Large-scale Computer Vision

Overview of the state of the art research, software and validation data sets in computer vision

E. Ranguelova
Netherlands eScience Center,
Science Park 140 (Matrix 1), 1098 XG Amsterdam, the Netherlands

August 12, 2015

Contents

1 Introduction

The goal of this document is to present a focused partial overview of the state of the art in large scale computer vision (CV). It aims at identifying areas of expertise in CV which are required or expected to be required in e-science projects at the Netherlands eScience Center (NLeSC) as well as defining own research line(s) within the eScience Technology and (applied) research Platform (eStep) at the NLeSC.

1.1 What is Computer Vision?

Computer vision is the science and technology of machines that can see. As a scientific discipline, CV is concerned with the theory for building artificial systems that obtain information from images. It is a type of Artificial intelligence and intercepts a broad range of disciplines such as Optics, Robotics, Image Processing and Pattern Recognition.

1.1.1 Why is CV difficult?

A computer vision researcher always faces the problem to explain why automating vision is a very hard problem, while humans solve it with ease (powered by 30% of the cortex). It is also important to explain the limitations of CV to the potential user of a CV system, especially to scientists from other disciplines. The short answer is: because **seeing is not perceiving**. This illustrated by Figure 1.

Figure 1: Human vision vs Computer vision.

Below some factors for deeper understanding of this difficulty are given ([76]):

1. **Loss of information 3D → 2D**. This is a phenomena which occurs in typical image capture device such as camera or a single eye. The corresponding mathematical model is the one of perspective projection- it maps points along rays, but does not preserve angles and collinearity.

2. **Interpretation** of images. Humans use previous knowledge and experience along with the current observation, while a single image is often the only information the computer has. This is one of the main motivation behind using machine learning in CV.

3. **Noise** is inherently present in each measurement in the real world.

4. **Large data volumes**. This is a main challenge in large scale CV systems.

5. **Brightness measurements** comes by complicated image formation physics. The radiance (brightness, image intensity) depends on irradiance (light source type, intensity and position), observer's position, surface geometry and reflectance properties.

6. **Local window vs. global view.** Usually image processing algorithms analyze a local part of an image (pixel, neighborhood, region), e.g. the computer sees the image trough e keyhole, which makes it hard to understand the global context.

1.2 Computer vision and other scientific disciplines

From projects and proposals at NLeSc, where the data for the object of scientific research are captured as 2D/3D images, some main applications of CV can be identified:

1. **Where is my object? (Localization).** For example, if the object of my study is a freely moving animal, while my camera is fixed somewhere in its habitat, can a CV system find automatically where (potentially) an animal appears on the recorded video, where most frames will probably be of no interest (also, can I keep only the meaningful data?)? Technically, the problem is how to automatically find the object of interest or reduce the data to be processed further, so they contain the object of interest (also efficient storage) in large collections of images/videos.

2. **Is my object the same? (Identification).** For example, if I am studying a specific animal, named King Kong, which shares habitat with other animals of the same species, and all the habitat is monitored at different times, can a CV system find me the images where King Kong appears on, as I'm interested only in his movements? Technically, the problem is to (semi-) automatically determine if the study object is the same in multiple instances of photographing it, usually at different times, in different environment and under changing viewing conditions or camera equipment.

3. **What is my object? (Classification).** For example, if King Kong is a gorilla sharing a habitat with other gorillas and chimps, I might like to separate the data of the chimps from those of the gorillas, and even identify new gorillas/chimps captured on camera, as I'm interested in the whole family, friends and enemies of King Kong. Hence, the problem is to (semi-) automatically classify the study object to one of possible categories. Usually the same camera equipment and modality are used to obtain the images/videos.

4. **Large scale.** The common challenge for these questions is how to efficiently answer them from *large scale* scientific images/videos collections.

At NLeSC there have been projects which illustrate these types of questions to be addressed. In the systems biology project, "Using big data solutions to understand worm behavior", the object of research is the *C.elegans* worm (see Figure 2). The source data are high resolution and long videos capturing the behavior of the worm. The first step is to *localize* precisely the worm in the large volume of imaging data.

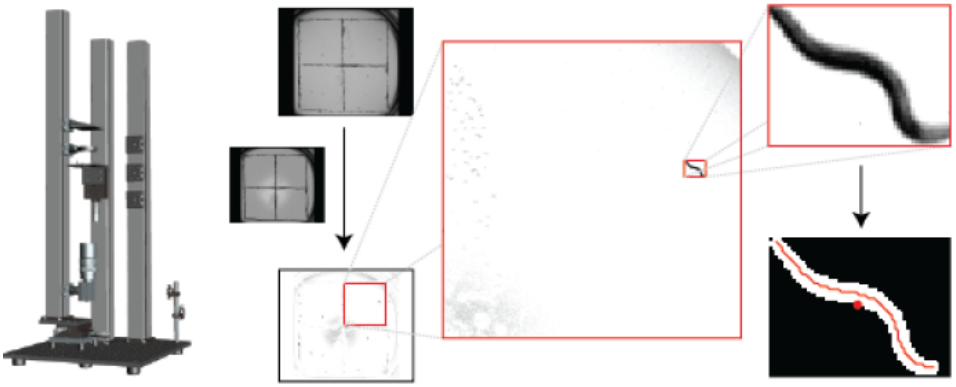


Figure 2: Behavioral analysis of Caenorhabditis elegans (roundworm) using video recordings.

This is a good example of how the object of interest is often studied in a controlled environment and one can assume that the most prominent (*salient*) object captured on images/videos is the object of interest. In this case, the lab plate ensures relatively uniform background where one can easily find the object of interest in foreground, the worm. The challenge is to find it automatically and to process efficiently large amounts of data.

The localization problem is often related to the *segmentation* problem, which is most challenging in the medical imaging domain (BiomarkerBoosting project and EYR4 project Lightpath for OCT imaging at NLeSc). Illustration of the difficulty can be seen in Figures 3 and 4. There is little perceptual difference between the hippocampus and the surrounding brain tissue as well as between some of the retina layers, making it difficult to distinguish them even by a human eye.

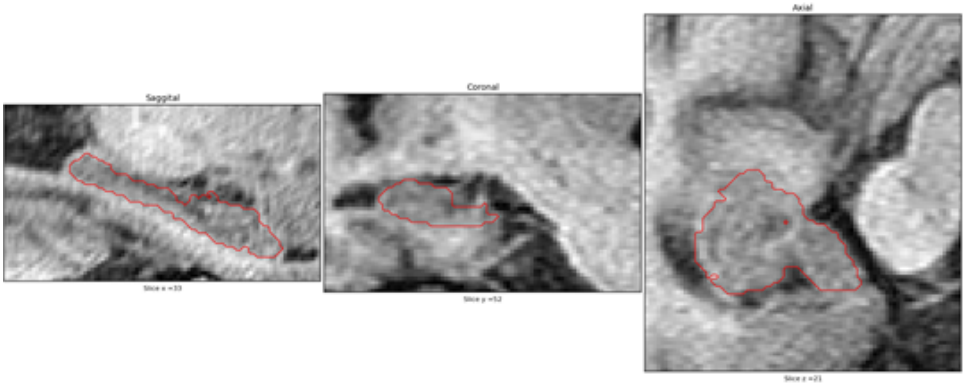


Figure 3: Hippocampus segmentation from a human brain MRI subvolume.

