Tech Stack

VisionDeck Pro is built on a modern, high-performance technology stack that integrates state-of-the-art deep learning, real-time computer vision, and an interactive web interface. The stack is optimized for accuracy, scalability, and ease of deployment across diverse environments.

1. Deep Learning & AI Frameworks

- **YOLOv8 (Ultralytics)**
  - Core object detection architecture used for recognizing all 52 playing cards.
  - Provides high-speed inference with fine-grained classification for suits and ranks.
  - Supports GPU acceleration and flexible model customization.
- **PyTorch**
  - Backbone deep learning framework powering model training, inference, and deployment.
  - Enables CUDA support for improved performance during real-time detection.
  - Used extensively in training utilities (train.py, val.py, predict.py).

2. Computer Vision & Image Processing

- **OpenCV (cv2)**
  - Handles webcam integration, frame capture, image preprocessing, and video pipeline.
  - Enables smooth real-time operation in the Streamlit UI environment.
- **Pillow (PIL)**
  - Used for image transformations and preprocessing during dataset creation and analysis.
- **NumPy**
  - Provides fast numerical operations essential for frame processing and model input formatting.

3. Frontend & Application Layer

- **Streamlit**
    - Web-based UI framework powering the VisionDeck Pro interface.
    - Supports real-time webcam access, snapshot capture, dynamic result display, and session state handling.
    - Enables deployment without needing traditional web frameworks or frontend coding.
- **Python 3.11.9**
    - Primary language for all backend logic, UI development, data processing, and integration scripts.
    - Ensures compatibility with modern libraries and optimized performance.

4. Dataset & Model Utilities

- **Custom Dataset Pipeline**
    - Utilizes synthetic (20,000+ images), real, and augmented datasets for robust training.
    - Structured in YOLO format for consistent annotation and loading.
- **Imgaug (Augmentation Library)**
    - Used to generate the 1,000-image augmented dataset, improving generalization.
    - Includes transformations like rotation, brightness shift, and noise injection.
- **Label Studio**
    - Annotation tool used for labeling real-world card images (hearts suit dataset).

5. Development, Testing & System Tools

- **Model Utility Scripts**
    - train.py — Full training pipeline.
    - val.py — Validation and metrics generation.
    - predict.py — CLI-based inference for testing.
    - test_connectivity.py, verify_setup.py — Debugging and environment validation.

- **Python Virtual Environment (venv)**
    - Ensures isolated and conflict-free dependency management.
- **CUDA (Optional)**
    - Enables GPU-accelerated inference for significantly faster detection performance.
    - Supported via PyTorch CUDA wheels.

6. File & Project Management

- **Project Structure organized into**:
    - demo_application/ — UI, utilities, and detection pipeline.
    - final_models/ — Pre-trained YOLOv8 models.
    - data/ — Datasets (synthetic, real, augmented, combined).
    - model_utils/ — Training and evaluation scripts.
    - presentations/ — Documentation and research materials.
- **Requirements.txt**
    - Consolidates all required libraries for easy setup and reproducibility.