Introduction to Computer Vision

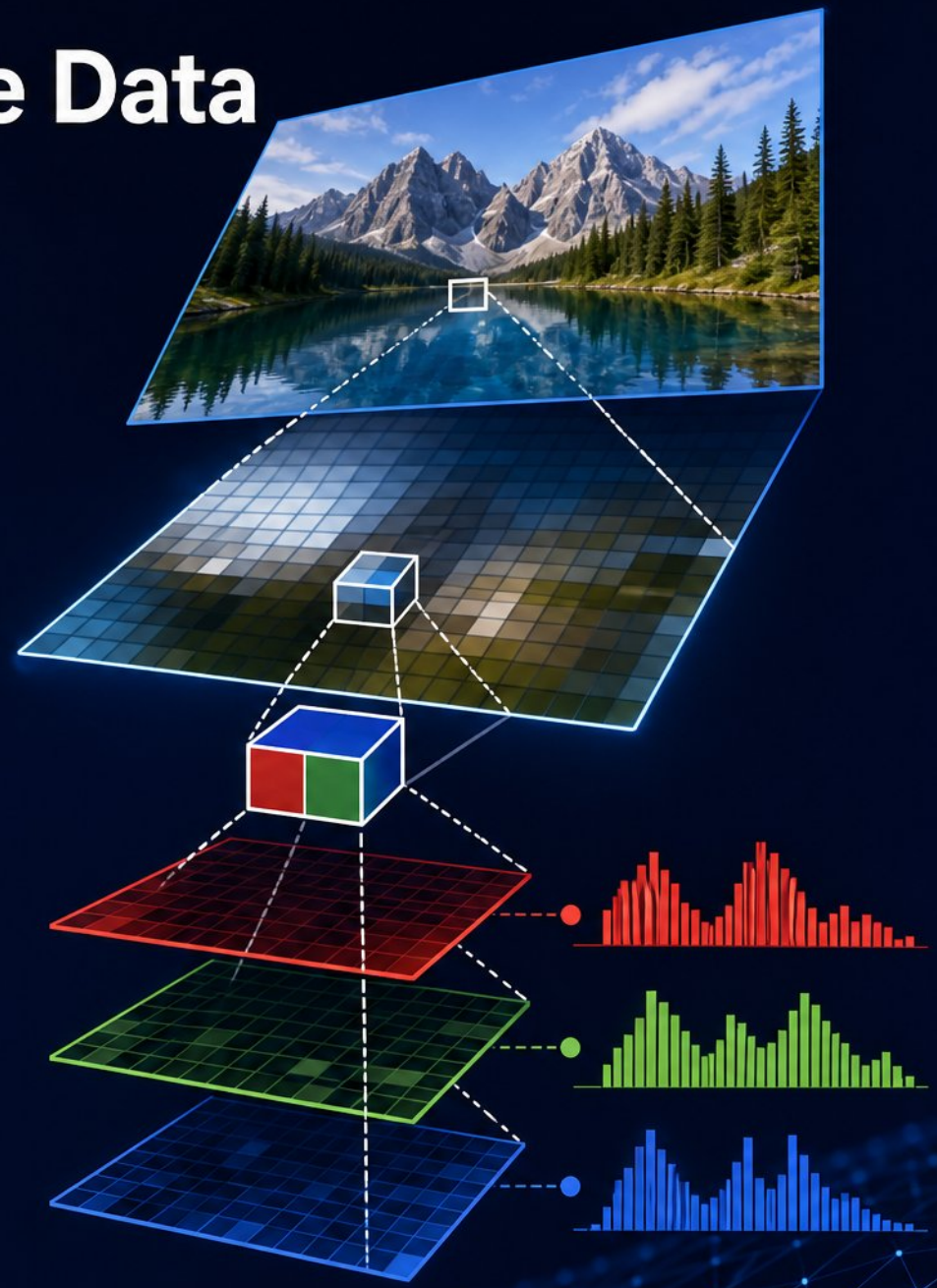
- Computer vision: AI that enables machines to interpret images, video, and real-time camera feeds
- Human vision uses biological eyes and a lifetime-trained brain; CV uses cameras, sensors, and ML models
- In 2026, the CV market exceeds $24 billion, moving from experimental to operational across all major industries
- Real-world impact: autonomous vehicles, medical image analysis, manufacturing quality control, retail automation, agriculture monitoring, document intelligence
- Course roadmap: images as data, pattern recognition, core vision tasks, model training, interpreting decisions, practical challenges, getting started