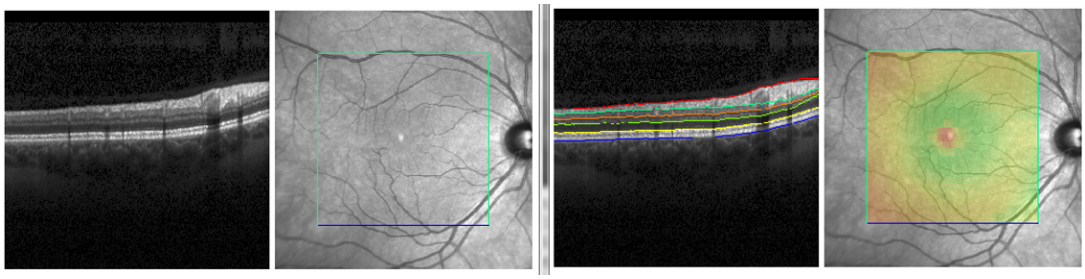


Figure 4: Optical Coherence Tomography (OCT) of retina layers segmentation: image data (left), image segmentation & retina thickness map (right).

Because of the complexity of the problem, the large number of imaging modalities, equipment and even image formats, usually specific algorithms are developed to segment different body organs and tissues. Since specific solutions are required, which are hard to generalize for other structures, modalities, even less across scientific domains, the image segmentation is left out of the scope of this document.

An example of the *identification* problem, is the individual photo-identification of wildlife. This is illustrated on Figure 5. Many species are individually recognizable by characteristic patterns or shape, i.e. by their phenotype.

Figure 5: Example species with sufficient spot pattering what could be useful for automated photo-identification: (a) whale shark (with reference area), (b) spotted tree frog, (c) northern quill, (d) Amazon spotted frog, (e) striped blue crow and (f) mangrove snake.

An example of the *classification* problem is shown on Figures 6 and 7. The question is to identify the tree species from microscopic images of wood samples. The *Acer* has a different appearance than *Toona*, but also within the family there are many species of *Toona* which differ- for example *Toona ciliata* differs subtly from *Toona sinesis* (Figure 7), hence the classification problem could be hierarchical and increasingly difficult.

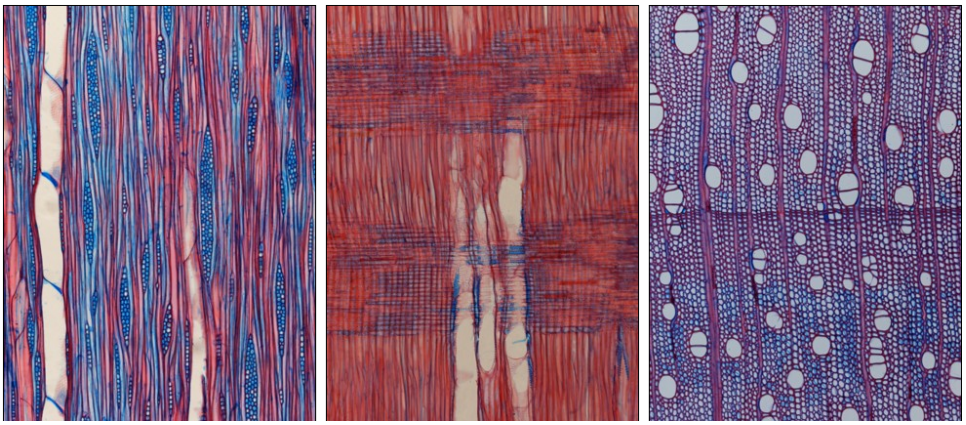


Figure 6: From left to right: tangential, radial and transversal tissue sections of stained mapple wood (genus Acer).

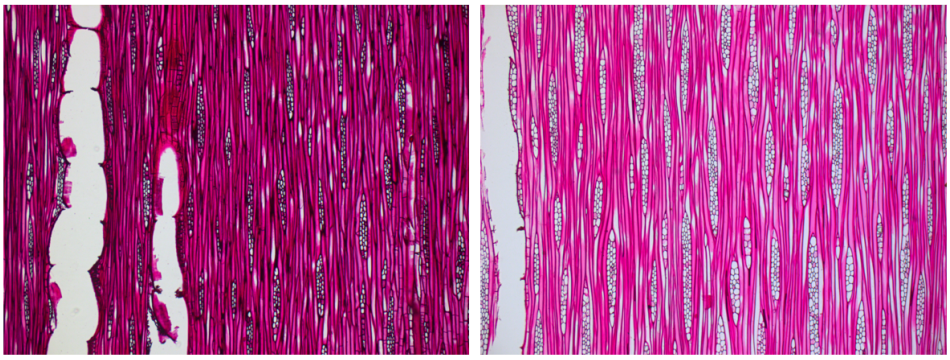


Figure 7: Tangential section of stained wood. Left: Toona ciliata, Right: Toona sinesis.

1.3 Computer vision topics

To address the related scientific challenges, four CV research questions can be defined:

1. **Visual salience:** How can the CV system determine automatically the most visual salient region(s) in an image?

2. **Object/scene identification:** How can the CV system automatically determine whether two images, potentially taken with different cameras under different viewing conditions and transformations, represent the same object/scene?

3. **Object/scene detection/classification:** How can the computer recognize automatically to what visual category the object/scene captured in an image belongs to?

4. **Large scale processing:** How can we perform saliency detection, identification and classification of large datasets.

1.3.1 Other applications

There are also numerous non-scientific applications related to the above research questions. For example *visual saliency* (1) in important in tasks such as:

- Automatic target detection (see Fig. 8)
- Robot/car navigation using salient objects
- Image and video compression
- Automatic cropping/centering images for display on small portable screens, etc.

Figure 8: Example of a saliency model detecting the vehicle as being the most salient object in a complex scene.

Some of the many applications related to *object/scene identification* (2) are:

- Stereo and wide-baseline matching
- Image panorama stitching/creation
- Automatic reconstruction of 3D scenes, etc.

Automatically *understating an object/scene* (3) b the means of image classification, have many applications like:

- Image search engines
- Organizing photo collections
- Autonomous driving
- Human machine interaction
- Digital forensic investigation, etc.

This is a complex and high-level computer vision task, with the goal of making machines see like humans and be able to infer both general principles as well as current situations from images. Example of a trained system for scene categorization is shown on Fig.9.

Predictions:

- **Type of environment:** outdoor
- **Semantic categories:** tower:0.50, bridge:0.25, viaduct:0.12,
- **SUN scene attributes:** man-made, clouds, openarea, naturallight, mostlyverticalcomponents, metal, vacationingtouring, nohorizon, directsunsunny, congregating

Figure 9: MIT Scene Recognition Demo.

In fact, the majority of CV researchers traditionally work on non-scientific applications, focusing on the third and most challenging problem. Addressing the problems faced by other domain scientists, while still conducting generic and widely applicable computer vision research, seems to fit best the strategy of NLeSC and eStep. Another important aspect, where NLeSC can contribute a lot with expertise, is the development of *large scale* CV systems, i.e. the last research question.

1.4 Structure

The overview is by no means complete, it rather tries to summarize the research in the field along the above topics in the last years. The document is structured along the main outputs of the CV research, namely, scientific research (some of the topics are explained) and publications in Section 2, software in Section 3 and datasets in Section 4. Some potential scientific applications are shown in Section 5. The conclusions and recommendations are given is Section 6.

The CV researchers can benefit most from the Research for overview of current research trends. The interested reader can also learn some of the terminology and methods in the field from that section. For development and testing the most useful sections are Software and Datasets. The engineers can start by looking at section Software. Readers interested in the application and directions of NLeSC are referred to sections Applications and Conclusions.

2 Research

This section explains some of the current CV research and gives pointers to recent publications about the three CV tasks: *visual saliency* (see section 2.1 Saliency), *object/scene identification*(section 2.2 Salient regions) and *image classification* (section 2.3 Convolutional Neural Networks) as well as some work on frameworks for *large-scale* CV systems (section 2.4 Large Scale CV Systems).

2.1 Saliency

The research question is how can the CV system determine automatically the most visually *salient* region(s) in an image?

Saliency refers to distinctiveness (standing out), catching the focus of attention, characteristic. Usually a salient object of interest is sought to be separated from the background. Automatic measures of saliency are based on measurable properties of the object, like texture, color, location or by simulating human visual attention.

