LIST OF PROJECTS

A.Y. 2020/2021

INSTRUCTOR: Prof. LAMBERTO BALLAN
TEACHING ASSISTANT: Dr. PASQUALE COSCIA

Examination methods

The student is expected to develop, in agreement with the instructor(s), a project. In addition, the student shall submit a written report about the project, addressing in a critical fashion all the issues dealt with during its development. During the exam students are asked to present and discuss their project and answer a few questions about the topics addressed in class.

Project's aim

The aim of the project is to apply the concepts you learned in class to a practical problem. You can apply machine learning and deep learning methods to a specific problem in your domain of interest. You can create new models or propose small variations to existing approaches. Some interesting problems are: image classification/segmentation, face recognition, object tracking/detection. You can use publicly available computer vision datasets or collect your own dataset (but this 2nd option is not recommended).

References

This is an incomplete list of popular datasets/benchmarks in computer vision:

- **ImageNet**: large visual database for visual recognition;
- **SUN Database**: scene recognition and object detection benchmarks with annotated scene categories and segmented objects;
- **Places Database**: scene-centric database with 205 scene categories and 2.5 millions of images with a category label;
- **NYU Depth Dataset v2**: video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect;
- **Microsoft COCO**: benchmark for image recognition, segmentation and captioning;
- **YFCC100M**: 100 million media objects, of which approximately 99.2 million are photos and 0.8 million are videos;
- **Labeled Faces in the Wild**: dataset of 13,000 labeled face photographs;
- **MPII Human Pose Database**: dataset for human pose estimation. It consists of around 25k images extracted from online videos;
- **YouTube Faces DB**: database of face videos designed for studying the problem of unconstrained face recognition in videos;
- **UCF101**: action recognition data set of realistic action videos, collected from YouTube, having 101 action categories;
- **HMDB-51**: dataset is a large collection of realistic videos from various sources, including movies and web videos;
- **ActivityNet**: large-scale video benchmark for human activity understanding;
- **Moments in Time**: one million videos for event understanding.

You might also look at publications from top-tier computer vision conferences:

- IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**)
- International Conference on Computer Vision (**ICCV**)
- European Conference on Computer Vision (**ECCV**)
- Neural Information Processing Systems (**NIPS**)
- International Conference on Learning Representations (**ICLR**)
- International Conference on Machine Learning (**ICML**)

The *Computer Vision Foundation* makes publicly available the research papers of top-tier computer vision conferences: https://openaccess.thecvf.com/.

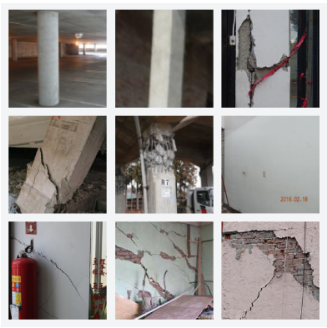
Assessment criteria

The project and the oral examination will be evaluated on the basis of the following criteria:
i) student's knowledge of the concepts, methods, and main technologies in computer vision and cognitive systems;
ii) ability of the student to master the technologies and to evaluate their performance in a proper way;
iii) student's capacity for synthesis, clarity, and abstraction, as demonstrated by the written report and project presentation.

In addition to your own project proposal, in the following you can find a list of topics and references.

***Note**: (i) References are purely indicative. The student(s) can use different datasets and architectures than the provided ones. (ii) Training deep neural networks on huge datasets may be time consuming and require adequate hardware. For this reason, using pre-trained networks or training your architecture on a subset of large datasets may be a preferable choice.*

1. Damage Detection

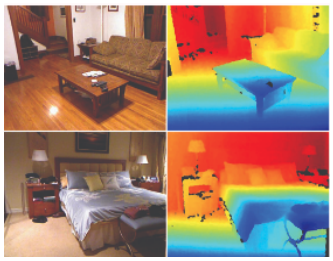


Post-earthquake damage surveys typically require experts to visually inspect buildings to assess their safety. This process can be time-consuming and require multiple inspections. Computer vision methods can be used to provide support to this process.