What Computer Vision Means in Practice

- CV interprets visual content; image processing just manipulates pixels
- Recognition, Reconstruction, Reorganization form the three core pillars
- Handles photos, video, medical scans, satellite, LiDAR, and documents
- Scales beyond humans: 10,000 inspections/hour; 500 camera feeds at once
- Industries: Auto, healthcare, manufacturing, retail, agriculture, security

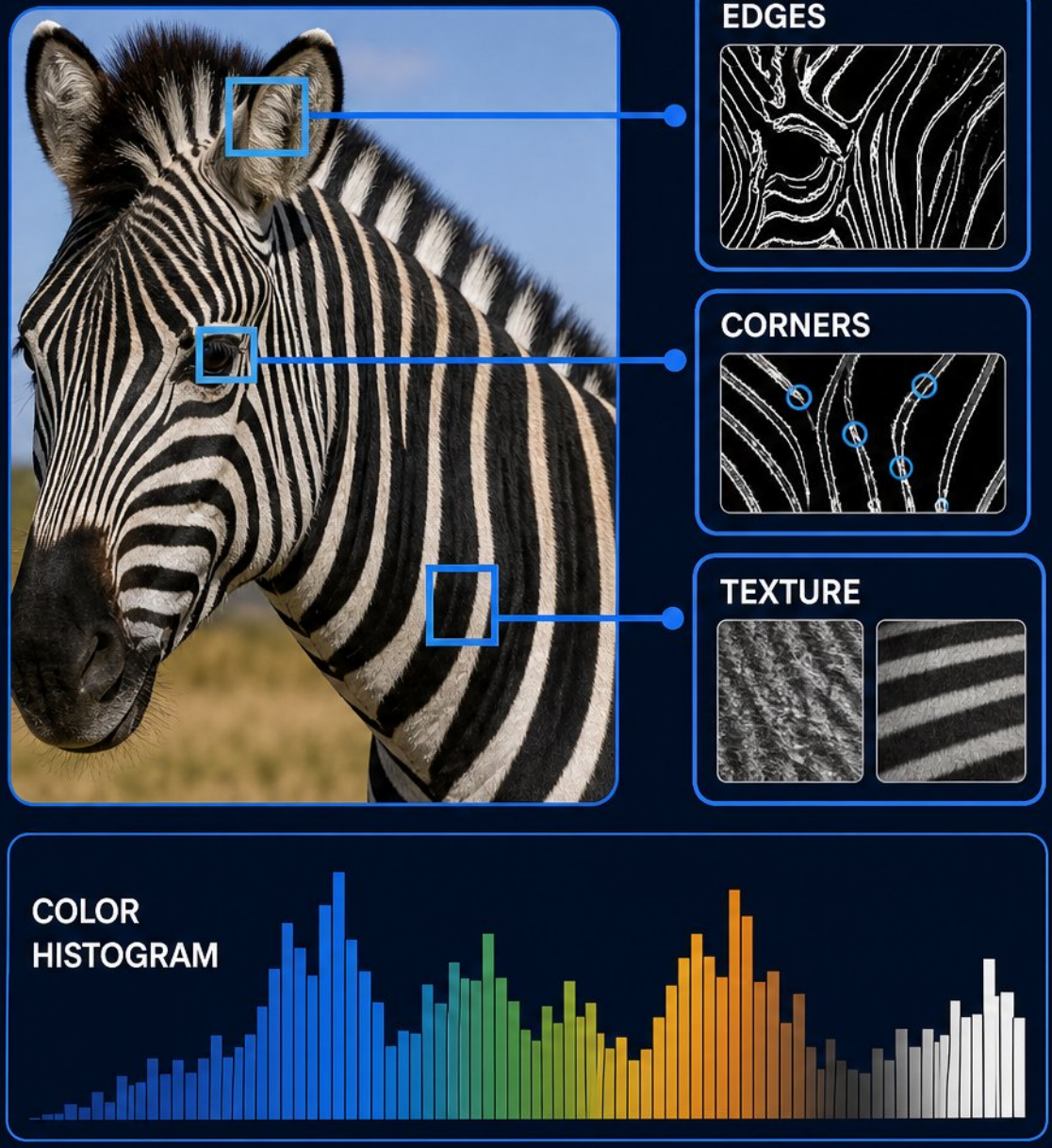
How Images Become Data

- A digital image is a grid of numbers encoding brightness and color at each pixel
- Grayscale uses one channel (0–255); RGB combines red, green, and blue channels
- Images are mathematical matrices — a 28×28 grayscale image is a 28×28 number array
- JPEG is lossy (photos); PNG is lossless with transparency — format affects task accuracy
- Preprocessing: resize, normalize, adjust contrast, and reduce noise for model-ready input



Visual Features and Patterns

- Visual features: distinctive patterns (edges, corners, textures) that serve as building blocks for understanding images
- Edge detection identifies boundaries where pixel intensity changes sharply, essential for recognizing shapes and objects
- Corners mark points where edges meet; texture captures repeating patterns like rough versus smooth surfaces
- Color histograms show global color distribution, useful for image matching and retrieval tasks
- Classic CV used handcrafted detectors (SIFT, ORB); modern deep learning automatically learns features from data

