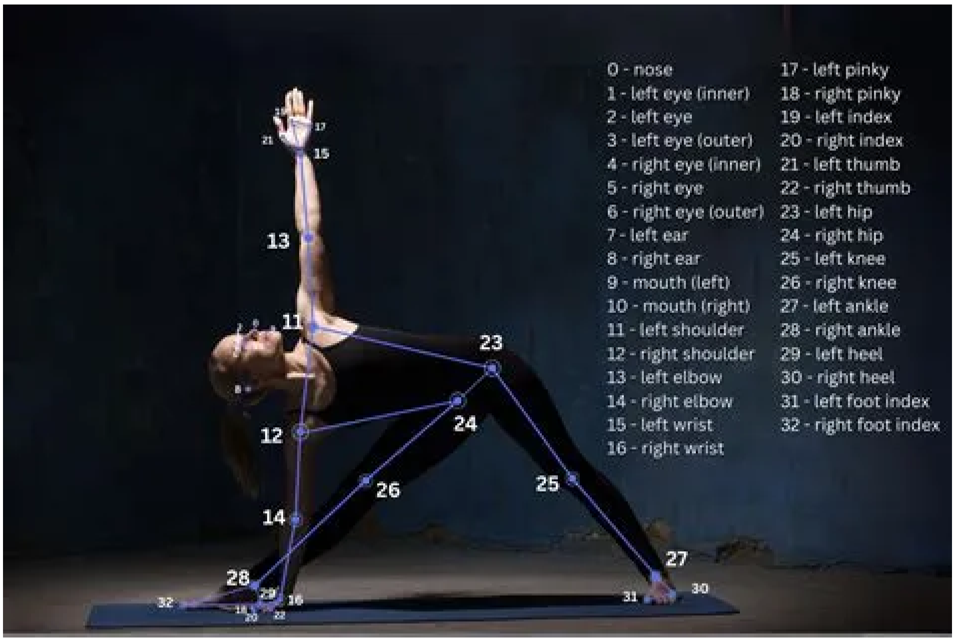


Real-Time Pose Tracking with MediaPipe: A Comprehensive Guide for Fitness Applications: Series 2

3 min read · Apr 14, 2025

Navdeep Sidana Follow

Listen Share

Real-Time Pose Tracking with MediaPipe: A Comprehensive Guide for Fitness Applications

Human pose estimation has evolved from complex research to accessible technology thanks to libraries like MediaPipe. This advanced computer vision technique identifies and tracks key points on the human body, enabling applications from fitness coaching to rehabilitation monitoring. The following article is the first in a two-part series exploring real-time pose tracking implementation and its practical applications in fitness technology.

Creating a Basic Real-Time Pose Tracker with MediaPipe and Python

Setting Up Your Environment

- **Required Libraries**
- OpenCV for video capture and image processing
- MediaPipe for pose detection capabilities
- NumPy for numerical operations and data handling
- **Development Prerequisites**
- Python 3.7 or higher installed on your system
- Basic understanding of computer vision concepts
- Familiarity with Python programming language

Implementation Pipeline

- **Initial Setup and Imports**
- Import necessary libraries (OpenCV, MediaPipe, NumPy)
- Initialize MediaPipe drawing utilities and pose module
- These components form the foundation of our pose detection system

```python
import cv2
import mediapipe as mp
import numpy as np
```

```
# MediaPipe Pose Estimation initialization
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose
```

- **Webcam Initialization**
- Configure video capture from default camera
- This creates a pipeline for receiving continuous frames for processing

```
cap = cv2.VideoCapture(0)
```

- **Pose Detection Configuration**
- Set confidence thresholds for detection and tracking
- These parameters balance accuracy with performance

