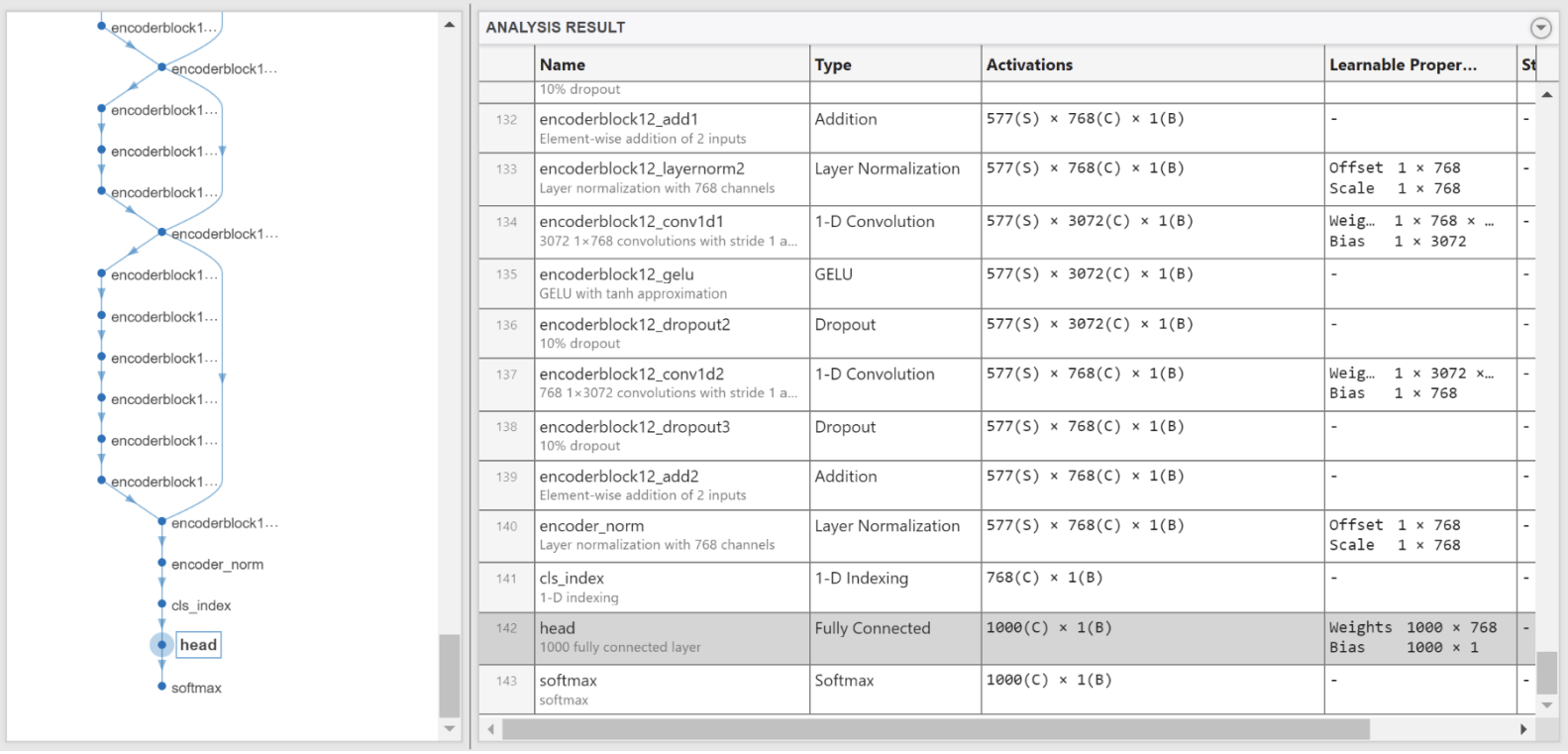
Model Analysis

- An image is treated as a sequence of patches and process it by using a standard Transformer encoder.
- The first layer of Vision Transformer linearly projects the flattened patches into a lower-dimensional space.
- After the projection, a learned position embedding is added to the patch representations.
- Self-attention allows ViT to integrate information across the entire image even in the lowest layers.
- Transformers show impressive performance from the scalability and large scale self-supervised pre-training.
- ViT matches or exceeds the state-of-the-art on many image datasets, but relatively cheap to pre-train.