A bottom-up visual saliency model, the *Graph-Based Visual Saliency (GBVS)* is proposed in [42]. The model is simple and biologically inspired, based on activation maps of certain feature channels followed by normalization and combination with other maps. The GBVS outperforms by far the classical algorithms of Itti & Koch [43]. For the software, see section 3.1.1 Saliency Map Algorithm.

The salient object detection is formulated as image segmentation problem in [52]. The object is separated from the background on the basis of several features including multi-scale contrast, center-surround histogram and color spatial distribution for the object description on several levels- locally, regionally and globally. The multi-scale contrast is the local feature, the center-surround histogram is the regional feature and the color spatial histogram- the global. These features are illustrated on Figure 10.

Figure 10: Examples of salient features. From left to right: input image, multi-scale contrast, center-surround histogram, color spatial distribution and binary salient mask by CRF.

A Conditional Random Field (CRF) is trained on these features. For the purposes of this research the authors have compiled a large-scale database, MSRA ([82]), presented in section 4.1.1 MSRA. The database is publicly available, while the software is not. The proposed methods, compared to two other algorithms "FG" (fuzzy growing) and "SM" (salient model as

computed by the SalientToolbox, described in section 3.1.2 SaliencyToolbox), tend to produce smaller and more focused bounding boxes.

Butko et al. propose fast approximation to a Bayesian model of visual saliency in [20]. The algorithm have been designed for efficiency in social robotics situations with the goal to orient a cameras as quickly as possible towards human faces. For the available software, see section 3.1.4 FastSalience.

In [1] the authors perform a frequency-domain analysis on five state-of-the-art saliency methods, and compared the spatial frequency content retained from the original image, which is then used in the computation of the saliency maps. This analysis illustrated that the deficiencies of these techniques arise from the use of an inappropriate range of spatial frequencies. Based on this analysis, they presented a frequency-tuned approach of computing saliency in images using low level features of color and luminance. The resulting saliency maps are better suited to salient object segmentation, with higher precision and better recall than the analyzed state-of-the-art techniques. For the available software, see section 3.1.3 Frequency-tuned Saliency.

In [96] the authors address a fundamental problem in saliency detection, namely, the small-scale background structures, which affect the detection. This problem occurs often in natural images. They propose a hierarchical framework that infers importance values from image layers with different scales. The approach is summarized in Figure 11.

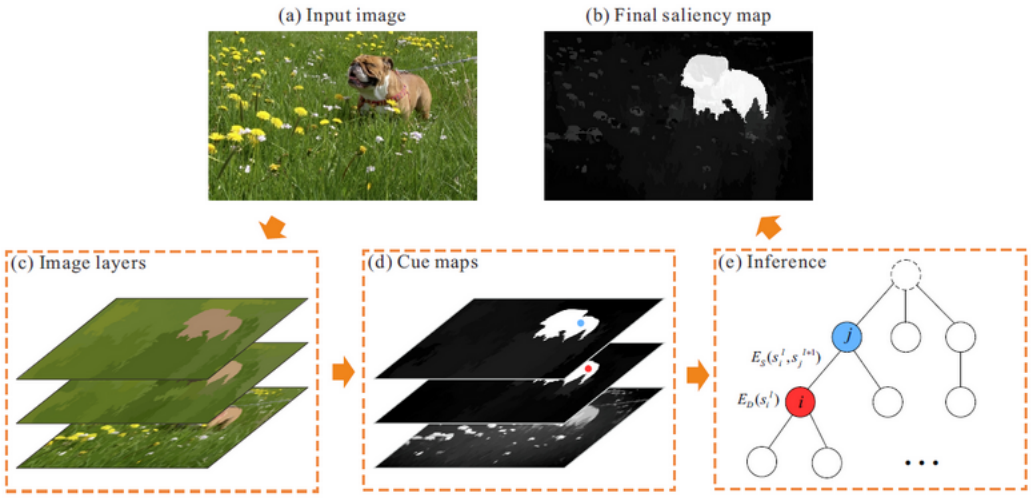


Figure 11: An overview of the hierarchical framework. Three image layers are extracted from the input, and then saliency cues from each of these layers are computed. They are finally fed into a hierarchical model to get the final results.

For the purpose of their research the authors made a new database available to the community, the Complex Scene Saliency Dataset (CSSD) and the Extended CSSD (ECSSD), described in Section 4.1.3 CSSD and ECSSD. The executable of their software is also available from the project link ([95]), but not the source code. The authors report better performance of their method compared to 11 other state-of-the-art methods.

Recently, a method for statistical textural distinctiveness for detecting saliency in natural images have been proposed [72]. Textural representations are extracted and a sparse models learned. Next, a weighted graphical model is constructed and used to distinguish between all texture atom pairs.

2.2 Salient regions

For the *object/scene identification* task, the main question is whether two images taken at different times or under different conditions depict the same object/scene. One approach is to compare sets of local characteristic features, extracted reliably and independently from the two images.

Detecting automatically *salient*, e.g. interesting, distinct, characteristic and repeatably findable regions from an image is a major research topic in the area of image *feature extraction*. The salient regions are one type of features, which can offer useful representation of images, along with interest points, edges, ridges etc. Usually the first step is automatically extracting salient regions using an salient/interest region *detector* followed by describing each region/patch using a region *descriptor*. This is illustrated in Figure 12.

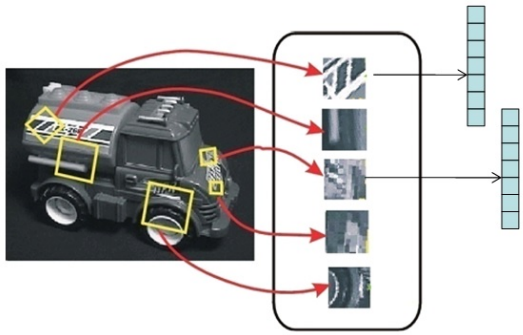


Figure 12: Extracting descriptors from detected keypoints/regions.

As a final step, two sets of region descriptors are compared and *matched* in order to establish correspondences between the images.

One very important and desired property of such region detection is *affine covariance* (often referred to as *affine invariance*). The affine covariance refers to the requirement that these regions should correspond to the same pre-image for different viewpoints and geometric transformations. This concept is shown on Figure 13. The figure illustrates that an affine salient regions detector has automatically and independently detected the same pre-image regions on both images. Some detectors provide arbitrarily shaped regions, others detect elliptical regions. For the purpose of comparison, the (equivalent) elliptical regions are usually used and displayed.

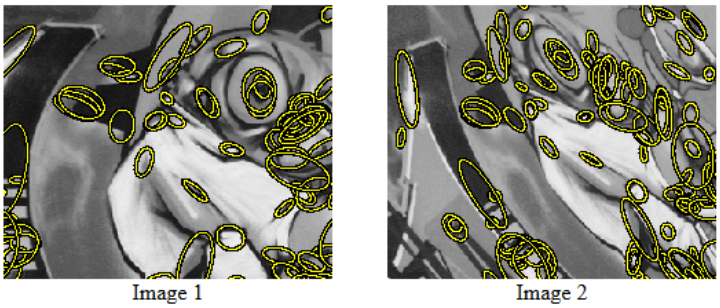


Figure 13: Example of affine regions detection. Image2 is affinely transformed version of Image1.

2.2.1 Detectors

A decade ago, a seminal paper by the Visual Geometry Group in Oxford, compared the existing affine-covariant region detectors in [57]. A clear conclusion of this comparative study is that the *Maximally Stable Extremal Regions (MSER)* detector ([56]) is the winner in many of the test scenarios and since then, the MSER detector has become a de-facto standard in the field (for example is now an integral part of the MATLAB Computer Vision System Toolbox). For common implementations of MSER, the reader is referred to section 3.2.1 Detectors.

