# **Deep Learning vs. Traditional Computer Vision**

## **What is Deep Learning and Traditional Computer Vision**

Deep learning and traditional computer vision are two different approaches to solving problems in computer vision.

Traditional computer vision typically involves designing algorithms that process and analyze images based on handcrafted features such as edges, corners, and texture. These algorithms are often based on mathematical models and may involve techniques such as filtering, segmentation, and feature extraction. Traditional computer vision approaches can be effective in solving many image processing problems, such as object detection, tracking, and recognition, but may require a lot of domain knowledge and engineering effort to get good results.

On the other hand, deep learning uses neural networks to learn features and patterns directly from data. Deep learning models are typically trained on large datasets and can learn to recognize complex patterns and features, such as textures, shapes, and objects. Deep learning has achieved state-of-the-art performance on a wide range of computer vision tasks, such as image classification, object detection, and segmentation, and has enabled breakthroughs in areas such as self-driving cars and facial recognition.

In general, deep learning can be more effective than traditional computer vision when dealing with complex and varied datasets. However, deep learning requires large amounts of training data and computational resources and may not be suitable for all applications. Traditional computer vision approaches can be more interpretable and may be better suited for tasks that require domain-specific knowledge and expertise.

## **A Comparison of Deep Learning and Traditional Computer Vision**

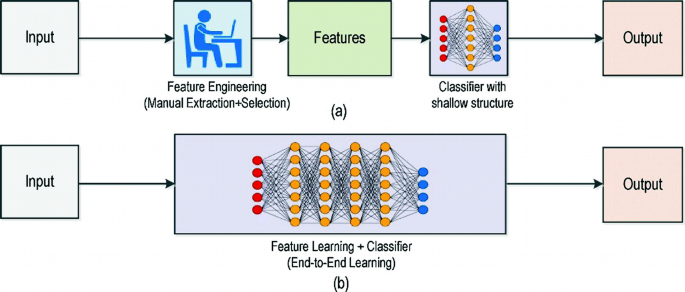
### **What is Deep Learning**

Deep learning is a subset of machine learning that uses neural networks with many layers to learn representations of data. It involves building a hierarchy of increasingly abstract features, with each layer transforming the input data into more complex representations. Deep learning has been successful in a wide range of applications and has achieved state-of-the-art performance in many tasks. It requires large amounts of labeled data and significant computational resources for training, but once trained, deep neural networks can make predictions quickly and accurately on new data.

### **Advantages of Deep Learning**

Deep learning has several advantages, including:

1. Ability to learn from unstructured data: Deep learning algorithms can learn from unstructured data such as images, audio, and text, which are difficult to analyze using traditional machine learning techniques.
2. Better accuracy: Deep learning models can achieve higher accuracy in many tasks than traditional machine learning models, especially in complex problems such as image and speech recognition.
3. Automated feature engineering: Deep learning models can learn features automatically from the input data, eliminating the need for manual feature engineering. This makes it possible to build models faster and more efficiently.
4. Scalability: Deep learning models can handle large amounts of data and can be scaled easily by adding more computing resources.
5. Generalization: Deep learning models can generalize well to new data, which means that they can perform well on data that they have not been trained on.
6. Robustness: Deep learning models can handle noisy data and can adapt to changes in the data distribution over time, making them more robust and reliable.
7. Flexibility: Deep learning models can be used in a variety of applications, including image and speech recognition, natural language processing, and autonomous systems.



**Fig. 1.** (a) Traditional Computer Vision workflow vs. (b) Deep Learning workflow.

Diagram, engineering drawing

Description automatically generated

**Fig. 2**. Building blocks of a CNN

### **Advantages of Traditional Computer Vision Techniques**

Traditional computer vision techniques also have several advantages, including:

1. Interpretable: Traditional computer vision techniques often involve hand-engineered features that are easily interpretable by humans. This makes it easier to understand why a model made a certain prediction.
2. Efficiency: Traditional computer vision techniques are often computationally efficient and can be run on lower-end hardware, making them more accessible to small businesses or individuals with limited resources.
3. Small datasets: Traditional computer vision techniques can be effective even with smaller datasets, which can be beneficial for applications with limited data availability.
4. Robustness: Traditional computer vision techniques can be more robust to changes in the input data, making them more reliable in situations where the data is noisy or incomplete.
5. Domain-specific: Traditional computer vision techniques can be customized for specific applications, such as detecting defects in manufacturing or analyzing medical images, making them more accurate and effective for those specific tasks.
6. Ease of implementation: Traditional computer vision techniques are often easier to implement and deploy than deep learning models, which can require specialized hardware and significant computing resources.
7. Well-established: Traditional computer vision techniques have been used in industry and research for many years and have a well-established body of research and expertise surrounding them.

## **Challenges for Traditional Computer Vision**

Traditional computer vision techniques have advantages such as being efficient, interpretable, and domain specific. However, they also face challenges such as limited ability to learn, scalability, and generalization. They may not be suitable for complex tasks like image and speech recognition and have limited flexibility compared to deep learning models.

