Deep Learning with Keras and TensorFlow

Transfer Learning

Learning Objectives

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

- Understand the concept of transfer learning and differentiate between positive and negative transfer learning scenarios.
- Utilize transfer learning to achieve higher accuracy in specific deep learning applications with limited datasets.
- Implement a custom transfer learning model using pre-trained architectures and fine-tune it for specific tasks, such as image classification and object detection.
- Assess the factors to consider when selecting pre-trained models for different tasks

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.

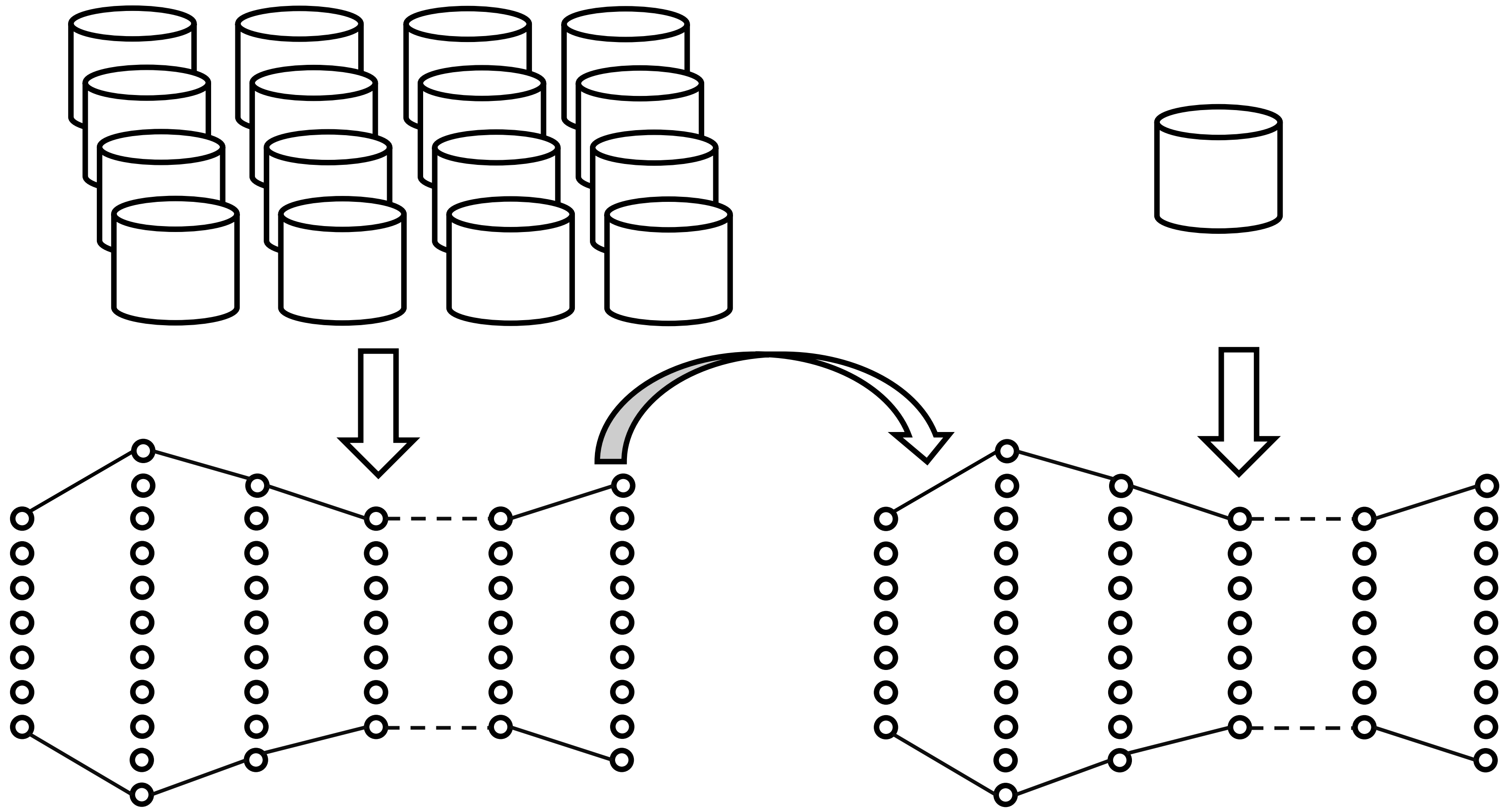
The company decides to use transfer learning, leveraging a pre-trained model developed for a similar task of image classification. The pre-trained model is modified and fine-tuned to detect rare diseases. The model is tested and achieves higher accuracy compared to a model trained from scratch.

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

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:

- Reuse the pre-trained models
- Retrain 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

In a classroom environment, both physical and virtual, a teacher's ability to gauge student engagement is crucial. Traditionally, this has been done through direct observation, which can be subjective and inconsistent.

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

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 at 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 of a different, unrelated task.