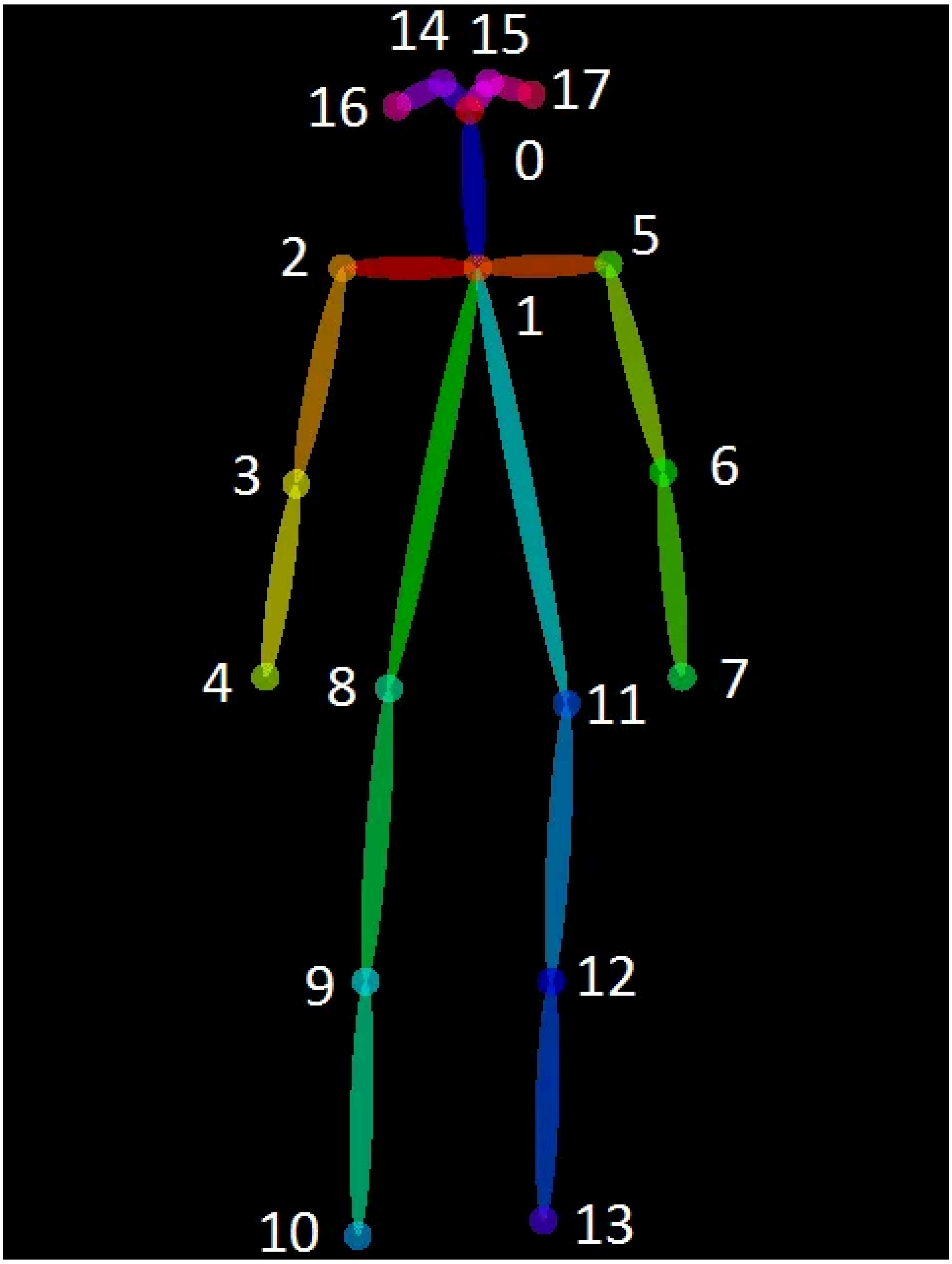
```
with mp_pose.Pose(min_detection_confidence=0.3, min_tracking_confidence=0.1) as
```

- **Processing Loop Implementation**
- Capture frames from webcam feed

- Convert color space for MediaPipe compatibility (BGR to RGB)
- Process image through pose detection model
- Convert back to BGR for display purposes
- This loop manages the continuous stream of video frames

```
while cap.isOpened():
    ret, Image = cap.read()
    Image = cv2.cvtColor(Image, cv2.COLOR_BGR2RGB)
    Image.flags.writeable = False
    results = pose.process(Image)
    Image.flags.writeable = True
    Image = cv2.cvtColor(Image, cv2.COLOR_RGB2BGR)
```

