MLAI – 2023/2024

MLAI 504
NEURAL NETWORKS & DEEP LEARNING

Dr. Zein Al Abidin IBRAHIM

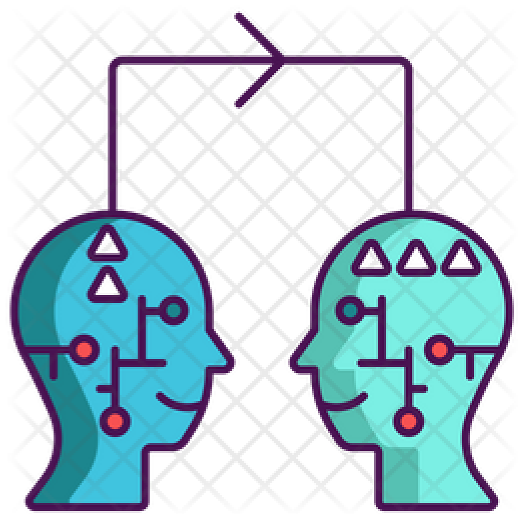
zein.ibrahim@ul.edu.lb

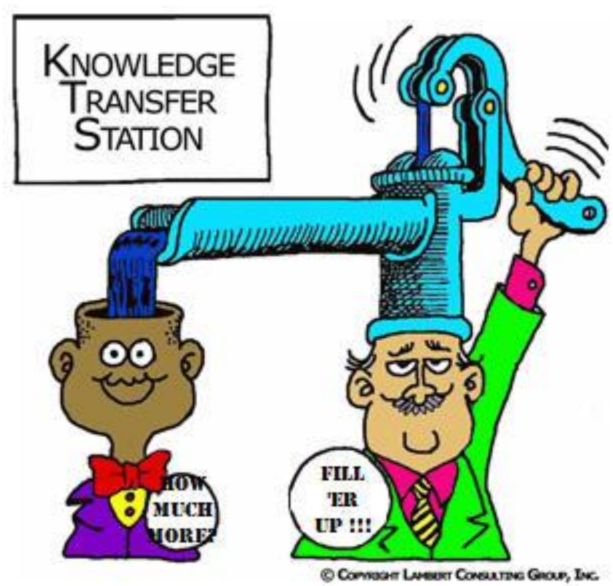
TRANSFER LEARNING

DEEP LEARNING

DEFINITION

- **Definition**: Transfer learning reuses knowledge gained from a pre-trained model on a new task, saving time and resources.
- **Think of it as**: Building upon existing expertise instead of starting from scratch.