References:

[1] https://apps.peer.berkeley.edu/phi-net/

2. Depth Maps Estimation from RGB Images



The goal of this project is to estimate a depth map from a single RGB image.

References:

[1] https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

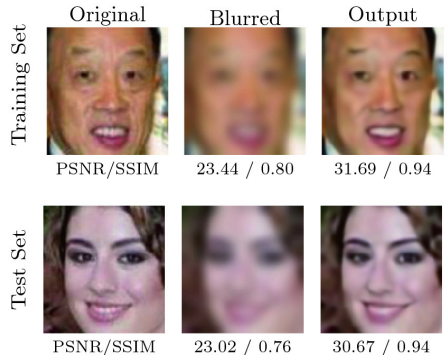
3. Plant Disease Detection

Plant diseases and pests identification using computer vision techniques.

References:

[1] https://github.com/spMohanty/PlantVillage-Dataset

4. Obfuscated Human Faces Reconstruction



Reconstruct human faces from obfuscated images using, for example, the "Labeled Faces in the Wild" dataset. OpenCV could be used to extract faces from images. Metrics that can be used: peak signal to noise ratio (PSNR) and structural similarity (SSIM). Losses that can be used: pixel loss and perceptual loss. Obfuscating techniques: pixelation, Gaussian blurring, etc.

References:

[1] https://arxiv.org/pdf/1908.08239.pdf

5. Image Segmentation

Image segmentation is an important topic in computer vision with a wide range of practical applications (e.g., space optimization, mobility improvement or autonomous driving). With a segmented image, an autonomous agent could be able to recognize buildings, trees, streets or crossroads

Your task is to perform image segmentation.

References:

[1] https://arxiv.org/abs/1411.4038

[2] https://arxiv.org/pdf/1511.00561.pdf

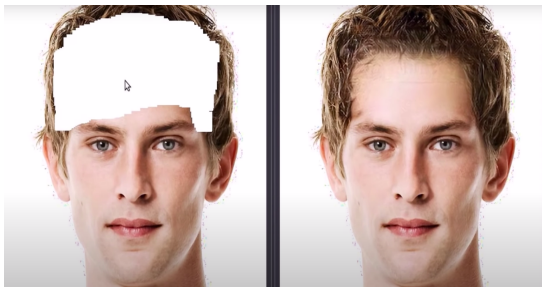
6. Human Pose Estimation

Estimate the location of keypoints of human bodies. As evaluation metrics, Percentage of Correct Parts (PCP) and Percentage of Detected Joints (PDJ) can be considered. A pipeline that jointly considers human pose estimation and action recognition can be implemented.

References:

[1] http://sam.johnson.io/research/lsp.html

7. Image Inpainting

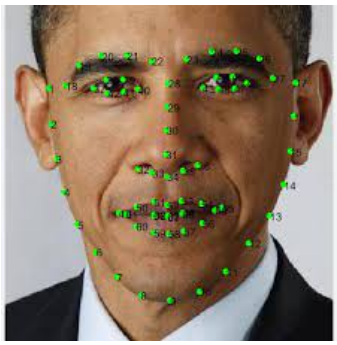


Inpainting refers to the process of filling in portions of images that are damaged, deteriorated, or missing. Non-blind inpainting uses the image and the restoring location, i.e., the inputs are the ground-truth image and the image with the missing/damaged part. Blind-inpainting does not use any locations to restore the original image, i.e., the image with missing/damaged part is provided along with the missing/damaged patch. Losses that can be used: Euclidean or Softmax.

References:

[1] https://arxiv.org/pdf/1601.06759.pdf

8. Facial Keypoints Detection



Facial key points detection is a core technique in several domains (e.g., face recognition, face morphing, …). Your task is to correctly localize facial key points from images.

