Deep Learning

Transfer Learning

Learning Objectives

By the end of this lesson, you will be able to:

- Implement pre-trained models according to the task at hand
- Analyze the advantages of transfer learning
- Apply the transfer learning technique to projects
- Define the pre-trained model

Business Scenario

A healthcare company is looking to develop a deep learning model to detect rare diseases in medical images. However, due to the limited availability of data, training a model from scratch is not feasible.

The company decides to use transfer learning and leverages a pre-trained model that was developed for a similar task of image classification. The pre-trained model is modified and fine-tuned to detect rare diseases. The model is tested and achieves higher accuracy compared to a model trained from scratch.

The use of transfer learning saved time and resources while improving the accuracy of the model. This approach can be applied in various industries, where pre-trained models can be utilized to speed up the development of new deep learning models.

Introduction to Transfer Learning

Discussion

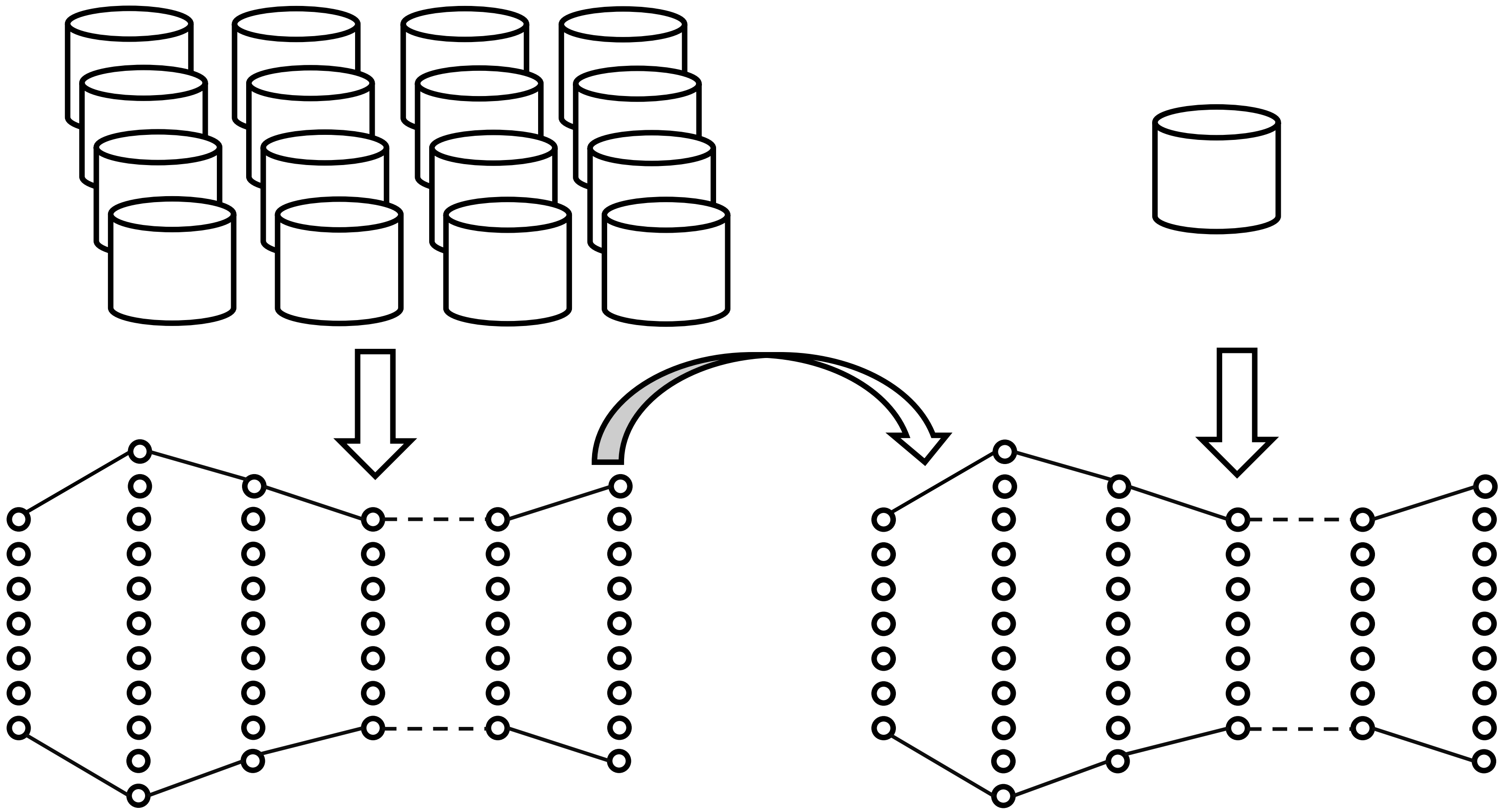
Discussion: Transfer Learning

Duration: 10 minutes

- What is transfer learning?
- How does transfer learning work?

What Is Transfer Learning?

It is a deep learning technique where a model developed for one task is reused as the starting point for another model.

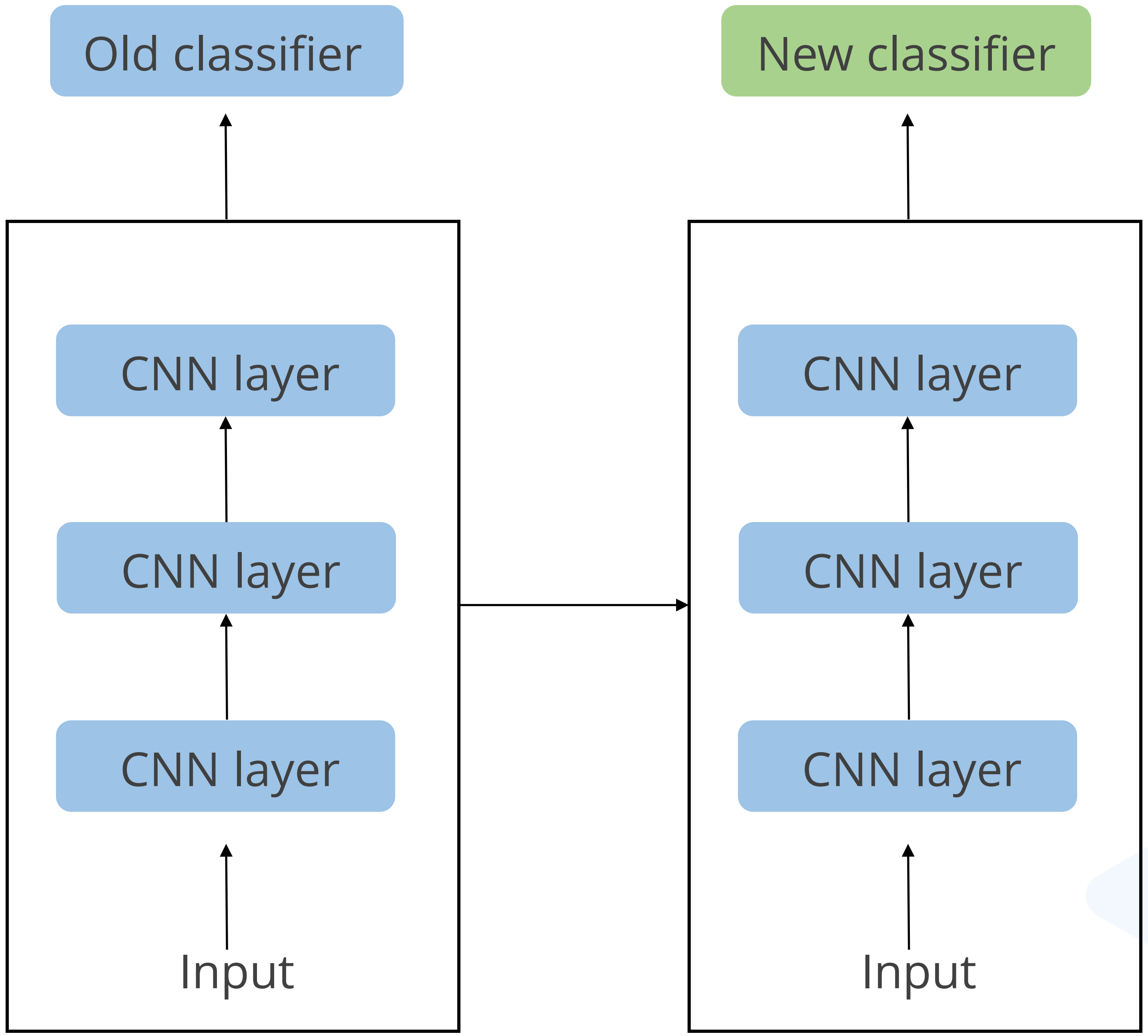


The latter part of this pre-trained model is then fine-tuned or adapted to suit the specific requirements of the new task.