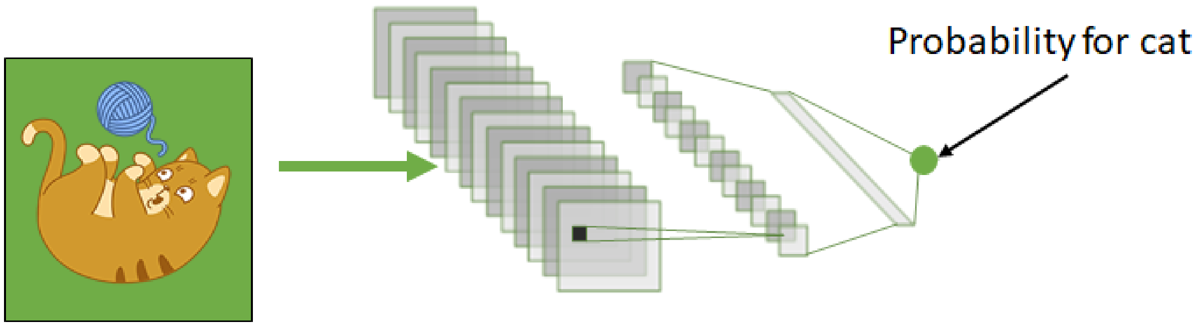
ADVANTAGES OF TRANSFER LEARNING

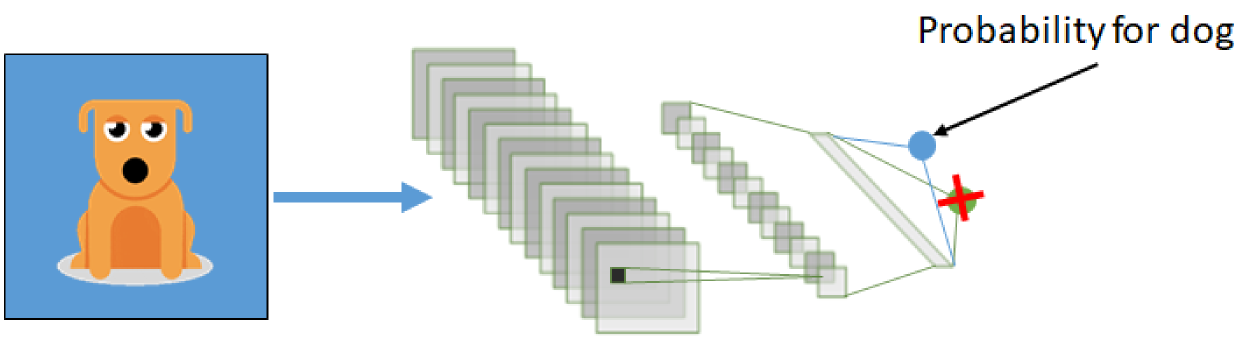
- **Faster Training**: Pre-trained models have already learned low-level features, reducing training time for new tasks.
- **Less Data Required**: Smaller datasets suffice for fine-tuning, making it ideal for scenarios with limited data.
- **Improved Performance**: Leveraging pre-trained knowledge often leads to better accuracy and generalization compared to training from scratch.

POTENTIAL DRAWBACKS

- **Domain mismatch**: Pre-trained knowledge might not perfectly translate to the new task, leading to suboptimal performance.
- **Negative Transfer**: Pre-trained biases can hinder learning on the new task, requiring careful selection and adaptation.
- **Computational Overhead**: Fine-tuning a pre-trained model can still be computationally expensive, especially for large models.

