AIMS Cameroon

Structured Master's Programme, Data Science Stream

Computer Vision

Yusuf Brima, Ph.D.

Academic Year 2026–2027

Abstract

The field of Computer Vision, that is, building computational algorithms that understand the visual perceptual domain, has made significant progress over the years, particularly with the rise of modern statistical learning-based approaches. The field has come a long way from its earlier use of domain knowledge for *crafting hand-engineered descriptors or features* to the use of state-of-the-art deep learning approaches, made possible primarily by the availability of massive amounts of data and compute. The field has since found applications across all walks of life, from smart agriculture and precision medicine to robotics, among others.

Thus, this course is geared towards understanding the foundational principles and practices of the field and how to build towards and leverage current frontier methodologies and tools within it. It is aimed at providing the requisite skillset suitable for both research in the field as well as its translation into practice. The course covers the continuum from visual perception in humans to its representation and understanding in machines. This includes rudimentary digital signal processing through to modern-day *end-to-end deep learning*, with the course culminating in capstone projects to reinforce learnt knowledge into real-world problem-solving skillsets.

Reference Materials

Core Texts

1. R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022.
2. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018.

Foundational and Frontier Papers

1. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
2. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
3. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *NeurIPS*, 2015.
4. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
5. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
6. Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Cham: Springer international publishing.
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2020. Generative adversarial networks. Communications of the ACM, 63(11), pp.139-144.
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Required Software and Setup

Core Environment

- Python 3.10 or later
- A package/environment manager: `conda` (Anaconda or Miniconda) **or** `venv` with `pip`
- Jupyter Notebook / JupyterLab
- Git (for cloning course repositories and submitting practicals)
- A code editor: VS Code (recommended) or PyCharm

Scientific and Data Libraries

- `numpy`
- `pandas`
- `matplotlib` and `seaborn`
- `scikit-learn`
- `scikit-image`

Computer Vision and Deep Learning

- `opencv-python` (OpenCV)
- `Pillow` (PIL)
- `torch` and `torchvision` (PyTorch $\geq$ 2.0)
- `ultralytics` (YOLO12)
- `albumentations` (image augmentation)
- `timm` (pretrained vision models)

Hardware and Compute

- A laptop/workstation with at least 8 GB RAM (16 GB recommended)
- A CUDA-capable GPU is **not required**; a free Google Colab or Kaggle Notebooks account is sufficient and recommended for GPU-intensive labs
- Those with a local NVIDIA GPU should additionally install the appropriate CUDA Toolkit and cuDNN matching their PyTorch build