A. Dosovitskiy, et al. (2021) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR

ViT: Vision Transformer

MATLAB: Vision Transformer(ViT)

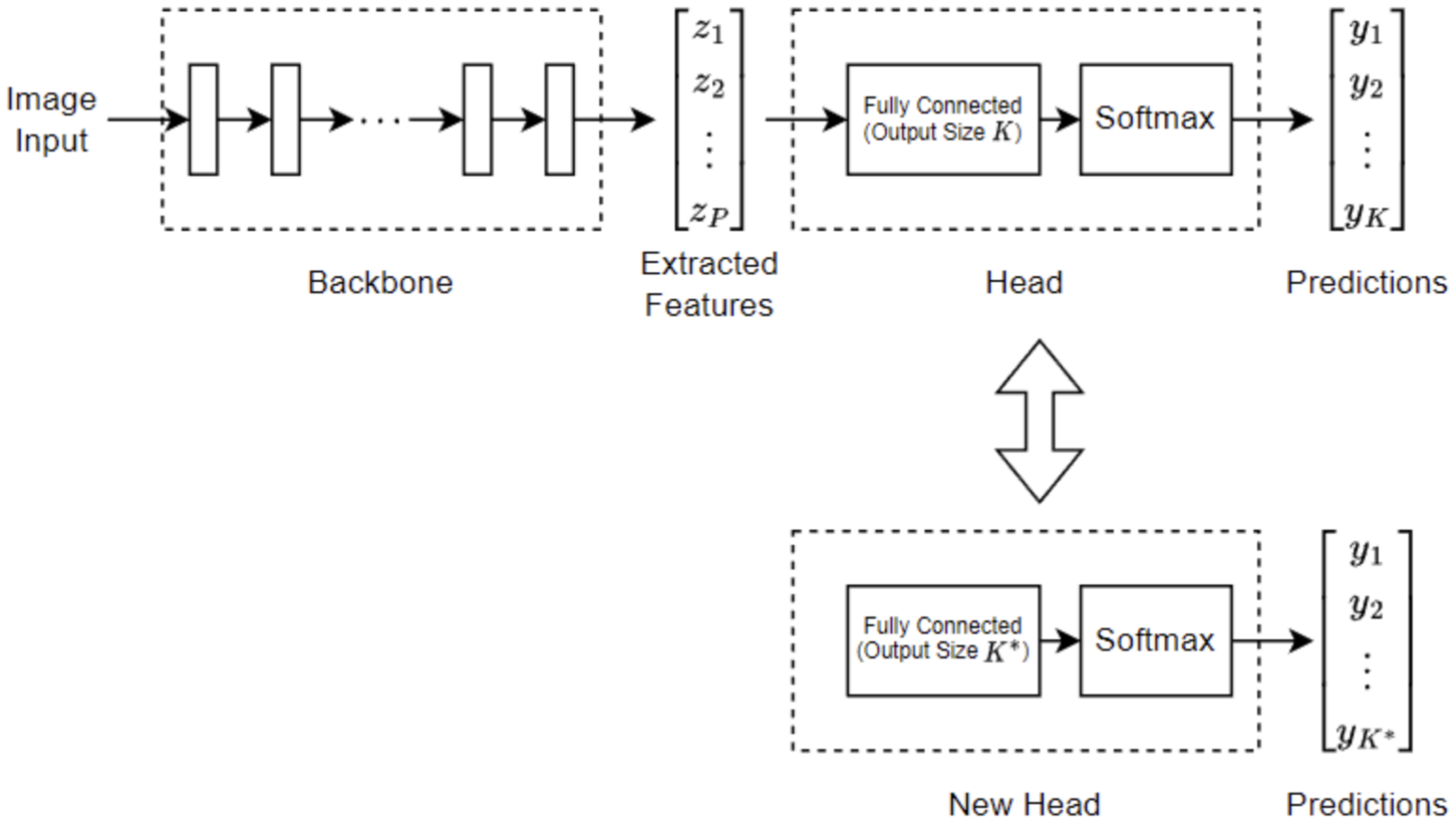


ANALYSIS RESULT

	Name	Type	Activations	Learnable Proper...	St
	10% dropout				
132	encoderblock12_add1 Element-wise addition of 2 inputs	Addition	577(S) × 768(C) × 1(B)	-	-
133	encoderblock12_layernorm2 Layer normalization with 768 channels	Layer Normalization	577(S) × 768(C) × 1(B)	Offset 1 × 768 Scale 1 × 768	-
134	encoderblock12_conv1d1 3072 1×768 convolutions with stride 1 a...	1-D Convolution	577(S) × 3072(C) × 1(B)	Weig... 1 × 768 × ... Bias 1 × 3072	-
135	encoderblock12_gelu GELU with tanh approximation	GELU	577(S) × 3072(C) × 1(B)	-	-
136	encoderblock12_dropout2 10% dropout	Dropout	577(S) × 3072(C) × 1(B)	-	-
137	encoderblock12_conv1d2 768 1×3072 convolutions with stride 1 a...	1-D Convolution	577(S) × 768(C) × 1(B)	Weig... 1 × 3072 ×... Bias 1 × 768	-
138	encoderblock12_dropout3 10% dropout	Dropout	577(S) × 768(C) × 1(B)	-	-
139	encoderblock12_add2 Element-wise addition of 2 inputs	Addition	577(S) × 768(C) × 1(B)	-	-
140	encoder_norm Layer normalization with 768 channels	Layer Normalization	577(S) × 768(C) × 1(B)	Offset 1 × 768 Scale 1 × 768	-
141	cls_index 1-D indexing	1-D Indexing	768(C) × 1(B)	-	-
142	head 1000 fully connected layer	Fully Connected	1000(C) × 1(B)	Weights 1000 × 768 Bias 1000 × 1	-
143	softmax softmax	Softmax	1000(C) × 1(B)	-	-