After this comparative study, several researchers have proposed improvements to the MSER detector, though none of them increased the performance drastically. An MSER color extension is proposed in [41]. The author calls his detector *Maximally Stable Color Region (MSCR)*. In comparison to the original MSER detector, the simple color MSER extension (MSER3) and a color blob detector, the MSCR performs the best in most cases on the well-known Visual Geometry Group in Oxford test image sets with known homographies ([83]).

The MSER detector has been also extended in 3D to *Maximally Stable Volumes (MSVs)* in [26]. The MSVs have been used to successfully segment 3D medical images and paper fiber networks.

A *Structure-Guided Salient Region detector (SGSR)* is introduced in [38]. It is based on entropy-based saliency theory and shows competitive performance.

Another enhancement of MSER with the Canny edge detector is introduced in [87]. The dilation operator is used on the detected edges to remove the ambiguous ones, which makes the interest regions more representative. The improved MSER shows better performance than the original MSER in image classification in the bag of words framework, though original comparison of repeatably ([57]) is not presented.

In the context of humpback whale identification, Ranguelova et al. have proposed *Morphology based Stable Salient Regions (MSSR)* detector [66, 67] . MSSR is not better than MSER on repeatably, but produces less number of regions (which is important in the matching step) and better represents human salient perception. Within the eStep, NLeSC's technology platform, some research is ongoing to improve MSSR.

Links to the implementations of the detectors are given in section 3.2.1 Detectors.

2.2.2 Descriptors

Another important performance evaluation paper from the Oxford Vision group has compared the salient region descriptors [58]. The conclusion of the performance evaluation was that the *Gradient Location-Orientation Histogram (GLOH)* detector was performing best (recall and precision) followed closely by the *Scale Invariant Feature Transform (SIFT)* descriptor [54]. The SIFT descriptor has been the most popular and widely used descriptor in CV for over a decade.

The idea behind SIFT Transform is finding the extrema from all possible scales and image locations (*scale space*). From these potential interest points only the most stable points are selected. One or more orientations are assigned to a point based on the local gradient directions, while invariance to transformations is achieved by computing these orientations from all possible transformed versions of the data. This histogram of (usually 128) orientations are the keypoint descriptor (Figure 14). The SIFT descriptor is example of descriptors based on *Histogram of Gradients (HoG)*. For applying SIFT descriptor on already detected salient regions, only the descriptor steps are performed.

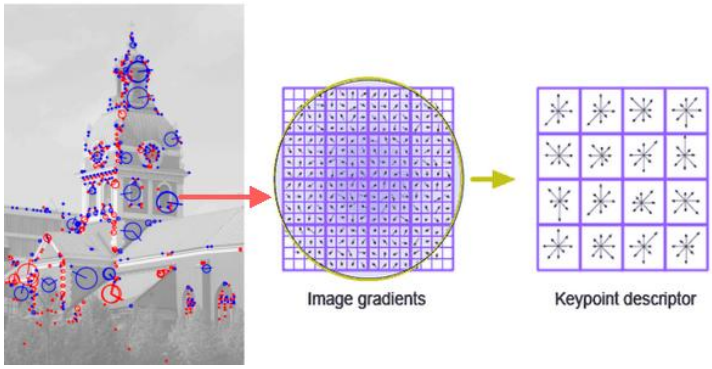


Figure 14: SIFT descriptor steps. From left to right: keypoint detection; image gradients; descriptor.

An algorithm for GPU-based video tracking and matching has been published at [74]. There are many variants of the SIFT descriptor, all based on floating point arithmetic, for example, SURF[15] and also GLOH [58], designed to improve distinctiveness.

Other alternatives to SIFT are *FAST* [68], which compares pixels on a ring centered at a feature point. *ORB* [69] extends FAST by computing orientations based on the intensity centroid moment. Rotation invariance is also achieved with the MRRID and MROGH descriptors, [37] by pooling local features based on their intensity orders. In the same group is the *Local Intensity Order Pattern (LIOP)* descriptor [88].

Another important class of descriptors which offer computationally more tractable solution to region/patch description, are the *binary descriptors*. Unlike the HoG type of descriptors and their expensive computation of gradients, a set of simple binary tests are used with the Hamming distance. Examples of such descriptors are *Binary Robust Independent Elementary Features (BRIEF)*[21], *Binary Robust Invariant Keypoints (BRISK)* [49], etc. A binary descriptor is composed of three parts:

1. **Sampling pattern**. Where to sample points in the region around the interest point.
2. **Orientation compensation**. Measure the orientation of the keypoint and rotate it to compensate for rotation.
3. **Sampling pairs**. Which pairs to compare when building the final descriptor.

For all possible sampling pairs a binary test is performed. For each pair (p_1, p_2), if the intensity at point p_1 is greater than the intensity at point p_2, the descriptor value for that pair is 1, otherwise 0 (Figure 15). A very good tutorial on binary descriptors can be found online.

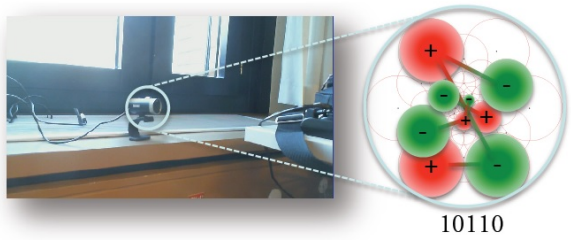


Figure 15: Binary descriptor.

Another more recent performance paper compares their performance also to the established descriptors like SIFT or SURF [59]. The main conclusions of the evaluation are:

1. The real valued descriptors such as LIOP, MRRID and MROGH outperform SUFR and SIFT both in precision and recall, although with low efficiency.

2. The binary descriptors are very efficient for time-constrained applications with good matching accuracy.

3. The binary descriptors are very fast with low memory requirements (32 bytes for BRIEF and ORB or 64 bytes for BRISK). These provide comparable precision/recall to SIFT/SURF.

A very recent descriptor, *Binary Online Learned Descriptor (BOLD)*, [13] has been proposed. It combines the advantages of an efficient binary descriptor with the improved performance of learning-based descriptors. The binary tests are *learned* from the content of the patches themselves minimizing the inter and intra-class distances leading to a more robust descriptor. BOLD has a good performance both in terms of accuracy as well as efficiency.

Links to the implementations of the detectors are given in section 3.2.2 Descriptors.

2.2.3 Matching

The extracted descriptors from the detected regions (as blobs or around keypoints) are usually compared via nearest neighborhood (NN) search. Much research is done for developing algorithms for faster matching (e.g. the fast Viola-Jones object detecor [81]). The most popular approaches are using kd-trees and hashing methods (see [59] and the references within). For binary descriptors matching is performed efficiently using Hamming distance.

2.3 Convolutional Neural Networks

The current trend of research to solve the most challenging task in CV, object/scene classification, is deep learning. *Deep Machine Learning* has been declared the new frontier in artificial intelligence research in 2010, [6]. It has been inspired by the findings of neuroscience that the neocortex, which is associated with many cognitive abilities, does not explicitly pre-process sensory signals, but rather allows them to propagate through a complex hierarchy that, over time, learn to represent observations based on their regularities [48, 47]. This motivated the emergence of the deep machine learning, which focuses on computational models that exhibit similar characteristics. The main goal of deep learning is to train multi-layered (deep) hierarchical network on large set of observations to extract signals from this network fed to a relatively simple classification engine for the purpose of robust (invariant to a diverse range of transformations and distortions) pattern recognition.

The *Convolutional Neural Networks (CNN)* are a type of deep learning networks [17]. CNNs are a family of multi-layer neural networks particularly designed for use on two-dimensional data, such as images and videos. In CNNs, small portions of the image (a local receptive field) are treated as inputs to the lowest layer of the hierarchical structure. Information generally propagates through the different layers of the network whereby at each layer digital filtering is applied in order to obtain salient features of the data observed. The method provides a level of invariance to shift, scale and rotation as the local receptive field allows the neuron (processing unit) access to low-level features such as oriented edges or corners. The filtering is a *convolution* with filter, followed by sub-sampling to further decrease the dimensionality and provide invariance to shifts (Figure 16).

