

## Self Learning Checklist for AI & Computer Vision (Beginner to Advanced)



**IMPORTANT: This is the checklist not a 4 year study plan. Users must work around the curriculum and use what best works for them. Make sure to learn hands-on skills with the learning resources and make fun projects along the way.**

## **Building Foundations (approx $\approx$ 3 months )**

### **Goal:**

Create strong foundations in:

1. MATH
2. PROGRAMMING
3. ML foundations(only conceptual)

## **Programming (Python Mastery)**

### **Topics:**

- ☐ Python basics
- ☐ OOP
- ☐ NumPy
- ☐ Pandas
- ☐ Matplotlib
- ☐ Basic algorithms
- ☐ Git & GitHub

### **Resources:**

❖ freeCodeCamp :

[Object Oriented Programming in Python \(Full Course\)](#)

[Git and GitHub for Beginners - Full Course](#)

[Scientific Computing with Python Certification](#)

[Learn Python - Full Course for Beginners \[Tutorial\]YouTube · freeCodeCamp.org11 Jul 2018](#)

[Python NumPy Tutorial for BeginnersYouTube · freeCodeCamp.org2.1M+ views · 6 years ago](#)

[Pandas & Python for Data Analysis by Example – Full Course ...YouTube · freeCodeCamp.org557.4K+ views · 2 years ago](#)

[Learn Matplotlib in 1 hour!](#)

- ❖ Corey Schafer
- ❖ Tech With Tim

### **Tools:**

- VS Code
- Jupyter Notebook
- GitHub

## **Math for AI**

### **Topics:**

- Linear Algebra (vectors, matrices)
- Calculus (derivatives, gradients)
- Probability & Statistics
- Eigenvalues
- Chain rule (VERY important for CNNs)

### **Resources:**

- Khan Academy
- 3Blue1Brown (Essence of Linear Algebra). Can

## **Machine Learning Basics**

### **Topics:**

- Supervised vs Unsupervised learning
- Loss functions
- Gradient Descent

- Overfitting
- Regularization

**Course:**

- Stanford-Andrew Ng ML Course
- StatQuest

**Framework:**

- Scikit-Learn
- 

## Deep Learning Foundations (3–6 Months)

Understand Neural Networks deeply.

### Neural Networks

**Topics:**

- Perceptron
- Activation functions (ReLU, Sigmoid)
- Backpropagation
- Optimization (Adam, SGD)
- Dropout
- Batch normalization

**Tools:**

- PyTorch
- TensorFlow

**Resources:**

- DeepLearning.AI
  - PyTorch Tutorials
  - TensorFlow
-

# Computer Vision Core (6–10 Months)

Now we enter serious territory :

## Image Processing Basics

### Topics:

- Image representation
- RGB, Grayscale
- Convolution
- Filters
- Gaussian Blur
- Sobel Operator
- Canny Edge Detection
- Histogram Equalization

### Tool:

- OpenCV

### Resource:

- OpenCV
- Murtaza's Workshop

## Convolutional Neural Networks (CNN)

### Topics:

- Convolution layers
- Padding & Stride
- Pooling layers
- Feature maps
- Fully connected layers
- Transfer Learning
- ResNet
- VGG
- EfficientNet

## Projects:

- MNIST digit classifier
- CIFAR-10 image classifier
- Custom dataset classifier

## Resources:

- Stanford University CS231n
  - Aladdin Persson
- 

# Object Detection (10–14 Months)

Detects objects in real-time.

## Concepts:

- Bounding boxes
- IoU (Intersection over Union)
- Non-Max Suppression
- Anchor boxes

## Models:

- R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO (v5/v8)
- SSD

## Tools:

- Ultralytics YOLO
- PyTorch

## Resource:

- Ultralytics
- Roboflow

## Projects:

- Real-time face detector
  - Helmet detection
  - Traffic sign detector
- 

## Image Segmentation (14–18 Months)

Pixel-level classification.

## Types:

- Semantic Segmentation
- Instance Segmentation
- Panoptic Segmentation

## Models:

- U-Net
- Mask R-CNN
- DeepLabV3

## Projects:

- Medical image tumor segmentation
- Road segmentation for self-driving
- Satellite image land detection

## Advanced Topics (Optional Elite Level)

- Vision Transformers (ViT)
- GANs
- Diffusion Models
- Self-supervised learning
- Multi-modal AI (Vision + Language)
- Edge AI (Deploying on Raspberry Pi / Jetson Nano)



# Essential Tools Stack

Category	Tools
Coding	Python
Deep Learning	PyTorch
Vision	OpenCV
Data	NumPy, Pandas
Visualization	Matplotlib
Deployment	Docker
Experiment Tracking	Weights & Biases

## Final Portfolio Projects

To be considered advanced:

1. Real-time Object Detection System
2. Custom-trained Segmentation Model
3. Edge Device Deployment
4. Research paper reproduction



5. Kaggle competition

## Weekly Structure (Example)

- 5 days learning
- 2 days building
- 1 project per month
- 1 Kaggle competition every 3 months

## If You Stay Consistent

In 12–18 months:

You'll be able to:

- Build production-ready AI vision systems
- Understand research papers
- Apply for AI/ML internships
- Publish projects on GitHub