Transfer Learning in Deep Learning Model

The key aspects of transfer learning in the context of deep learning models include:

- Reuses the pre-trained models
- Retrains the latter layers for new tasks
- Leverages learned features for a wide range of tasks, including but not limited to object recognition



Why Is Transfer Learning Used?

It is used in deep learning for several reasons:

1. **Faster training:** Speeds up training by reusing pre-trained model knowledge

2. **Improved performance:** Improves performance with less training time and resources

3. **Handling small datasets:** Is particularly useful when the new task has a small dataset

Why Is Transfer Learning Used?

It is used in deep learning for several reasons:

4. **Domain adaptation:** Allows models to adapt to new domains or different data distributions

5. **Resource efficiency:** Reduces data annotation needs by utilizing pre-trained models

6. **Transfer of knowledge:** Facilitates the insights gained from one task to another

Transfer Learning: Example

Consider a teacher in a classroom who wants to detect if the students are listening to the lecture or if they're bored

- It enables the detection of facial expressions and emotions.
- It also allows for fine-tuning pre-trained models to recognize classroom engagement in virtual meetings.

Transfer Learning: Example

To solve facial expression recognition, first, gather a dataset of the facial expressions of various people.

It consists of various facial expressions, such as:

Angry

Sad

Happy

Disgusted

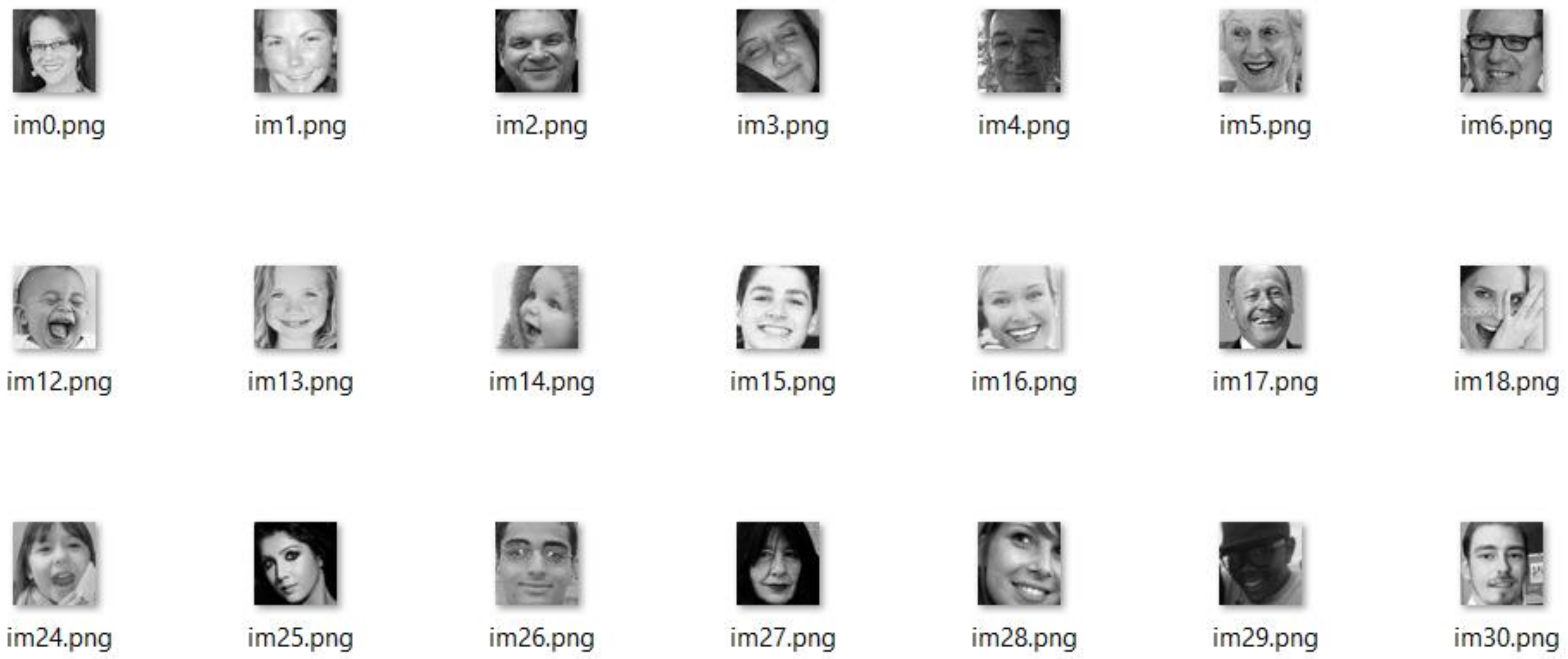


Image Source: https://datarepository.wolframcloud.com/resources/FER-2013

Transfer Learning: Example

After gathering the dataset, create a custom model to train a classifier to classify all the expressions

Demo

```
from tensorflow.keras.datasets import fer2013

# Load the FER2013 dataset
(train_images, train_labels), (test_images, test_labels) = fer2013.load_data()

# Build the custom model
def custom model():
 model = Sequential()

# Convolutional 2D layers with 32 filters, a kernel size of (3,3), and ReLU
activation function. It takes an input shape of (48,48,1), representing a
grayscale image of size 48x48.
 model.add(Conv2D(32, kernel_size=(3,3), activation='relu',
input_shape=(48,48,1)))
 model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
```

Transfer Learning: Example

MaxPooling 2D layer reduces the spatial dimensions of the feature maps using a pooling window of size (2,2).

Demo

```
model.add(MaxPooling2D(pool_size=(2,2)))

# Dropout layer helps prevent overfitting by randomly dropping out a fraction of the input units during training.
 model.add(Dropout(0.25))

 model.add(Conv2D(128, kernel_size=(3,3), activation='relu'))
 model.add(MaxPooling2D(pool_size=(2,2)))
 model.add(Conv2D(128, kernel_size=(3,3), activation='relu'))
 model.add(MaxPooling2D(pool_size=(2,2)))
 model.add(Dropout(0.25))
```

Transfer Learning: Example

Save and load the model weights

Demo

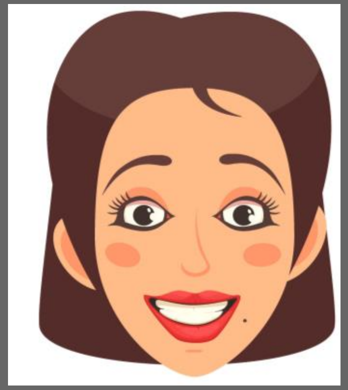
```
 model.add(Flatten())
# Dense layer with 256 units and ReLU activation is added.
 model.add(Dense(256, activation='relu'))
# Dropout layer with a dropout rate of 0.5 is included.
 model.add(Dropout(0.5))
# Dense layer with 3 units and a softmax activation function to obtain
probability distributions over the classes.
 model.add(Dense(3, activation='softmax'))

 return model

# Create the custom model
model = custom_model()
# Save and load the model weights to 'model.h5'
model.save_weights('model.h5')
model.load_weights('Emotion_Data/Models/model.h5')
```

Transfer Learning: Example

Prepare an image for prediction

Demo

```
# Load the image of a happy face using OpenCV
out_img = cv2.imread('Emotion_Data/Demo/happy.png')
# Resize the image to match the input shape of the model
out = cv2.resize(out_img, (48,48))
out = out[ :, :, 0]
plt.imshow(out)
```



```
0: Happy
1: Sad
2: Angry
# Make predictions on the preprocessed image
predictions = model.predict(np.expand_dims(out, 0)) predicted_class =
np.argmax(predictions) print(predicted_class)

0

#Predicted class index is 0, which corresponds to the "Happy" facial expression.
```

Scenarios of Transfer Learning

There are two scenarios of transfer learning:

- Positive transfer learning
- Negative transfer learning

Positive Transfer Learning

Positive transfer learning refers to a situation where knowledge or experience gained from one task improves performance on a different, related task.

Example

A model trained to detect one type of cancer cell may also perform well in detecting variants of those cancer cells in the future.

Negative Transfer Learning

Negative transfer learning refers to a situation where knowledge or experience gained from one task hinders performance on a different, unrelated task.