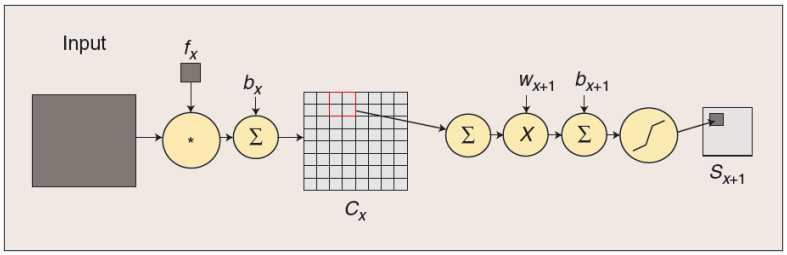


Figure 16: The convolution and sub-sampling. Convolving an input (image for the first stage or feature map for later stages) with a trainable filter f_x and adding a trainable bias b_x to produce the convolution layer C_x. The sub-sampling is summing a neighborhood (4 pixels), weighting by scalar w_{x+1}, adding trainable bias b_{x+1}, and passing through a sigmoid function to produce a twice smaller feature map S_{x+1}.

The convolution an sub-sampling can be performed arbitrary number of times. A conceptual example of CNNs is shown on Figure 17. CNNs create their invariance to object translations by a *"feature pooling "* (the S layers in Figure 17).

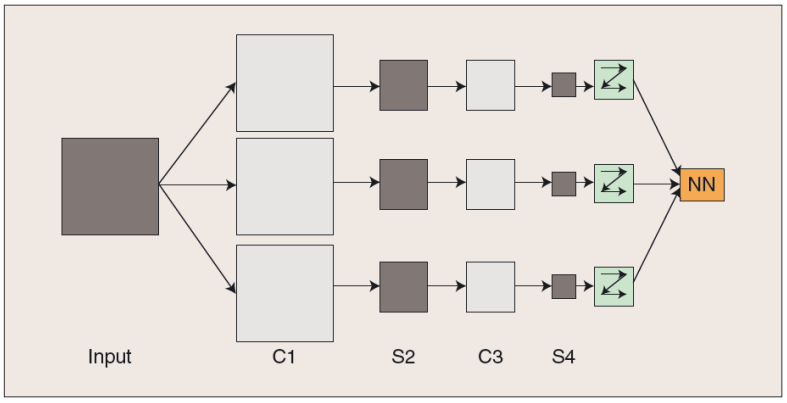


Figure 17: Conceptual example of CNN. The input image is convolved with 3 trainable filters and biases as in Figure 16 to produce 3 feature maps at the C_1 level. Each group of 4 pixels in the feature maps are added, weighted, combined with a bias, and passed through a sigmoid function to produce the 3 feature maps at S_2. These are again filtered to produce the C_3 level. The hierarchy then produces S_4 in a manner analogous to S_2. Finally these pixel values are rasterized and presented as a single vector input to the conventional neural network at the output.

The close relationship between the layers and spatial information in CNNs makes them well suited for image processing and understanding, and they generally perform well at autonomously extracting salient features from images. Since then, especially after the CNNs performed better than the hand-crafted features in the ImageNet challenge [70], a lots of research has been done in CNNs in CV. Here only few key papers are mentioned. For a starting point to the large amount of publications on the topic the reader is referred to the reading list of the DeepLearning.net. Also, all papers from the last 3 years of the largest CV conference Computer Vision and Pattern Recognition (CVPR) are available online.

Despite the CNN success, to a large extend they are still 'black boxes' and researchers try to get deeper insight of the internal working of the network, namely to understand the representations that are learned by the inner layers of these deep architectures. As scenes are composed of objects, the CNN for scene classification automatically discovers meaningful objects detectors, representative of the learned categories. With object detectors emerging as a result of learning to recognize scenes,[99] demonstrates that the same network can perform both scene recognition and object localization in a single forward-pass (Figure 18), without explicitly learning the notion of objects.

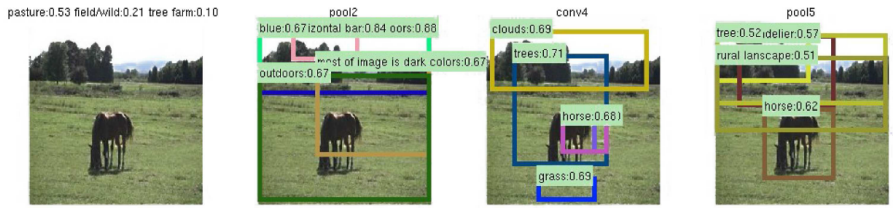


Figure 18: Interpretation of a picture by different layers of the Places-CNN (4.4.4 Places) using semantic tags given by humans. The first shows the final layer output of Places-CNN. The other three show detection results along with the confidence based on the units activation and the semantic tags.

Another approach for trying to understand the CNNs is by "fooling" them! Researchers have shown that it is easy to produce images that are completely unrecognizable to humans, but that CNNs believe to be recognizable objects with 99.99% confidence (e.g. applying the label lion with certainty to an image that humans perceive as white noise static). These are called fooling images (Figure 19), [62]. These results shed light on interesting differences between human vision and current CNNs, and raise questions about the generality of CNN computer vision.

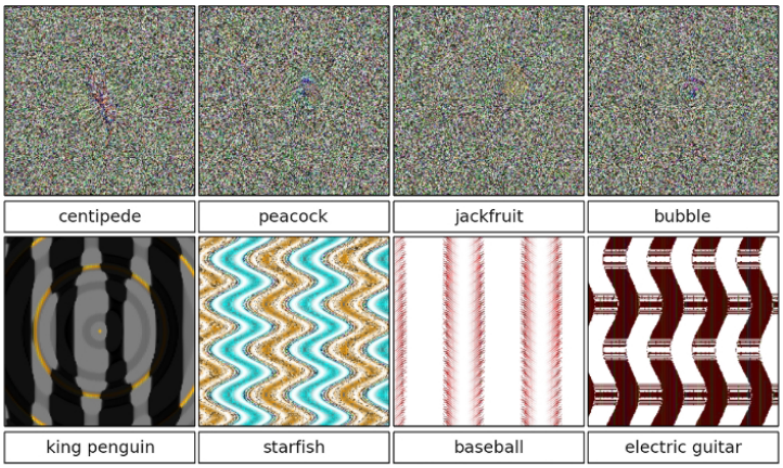


Figure 19: Evolved images that are unrecognizable to humans, but that CNN trained on ImageNet believe with 99.6% certainty to be a familiar object. This result highlights differences between how CNNs and humans recognize objects. Images are either directly (top) or indirectly (bottom) encoded.

Another fundamental issue with the CNN is the need for many *labeled* observations for training. This could be a serious obstacle for the application of CNNs. Interesting trend are the *never ending learning systems*. For example, *NEIL (Never Ending Image Learner)* is a computer program that runs 24 hours per day and 7 days per week to automatically extract visual knowledge from Internet data. For more details see [22].

Recently, an interesting article which compares the matching of salient regions with CNNs with matching with SIFT descriptor have been published [39]. The authors have also recorded a new larger dataset, compared to the only 48 images in the comparative study paper, covering larger class of image transformations with more images (see 4.3 Salient Regions Datasets). The goal of the paper is to study the regions/patches descriptors, not to evaluate detectors. For the detection step, the standard MSER detector have been used. The paper concludes that the CNN trained features are consistently better than the SIFT descriptor, for the price of a higher computational cost.

Software packages supporting CNNs are given in Section 3.3 Deep Learning.