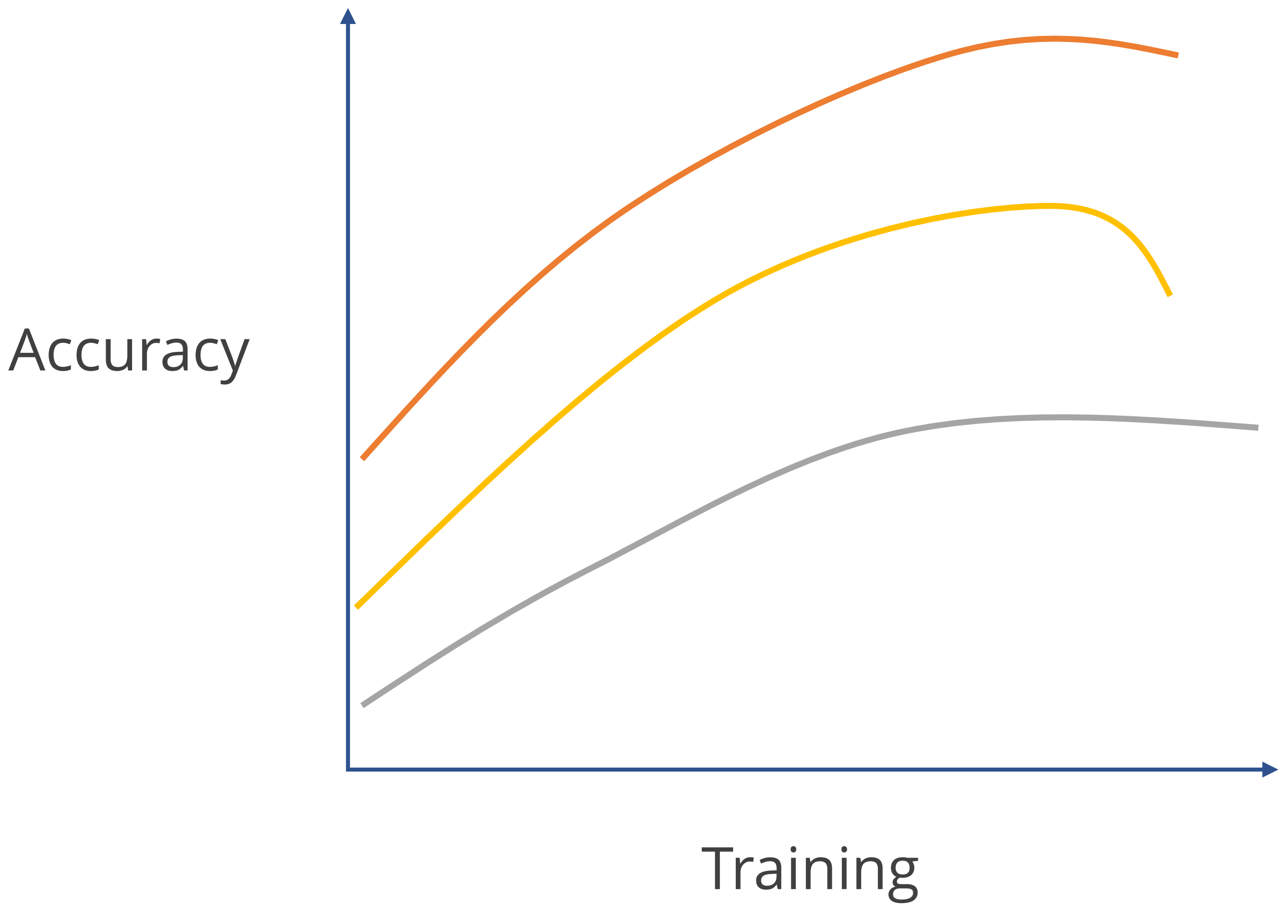
Example

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

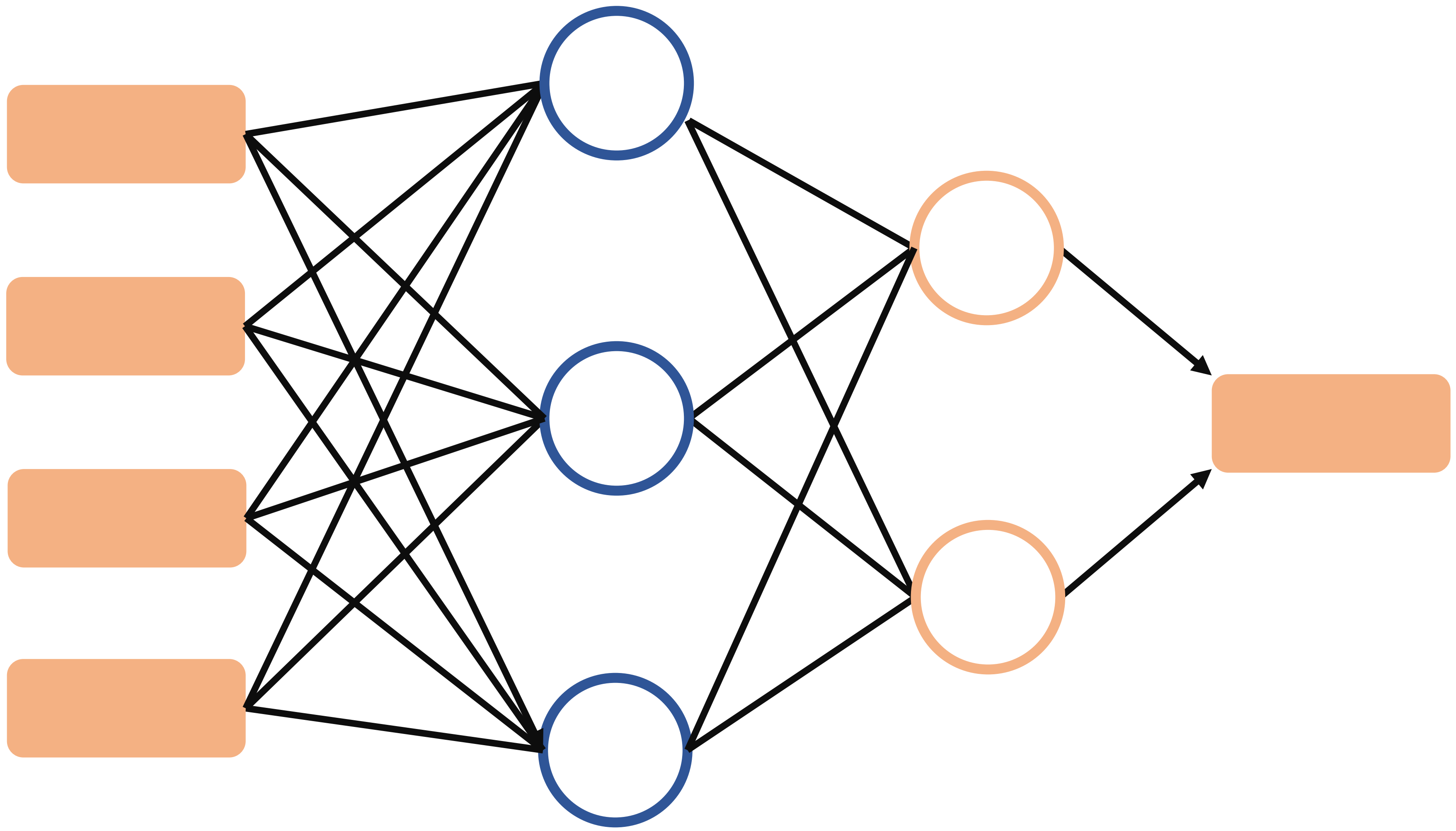
Pre-trained model → Pre-existing neural network that has been trained on a large dataset

