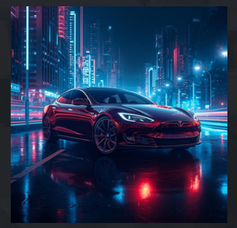
deep learning pretrained model

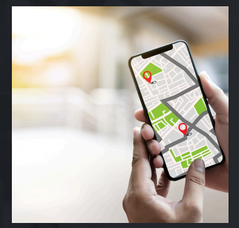
presented by
Madhusudhanan Jeyaram

A pretrained model is a deep learning model that has already been trained on a large dataset — so it already knows a lot about patterns, shapes, or language.
You can reuse it and fine-tune it for your own smaller task.

Pretraining
Global Data



Fine Tunning
YOUR LOCAL ROADS



CUSTOMIZED AI

Introduction

DATA → PRETRAINED → KNOWLED TRANSFER → CUSTOMIZED AI

Domain Types

Domain	Main Tasks	Example Models
Vision	Image Classification, Object Detection, Segmentation	ResNet, YOLO, Mask R-CNN, EfficientNet
Natural Language Processing (NLP)	Text Classification, Summarization, Translation, Q&A	Whisper, Wav2Vec2, DeepSpeech BERT, GPT, T5, RoBERTa
Speech (Audio)	Speech-to-Text, Emotion Detection, Sound Classification	Whisper, Wav2Vec2, DeepSpeech
Pose / Activity Recognition	Human Pose Estimation, Gesture Detection	OpenPose, MoveNet, BlazePose
Time Series / Forecastingn	Trend Prediction, Anomaly Detection	DeepAR, Temporal Fusion Transformer
Multimodal (Vision + Language)	Image Captioning, Text-to-Image, VQA	CLIP, BLIP, Flamingo

Core Task of Vision

Recognition - Image Classification

Identify what is in an image
Example: ResNet, VGG16, EfficientNet

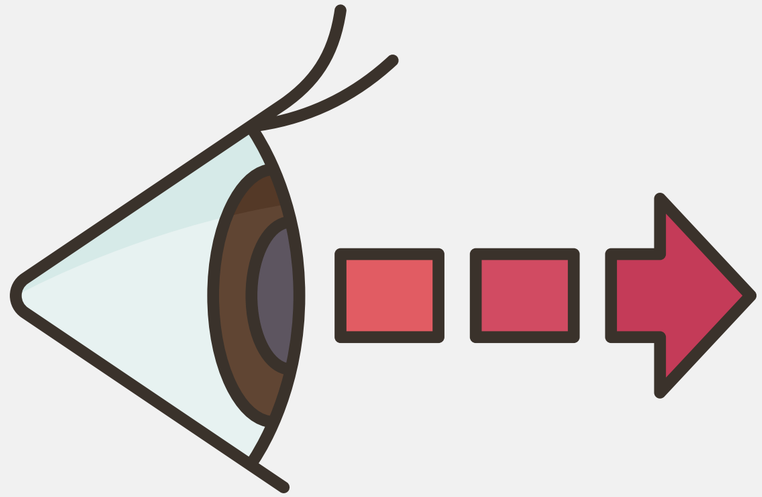
Localization - Object Detection

Find where each object is (bounding boxes)
Example: YOLO, SSD, Faster R-CNN

Pixel-Level Understanding - Semantic Segmentation

Label each pixel by object type
Example: U-Net, DeepLabV3, FCN

Vision models can see, detect, understand, recreate, and even imagine visuals.



Core Tasks - Cont..

Instance Level Understanding - Instance Segmentation

Separate different instances of the same class
Example:Mask,R-CNN, SOLOv2

Scene Understanding - Scene Classification

Recognize the type of scene (beach, city, forest)
Example: PlacesCNN, ResNet

Pose Estimation - Detect human joints & body posture

Detect human joints & body posture

Sample Code

```python
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# 1️⃣ Load Pretrained Model (without top layer)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,224,3))

# 2️⃣ Freeze pretrained layers
for layer in base_model.layers:
    layer.trainable = False

# 3️⃣ Add custom classification head
model = Sequential([
    base_model,
    Flatten(),
    Dense(128, activation='relu'),
    Dense(2, activation='softmax')  # 2 classes: cat/dog
])

# 4️⃣ Compile
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# 5️⃣ Prepare data (from folders)
train_gen = ImageDataGenerator(rescale=1./255).flow_from_directory(
    'data/train', target_size=(224,224), class_mode='categorical'
)

# 6️⃣ Train model
model.fit(train_gen, epochs=3)
```

Thank you!

madhusudhanan.jayaram@gmail.com
email address

https://madhusudhanan.online/
website