2.4 Large Scale CV Systems

One of the manifestations of the Big Data era is the proliferation of massive visual data. The key challenges in extracting a value from these data are:

- **Scalability.** This is the key challenge in the world of Big Data. Lack of infrastructural support leaves researcher repeatedly facing the same difficulties when developing and porting distributed CV algorithms.
- **Provably Correct Parallel/Distributed Implementations.** Designing and implementing efficient and correct parallel computer vision algorithms is extremely challenging. Some tasks like extracting statistics from image collections are embarrassingly parallel, i.e. can be parallelized simply by distributing the images to different machines. Unfortunately, most tasks in CV and machine learning such as training a face detector are not embarrassingly parallel (there are data and computational dependencies between images and various steps in the algorithm).
- **Reusability.** Computer vision researchers have developed vision algorithms that solve specific tasks but software developers building end-to-end system find it extremely difficult to integrate these algorithms into the system due to different software stacks, dependencies and different data format.

Researchers are looking more and more at these challenges.

2.4.1 MapReduce

MapReduce is a programming model and its implementation for processing large datasets, introduced by Google, [25]. The user specifies the computations in terms of *map* and *reduce* functions, and the system automatically parallelizes the computation across large clusters of machines at run-time. Generally CV algorithms are applied on one or more images, consisting of pixels with some set of potentially varying parameters. Parallelism can be exploited on any of these levels, across parameters the task is embarrassingly parallel (i.e.independant) and on image or pixel level,the parallelization depends on whether the algorithm operates independently (e.g. SIFT, face detection).

For example [90] addresses a web-scale multimedia mining task using MapReduce. The paper describes the low level implementation of several computer vision algorithms relevant for mining applications: classifier training, sliding windows, clustering, bag-of-features, background subtraction and image registration. The paper gives guidelines and algorithms of how the mappers and reducers can be implemented for these CV tasks. For example, for classifier training, the mapper performs the feature computation in parallel, the features are collected for the reducer where the classifier training is performed.

A further development in that direction is the *Hadoop Image Processing Interface (HIPI)* library [8]. The undergraduate thesis of C. Sweeney is the publication about HIPI [7]. The goal of HIPI is to provide a simple and clean interface for high-throughput distributed image processing on the MapReduce platform. Figure 20 illustrates a typical MapReduce/HIPI workflow.

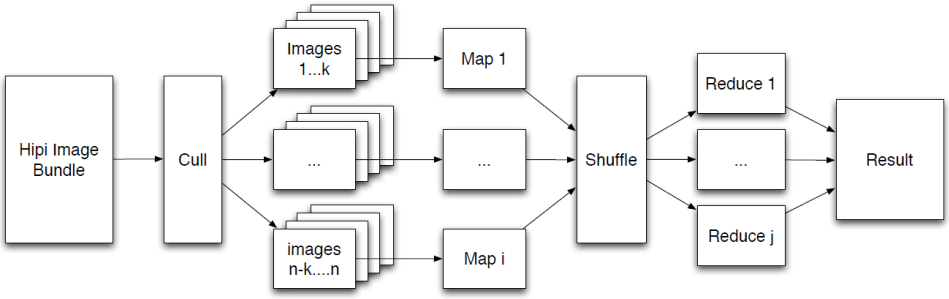


Figure 20: A typical MapReduce pipeline using HIPI with n images, i map nodes and j reduce nodes.

HIPI is designed to abstract Hadoop's functionality into an image-centric system and to serve as a tool for efficient use of Hadoops MapReduce for image processing and computer vision. For the MapReduce/HIPI software, see Section 3.4.1 MapReduce.

2.4.2 Cognitive Computer Vision Systems

Apart from traditional research of CV algorithms, in the last decades much research has been done into integrating such algorithms into larger systems. For example, *cognitive computer vision (CVS) systems* does not only involve CV, but also *machine learning* and reasoning to extend priory knowledge or verify the consistency of results. An overview paper studies the requirements and developments for large scale CVS, [92]. The existing systems have been evaluated along *functional, non-functional and other requirements* as illustrated on Figure 21.

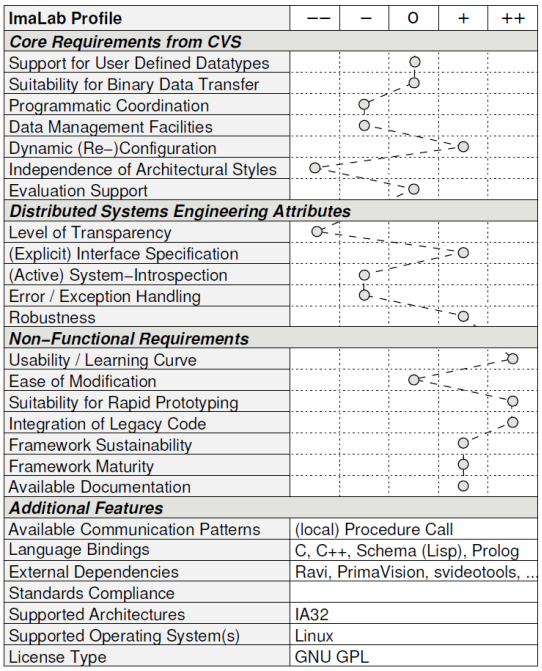
The study groups the frameworks in 4 major classes based on the evaluation.

1. *Visual Image Processing Environments*, e.g.ImaLab (see Figure 21) provides facilities to quickly develop pipelines and allows rapid prototyping, easy GUI. They lack support for distributed architectures.

2. *Middleware approaches*, e.g. COBRA implementation TAO. Very generic and powerful, but complicated use. Also do not support specific CV domain requirements.

3. *Robotics domain*, e.g.SmartSoft or OROCOS. Satisfy many CV requirements, including distribution and transparency. However, their integration is largely coupled with the robot control components.

4. *CVS Frameworks*, **XCF** [91] focuses on data management facilities, distribution and rapid prototyping, **zwork** [65] mainly deals with the programmatic coordination and dynamic reconfiguration in the control aspect of CVS.



ImaLab Profile	−−	−	o	+	++
Core Requirements from CVS					
Support for User Defined Datatypes					
Suitability for Binary Data Transfer					
Programmatic Coordination					
Data Management Facilities					
Dynamic (Re–)Configuration					
Independence of Architectural Styles					
Evaluation Support					
Distributed Systems Engineering Attributes					
Level of Transparency					
(Explicit) Interface Specification					
(Active) System–Introspection					
Error / Exception Handling					
Robustness					
Non–Functional Requirements					
Usability / Learning Curve					
Ease of Modification					
Suitability for Rapid Prototyping					
Integration of Legacy Code					
Framework Sustainability					
Framework Maturity					
Available Documentation					
Additional Features					
Available Communication Patterns	(local) Procedure Call				
Language Bindings	C, C++, Schema (Lisp), Prolog				
External Dependencies	Ravi, PrimaVision, svideotools, ...				
Standards Compliance					
Supported Architectures	IA32				
Supported Operating System(s)	Linux				
License Type	GNU GPL				

Figure 21: Graphical evaluation scheme for CVS (ImaLab). Each assessment category has 5 step scale: '- -'- impossible, '-'- difficult, 'o'- neutral, '+'- supported and '++'- automatic.

Therefore, the system engineers will have to carefully identify their needs and project requirements and correspondingly decide for a most suitable framework.

2.4.3 CloudCV

A very recent development, which tackles the large-scale CV challenges, is the CloudCV, [2]. The goal is to democratize computer vision, i.e. make the state-of-the-art distributed CV algorithms also available to non-experts (i.e. people who are not CV researchers or Big Data experts). Therefore, the platform aims at three different audiences: computer vision researcher, scientists which are not CV experts and non-scientists.

CloudCV consists of a group of virtual machines running on Amazon Web Services capable of running large number of tasks in a distributed and parallel setting. Popular datasets are already cached on these servers to facilitate researchers trying to run popular computer vision algorithms on these datasets. For custom datasets the users can access these services through a web interface for uploading smaller sets, or use Python and Matlab APIs for larger datasets. An overview of the system is shown on Figure 22.