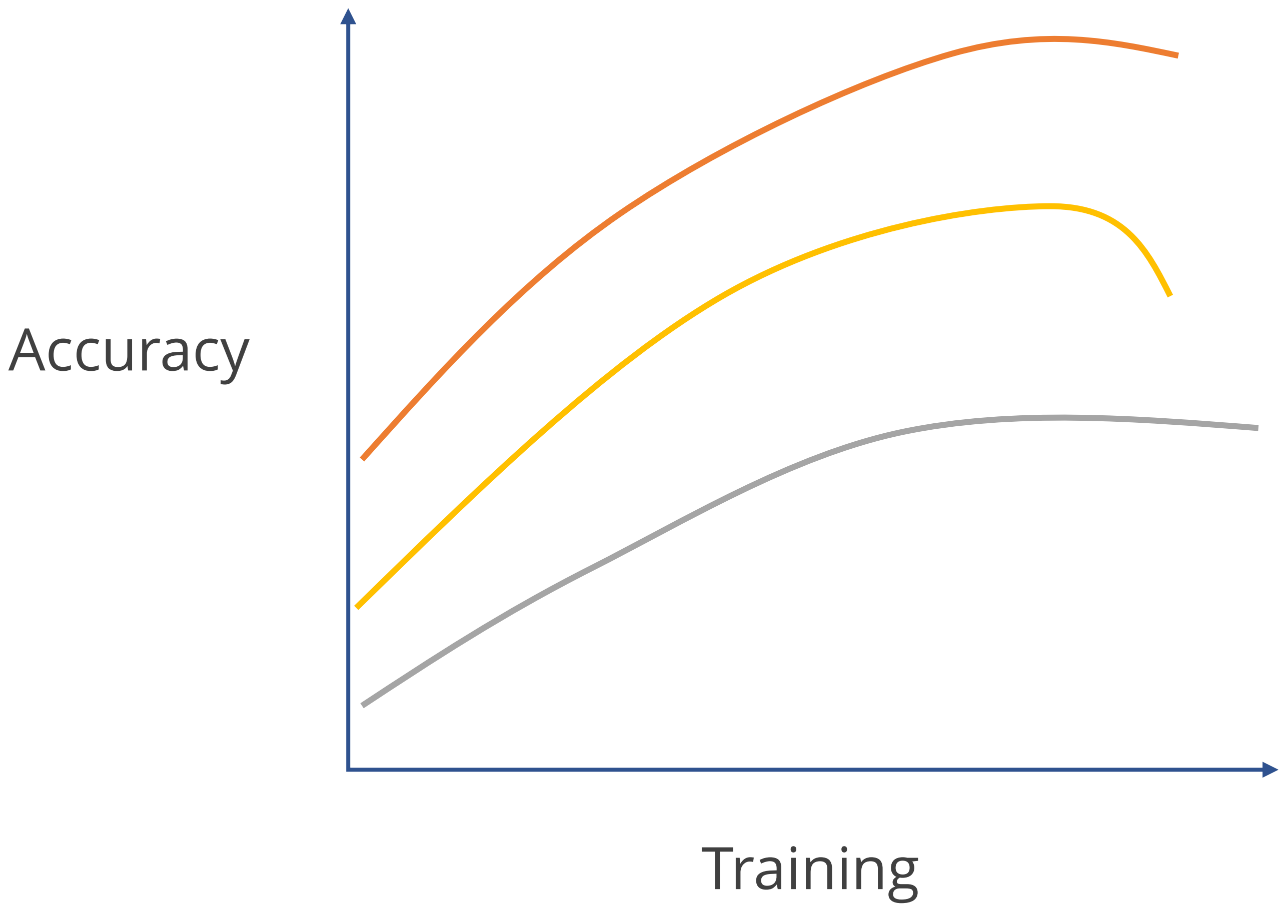
Example

A model trained on the MNIST dataset cannot perform well at detecting Chinese digits.

If negative transfer learning is observed, it may be beneficial to conduct further training.

Positive and Negative Transfer Learning

The following image shows how the two pre-trained models are:



Positive transfer learning → Improves a pre-trained model's performance on a new problem

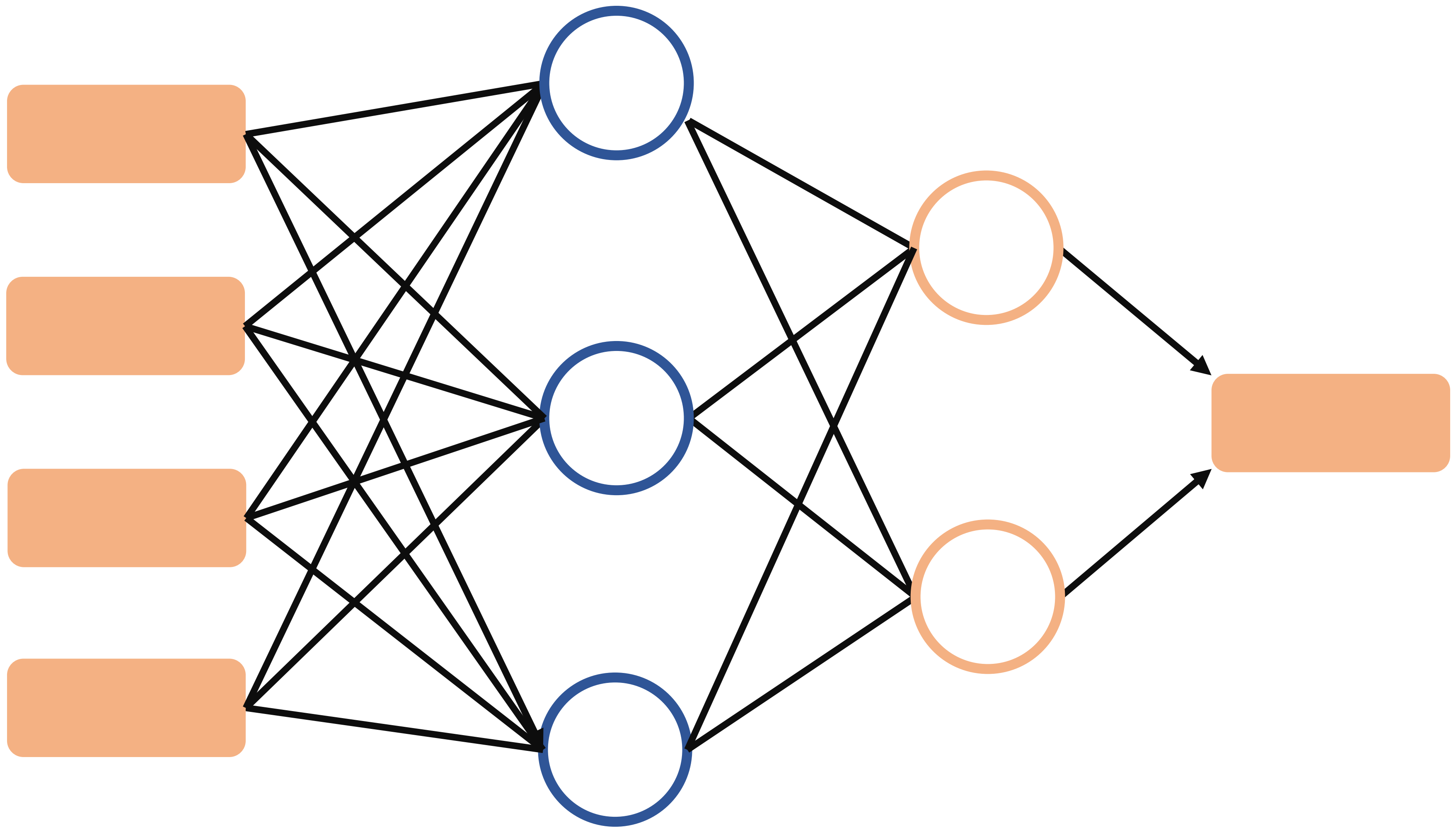
Pre-trained model → Pre-existing neural network that has been trained on a large dataset

Negative transfer learning → Deteriorates the original model's performance on a new problem

How to Select Pre-trained Models?

