---description: "Hands-on demo comparing CNNs and Vision Transformers (ViTs) for image classification."

keywords: [deep learning, CNN, Vision Transformers, image classification, computer vision]

Hands-On Demo: Comparing CNNs and Vision Transformers (ViTs) for Image Classification

This hands-on demo aims to provide a comprehensive understanding of both CNNs and ViTs, enabling participants to implement and compare these architectures using practical examples. By the end of this session, participants will be equipped with the skills to choose and apply the appropriate model for various visual tasks, enhancing their ability to develop cutting-edge visual applications.

CNNs vs. ViTs: A Comparative Overview

Aspect	Convolutional Neural Networks (CNNs)	Vision Transformers (ViTs)
Architecture	Utilizes convolutional layers to capture spatial hierarchies.	
Strengths	Efficient in processing local features; excels in image classification and object detection.	Scalable to large datasets; effective in capturing global context.
Weaknesses	Limited in capturing long-range dependencies; may require extensive data augmentation.	Requires large datasets for training; computationally intensive.
Applications	Widely used in image classification, object detection, and segmentation.	Emerging in image classification, segmentation, and other vision tasks.
Performance	Generally performs well on smaller datasets; robust to variations in images.	Outperforms CNNs on large-scale datasets; excels in capturing complex patterns.

Practical Implementation

In this hands-on demo, we will implement both CNN and ViT architectures using the Hugging Face Transformers library. We will utilize a preprocessed dataset, split it into training and testing sets, and train both models to compare their performance on image classification tasks. The demo will cover the following steps:

1. **Dataset Preparation**: Load and preprocess the dataset, creating training and testing splits.
2. **Model Selection**: Choose appropriate CNN and ViT architectures from the Hugging Face model hub.
3. **Training**: Train both models using the training dataset, monitoring performance metrics such as accuracy and loss.
4. **Evaluation**: Evaluate both models on the testing dataset, comparing their performance and analyzing results.
5. **Reflection**: Discuss the strengths and weaknesses of each architecture based on the observed results, and explore potential applications in real-world scenarios.