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

How to Select Pre-trained Models?

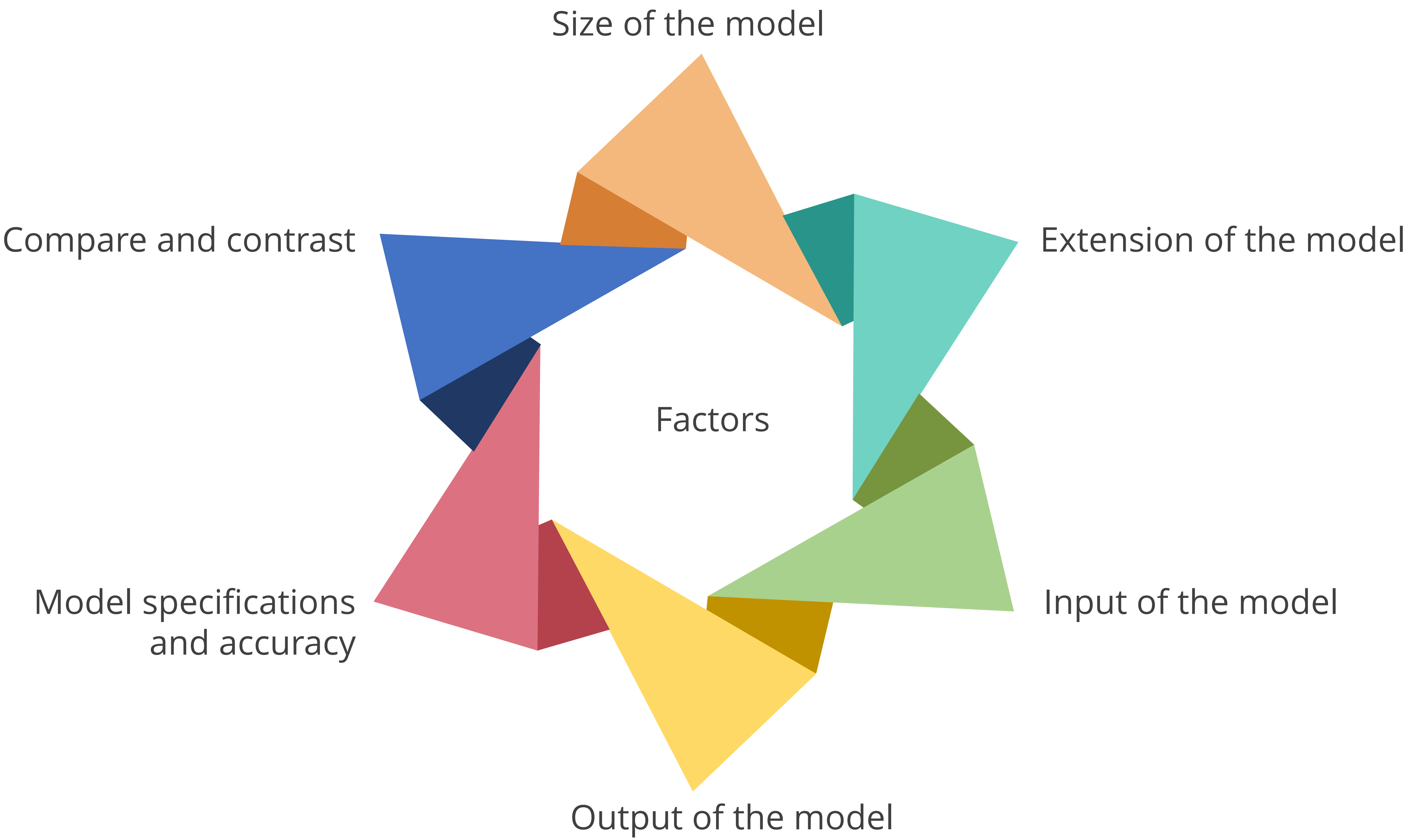
Pre-trained Models

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 capacity.