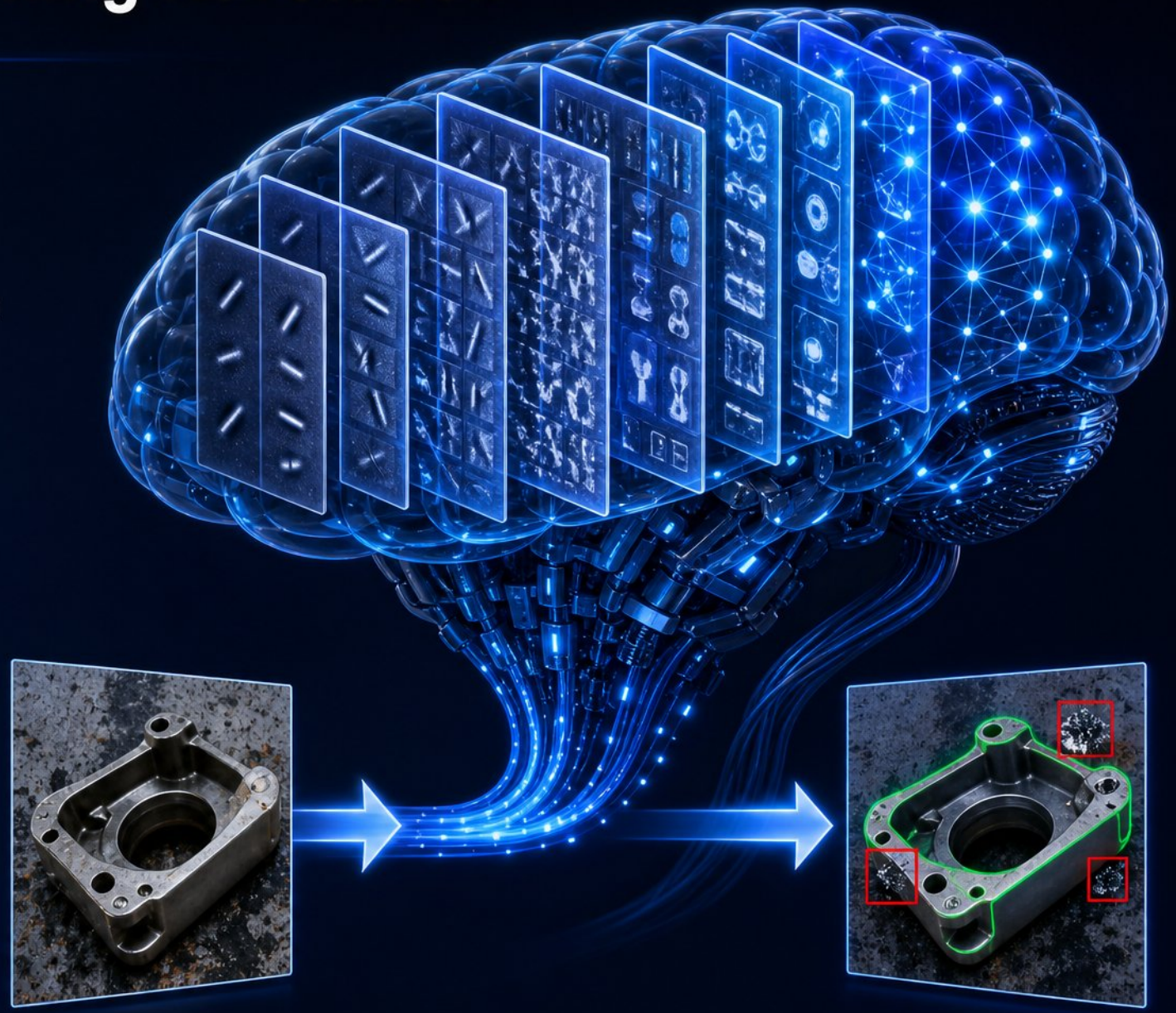


Core Vision Tasks Explained

- Image classification assigns a single label (e.g., "cat," 97% confidence)
- Object detection adds bounding boxes around multiple objects in one image
- Semantic segmentation labels every pixel by category (road, car, sky)
- Instance segmentation separates individual objects (car #1 vs. car #2)
- Other key tasks: pose estimation, OCR, object tracking, scene understanding