- **Visualization and Display**
- Draw detected pose landmarks and connections
- Display the processed image with annotations
- The visual feedback confirms proper detection and tracking



```
mp_drawing.draw_landmarks(Image, results.pose_landmarks, mp_pose.POSE_CONNECTIO
                          mp_drawing.DrawingSpec(color=(245, 117, 66), thicknes
                          mp_drawing.DrawingSpec(color=(245, 66, 230), thicknes
cv2.imshow('Media pipe pose detection', Image)
```

- **Graceful Termination**
- Implement keyboard interrupt (q key) for controlled exit
- Release resources and close windows properly
- This ensures clean program termination without resource leaks

```
if cv2.waitKey(10) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
```

Technical Insights on MediaPipe Pose

- **Keypoint Detection System**
- MediaPipe’s pose model identifies 30 distinct keypoints on the human body
- These points include major joints, facial landmarks, and torso markers
- The comprehensive keypoint system enables detailed pose analysis
- **Real-Time Processing Capability**
- The implementation processes frames sequentially with minimal latency
- Adjustable confidence thresholds allow balancing between speed and accuracy
- This real-time capability is essential for interactive applications
- **Landmark Visualization**
- Custom drawing specifications enhance visual clarity
- Different colors distinguish landmarks from connections
- These visualization options improve user understanding of detected poses

```
# Full Code
# %pip install opencv-python
# %pip install mediapipe

import cv2
import mediapipe as mp
import numpy as np

# MEdiapipe Pose Estimation
mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose

cap = cv2.VideoCapture(0)
with mp_pose.Pose(min_detection_confidence=0.3, min_tracking_confidence=0.1) as
    while cap.isOpened():
        ret, Image = cap.read()
        Image = cv2.cvtColor(Image, cv2.COLOR_BGR2RGB)
        Image.flags.writeable = False
        results = pose.process(Image)
        Image.flags.writeable = True
        Image = cv2.cvtColor(Image, cv2.COLOR_RGB2BGR)  # Re-enable color conve
        mp_drawing.draw_landmarks(Image, results.pose_landmarks, mp_pose.POSE_C