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

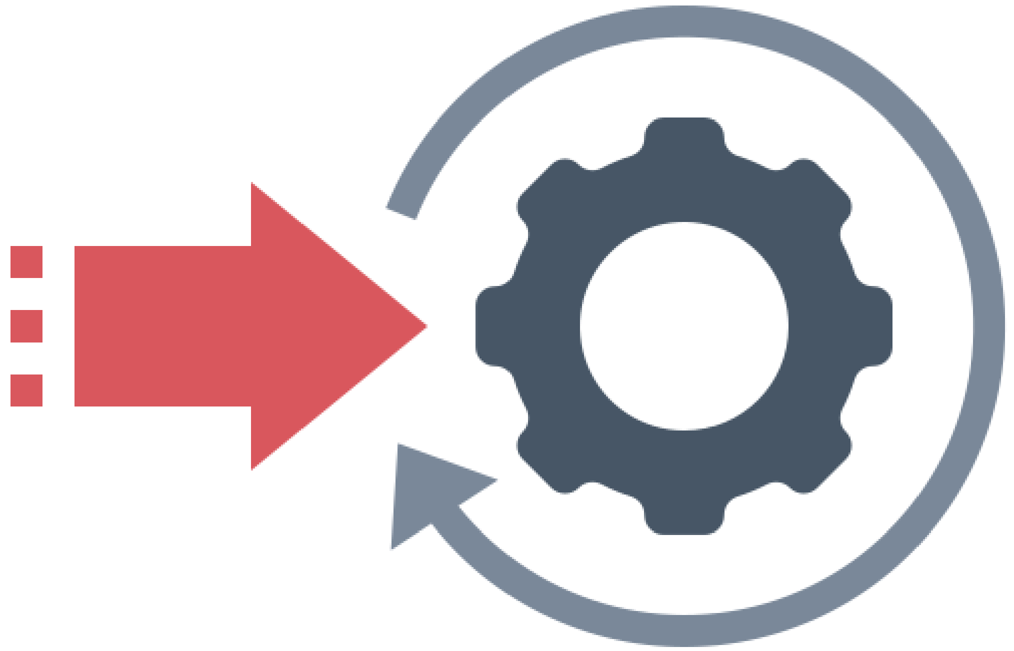
It reflects the framework on which the model was trained.

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

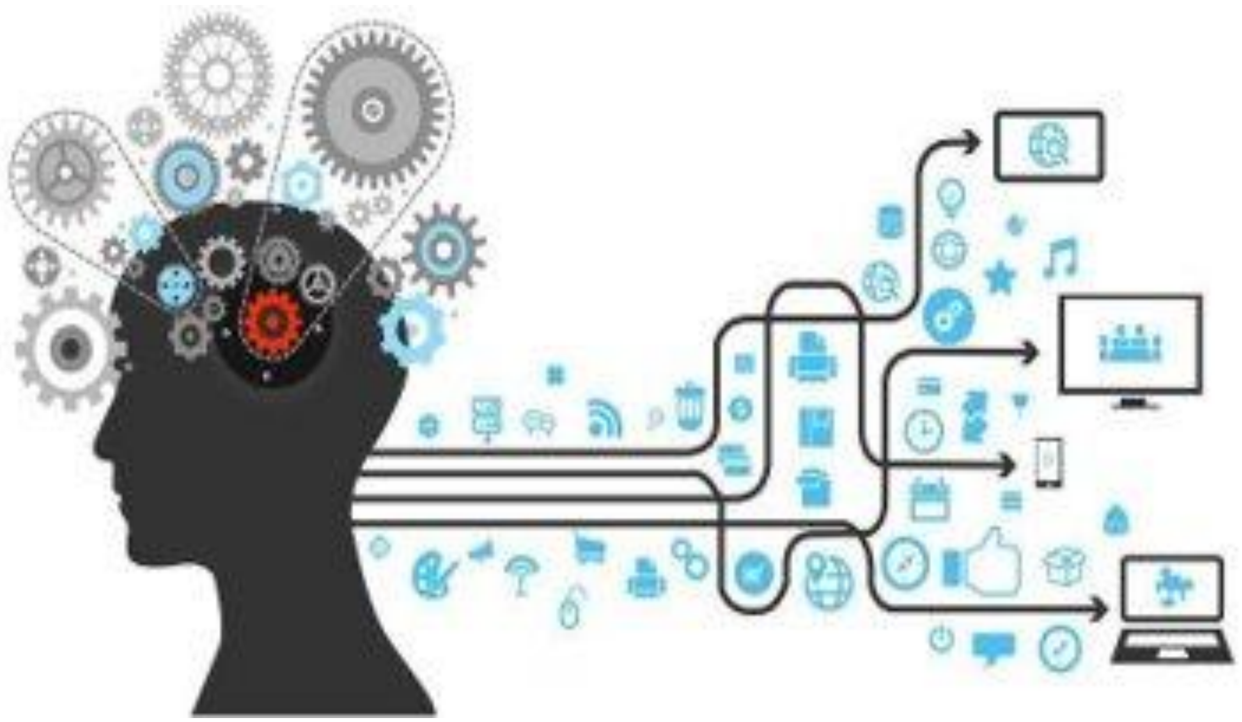
Output of the model

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: 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:

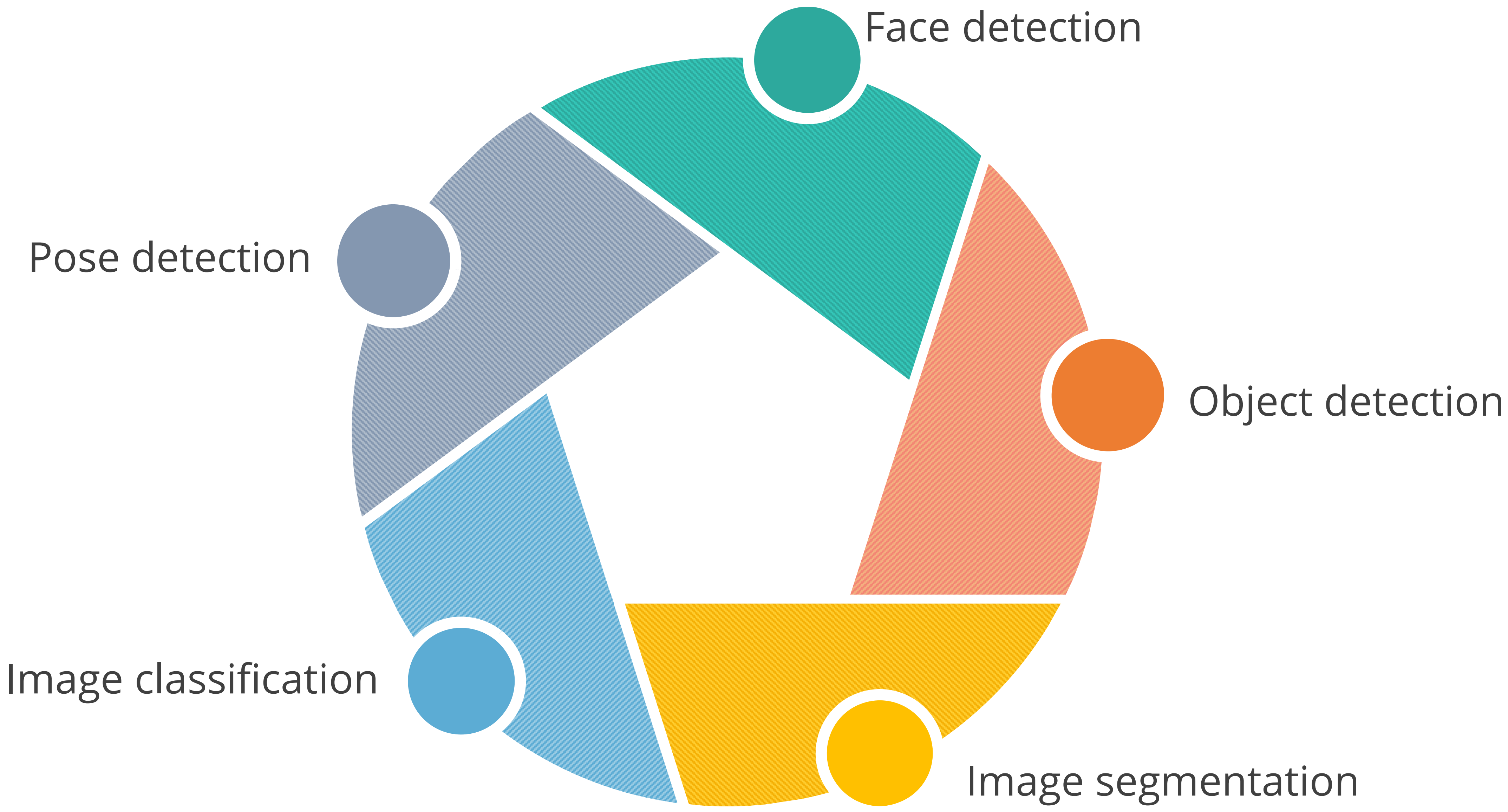
Speed: Model's prediction time.

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

There are various tasks in the image domain for which pre-trained models are available. The tasks mentioned are:



Pre-trained Model Lists: Face Detection

Face detection

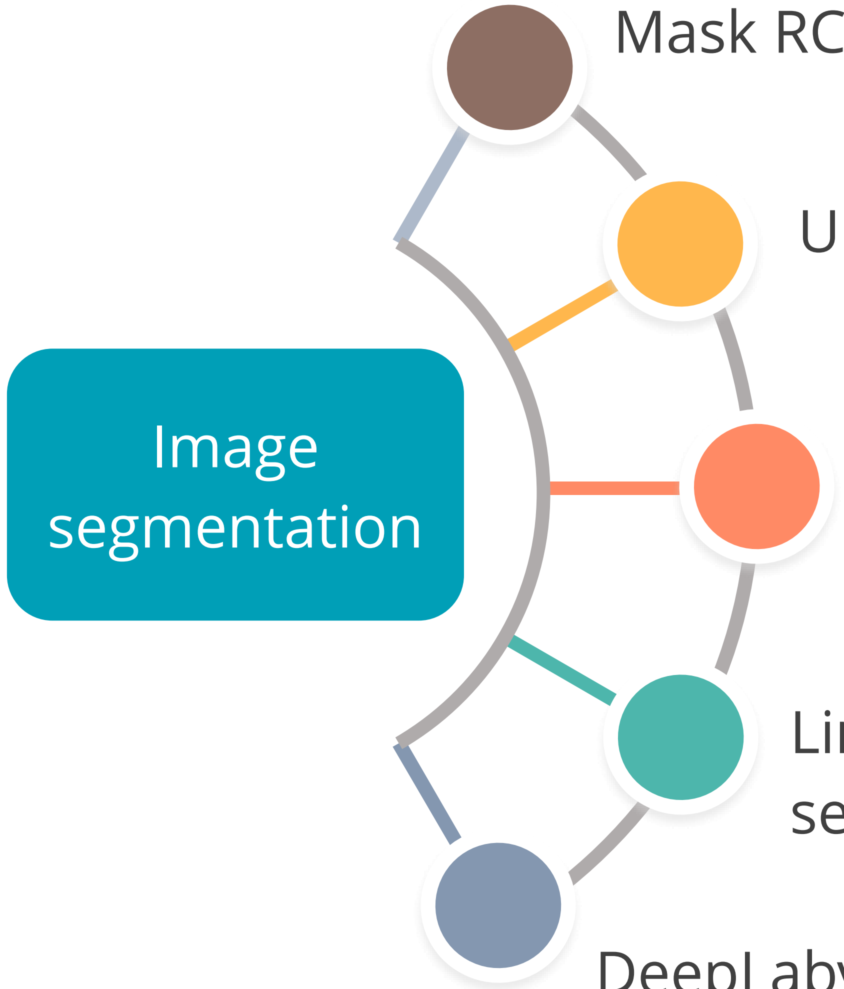
- MTCNN (Multi-task Cascaded Convolutional Networks) is a deep learning model specifically designed for face detection.
- Inception-ResNet is a hybrid model that combines the Inception and ResNet architectures.
- MobileNet is quick and effective for smartphones with limited resources.

