INNOPOLIS UNIVERSITY

To: M24-RO-01 (master students)

From: Alexei Kornaev, Kirill Yakovlev

Subject: Course on Computer Vision (CV-2025)

Date: March 4, 2025

---

1 Lectures

1. **Intro to Computer Vision** [8, 26, 18, 21, 25, 12, 16]
   - **Theory:** overview of computer vision, it's history, applications, and challenges; image formation; basics of deterministic image processing using filtering, edge detection, and feature extraction; the simplest algorithms of CV for image classification.
   - **Skills:** OpenCV tools for image preprocessing and automatic feature extraction, Pytorch framework, Pytorch Lightning as a tool to simplify the code, and ClearML tool to organize the experiments; comparison of simple algorithms with and without preliminary feature extraction.
2. **Convolutional Neural Networks** [22, 8, 1, 6, 7, 10, 15]
   - **Theory:** Intuition behind convolutions, filters, kernels, and feature maps. Padding, pooling, and striding. Tips and tricks: residual connections, batch normalization. Backpropagation in CNNs and optimization techniques. Architectures: AlexNet, VGG, ResNet, DenseNet, Googlenet, ConvNext, etc. Transfer learning and fine-tuning pre-trained models.
   - **Skills:** Implementation of CNNs using PyTorch on datasets like MNIST and CIFAR-10. Apply transfer learning to adapt pre-trained models (e.g., ResNet, ConvNeXt) to new tasks. Fine-tuning a pre-trained CNN on a custom dataset.
3. **Visual transformers** [22, 28, 5]
   - **Theory:** Transformer architecture: embeddings, positional encoding, self-attention mechanism, multi-head attention, classification block. Vision Transformers (ViTs): patch embeddings, transformer blocks, and classification heads. Comparison of CNNs and ViTs in vision tasks.
   - **Skills:** Implement a Vision Transformer using PyTorch or Hugging Face. Fine-tune ViTs for image classification or segmentation tasks. Compare the performance of ViTs and CNNs on a benchmark dataset.
4. **Assignment # 01**
5. **Segmentation and Object Detection** [22, 2, 27, 9, 11]
   - **Theory:** Semantic segmentation: U-Net, DeepLab, and Mask R-CNN. Object detection: YOLO, Faster R-CNN, and SSD. Instance segmentation and panoptic segmentation. Segment anything model (SAM).

- **Skills:** Implement segmentation models using PyTorch or TensorFlow. Train an object detection model on COCO or Pascal VOC datasets. Perform instance segmentation on a custom dataset.

6. **Maps of Depth and Landmarks Detection** [1, 22, 19, 13]

   - **Theory:** Depth estimation: stereo vision, monocular depth estimation, and LiDAR. Landmark detection: facial landmarks, pose estimation, and keypoint detection.
   - **Skills:** Implement depth estimation using stereo images or monocular methods. Detect facial landmarks using pre-trained models (e.g., dlib or MediaPipe). Build a depth map from a stereo camera setup.

7. **Face Recognition** [29, 22]

   - **Theory:** Face recognition: Traditional methods and deep learning methods. Challenges: pose variation, lighting, occlusion, ethics, adversarial attacks. Architectures (Siamese networks, FaceNet, and ArcFace) and loss functions (contrastive loss, triplet loss, ArcFace loss), and metrics in face recognition.
   - **Skills:** Train a face recognition model using FaceNet or ArcFace. Taking part in CV competitions using Kaggle platform. Implement face detection and recognition in real-time using OpenCV. Build a face recognition system for attendance tracking.

8. **Midterm Exam**

9. **3D Image Processing** [22, 2, 24]

   - **Theory:** 3D reconstruction: structure from motion (SfM) and multi-view stereo. 3D CNNs in medical image processing.
   - **Skills:** Reconstruct 3D models from multiple images using Open3D or PCL. 3D images processing. Hands-on: Build a 3D model from a sequence of images.

10. **Cloud of Points Processing** [1, 23, 14]

    - **Theory:** Point cloud representation: voxelization, octrees, and graph-based methods. Deep learning on point clouds: PointNet, PointNet++, and DGCNN.
    - **Skills:** Implement PointNet for point cloud classification. Perform point cloud segmentation using DGCNN. Hands-on: Classify objects in a LiDAR point cloud dataset.

11. **Video Data Processing** [22]

    - **Theory:** Video analysis: optical flow, action recognition, and video summarization. Temporal modeling: RNNs, LSTMs, and 3D CNNs. Pytorch video library.
    - **Skills:** Extract optical flow from video sequences using OpenCV. Train a 3D CNN for action recognition on UCF101 or Kinetics datasets. Hands-on: Build a video summarization system.

12. **Object Tracking on Videos** [22, 3, 17]

    - **Theory:** Tracking algorithms: Kalman filters, particle filters, and deep learning-based methods. Challenges: occlusion, scale variation, and real-time processing.
    - **Skills:** Implement object tracking using SORT or DeepSORT. Evaluate tracking performance on MOTChallenge datasets. Hands-on: Track multiple objects in a video stream.