## **Mixing Hand-Crafted Approaches with DL for Better Performance**

Mixing hand-crafted approaches with deep learning can lead to better performance in computer vision tasks. By combining the interpretability and domain-specific expertise of traditional computer vision techniques with the automatic feature extraction and generalization capabilities of deep learning models, it is possible to achieve higher accuracy and better robustness in complex tasks. For example, using hand-crafted features as input to a deep neural network can help the model learn faster and with less data, while also providing insights into the feature extraction process. Additionally, traditional computer vision techniques can be used to preprocess the data, removing noise or artifacts before feeding it into a deep learning model, which can improve its performance. This hybrid approach can also address some of the limitations of each technique, such as the limited scalability of traditional computer vision and the limited interpretability of deep learning models.

## **Overcoming the Challenges of Deep Learning**

There are several ways to overcome the challenges of deep learning, including:

1. Data quality: Ensuring high-quality data is critical for the success of deep learning models. This can be achieved by using data augmentation techniques, such as rotation or scaling, to increase the amount of training data and improve its diversity.
2. Computational resources: Deep learning models require significant computational resources, which can be expensive. One way to address this challenge is to use cloud-based services that provide access to powerful hardware without the need for upfront investments.
3. Overfitting: Deep learning models are prone to overfitting, which can lead to poor performance on new data. This can be addressed by using regularization techniques, such as dropout or weight decay, to prevent the model from memorizing the training data.
4. Explain ability: Deep learning models can be difficult to interpret, making it challenging to understand how the model arrived at a particular decision. This challenge can be addressed by using techniques such as attention mechanisms or visualizations to provide insight into the model's decision-making process.
5. Generalization: Deep learning models may not generalize well to new data or situations outside of the specific domain they were trained on. This can be addressed by using transfer learning, where a pre-trained model is fine-tuned on a new dataset, or by using domain-adaptive techniques, which allow the model to adapt to new domains.
6. Interpretability: Deep learning models can be difficult to interpret, making it challenging to understand how the model arrived at a particular decision. This challenge can be addressed by using techniques such as attention mechanisms or visualizations to provide insight into the model's decision-making process.
7. Model architecture: Selecting the appropriate model architecture is critical for the success of deep learning models. This can be achieved by using techniques such as hyperparameter tuning or neural architecture search to find the best architecture for a particular task.

## **Making Best Use of Edge Computing**

Edge computing is a computing paradigm that involves processing data and performing computation at the edge of the network, closer to where the data is generated. This approach can have several benefits, including reduced latency, improved reliability, and increased security. Here are some ways to make the best use of edge computing:

1. Data preprocessing: Preprocessing data at the edge can help reduce the amount of data that needs to be transmitted to the cloud or data center for processing. This can help reduce latency and bandwidth requirements, as well as improve the overall efficiency of the system.
2. Machine learning at the edge: Performing machine learning at the edge can help reduce latency and improve real-time decision-making. This can be achieved by deploying lightweight machine learning models or using techniques such as federated learning, where multiple edge devices collaborate to train a shared model.
3. Security: Edge computing can improve security by reducing the attack surface and enabling data to be processed and analyzed locally, rather than being transmitted over the network. This can be achieved by using techniques such as encryption, secure boot, and secure enclaves.
4. Resource management: Edge computing can help optimize resource usage by offloading computation from the cloud or data center to edge devices. This can help reduce costs and improve overall system efficiency.
5. Collaborative edge networks: Collaboration between edge devices can help improve overall system performance and enable new applications and use cases. This can be achieved by using techniques such as edge caching, edge analytics, and peer-to-peer communication.

Overall, making the best use of edge computing requires careful consideration of the specific use case, the available hardware and software resources, and the trade-offs between latency, bandwidth, security, and efficiency.

## **Problems Not Suited to Deep Learning**

While deep learning has achieved remarkable success in many areas, there are still several problems that are not well suited to deep learning. Here are some examples:

1. Small Datasets: Deep learning models require large amounts of data to learn effectively. If the dataset is too small, the model may overfit and perform poorly on new data. In such cases, traditional machine learning techniques or data augmentation may be more appropriate.
2. Interpretable Decision Making: Deep learning models can be difficult to interpret, making it challenging to understand how the model arrived at a particular decision. If interpretability is essential, traditional machine learning models or rule-based approaches may be more appropriate.
3. Complex Decision Rules: Deep learning models may struggle with problems that require complex decision rules or logical reasoning. In such cases, symbolic reasoning or knowledge-based systems may be more appropriate.
4. Resource Constraints: Deep learning models require significant computational resources, which can be challenging to obtain in resource-constrained environments. If resources are limited, traditional machine learning techniques or rule-based systems may be more appropriate.
5. Non-numeric Data: Deep learning models typically require input data to be represented as numeric vectors. If the data is not easily represented in this way, such as text or graphs, specialized techniques such as natural language processing or graph neural networks may be more appropriate.

Overall, deep learning is a powerful technique that has achieved remarkable success in many areas, but it is not a one-size-fits-all solution. The appropriate approach depends on the specific problem at hand and the available data and resources.