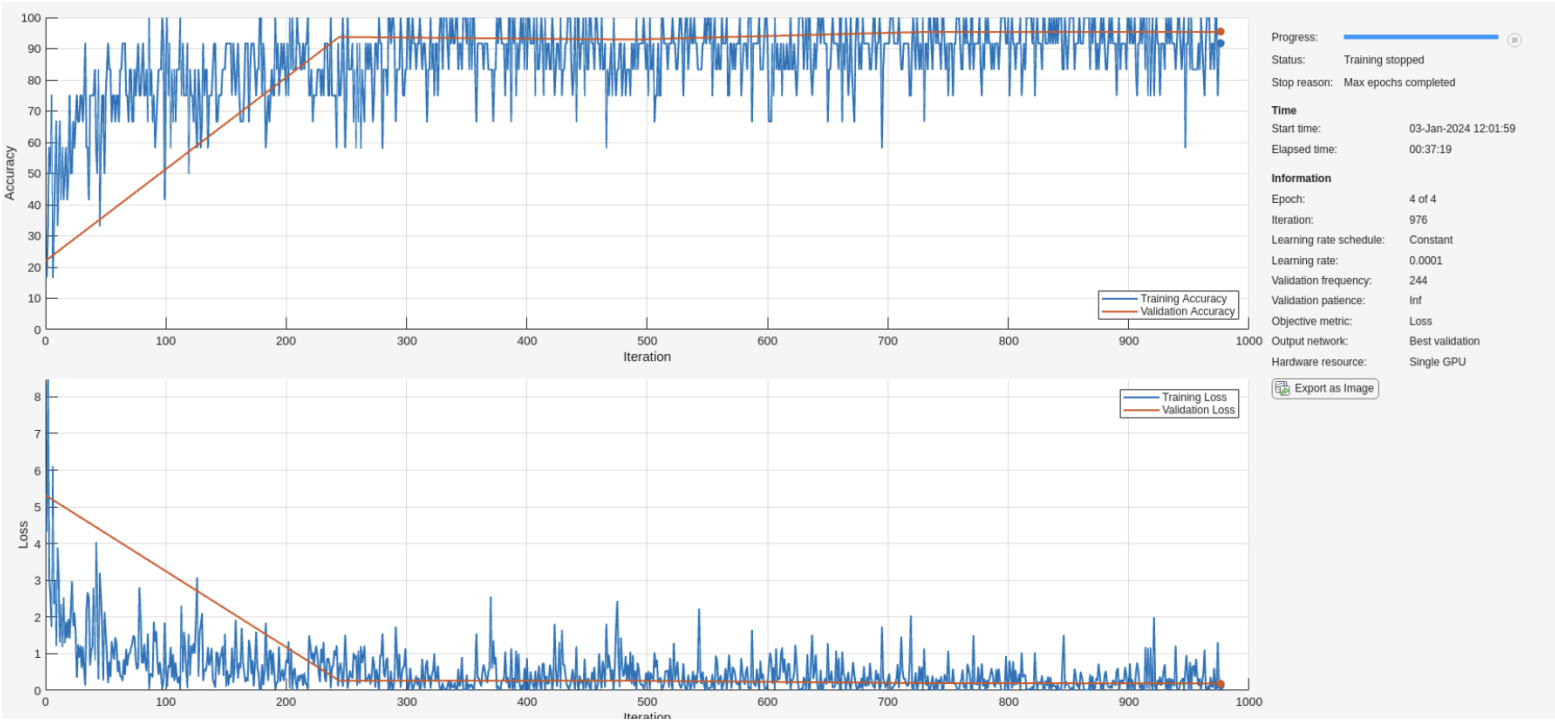
https://au.mathworks.com/help/deeplearning/ug/train-vision-transformer-network-for-image-classification.html

MATLAB: Vision Transformer(ViT)



https://au.mathworks.com/help/deeplearning/ug/train-vision-transformer-network-for-image-classification.html

MATLAB: Vision Transformer(ViT)