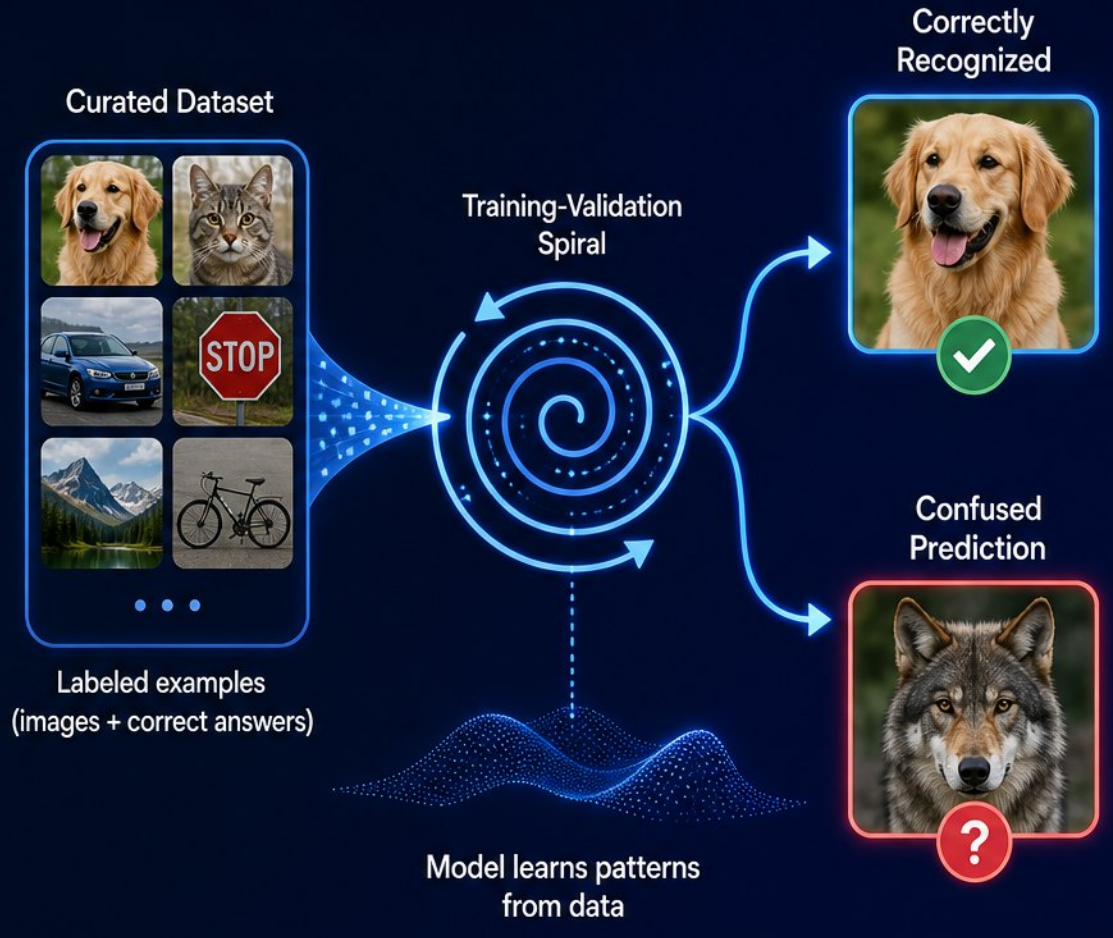
The Deep Learning Revolution

- Traditional CV used handcrafted features like SIFT and edge detectors
- Handcrafted methods fail with complex backgrounds or subtle defects
- CNNs apply learnable filters that automatically discover visual patterns
- In 2026: CNNs for speed, Vision Transformers for global accuracy
- Models now learn directly from data, achieving superhuman task performance