                                  mp_drawing.DrawingSpec(color=(245, 117, 66),
                                  mp_drawing.DrawingSpec(color=(245, 66, 230),
        cv2.imshow('Media pipe pose detection', Image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

Thanks for reading the article this far. If you like this, Please clap 93 Times and build an app to count the number of claps and I will see you in next article doing dead lifts or Fitness Exercises, code for this is available on **Github**. Happy Learning

Mediapipe Opencv Python Vision Projects

Follow

Written by Navdeep Sidana

43 followers · 49 following

Data Science | Artificial Intelligence | Machine Learning | Big Data | Fintech | Gaming | Healthcare | Software Developer | Blockchain |

No responses yet

Write a response

What are your thoughts?

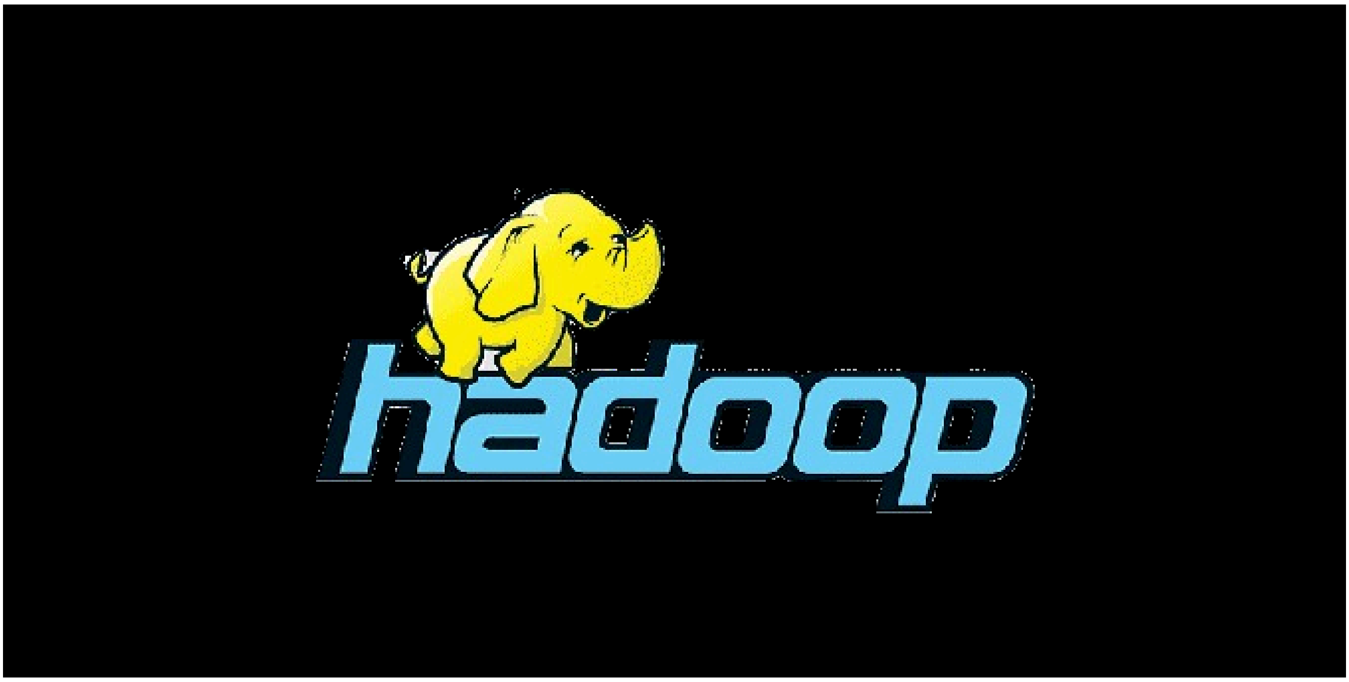
More from Navdeep Sidana

Navdeep Sidana

Before the birth of of Node.js

Node.js is a cross-platform and free to use server-side JavaScript runtime environment created from Google's V8 JavaScript engine used in...