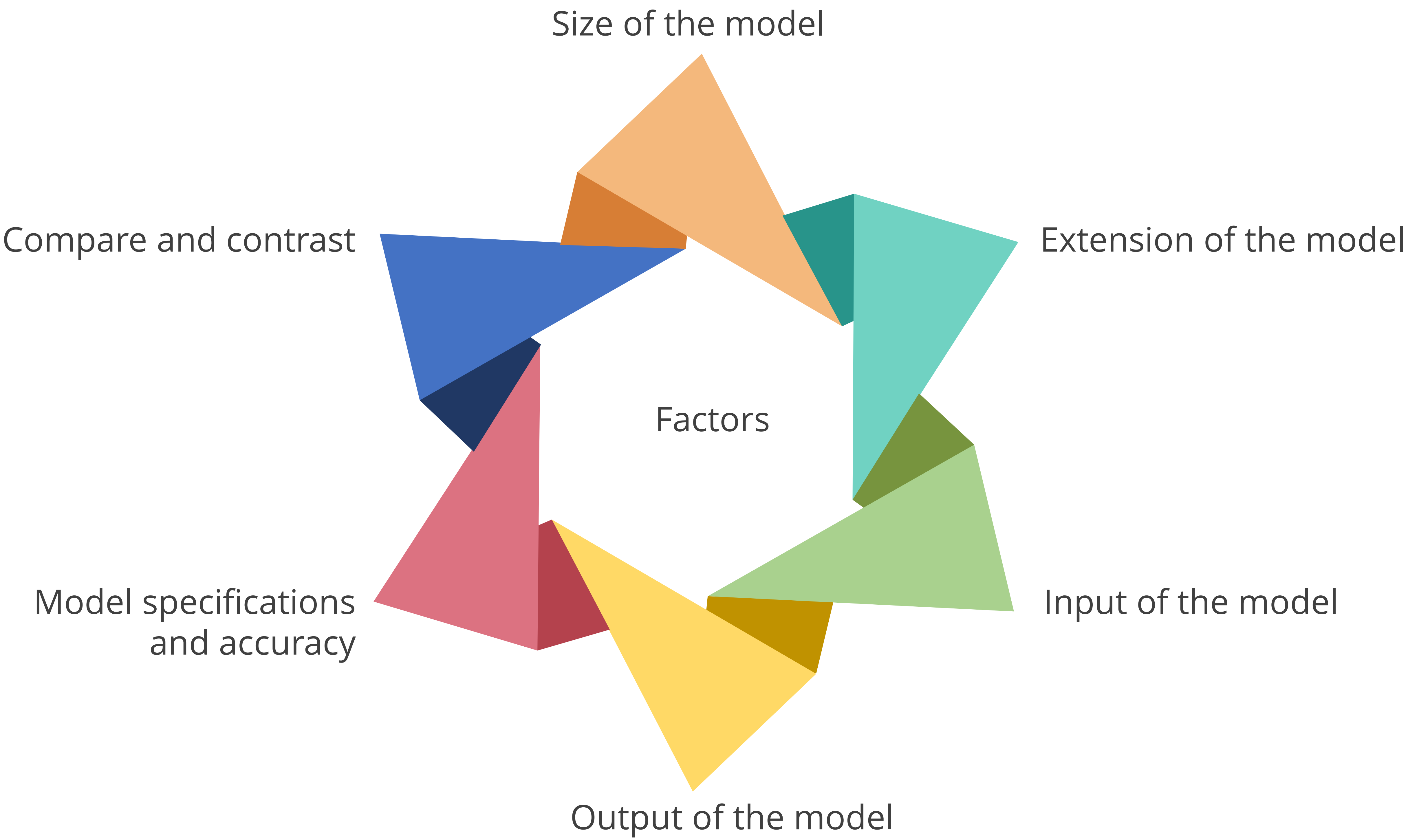
Pre-trained Models

Pre-trained models are pre-built deep learning models trained on large datasets, enabling efficient transfer learning for improved performance on new tasks.



Factors Considered to Choose a Pre-trained Model

The factors considered when choosing a pre-trained model are:



Factors of Pre-trained Model: Size of the Model

Size of the model

It is the most crucial part of a model, as it determines the system storage occupation.

For object detection with an edge device, a small model is preferable to a heavy model.

Factors of Pre-trained Model: Extension of the Model

Extension of the model

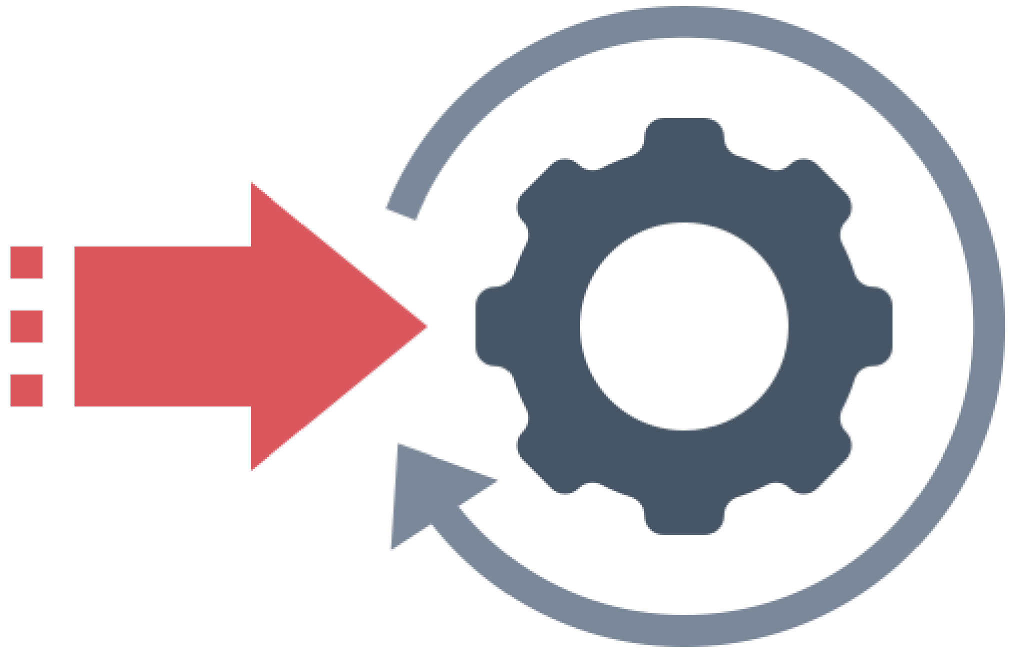
It reflects the framework with which the model was trained.

If the model was trained with TensorFlow, the file extension is typically .h5, and if it was trained with PyTorch, it is typically .pth.

The choice of a pre-trained model depends on the framework being worked on.

Factors of Pre-trained Model: Input of the Model

Input of the model	Each model has its own input requirements, which should be ensured in the preprocessing phase.



Factors of Pre-trained Model: Output of the Model

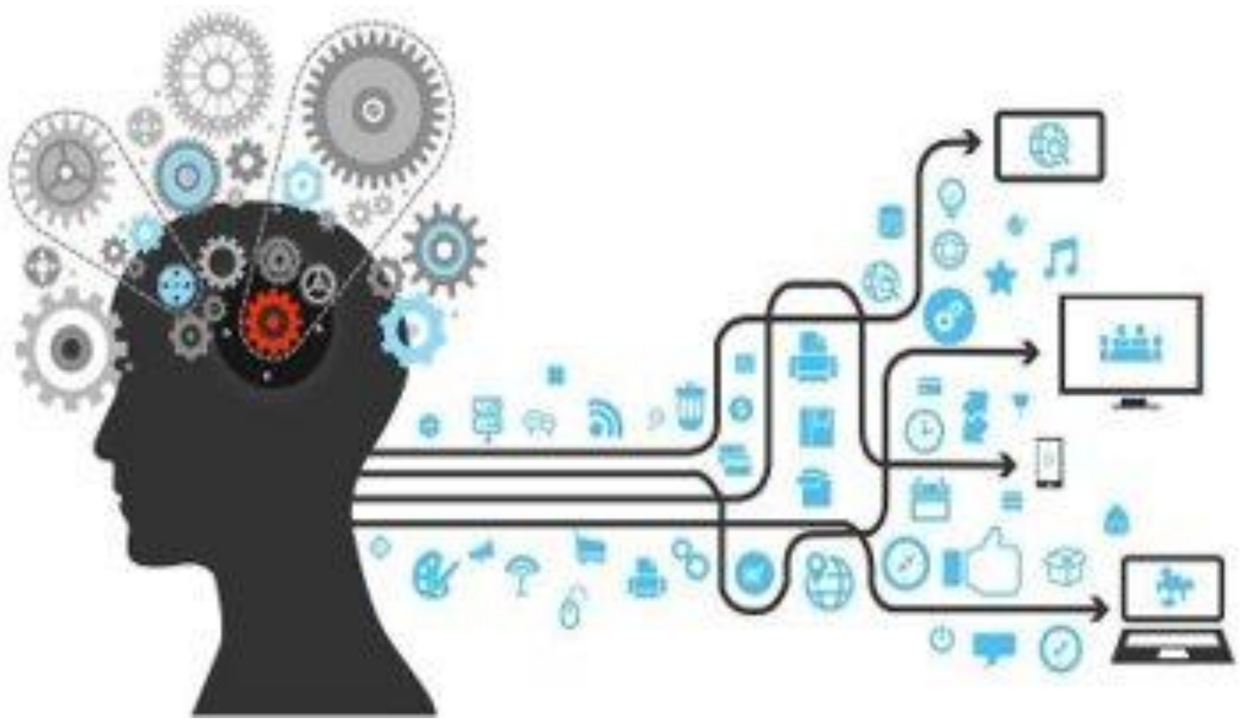
Output of the model

After successful input processing, the model outputs can be interpreted to provide the desired result.