References:

[1] https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_iccv_2013_300_w.pdf

[2] https://arxiv.org/pdf/1704.04023.pdf

9. Image Geolocalization

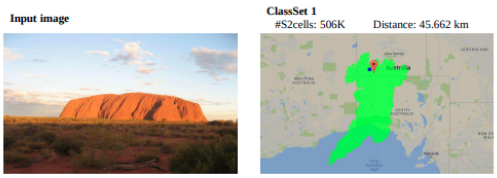


Image geolocalization is a very challenging task. In a nutshell, the task is to predict/assign the right GPS coordinate to a given image.

References:

[1] https://openaccess.thecvf.com/content_ECCV_2018/papers/Paul_Hongsuck_Seo_Enhancing_Image_Geolocalization_ECCV_2018_paper.pdf

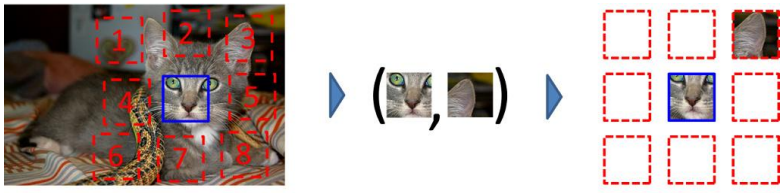
10. Image Colorization

Given a grayscale photograph as input, this paper attacks the problem of hallucinating a plausible color version of the photograph. The system is implemented as a feed-forward pass in a CNN at test time and is trained on over a million color images. We evaluate our algorithm using a "colorization Turing test", asking human participants to choose between a generated and ground truth color image.

References:

[1] https://richzhang.github.io/colorization/
[2] https://arxiv.org/abs/1603.08511
[3] https://arxiv.org/pdf/1908.01311.pdf

11. Unsupervised Visual Representation Learning by Context Prediction



This work explores the use of spatial context as a source of free and plentiful supervisory signal for training a rich visual representation. Given only a large, unlabeled image collection, we extract random pairs of patches from each image and train a CNN to predict the position of the second patch relative to the first. To perform this task correctly the model should to learn to recognize objects and their parts.

References:

[1] http://graphics.cs.cmu.edu/projects/deepContext/

[2] https://arxiv.org/abs/1505.05192

[3] https://arxiv.org/abs/1603.09246

12. Human Counting

Shopping malls, airports and public transportation typically require to count the number of people monitored by RGB cameras for several reasons. For example, it is frequently required to count the percentage of visitors who buyed a specific product, measure the occupancy in buses or trains, or control the crowd.

Your task: Detect the number of people in RGB images or videos. Some of the domains could be sport, security or retail.

References:

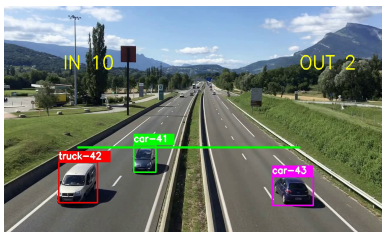
[1] https://www.kaggle.com/fmena14/crowd-counting

[2]https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Zhang_Single-Image_Crowd_Counting_CVPR_2016_paper.pdf

[3] https://arxiv.org/pdf/1802.10062.pdf

[4] https://github.com/gjy3035/Awesome-Crowd-Counting

13. Vehicle Counting



Traffic analysis typically requires counting the number of vehicles that travel from a road. An additional requirement is to classify the types of vehicles. For example, we could need to classify buses and cars, or light or heavy motor vehicles.

Your task: Detect the number of vehicles in RGB images or videos. You can also add a classification step to classify the types of detected vehicles.

References:

[1] https://www.youtube.com/watch?v=sRTqwFXYvs8

[2] https://www.youtube.com/watch?v=O84FlZnP0qs

Additional Material

1. List of computer vision datasets: https://github.com/xiaobai1217/Awesome-Video-Datasets
2. List of robotics datasets (be sure to select a computer vision-oriented dataset/task): https://sunglok.github.io/awesome-robotics-datasets/