13. **Multimodal Data Processing: Image and Text Classification** [30, 31, 22]
    - **Theory:** Vision-language models: CLIP, BLIP, and Flamingo. Applications in image captioning and visual question answering.
    - **Skills:** Fine-tune CLIP for image-text matching tasks. Hands-on: Generate captions for images using BLIP.
14. **A lecture of the student's choise**
    - **Multimodal Data Processing: Image recognition and control** []
    - **Approaching Artificial General Intelligence (AGI)** []
    - **Pose Estimation** []
    - **3D Reconstruction from 2D Images or Videl** []
    - **Generative Models** []
    - **Scene Understanding** []
    - **Visual Question Answering (VQA)** []
    - **Mixing and Tuning of the Models** []

2 Practical sessions

1. **Feature Extraction and Machine Learning** [12, 16]
2. **Convolutional Neural Networks** [12]
3. **Visual Transformers** [12]
4. **Assignment # 01**
5. **Segmentation and Object Detection** [12]
6. **Maps of Depth** [12]
7. **Face recognition** [12]
8. **Midterm Exam**
9. **3D Image Processing** [12]
10. **Cloud of Points Processing** [12, 1]
11. **Video Data Processing** [12]
12. **Object Tracking on Videos** [12]
13. **Multimodal Data Processing: Image and Text classification** [12]
14. **Deploy of a CV model** [12]
15. **Defend of the projects**

3 Assignments

1. **Computer Vision with Real-World Data**

   - **Challenge**: It is human nature to make mistakes. How, then, can the accuracy of the trainable models be improved?
   - **Task**: Design and implement algorithms and models that operate with noisy data (noise in labels, and/or out-of-distribution domain), measure their performance and formulate recommendations on operating with noisy data
   - **Dataset**: CIFAR-10N [20]
   - **Skills**: operating with noisy data, comparison of different loss functions and network architectures, analysis of experimental results

2. **Multimodal Scene Understanding for Robotics**

   - **Challenge**: Robots need to understand their environment using multiple sensors (e.g., cameras, depth sensors) to navigate and interact effectively.
   - **Task**: Create a system that combines RGB images and depth data to perform scene segmentation or object detection for robotic navigation.
   - **Dataset**: Use the **NYU Depth V2** dataset [4], which provides RGB-D images for indoor scenes.
   - **Skills**: Depth estimation, semantic segmentation (e.g., U-Net, DeepLab), and multimodal fusion techniques.

4 Group Project (a paper project, bonus track)

The course project is an opportunity for you to apply the concepts learned in class to a problem aligned with your interests. As this course is part of the **Robotics and Computer Vision Master's Program**, your project should reflect this direction. Projects generally fall into two tracks:

- **Applications**: If you have a background or interest in a specific domain, we encourage you to apply computer vision methods to solve a real-world problem in your field. Identify a practical challenge and address it using **computer vision** and/or **multimodal data processing** techniques.
- **Methods**: You can develop a new model (algorithm) or adapt existing ones to tackle vision or multimodal tasks. This track is more advanced and can potentially lead to publishable work.

This is a **Computer Vision** course, so your project must involve **visual data (pixels)** or **multimodal data** (e.g., combining visual, textual, or sensor data). Projects purely focused on non-visual domains, even if they use convolutional networks, are not suitable.

5 Midterm

A Kaggle competition.

6 Final Exam

A test.

References

[1] M. Artemyev and A. Ashukha. *Handbook on Machine Learning (in Russian)*. Yandex, 2024. URL https://education.yandex.ru/handbook/ml.

[2] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[3] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

[4] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[7] J. Howard and S. Gugger. *Deep Learning for Coders with Fastai and Pytorch: AI Applications Without a PhD*. O'Reilly Media, Incorporated, 2020. ISBN 9781492045526. URL https://books.google.no/books?id=xd6LxgEACAAJ.

[8] Hugging Face, CV course. Computer vision course by hugging face community. https://huggingface.co/learn/computer-vision-course/unit0/welcome/welcome. Accessed: March 4, 2025.

[9] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

[10] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151: 107398, 2021.

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[12] A. Kornaev and K. Yakovlev. Cv-2025, the course repository. https://github.com/Mechanics-Mechatronics-and-Robotics/CV-2025. Accessed: March 4, 2025.

[13] A. Kumar, T. K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8236–8246, 2020.

[14] J. Li, H. Qin, J. Wang, and J. Li. Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and ccd camera. *IEEE Transactions on Industrial Electronics*, 69(3):2708–2717, 2021.

[15] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[16] Lighthing. Pytorch lightning introduction. a colab notebook. https://colab.research.google.com/drive/1Mowb4NzWlRCxzAFjOIJqUmmk_wAT-XP3. Accessed: March 4, 2025.

[17] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim. Multiple object tracking: A literature review. *Artificial intelligence*, 293:103448, 2021.

[18] D. Merkulov. Seminar on optimization at mipt. https://mipt21.fmin.xyz/. Accessed: March 4, 2025.

[19] Y. Ming, X. Meng, C. Fan, and H. Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021.

[20] Papers&Code, CIFAR-10N. Learning with noisy labels. https://paperswithcode.com/task/learning-with-noisy-labels. Accessed: March 4, 2025.

[21] S. J. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[22] S. J. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL http://udlbook.com.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. URL https://arxiv.org/abs/1612.00593.

[24] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás. 3d deep learning on medical images: a review. *Sensors*, 20(18):5097, 2020.

[25] S. Sridhar. Computer vision, csci 1430, fall 2024. https://browncsci1430.github.io/index.html#schedule-content. Accessed: March 4, 2025.

[26] R. Szeliski. Computer vision. algorithms and applications, 2022. URL https://szeliski.org/Book/.

[27] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[29] M. Wang and W. Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021.

[30] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[31] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.