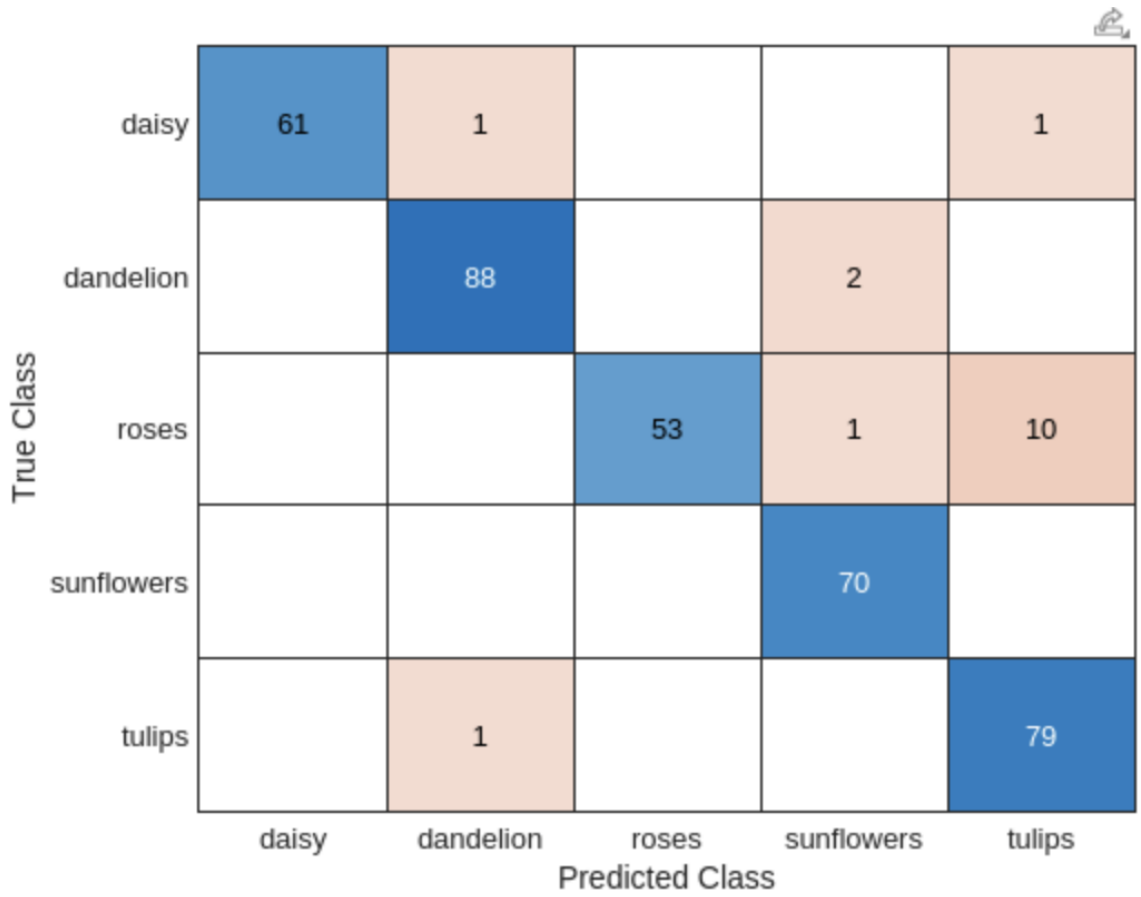


https://au.mathworks.com/help/deeplearning/ug/train-vision-transformer-network-for-image-classification.html

MATLAB: Vision Transformer(ViT)

https://au.mathworks.com/help/deeplearning/ug/train-vision-transformer-network-for-image-classification.html

MATLAB: Confusion Matrix



https://au.mathworks.com/help/deeplearning/ug/train-vision-transformer-network-for-image-classification.html

Evaluations

- Training set
- Test set
- Ground truth
- Precision: $p = \frac{TP}{TP+FP}$
- Recall: $r = \frac{TP}{TP+FN}$
- F-measure: $F = \frac{2 \cdot p \cdot r}{p+r}$
- G-measure: $G = \sqrt{p \cdot r}$

Note: F-measure is the harmonic mean (average) of recall and precision, G-measure is the geometric mean (average).

Questions?

Questions?

What is the relationship between RNNs and Transformers?

1. Like RNNs, transformers were designed to handle sequential input data.
2. Unlike RNNs, transformers do not necessarily process the data in order.
3. Transformers allow for more parallelisation than RNNs.
4. None of the given options.

The wrong answer is:___

Questions?

Learning Objectives

- Derive solutions for particular robotic vision and visual control tasks characterised by specifics of image data and deep learning algorithms.
- Critically evaluate the performance of robotic vision with deep learning algorithms, bench mark data, performance measures, and ways to define ground truth.
- Examine opportunities of using robotic vision as a part of complex robotic systems and applications.