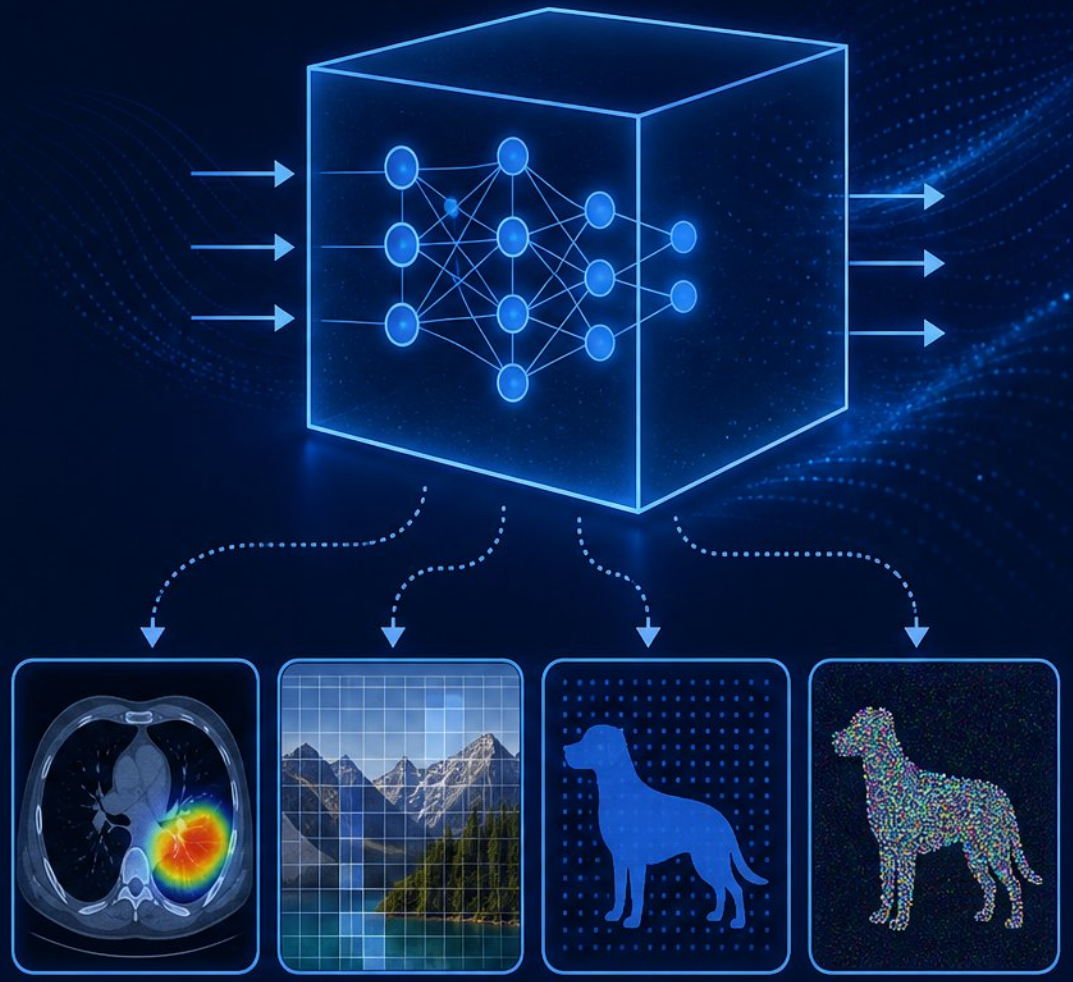
How a Model Learns to See

- **Learn from labeled examples** — images paired with correct answers
- **Split data:** training (60-80%), validation (10-20%), final test (10-20%)
- **Loss function:** a score measuring prediction error, minimized through training
- **Overfitting:** memorizing training data but failing on new images
- **Generalization:** perform reliably on real-world data never seen before



Interpreting Model Decisions

- - Blind trust in 'black box' models is risky — interpretation is essential for safety, fairness, and debugging.
- - Confidence scores can mislead: a model may be wrong with 95% probability, so always contextualize outputs.
- - Saliency maps highlight image regions a model focused on, like abnormal tissue in a medical scan.
- - Models often cheat using spurious cues (backgrounds, watermarks) instead of learning the actual object.
- - Adversarial perturbations: tiny, invisible changes can flip a model's prediction entirely.



Understanding Model Bias

- Models inherit biases from training data reflecting historical inequalities and underrepresentation.
- VLMs default to male for ambiguous inputs; high-income prompts generate lighter-skinned faces.
- Proxy bias: models use shortcuts like visible teeth for "smiling" instead of causally valid features.
- Detection via counterfactual testing, subgroup analysis, and benchmark audits (FOCUS, REFLECT, VIGNETTE).
- Mitigation through curated datasets, neuron-level intervention, and pruning biased subnetworks.

Practical Challenges in Real-World Deployment

- Lighting, angles, and occlusion can dramatically degrade model performance
- Training on studio photos fails when real factory feeds are grainy or dark
- High-quality labels require costly expert time (engineers, radiologists)
- Edge devices demand compressed models that maintain accuracy and speed
- Models degrade as the world changes—drift requires continuous retraining

Ethics, Privacy, and Regulation

- Facial recognition: >99.9% accuracy, yet raises surveillance & civil liberty concerns
- VLMs infer sensitive attributes (location, identity) from seemingly innocuous images
- Protective tech: ImageProtector perturbations & Unsafe2Safe anonymization pipelines
- EU AI Act mandates transparency, fairness docs & robustness for high-risk systems
- Privacy criteria and bias auditing must be designed in from the start — not retrofitted