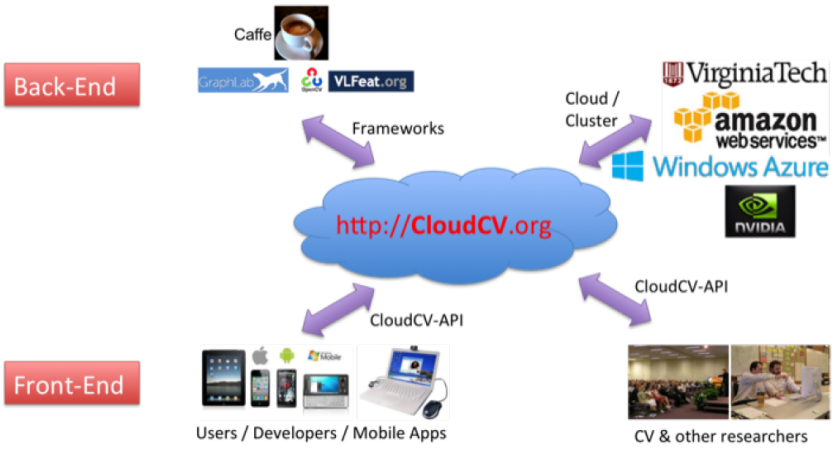


Figure 22: Overview of CloudCV.

The paper presents the back-end infrastructure (web servers, distributed processing and job schedulers), the used deep learning framework and the front-end platforms (web interface, Python and MATLAB APIs). It also presents in detail the CloudCV functionalities, namely classification, feature extraction, finding VIP in group images, gigapixel image stitching. A very concrete future plan is to integrate the *Deep Learning GPU Training System (DIGITS)*, [63] with CloudCV.

3 Software

The saliency detection (see section 3.1 Saliency), salient regions detection and matching (section 3.2 Salient regions), (pre-trained) CNN architectures (section 3.3 Deep Learning)and large scale CV frameworks (section 3.4 Distributed software for CV), developed by the researchers are often made available to the community. References to the research which led to the development of the software, can be found in the corresponding parts of Section 2 Research.

3.1 Saliency

3.1.1 Saliency Map Algorithm

The algorithms for the classical Itti [43] saliency maps as well as the GBVS maps [42] are available as MATLAB source code at the Saliency Map Algorithm page, [29].

3.1.2 SaliencyToolbox

The SaliencyToolbox, [84] is a collection of Matlab functions and scripts for computing the saliency map for an image, for determining the extent of a proto-object, and for serially scanning the image with the focus of attention.

3.1.3 Frequency-tuned Saliency

The code used in the CVPR 2009 paper "Frequency-tuned Salient Region Detection" ([1]) is accessible through the online presentation of the work at [31].

3.1.4 FastSalience

One of the few software tools which are not implemented in MATLAB is part of the Nick's Machine Perception Toolbox (NMPT), [19].FastSalience is an implementation of the "Fast Salience Using Natural-statistics" algorithm from Butko et al., [20].

3.2 Salient regions

The software used for performance evaluation and comparison of salient region detectors and descriptors [57] as well as the test datasets (4.3) can be found online from the Oxford Vision Group page ([83]). The comparison scripts are MATLAB code, while some of the detectors (IBR,EBR, MSER) are available only as executables. Specific detector and descriptor software is given below.

3.2.1 Detectors

The most popular affine covariant region detector *Maximally Stable Extremal Regions MSER* ([56]) has several available implementations, most popular of which are:

1. **MATLAB Computer Vision Systems Toolbox**. The MSER detector is implemented as a function detectMSERFeatures in the MATLAB CVS Toolbox since release R2012a.

2. **OpenCV**. There is MSER class descendant from the FeatureDetector class in the Feature Detection and Description functionality in the open source OpenCV library.

3. **Vision Lab Features Library (VLFeat)**,[77]. The MSER is part of the VLFeat library.

The software for the *Maximally stable color region (MSCR)* detector, [41], is available at the Forssen's homepage [40]. No code for the *Maximally Stable Volumes (MSVs)*, [26] and for the *Structure-Guided Salient Region detector (SGSR)*, [38] or the enhanced MSER with Canny,[87] was found online. The code for the *Morphology based Stable Salient Regions (MSSR)* detector, Ranguelova et al. [66, 67] is available in the (for now private) git NLeSC repository.

3.2.2 Descriptors

There are numerous implementations of the *SIFT* descriptor, [54]:

1. **Lowe's own executable** is available at the Demo Software SIFT Keypointpage, [53].

2. **Vision Lab Features Library (VLFeat)**,[77]. The SIFT is part of the VLFeat library. It includes implementations of both the SIFT detector and descriptor.

3. **OpenCV**. This OpenCV tutorial explains the SIFT Python API in OpenCV.

4. **GPU** implementation of SIFT, [74] is available at SiftGPU, [93].

The related HoG detector,*SURF* is implemented as a function detectSURFFeatures in the MATLAB CVS Toolbox since release R2011b. The SURF Tutorial describes the SURF Python API in OpenCV.

Many *binary descriptors* have also become standard in many Computer vision software libraries:

1. **MATLAB's Computer Vision Systems (CSV) Toolbox** contains many feature detectors and their descriptors (Harris, FAST, FREAK, BRISK in addition to SURF and MSER) and possibility to match and display matched features. The Feature Detection and Extraction functionality contains binary descriptors since release R2013a.

2. **VLFeat Library** supports three local descriptors- SIFT, LIOP and raw patches (from which any other descriptor can be computed). Their use with the detectors is described in the Covariant feature detectors tutorial .

3. **OpenCV library** contains DescriptorExtractor class which supports the following descriptors- SIFT, SURF, BRIEF, BRISK, ORB and FREAK. It supports also descriptor matchers.

For the more recent descriptors authors often publish code along with their paper, e.g. the code for the *BOLD*, [13] can be obtained online.

3.3 Deep Learning

With the excellent performance of CNNs on the ImageNet classification dataset and many other recognition tasks, there is a boom of development of software tools implementing deep learning and CNNs. The tools are often free and open source. At DeepLearning.net there is an extensive list of such packages/libraries. Here, only the most popular are presented:

3.3.1 Caffe

Caffe ([35]) is BSD2-Clause license modular framework for deep learning developed by the Berkeley Vision and Learning Center (BVLC). The framework is a C++ library with Python and MATLAB bindings fr training and deploying general-purpose CNNs and other deep models on commodity architectures. It is a very popular framework, both in academia and industry due to its speed performance- it can process $60M$ images per day with single NVIDIA K40 GPU. There is a large community of user and user groups and contributors on GitHub. A technical report for Caffe can be found at it's git repository. It is the most popular open source project on computer vision and deep learning.

3.3.2 Torch7

Torch ([33]) is a scientific computing framework with wide support for machine learning algorithms. It is efficient due to the underlying C/CUDA implementations. It has interfaces to C, via LuaJIT, linear algebra routines, neural networks and numeric optimization routines. It runs on Mac OS X and Ubuntu 12+. There is a large community of contributors and users and the developers and maintainers are from Facebook AI Research, Google DeepMind, Twitter etc.

3.3.3 Theano

Theano ([32]) is a Python library allowing you definition, optimization, and evaluation of mathematical expressions involving multi-dimensional arrays efficiently. It supports transparent usage

of GPU. [18] and [14] are the initial publications about the library with a new academic publication coming up nearly every year. Many DeepLearning tutorials are based on Theano (Theano tutorial could be found online.

3.3.4 Keras

Keras ([60]) is a highly modular NN library in the spirit of Torch, written in Python, that uses Theano under the hood for optimized tensor manipulation on GPU and CPU. It was developed with a focus on enabling fast experimentation and seems to be suitable for beginners for easy and fast prototyping. It is available under the open source MIT license.