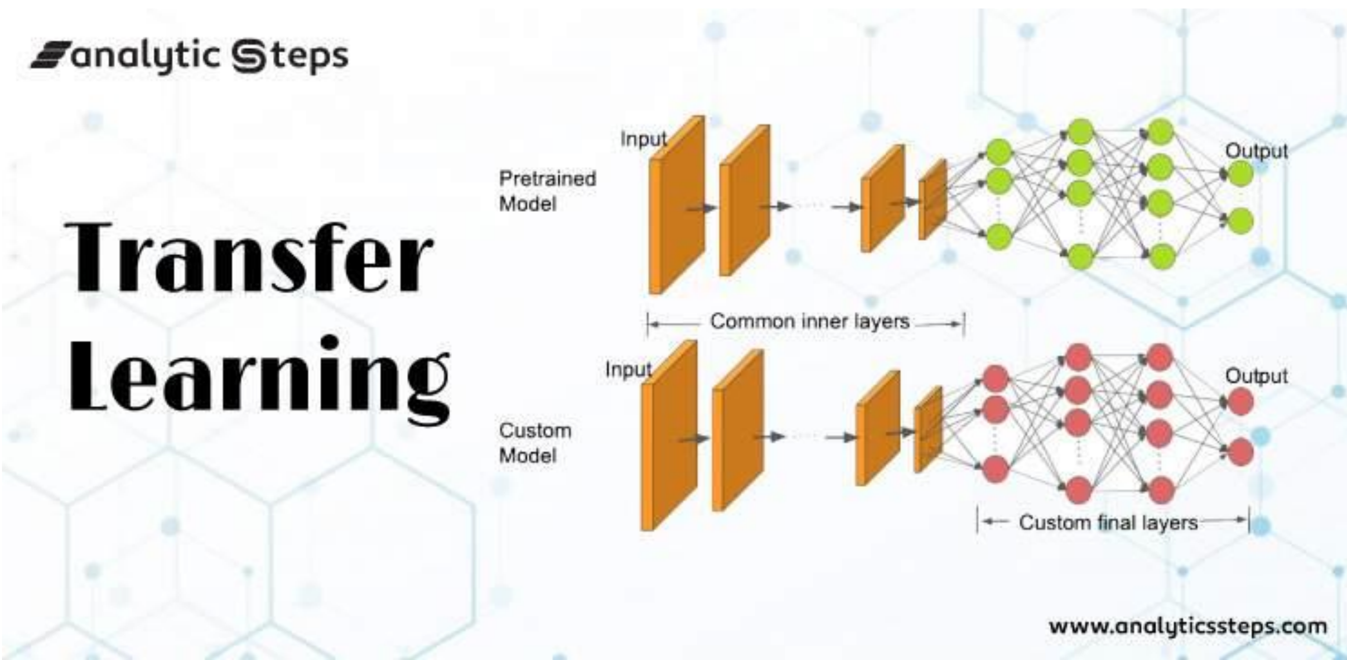
EXAMPLE



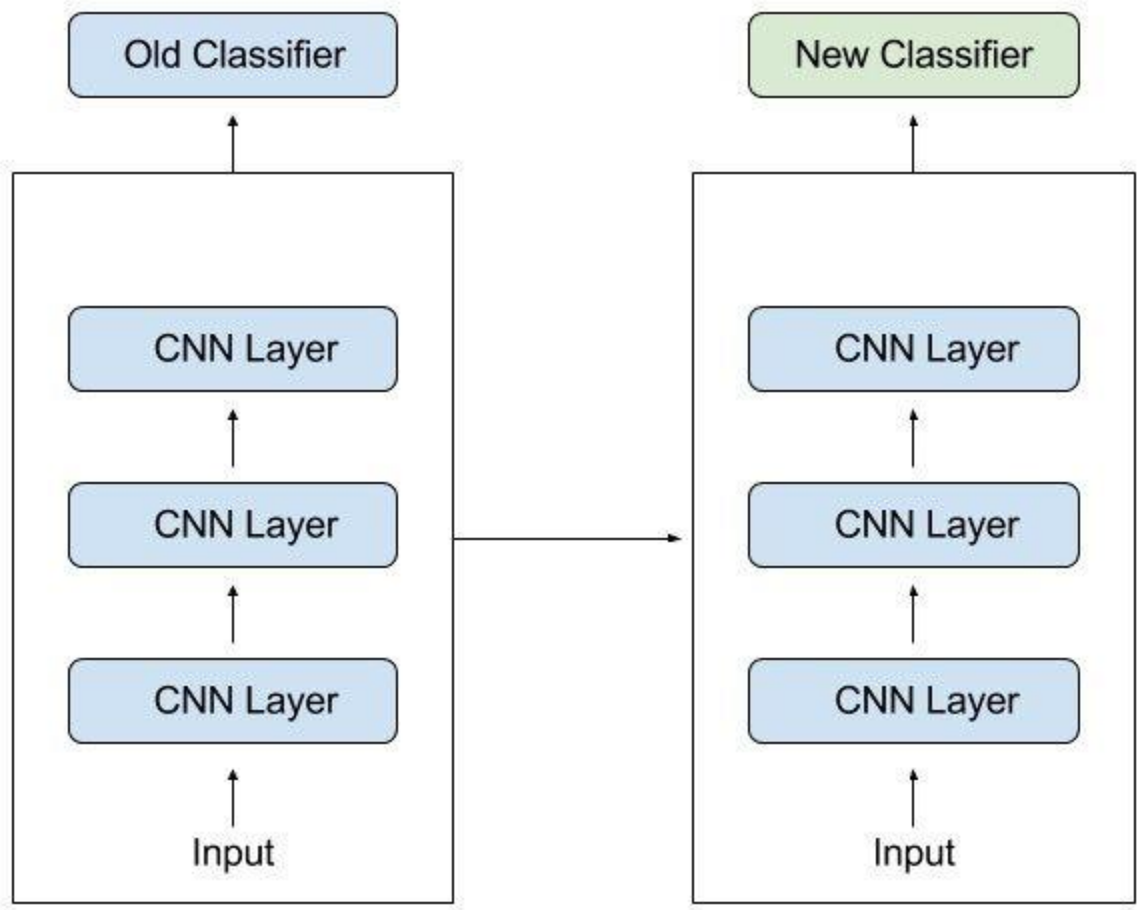


HOW IT WORKS?

- In computer vision → The neural network aims to:
  - detect edges in the first layer(s),
  - detect forms in the middle layer(s),
  - and task-specific features in the latter layers.
- early and central layers are employed in transfer learning, and the latter layers are only retrained



HOW IT WORKS?


Old Classifier
CNN Layer
CNN Layer
CNN Layer
Input
New Classifier
CNN Layer
CNN Layer
CNN Layer
Input