Pre-trained Model Lists: Object Detection

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 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 Classification

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: Pose Detection

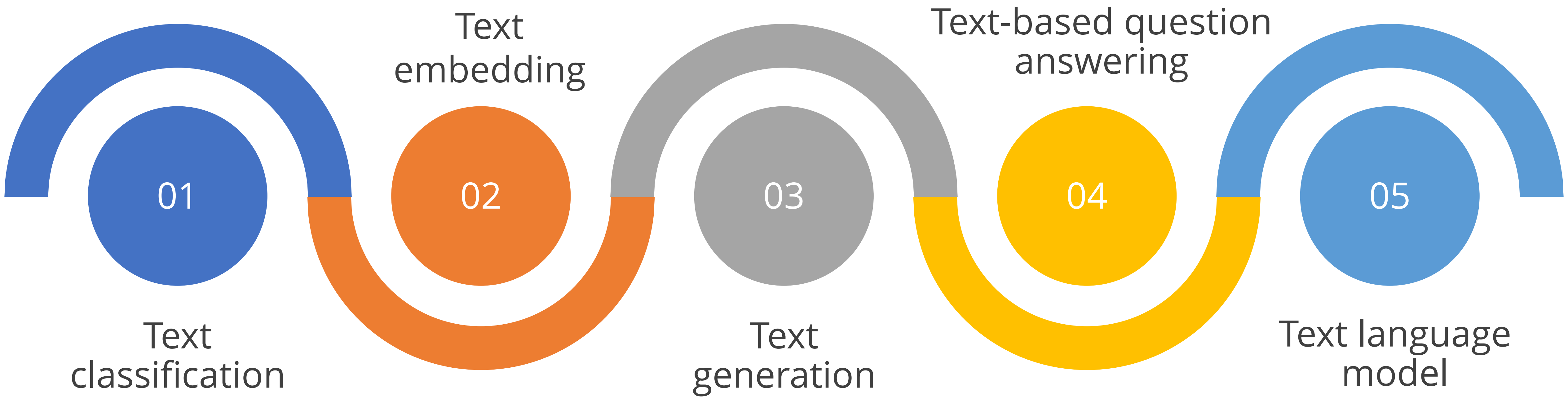
Pose detection

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

There are various tasks in the text domain for which pre-trained models are available. The tasks mentioned are:



Pre-trained Model Lists: Text Classification

The following are the text classification models:

01

XLNet: (eXtreme Language Model) uses permutation-based training to improve contextual learning, suitable for tasks like sentiment analysis and spam detection.

02

ERNIE: (Enhanced Representation through Knowledge Integration) integrates structured knowledge, outperforming BERT and XLNet in various benchmarks, making it ideal for relation extraction and sentiment analysis.

Pre-trained Model Lists: Text Embedding

The following are the text embedding models:

01 BERT: known for its bidirectional training and contextual understanding. It is used in NER, question answering, and sentiment analysis.

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 Generation

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-Based Question Answering

The following is a text-based question answering model:

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 Language Model

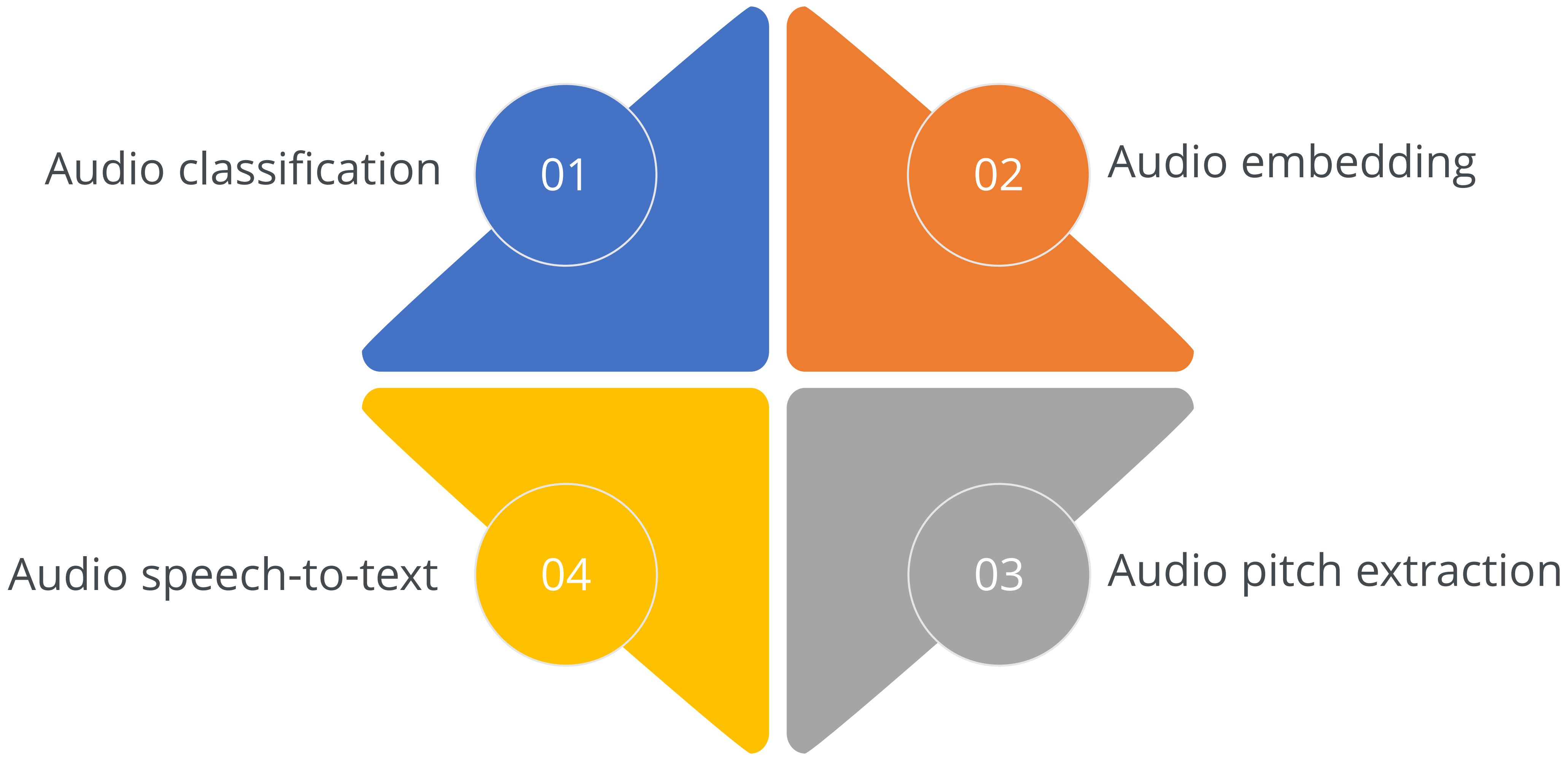
The following are the text language models:

01 GPT-4: Superior in handling longer texts, multilingual support, and factual accuracy; useful for language translation and summarization.

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

Pre-trained Model Lists: Audio Domain

There are various tasks in the Audio domain for which pre-trained models are available. The tasks mentioned are:



Pre-trained Model Lists: Audio Classification

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 Embedding

Audio embedding

- TRILL (Transferable and Interpretable Learning for Language) is an audio embedding model that learns transferable representations from speech data.
- OpenL3 is an open-source Python library that computes deep audio and image embeddings. It is based on the Look, Listen, and Learn (L3) approach, which uses both audio and visual data to learn useful representations.

Pre-trained Model Lists: Audio Pitch Extraction

Audio pitch extraction

- CREPE (Convolutional REpresentation for Pitch Estimation) is a deep convolutional neural network designed for pitch estimation directly from time-domain waveform inputs. It processes raw audio signals, making it robust to various types of noise and distortion.

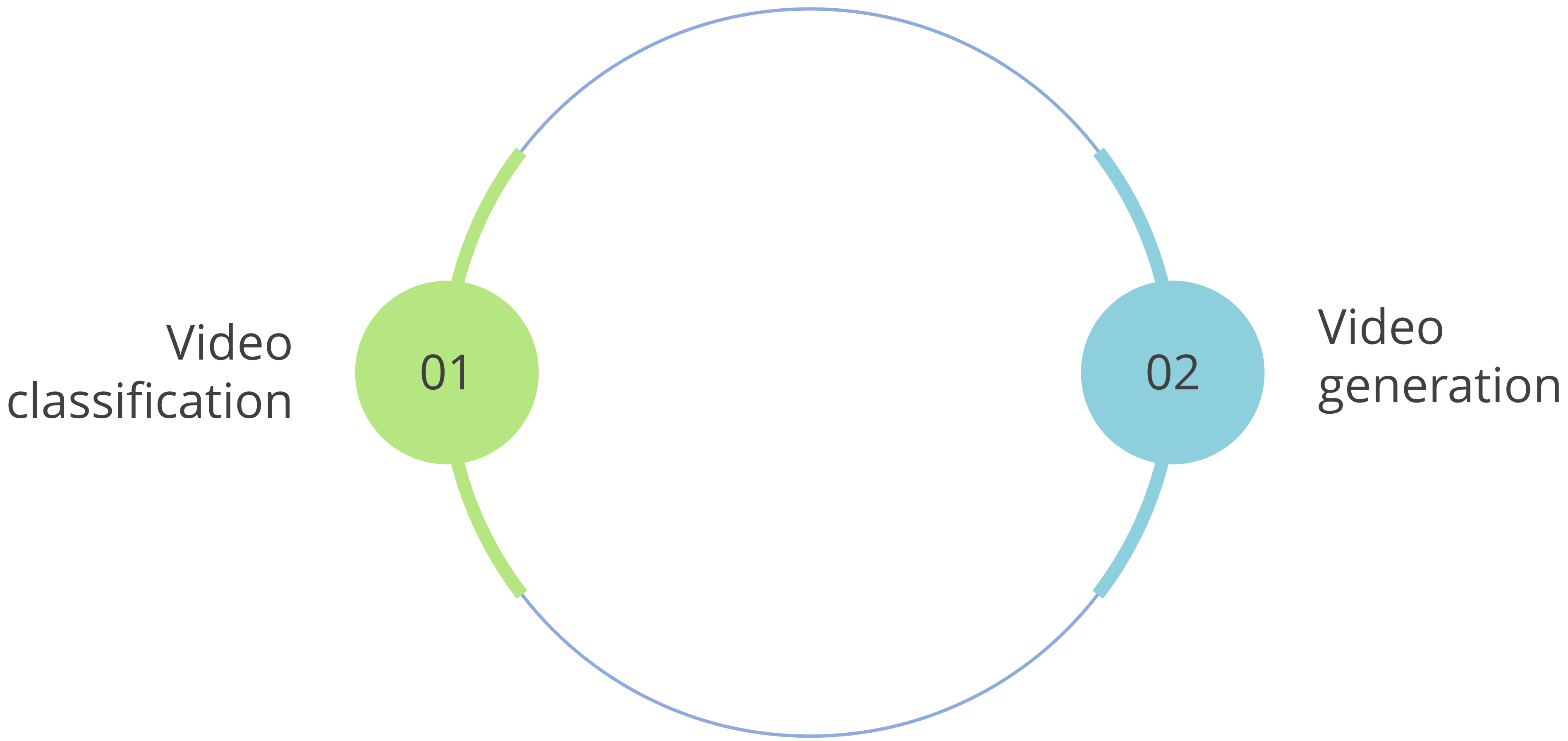
Pre-trained Model Lists: Audio Speech-To-Text

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

There are various tasks in the video domain for which pre-trained models are available. The tasks mentioned are:



Pre-trained Model Lists: Video Classification

The following are the video classification models:

VideoMAE (Masked Autoencoder for Video) is a video classification model using a masked autoencoder architecture.

