PROJECT IDEAS / REFERENCES

LECTURER: Prof. LAMBERTO BALLAN
TEACHING ASSISTANT: FILIPPO ZILIOTTO

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 computer vision and machine (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 are encouraged to use publicly available datasets, or you can collect your own dataset (but this 2nd option is not recommended).

Computer vision datasets

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;
- **Labeled Faces in the Wild**: dataset of 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.

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 (**NeurIPS**)
- International Conference on Learning Representations (**ICLR**)

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

Assessment criteria

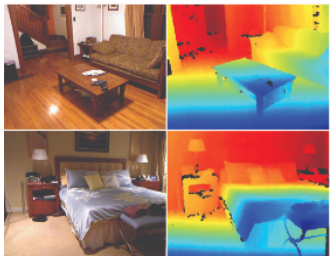
The project and the oral examination will be evaluated on the basis of the following criteria:
i) student's knowledge of the main concepts, methods, and technologies in computer vision and cognitive systems;
ii) ability of the student to master the technologies and to evaluate their performance in a 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, you can find a list of computer vision tasks and corresponding references in the following pages.

***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.*

PROJECT IDEAS

Depth Maps Estimation from RGB Images

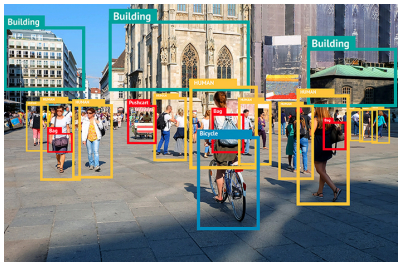


The goal of this project is to estimate a depth map from a single RGB image.

References:

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

Object Detection



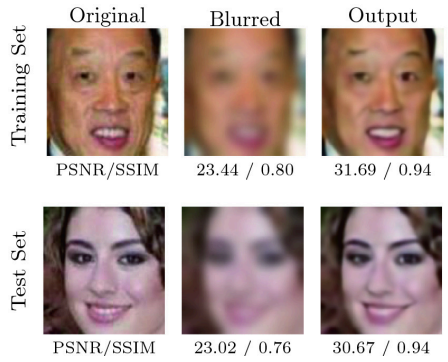
Object detection deals with detecting instances of objects, such as humans, cars or buildings, in images or videos. It is mainly used for video-surveillance, autonomous vehicles and many other computer vision tasks.

References

[1] https://github.com/amdegroot/ssd.pytorch

[2] https://github.com/Capino512/pytorch-rotation-decoupled-detector

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

[2] https://arxiv.org/pdf/1501.00092.pdf

[3] https://arxiv.org/pdf/1707.02921.pdf

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

[2] https://arxiv.org/pdf/1804.06208.pdf

[3] https://github.com/microsoft/human-pose-estimation.pytorch

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 parts. Blind-inpainting does not use any prior information about locations to restore in the original image, which is useful to remove scratches or age-related wear from vintage photos. Losses that can be used: Euclidean or Softmax.

References:

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

[2] https://github.com/WonwoongCho/Generative-Inpainting-pytorch

[3] https://github.com/akmtn/pytorch-siggraph2017-inpainting

[4] https://github.com/naoto0804/pytorch-inpainting-with-partial-conv

[5] https://github.com/JiahuiYu/generative_inpainting

Image Geolocalization

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

References:

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

[2] https://github.com/TIBHannover/GeoEstimation

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 machine-generated image and a ground truth color image.

References:

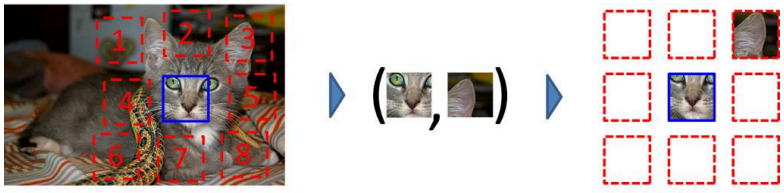
[1] https://richzhang.github.io/colorization/

[2] https://arxiv.org/abs/1603.08511

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

[4] https://arxiv.org/pdf/1908.01311.pdf

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.

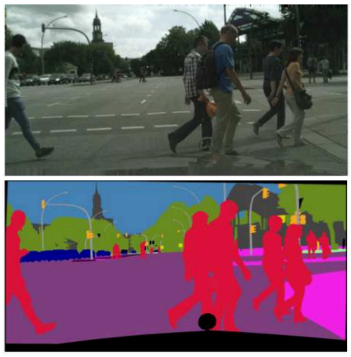
References:

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

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

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

Semantic Segmentation



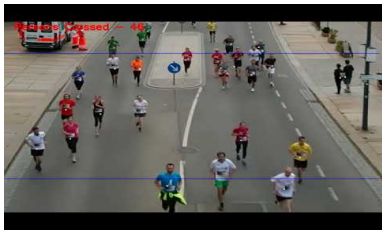
Semantic segmentation assigns a specific label to each pixel of an input image. It is useful for separating foreground from background, for virtual try-on of clothes/eyeglasses and self-driving cars.

References

[1] https://www.youtube.com/watch?v=NyLF8nHIquM

[2] https://openaccess.thecvf.com/content_cvpr_2018/papers/Yang_DenseASPP_for_Semantic_CVPR_2018_paper.pdf

Human Counting



Shopping malls, airports and public transportation typically require counting 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 sports, security or retail.

References:

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

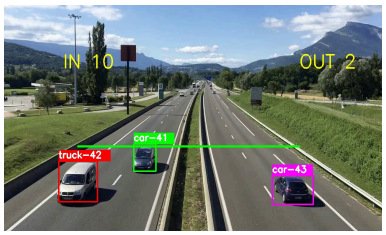
[2]https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Zhang_Single-Image_Crowd_Counting_CVPR_2016_paper.pdf

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

[4] https://github.com/gjy303

[5] https://github.com/gjy3035/Awesome-Crowd-Counting5/Awesome-Crowd-Counting

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:

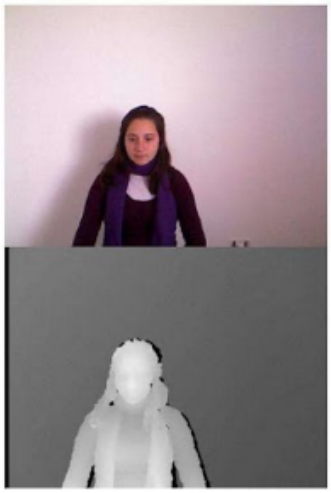
[1] https://www.youtube.com/watch?v=sRTqwFXYvs8

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

[3] https://www.youtube.com/watch?v=nt3D26lrkho

[4] https://www.youtube.com/watch?v=PNCJQkvALVc

Lip-reading from images



The aim of this project is to predict words or phrases from videos. This task can be solved with both machine learning and deep learning techniques (for example, you can use a CNN to extract features from video frames and process extracted features with an RNN). Pre-trained CNNs on human faces could improve classification results. Data augmentation techniques should be applied to increase the number of samples.

ID	Words	ID	Phrases
1	*Begin*	1	*Stop navigation.*
2	*Choose*	2	*Excuse me.*
3	*Connection*	3	*I am sorry.*

References

[1] https://sites.google.com/site/achrafbenhamadou/-datasets/miracl-vc1

Additional datasets

- List of computer vision datasets: https://github.com/xiaobai1217/Awesome-Video-Datasets

- List of robotics datasets (be sure to select a computer vision-oriented dataset/task): https://github.com/mint-lab/awesome-robotics-datasets