Factors of Pre-trained Model: Model Specifications and Accuracy

Model specifications and accuracy

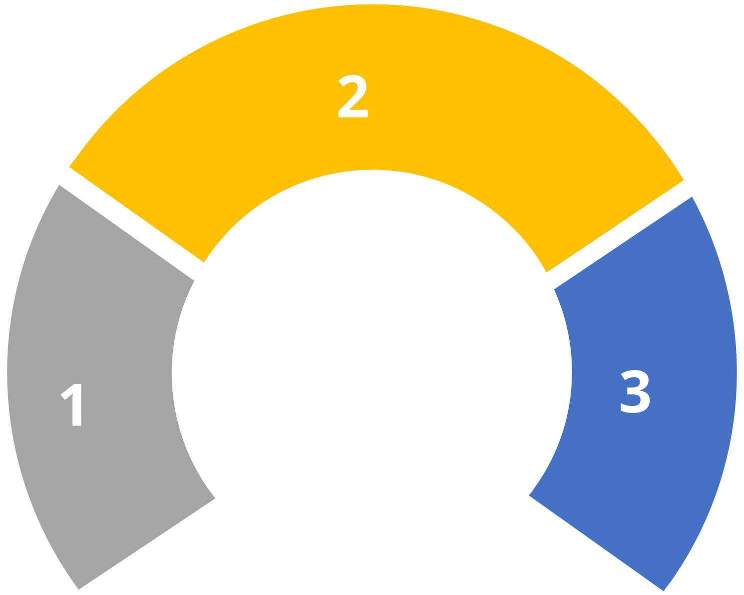
Specifications vary between pre-trained models based on the tasks to be performed.



Factors of Pre-trained Model: Model Specifications and Accuracy

The following concepts are key in evaluating object detection models:

Accuracy refers to how often the model's predictions are correct.



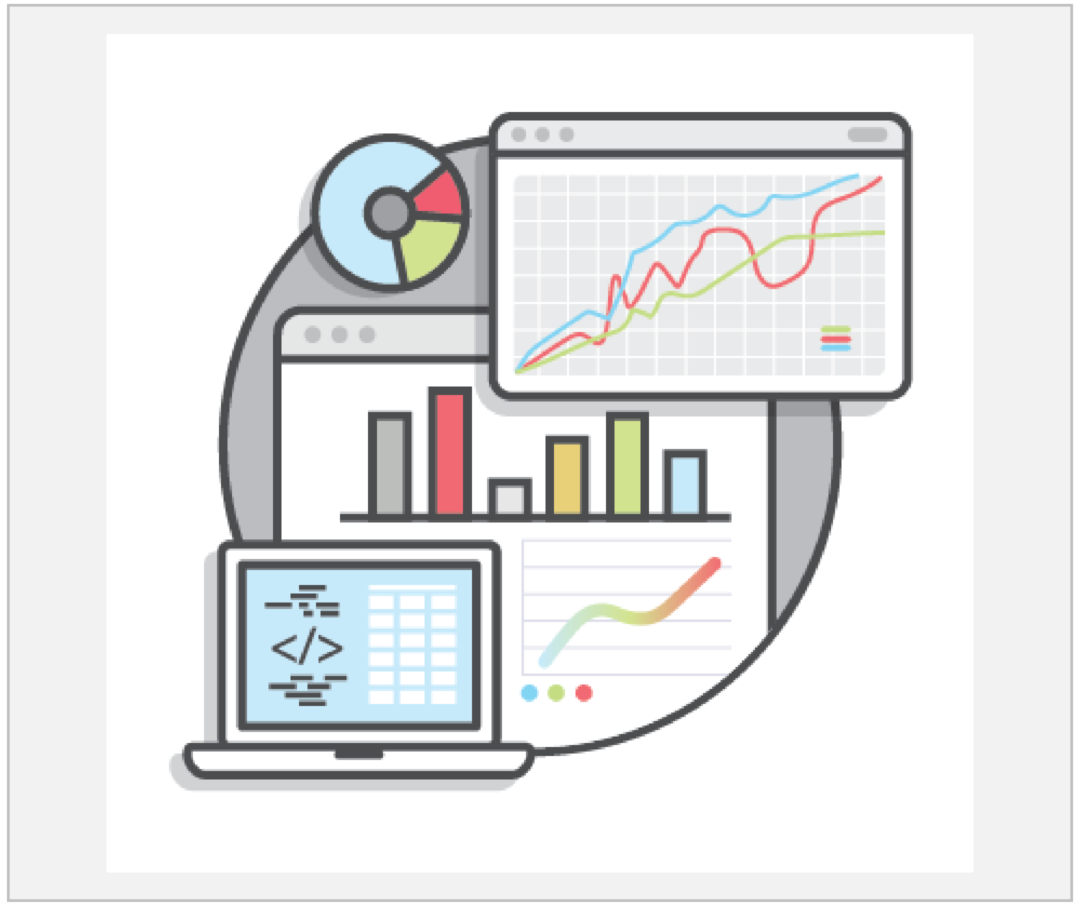
Average precision is a metric used to measure the accuracy of an object detector model.

Robustness to occlusion assesses the model's ability to handle partially visible objects in an image.

Factors of Pre-trained Model: Compare and Contrast

Compare and contrast

After evaluating all the factors, the models under consideration are compared.



Factors of Pre-trained Model: Compare and Contrast

To choose the best-fit model, the models can be compared based on:

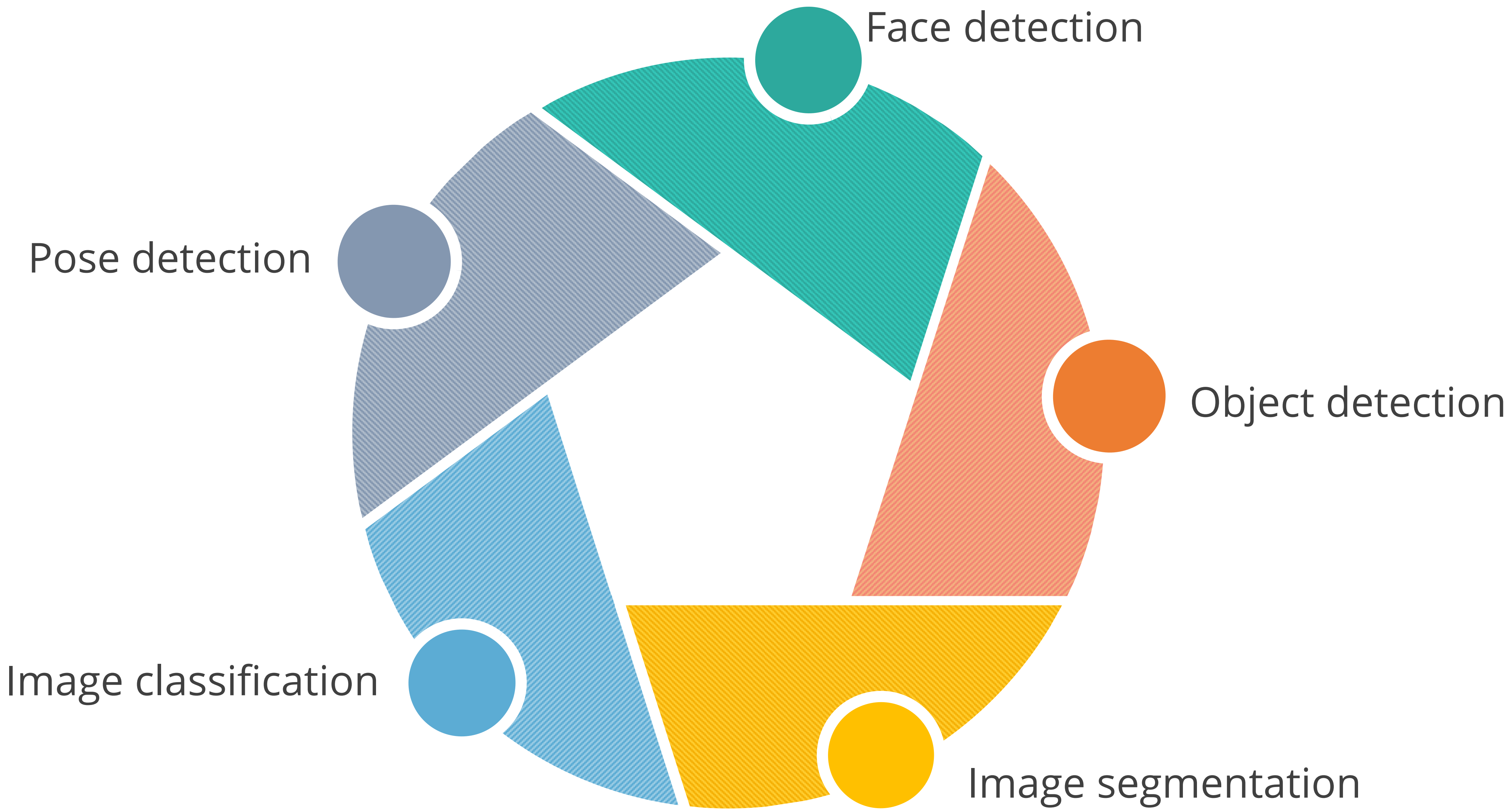
Speed: Model's prediction time.

Accuracy: Frequency of correct predictions, balanced with speed and size.

Size: Computational and memory demands of the model based on deployment constraints.

Pre-trained Model Lists

Following are some pre-trained models in the image domain:



Pre-trained Model Lists: Image Domain

Face detection

- MTCNN (Multi-task Cascaded Convolutional Networks) is a deep learning model specifically designed for face detection.
- Inception-ResNet is a hybrid model that combines the Inception and ResNet architecture.
- MobileNet is quick and effective on smartphones with limited resources.

Pre-trained Model Lists: Image Domain

Object detection

- Detectron2 is an object detection framework developed by Facebook AI Research.
- YOLOv5 (You Only Look Once) is an object detection algorithm known for its real-time processing speed.
- InceptionResNetV2 is a convolutional neural network architecture that combines the Inception and ResNet modules.

Pre-trained Model Lists: Image Domain

Image segmentation

- Mask RCNN is an object detection and instance segmentation model.
- UNet is a popular model architecture used for image segmentation tasks.
- MANet (Microscopy Adaptive Network) is a deep learning model designed specifically for microscopy image analysis tasks.
- LinkNet is a lightweight and efficient model architecture for semantic segmentation.
- DeepLabv3 is a widely adopted model for semantic image segmentation.

Pre-trained Model Lists: Image Domain

Image classification

- RegNetY is designed for high performance and computational efficiency in CNN architectures.
- ResNet-50 revolutionized computer vision with its deep architecture and skip connections.
- VGG-16 is known for its simplicity and effectiveness in image classification tasks with deep CNNs.
- MobileNet V2 is optimized for mobile and embedded vision applications with lightweight CNNs.
- EfficientNet achieves performance while being computationally efficient in CNN architectures.

Pre-trained Model Lists: Image Domain

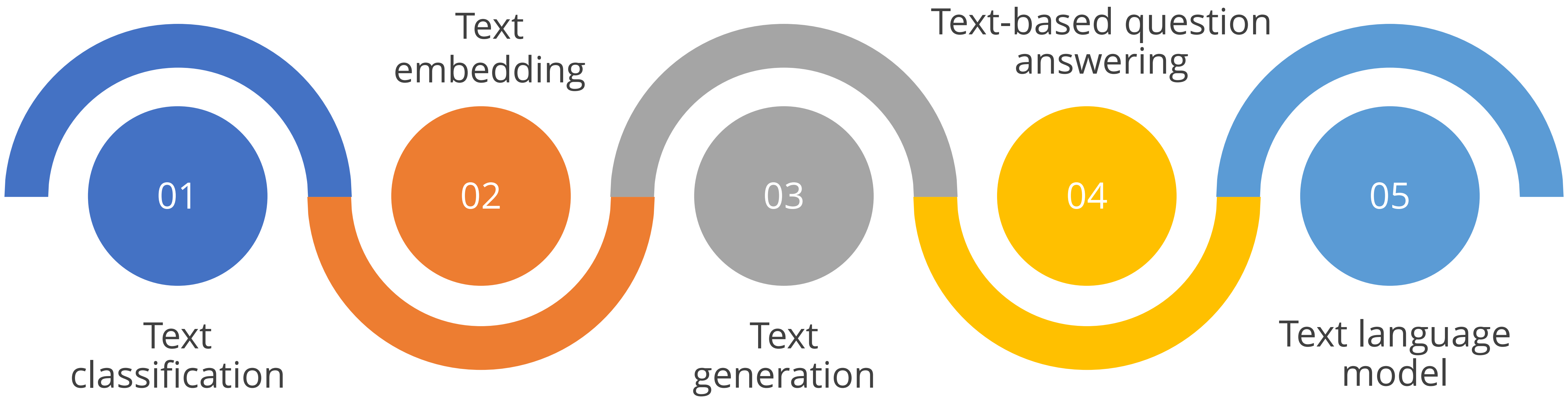
Pose detection

MoveNet is a lightweight pose estimation model designed for accurate human pose detection.

OpenPose is a popular framework for keypoint detection and action recognition.

Pre-trained Model Lists: Text Domain

Following are some pre-trained models in the text domain:



Pre-trained Model Lists: Text Domain

The following are the text classification models:

01 MobileBERT is a compact variant of BERT, optimized for mobile and resource-constrained devices.

02 Transformers are deep learning models that leverage self-attention to capture contextual information effectively.

Pre-trained Model Lists: Text Domain

The following are the text embedding models:

01 ULMFiT (Universal Language Model Fine-tuning) is an advanced text embedding model for diverse language processing tasks.

02 ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) is an efficient pre-training method with strong performance in text embedding tasks.

Pre-trained Model Lists: Text Domain

The following are the text generation models:

01 SmartReply is a text generation model developed by Google that provides automated suggestions for short message responses.

02 RoBERTa is a state-of-the-art text generation model based on the BERT architecture.

Pre-trained Model Lists: Text Domain

The following are the text-based question answering models:

01

TF2NQ (TensorFlow 2.0 Natural Questions) is a text-based question answering model specifically designed for the Natural Questions dataset.