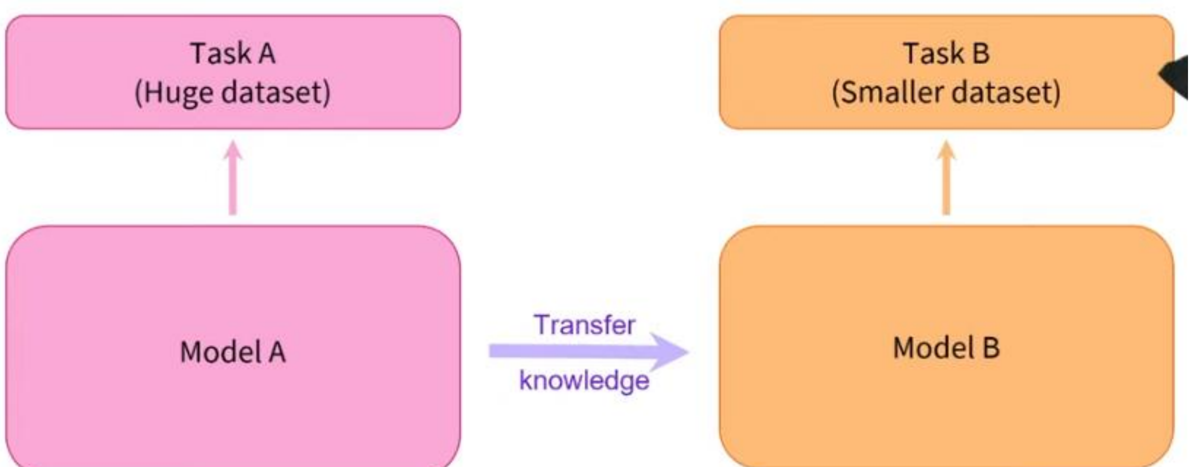
TRAINING FROM SCRATCH VS USING PRE-TRAINED MODEL

Training from scratch

Epoch	Validation Loss	Accuracy
1.0	0.626885	0.683824
2.0	0.643775	0.691176
3.0	0.599722	0.691176
4.0	0.699712	0.686275
5.0	0.766334	0.681373

Fine-tuning a pretrained model

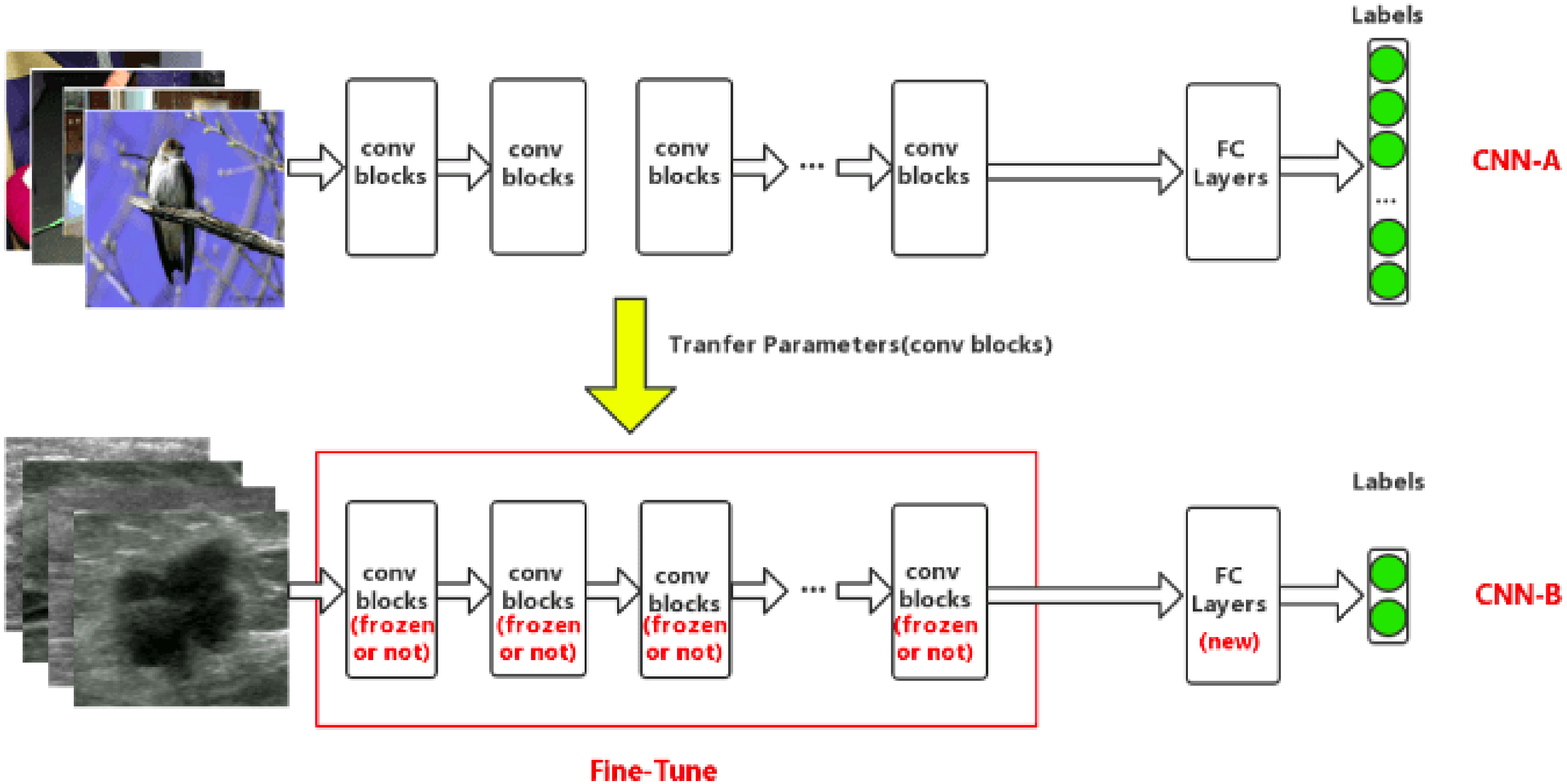
Epoch	Validation Loss	Accuracy
1.0	0.420394	0.813725
2.0	0.380642	0.830882
3.0	0.342212	0.865196