ViViT uses a transformer-based architecture specifically tailored for video classification. It processes video data by applying self-attention mechanisms to capture long-range dependencies.

Pre-trained Model Lists: Video Generation

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.

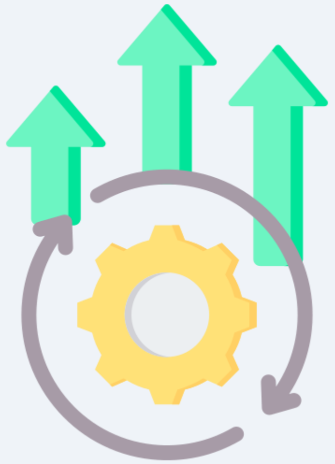
Tweening Conv3D is a video generation model that focuses on generating intermediate frames between two given frames.

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

Better model training using simulations instead of resource-intensive real-world environments

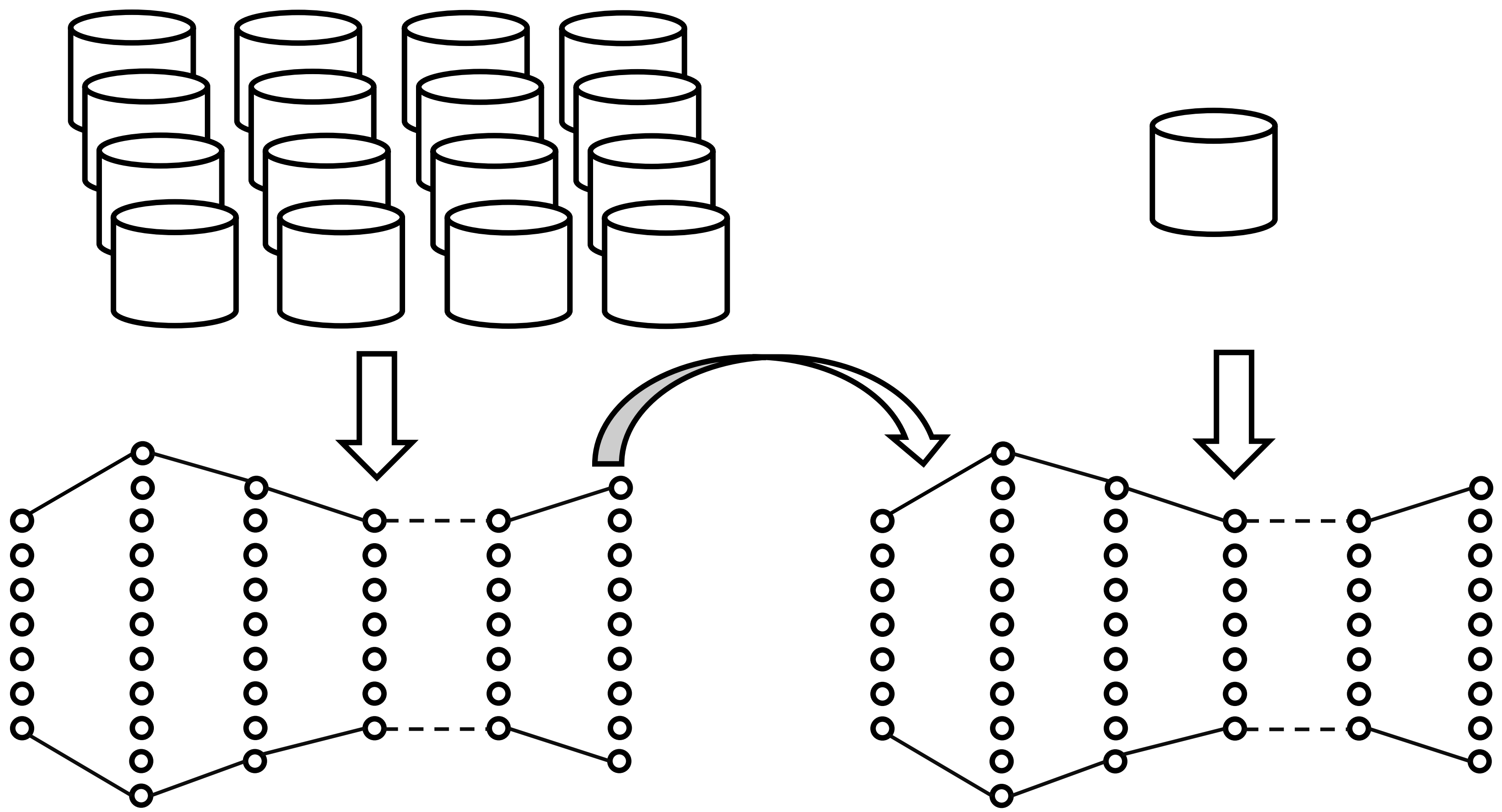
Advantages of Transfer Learning

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

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



This reduces the burden of retraining another algorithm from scratch.

Assisted Practice

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

- 9.04_Implementation of Transfer Learning

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

ASSISTED PRACTICE

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 for another task.
- There are two outcomes of transfer learning: positive transfer learning and negative transfer learning.
- 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 and 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 and 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 and 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!