Pre-trained Model Lists: Text Domain

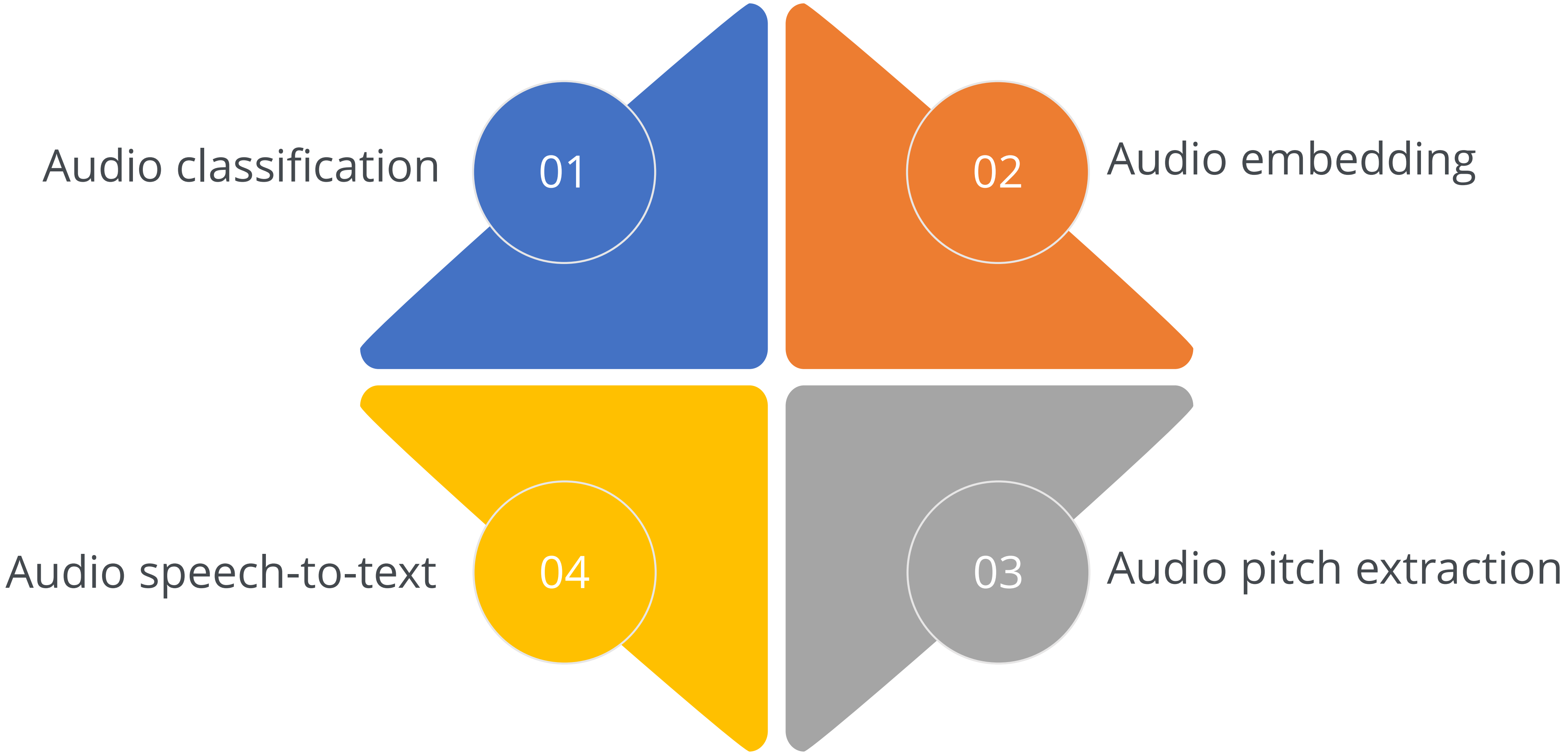
The following are the text language models:

01 Wiki-40B is a large-scale language model trained on a subset of Wikipedia text known as the *Wiki-40B* dataset.

02 Enformer is a text model with transformer-based architecture and enhanced long-range context handling.

Pre-trained Model Lists: Audio Domain

Following are some pre-trained models for audio classification and audio domain-related needs:



Pre-trained Model Lists: Audio Domain

Audio classification

- YAMNet (Yet Another Music Network) is designed to classify audio signals into a wide range of sound categories, including environmental sounds, musical instruments, and human actions.

Pre-trained Model Lists: Audio Domain

Audio embedding

- TRILL (Transferable and Interpretable Learning for Language) is an audio embedding model that learns transferable representations from speech data.
- TRILL-Distilled is a compact version of the TRILL model optimized for resource-constrained environments.

Pre-trained Model Lists: Audio Domain

Audio pitch extraction

- SPICE (Synchronized Pitch Estimation and Confidence Estimation) accurately estimates the pitch of each audio frame and provides a confidence measure indicating the reliability of the pitch estimation.

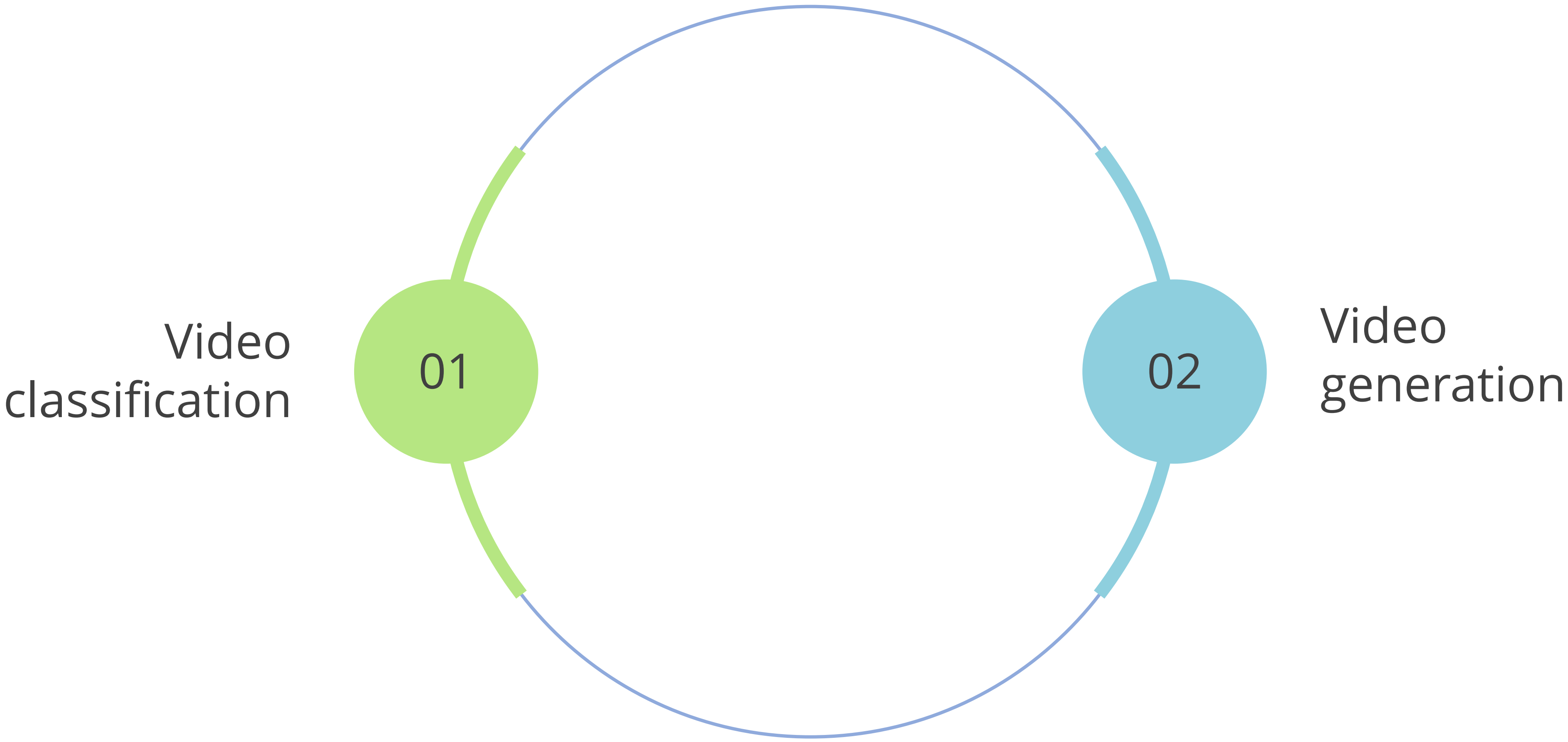
Pre-trained Model Lists: Audio Domain

Audio speech-to-text

- Wav2Vec converts audio speech signals into textual representations.
- Wav2Vec2 results in various speech recognition benchmarks and is widely used in industry and academia.
- Wav2Vec2-Robust is a variant of the Wav2Vec2 model that is specifically designed to handle noisy and challenging audio conditions.

Pre-trained Model Lists: Video Domain

Following are some pre-trained models for video-related needs:



Pre-trained Model Lists: Video Domain

The following are the video classification models:

I3D-Kinetics-400 is a video classification model based on the Inflated 3D ConvNet (I3D) architecture.

I3D-Kinetics-600 is a variant of the I3D video classification model that is pre-trained on the Kinetics-600 dataset.

Pre-trained Model Lists: Video Domain

The following are the video generation models:

VideoFlow Encoder is a component of a video generation model that extracts high-level features from input video frames.

VideoFlow Generator is another component of a video generation model that takes the encoded features from the VideoFlow Encoder and generates new video frames.