HOW TO USE TRANSFER LEARNING

- Use pre-trained models without any modification
- Change the last layer responsible of the classification and train the new weights and freeze all the other weights in the other layers
- Same as 2 but re-train all the network
- Remove more than the last layer and add as much layers at the end as you want and train the weights of the added layers or all the weights as you want
  - You should have enough data and resources to train a complete network

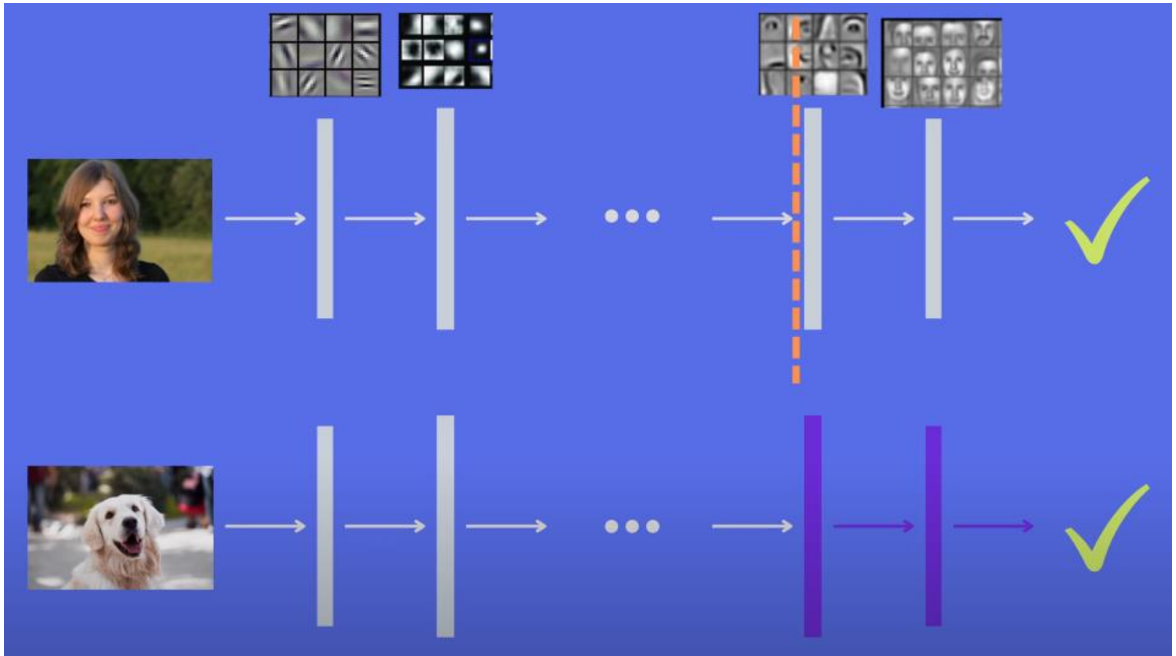
TYPES OF TRANSFER LEARNING



TYPES OF TRANSFER LEARNING

HOW TO USE TRANSFER LEARNING

- Task A → recognizing human faces
- Task B → recognizing dog faces.
- Here → removing the last layer is not enough because the before last layers detect high level features corresponding to humans not dogs so you remove more layers
  - Number of layers to remove depends on how much tasks are similar.



WHEN TRANSFER LEARNING MAKES SENSE

- To use transfer learning from task A to task B you should consider:
  - Tasks A and B have the same input
  - We have more data for task A than task B
  - Low level features from task A could be valuable for learning B

WHERE TO FIND PRE-TRAINED MODELS

- Deep learning frameworks
  - Tensorflow hub → https://www.tensorflow.org/resources/models-datasets
  - Keras applications → https://keras.io/api/applications/
  - Pytorch → https://pytorch.org/vision/stable/models.html
  - Hugging face → transformer-based models
  - Kaggle
  - Model zooz
  - …