Comprehensive **Roadmap** of **Machine Learning** for **Computer Vision**, organized by **Learning Priority** and **Necessary Prerequisites**.

This guide moves from fundamentals to advanced, so it can build layer by layer.

📚 **Complete Study Path for Computer Vision**

This roadmap is structured into progressive phases from foundation to frontier.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 🧱 **Phase 1: Prerequisites & Core Knowledge (Foundational Concepts)**   |  |  |  |  | | --- | --- | --- | --- | |  | These are must-know concepts before diving into computer vision:  **Goal**: Build mathematical and programming fluency.   |  |  | | --- | --- | |  | * 🧮 **Mathematics:**   + 📊 **Linear Algebra & Calculus**: Vectors, matrices, derivatives   + 📉 **Probability & Statistics**: Distributions, Bayes’ theorem, evaluation metrics * 🧠 **Core ML Principles**: Supervised vs. unsupervised learning, overfitting, bias-variance tradeoff, loss functions * 🐍 **Programming:** Python mastery (NumPy, Matplotlib, Pandas) * 🧪 **Evaluation Metrics:** Accuracy, precision, recall, F1-score |   **Priority Level**: 🚨 Essential **Prerequisite for**: All models and architectures | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 🏗️ **Phase 2: ML Model Fundamentals**   |  |  |  |  | | --- | --- | --- | --- | |  | Before working with images, understand basic model types:  **Goal**: Being familiar with Machine Learning.   |  |  | | --- | --- | |  | * 🔁 **Regression and Classification Models**: Logistic regression, decision trees, k-NN * 🕸️ **Gradient-Based Optimization**: Backpropagation, loss functions * 🎛️ **Feature Engineering**: Scaling, normalization, data splitting * 🧪 **Evaluation**: Accuracy, precision, recall, F1-score, confusion matrix |   **Priority Level**: 🚨 High **Prerequisite for**: Neural networks, CNNs | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **🔧 Phase 3: Classical Computer Vision**   |  |  |  |  | | --- | --- | --- | --- | |  | Before using ML and DL, Interacting with the image.  **Goal**: Understand rule-based image processing and feature detection.   |  |  | | --- | --- | |  | * 🖼️ **Introduction** to Computer Vision and image types (RGB, grayscale, etc.) * **📷 Image manipulation** with OpenCV (resize, crop, filters) * 📷 **Image Filters:** Edge detection (Sobel, Canny), blurring * 🔍 **Feature Extraction:** SIFT, SURF, ORB * 📐 **Image Geometry:** Homographies, Camera Calibration * 🧰 **Libraries:** OpenCV, PIL |   **Priority Level**: 🚨 Essential **Prerequisite for**: Neural networks, CNNs | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **🧠 Phase 4: Deep Learning Foundations**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Start exploring computer vision with these essentials:  **Goal**: Learn to train neural networks from scratch.   |  |  | | --- | --- | |  | * 🧱 **Neural Networks**: Feedforward networks, activation functions, backpropagation, training * 🧠 **Convolutional Neural Networks (CNNs)**: Convolutions, Filters, pooling, feature maps (activation maps) * 📦 **Frameworks**: PyTorch or TensorFlow for building models |   **Projects**:   |  |  | | --- | --- | |  | * Build a handwritten digit classifier (MNIST) * Create a custom image classifier |   **Priority Level**: ⭐ Critical **Prerequisite for**: Image classification, object detection, segmentation | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **🎯 Phase 5: Core CV (Computer Vision) Tasks**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Now ready to tackle real-world visual challenges:  **Goal**: Master practical computer vision applications.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | | **Task** | **Goal - Implement** | **Example Models/Techniques** | | --- | --- | --- | | 📷 Image Classification | Label whole image (TorchVision or Keras) | CNNs, ResNet, EfficientNet, ViT | | 🎯 Object Detection | Detect and localize objects in image (Detectron2) | YOLO, SSD, Faster R-CNN | | 🧩 Semantic Segmentation | Pixel-level class labeling (MMDetection) | U-Net, DeepLab | | 👥 Instance Segmentation | Segment individual objects | Mask R-CNN | | 🧑 Pose Estimation | OpenPose, HRNet (MediaPipe) | MediaPipe | | 👤 Face Recognition | FaceNet, ArcFace (InsightFace) | InsightFace | | 📜 Image Captioning | Generate descriptive text | CNN+RNN, Vision Transformers | | 🎨 Image Generation | Synthesize new images | GANs, Diffusion Models | |   **Projects**:   |  |  | | --- | --- | |  | * Detect objects in street scenes * Segment medical images |   **Priority Level**: 🔷 Advanced **Prerequisite for**: Application-specific CV systems | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **🧬 Phase 6: Advanced & Generative Models & Architectures**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Once fluent with vision tasks, expand into cutting-edge systems:  **Goal**: Explore emerging architectures and creative AI.   |  |  | | --- | --- | |  | * 📐 **Vision Transformers (ViT, Swin Transformer)**: Attention-based models for image data * 🔗 **Multimodal Learning**: Combine vision with language (e.g., CLIP, BLIP, Flamingo) * 🧬 **Generative Models**: GANs, VAEs, Diffusion-based Synthesis Models * 🧠 **Self-Supervised Learning**: Learn representations without labels |   **Projects**:   |  |  | | --- | --- | |  | * Train StyleGAN to generate portraits * Build an image captioning system with BLIP |   **Priority Level**: 🧠 Expert **Prerequisite for**: Research, AI applications, multimodal systems | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **🚀 Phase 7: Applications Deployment & Optimization**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Finally, bring it all into practice:  **Goal**: Productionize and refine vision systems.   |  |  | | --- | --- | |  | * 🧪 **Model Evaluation & Tuning**: Cross-Validation, Hyperparameter Search * ⚡ **Model Acceleration:** Quantization, Pruning * 📱 **Edge Deployment**: Using Models on Mobile or Embedded Devices (ONNX, TensorRT, TFLite) * 🔐 **Explainability & Ethics**: Bias detection, model transparency (Grad-CAM, SHAP) * 🧪 Hyperparameter Tuning: Optuna, Ray Tune * 🧰 **Toolkits**: OpenCV, Detectron2, MMDetection, HuggingFace |   **Projects**:   |  |  | | --- | --- | |  | * Deploy models to mobile * Create interpretable vision systems |   **Priority Level**: 🛠 Practical **Prerequisite for**: Production-level ML systems | |