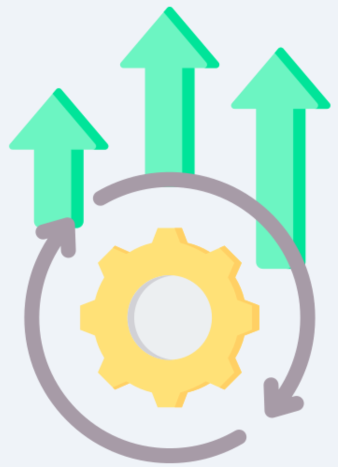
Tweening Conv3D is a video generation model that focuses on generating intermediate frames between two given frames.

Advantages of Transfer Learning

Advantages of Transfer Learning

Transfer learning brings a wide range of benefits to the development process of deep learning models, such as:

Reduced training time for models on similar datasets



Enhanced efficiency in deploying multiple deep learning models

Trains models using simulations instead of resource-intensive real-world environments

Advantages of Transfer Learning

Deep learning algorithms often require large datasets for effective training.

The key advantage of transfer learning is that it allows a model pre-trained on one dataset to be fine-tuned for other tasks, reducing the need for massive datasets each time.

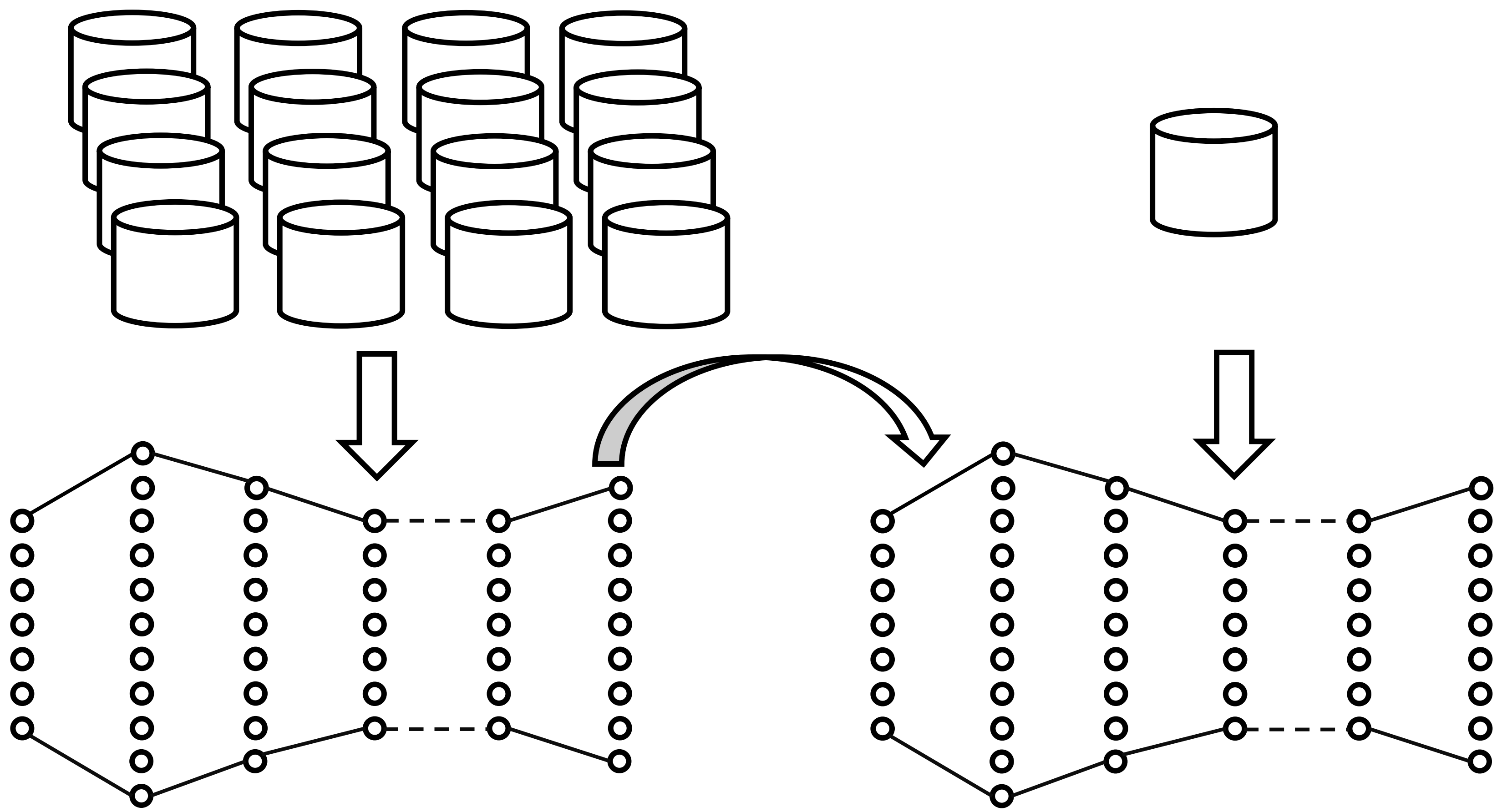
Advantages of Transfer Learning

Complex machine learning tasks can take a long time to properly train.

With transfer learning, each time a similar model is required, the model does not need to be trained from scratch.

Advantages of Transfer Learning

The time and resources spent on one model can be shared across different models.



This reduces the burden of retraining another algorithm from scratch.

Transfer Learning: Example

Simulations are created to replicate real-life environments and actions.

Consider a scenario where an agent needs to be trained to cross the road when the traffic signal is green

Create a 3D simulation to train an agent (model) for this

Integrate the model into a self-driving car system to easily infer the actions to be taken

Assisted Practices

Let’s understand the concept of transfer learning using Jupyter Notebooks.

- 8.05_Implementation of Transfer Learning

Note: Please refer to the Reference Material section to download the notebook files corresponding to each mentioned topic

ASSISTED PRACTICE

Discussion: Transfer Learning

Duration: 10 minutes

- What is transfer learning?

Answer: Transfer learning is a machine learning technique that leverages knowledge gained from training a model on one task to improve performance on a different but related task.

- How does transfer learning work?

Answer: Transfer learning works by using a pre-trained model, typically trained on a large dataset, as a starting point. The pre-trained model's learned features and weights are then utilized and fine-tuned on a new dataset or task, reducing the need for extensive training from scratch.

Key Takeaways

- Transfer learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on another task.
- There are two outcomes of transfer learning: positive transfer learning and negative transfer learning.
- In positive transfer learning, the model demonstrates improved performance on a new task compared to its original performance.
- In negative transfer learning, the performance of the model degrades when applied to a new problem compared to the original model.

Key Takeaways

- The factors to be considered when choosing a pre-trained model are size, extension, input, output, specification, and accuracy of the model.

- Transfer learning saves the time taken to train a model and improves the efficiency of the machine learning workflow for the deployment of multiple models.

Knowledge Check

Knowledge Check

1

What are the two scenarios of transfer learning?

A. Negative and positive

B. Good and bad

C. Easy and hard

D. Expensive and cheap

Knowledge Check

What are the two scenarios of transfer learning?

A. Negative and positive

B. Good and bad

C. Easy and hard

D. Expensive and cheap

The correct answer is **A**

The two scenarios of transfer learning are positive and negative transfer learning.

Knowledge Check

2

What are the factors to consider before choosing a pre-trained model?

A. Size of the model, type of model, and output requirements

B. Size of the model, extension of the model, input requirements, output requirements, model specifications and accuracy, and comparison

C. Speed of the model, type of model, and output requirements

D. Extension of the model, input requirements, output requirements, and model specifications and accuracy

Knowledge Check

2

What are the factors to consider before choosing a pre-trained model?

A. Size of the model, type of model, and output requirements

B. Size of the model, extension of the model, input requirements, output requirements, model specifications and accuracy, and comparison

C. Speed of the model, type of model, and output requirements

D. Extension of the model, input requirements, output requirements, and model specifications and accuracy

The correct answer is **B**

The factors to consider before choosing a pre-trained model are the size of the model, extension of the model, input requirements, output requirements, model specifications and accuracy, and comparison.

Knowledge Check

3

What is negative transfer learning?

A. The model's performance deteriorates with a new problem as compared to the original model.

B. The performance of the model on a new task is better than the performance of the original model.

C. The model is trained on multiple tasks simultaneously.

D. The model is not able to learn anything.

Knowledge Check

3

What is negative transfer learning?

A. The model's performance deteriorates with a new problem as compared to the original model.

B. The performance of the model on a new task is better than the performance of the original model.

C. The model is trained on multiple tasks simultaneously.

D. The model is not able to learn anything.

The correct answer is **A**

Negative transfer learning happens when the model's performance deteriorates with a new problem as compared to the original model.

Thank You!