Dec 7, 2023 23 1

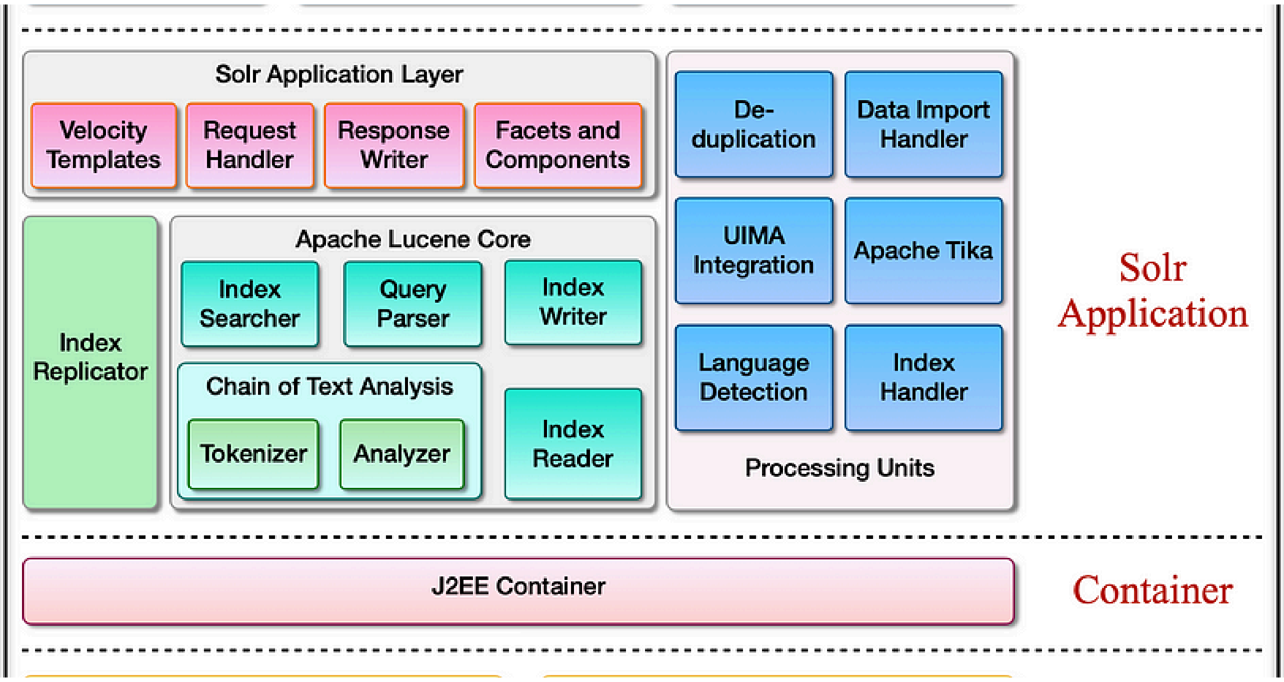


Navdeep Sidana

Installing latest Hadoop 3.4 on Ubuntu 2024: Easy Installation Guide

What is Apache Hadoop ?

Sep 3, 2024 2 1



Navdeep Sidana

How to Install and Get Started with Apache Solr 9.4. Big Data Engineering Series 1: Part 6

How to Install and Get Started with Apache Solr 9.4

Jan 16, 2024 5 1

Navdeep Sidana

Running models with Ollama step-by-step

Looking for a way to quickly test LLM without setting up the full infrastructure? That’s great because that’s exactly what we’re about to...

Apr 26, 2024 👏 25 💬 1

See all from Navdeep Sidana

Recommended from Medium

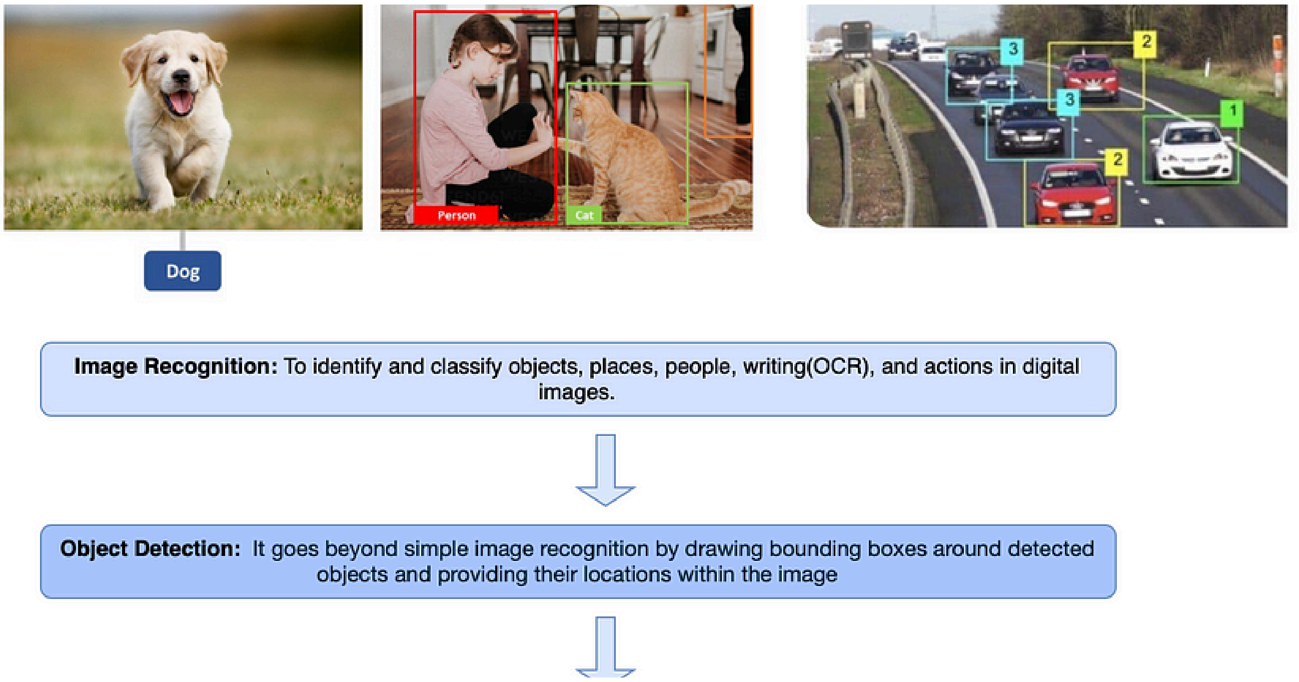


Image Recognition: To identify and classify objects, places, people, writing(OCR), and actions in digital images.

Object Detection: It goes beyond simple image recognition by drawing bounding boxes around detected objects and providing their locations within the image

In AI-ML Interview Playbook by Sajid Khan

Understanding Video Transformers for Object Recognition, Tracking and Detection