Current Trends and Frontiers (2026)

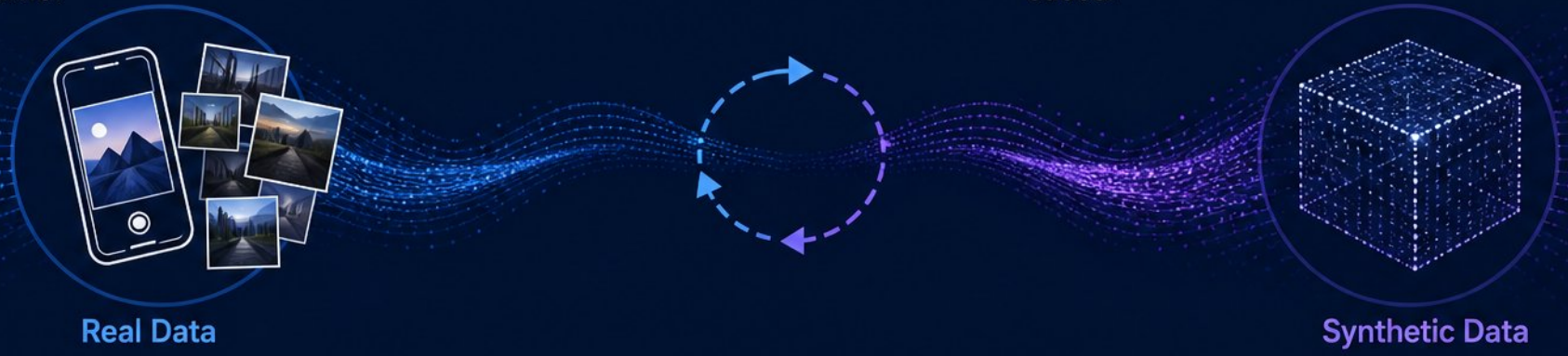
- Vision-language models like GPT-4V combine images and text for reasoning
- Vision Transformers excel with large datasets; CNNs still lead at edge
- 3D vision fuses stereo, LiDAR, and depth for robotics and AR/VR
- Document intelligence extracts data from invoices and contracts at scale
- Video understanding moves beyond detection to behavior prediction

Tools and Frameworks for Getting Started

- **Python ecosystem:** OpenCV, TensorFlow, PyTorch, scikit-image, Matplotlib, NumPy
- **Pre-trained models:** YOLO26 for detection, ResNet/EfficientNet for classification, SAM 2 for segmentation
- **Beginner platforms:** Google Colab (free GPU), Hugging Face Spaces (free deployment), Roboflow (labeling)
- **Learning path:** MNIST classification → transfer learning on custom data → real-time webcam detection
- **Community:** DataCamp, MachineLearningMastery, Kaggle competitions, runnable GitHub repos

Beginner Project Roadmap

1. **Capture Images**
   - Collect 200–500 phone images; diverse lighting beats sheer volume.
2. **Annotate Data**
   - Label with free tools (LabelImg, Roboflow); 2–3 hours is enough.
3. **Fine-tune Model**
   - Fine-tune a pretrained YOLO or MobileNet on free Colab GPU.
4. **Evaluate Model**
   - Evaluate on 20% hold-out data; report precision, recall, and failure cases.
5. **Deploy to Hugging Face Spaces**
   - Wrap in Streamlit or Gradio, deploy to Hugging Face Spaces.



Key Takeaways and Next Steps

- Computer vision transforms pixels into structured, actionable data at scale
- Powerful but fallible: bias, failure modes, and limitations demand scrutiny
- Ethics is not an afterthought — privacy, fairness, and compliance must be built in
- Next: specialize in detection, medical imaging, or vision-language models; build real projects
- Multimodal systems combining vision, language, and spatial AI are the frontier

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/d33a1017/public