How computer vision models are evolving to see, follow, and understand moving objects across video frames

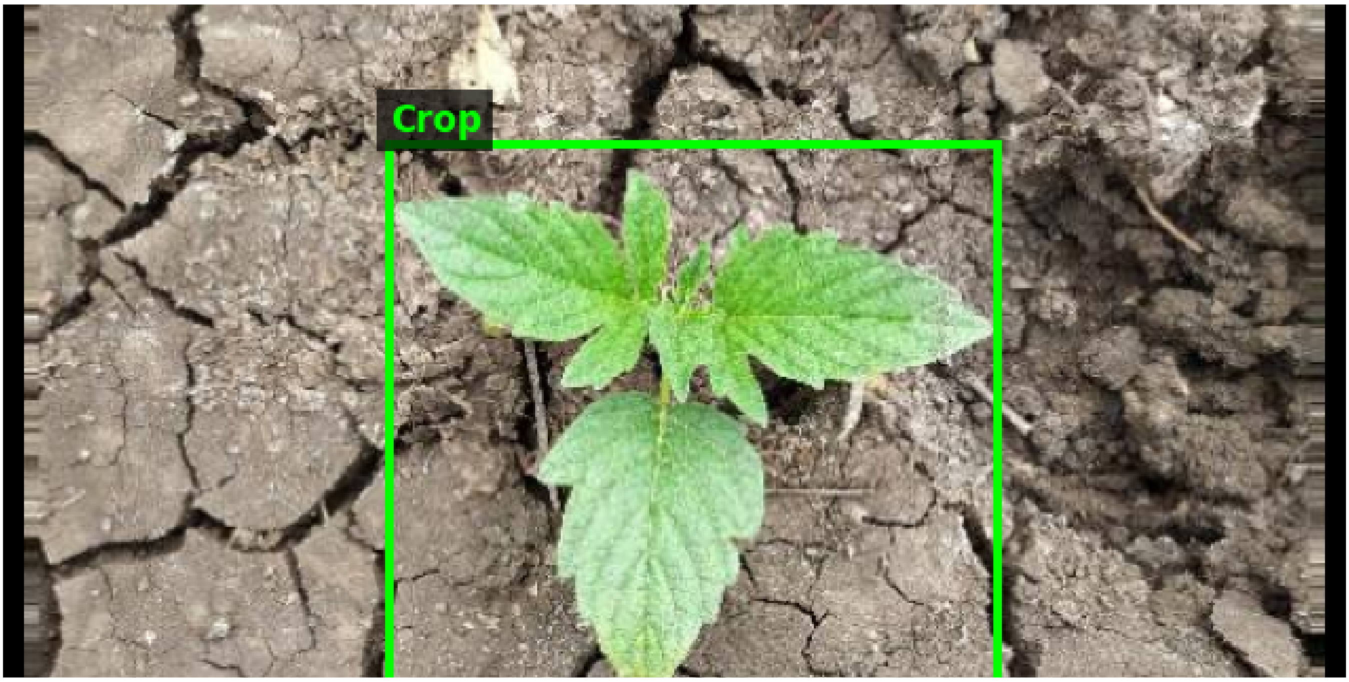
✦ 6d ago 👏 6

In TIER IV MEDIA by TIER IV

High-performance SAM2 inference framework with TensorRT

Keywords: SAM2, Instance Segmentation, TensorRT, ONNX Runtime, Optimization

Jun 25 2

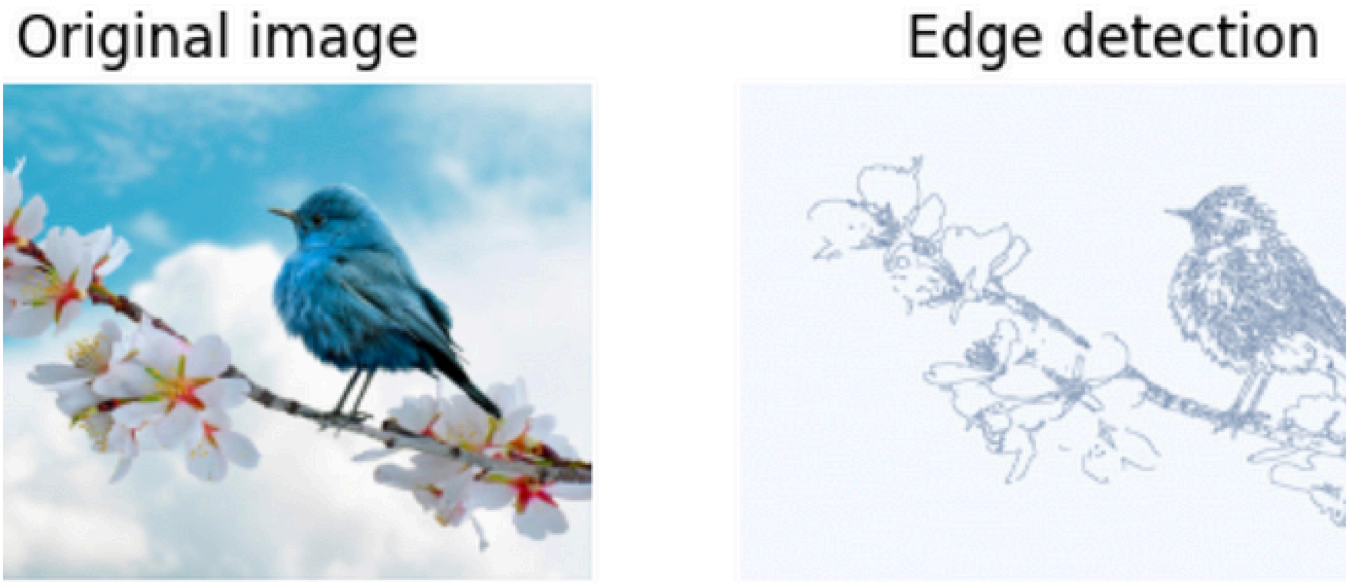


Brian Hulela

Understanding Image Annotation Formats for Object Detection

A Beginner's Guide to YOLO, COCO, Pascal VOC, and More

✦ Apr 1



In The ByteDoodle Blog by Hareesha Dandamudi

Canny Edge Detection

Image Processing with OpenCV

✦ Apr 1

In Generative AI by Ask With Ai

Top 10 Telegram Mini App Startup Ideas to Earn $4.2K/Month

A Hidden Startup idea No one’s talking about This (Save It!)

Jun 23 463 14

Sohail Saifi

Why Japanese Developers Write Code Completely Differently (And Why It Works Better)

I’ve been studying Japanese software development practices for the past three years, and what I discovered completely changed how I think…

✦ 5d ago 👏 5K 💬 151

See more recommendations