3.3.5 MATLAB Toolboxes

1. **DeepLearnToolbox**([64])is a Matlab/Octave toolbox for deep learning. Includes Deep Belief Nets, Stacked Autoencoders, Convolutional Neural Nets, Convolutional Autoencoders and vanilla Neural Nets. Each method has examples to help the starting process.

2. **MatConvNet** ([79]) is a MATLAB toolbox implementing Convolutional Neural Networks (CNNs) for computer vision applications. It is part of the VLFeat suite ([77]), which contains also salient feature extraction (see section 3.2.1 Detectors). MatConvNet is described in [78].

There are many other MATLAB software which provide CNN implementations, some can be found at the MATLAB FileExchange.

3.3.6 Deep Learning for Java

Deeplearning4j, [75] is a commercial-grade, open-source, distributed deep-learning library written for Java and Scala. Integrated with Hadoop and Spark, DL4J is designed to be used in business environments, rather than as a research tool. Deeplearning4j aims to be cutting-edge plug and play, more convention than configuration, which allows for fast prototyping for non-researchers. DL4J is customizable at scale. It is released under the Apache 2.0 license. It's main features include n-dimensional array class, GPU integration, Salable on Hadoop, Spark and Akka + AWS et al and a linear algebra library twice as fast as Numpy. There are several neural networks available, including CNNs.

3.3.7 Dataset annotation

A common problem of using Deep Learning is the need for having large annotated datasets (though also unsupervised approaches exist). This is an issue for large-scale imaging problems. Some software has been developed to tackle the issue.

LabelMe is a WEB-based image annotation tool that allows researchers to label images and share the annotations with the world. The images can be organized into collections, which can be nested. Images can also be uploaded into the system and shared.

The LabelMe MATLAB toolbox is used for interaction with the images and annotations in the LabelMe dataset, section 4.4 Object and Scene Recognition Datasets. The tool is described in this paper [71]. The toolbox also exists in 3D version, LabelMe3D which is described in [16]. There is also a mobile App version and instructions how could the labeling be outsourced using the Amazon Mechanical Turk.

3.4 Distributed software for CV

3.4.1 MapReduce

The example code for the MapReduce implementations of several CV tasks for data mining application, presented in [90] is available in GitHub repository hadoop_vision [89].

The *Hadoop Image Processing Interface (HIPI)* [8] is an open source library, developed by a team from the University of Virginia Graphics Lab.

3.4.2 StormCV

While Hadoop MapReduce is suitable for batch processing, Apache Storm,[5] a free and open source distributed real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing. Developers at TNO have developed StormCV,[10]. StormCV enables the use of Apache Storm for video processing by adding computer vision (CV) specific operations and data model. The platform enables the development of distributed video processing pipelines which can be deployed on Storm clusters. It is available under the Apache Software Foundation (ASF) license and some modules are under the OpenCV License.

3.4.3 CloudCV

The Large-Scale distributed CV as a Cloud service, CloudCV [9] is not available as a software package, it is Software as a Service (SaaS). It's current functionality includes image stitching, object detection, decaf-server, classification and VIP finding application. As such it could be useful link for scientists which need to perform tasks for which functionality is supported, especially interesting looks the object detection and classification services. Some related feature computation code code and datasets are available.

4 Datasets

The validation datasets (databases is the term used in the field), which are often gathered and used by the CV researchers to test the algorithms, are often made available to the community. Here links to the annotated segmentation and saliency datasets (see sections 4.1 Image Saliency Datasets and 4.2 Multimedia Datasets), interest regions (section 4.3 Salient Regions Datasets) and object and scenes classification datasets (section 4.4 Object and Scene Recognition Datasets) are given. Scientific image datasets are much harder to obtain access to (factors like privacy, cost of collecting and value of the data and competitiveness play an important role), so the focus here is only on the datasets, gathered for CV research purposes by the community.

4.1 Image Saliency Datasets

4.1.1 MSRA

The MSRA Database form the Visual computing group of Microsoft Research Asia [82] is the first large-scale labeled dataset made publicly available for training and evaluation. It contains two image sets. The first set consists of 20000 images labeled by three users, while the second set consists of 5000 images labeled by nine users. The labeling are available as bounding boxes. Figure 23 illustrates the dataset. The results of the proposed method by the authors of the dataset, have been published in [52].

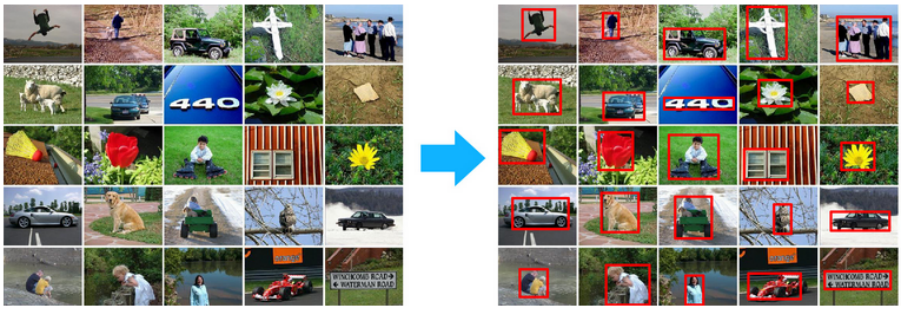


Figure 23: Examples of the MSRA dataset.

4.1.2 MSRA10k

This is an extension of the MSRA dataset, which addresses the coarse-grained limitation of the MSRA labeling (bounding boxes). The MSRA10k ([61]) dataset consists of 10000 randomly selected MSRA images for which a pixel-level saliency labeling is available. Figure 24 illustrates the dataset. This dataset is used by in a very recent paper in IEEE Transactions on PAMI [23]

Figure 24: Examples of the MSRA 10k dataset. First row: original images with ground truth rectangles from MSRA dataset. Second row: Ground truth with pixel accuracy.

and [30] (online resources with link to the software).

4.1.3 CSSD and ECSSD

Although images from MSRA-1000 [1] have a large variety in their content, background structures are primarily simple and smooth. To represent the situations that natural images generally fall into, the Complex Scene Saliency Dataset (CSSD) [94] was proposed in [96] with 200 images. They contain diverse patterns in both foreground and background. The labeling has done by five helpers. These images were collected from the BSD300 (later extended to BSD500, [55]), VOC dataset [36] and internet.

Later, the CSSD was extended to a larger dataset (ECSSD) of 1000 images, which includes many semantically meaningful and structurally complex images for evaluation. The images are acquired from the internet and five helpers were asked to produce the ground truth masks. Examples of the images in the dataset can be seen on Figure 25.

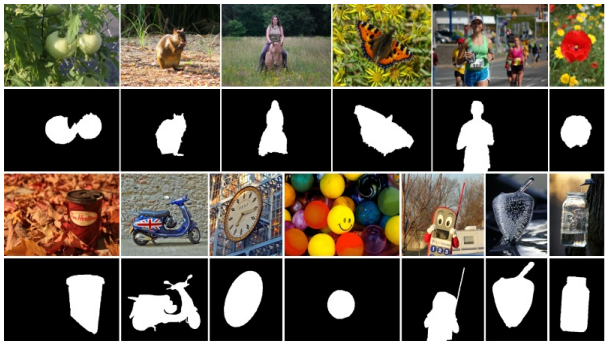


Figure 25: Examples of the ECSSD dataset.

4.1.4 DUT-OMRON

The Dalian University of Technology and the Omron Corporation introduced in the DUT-OMRON dataset [97] consisting of 5168, manually selected from more than 140000 images. They are re-sized to $400 \times x$ or $x \times 400$, where $x < 400$. They contain one or more salient objects with relatively complex background. Five people have labeled the pixel-wise ground truth along with bounding box and eye-fixation. The dataset is illustrated on Figure 26. The results of the experiments on the collected dataset were published in [98].

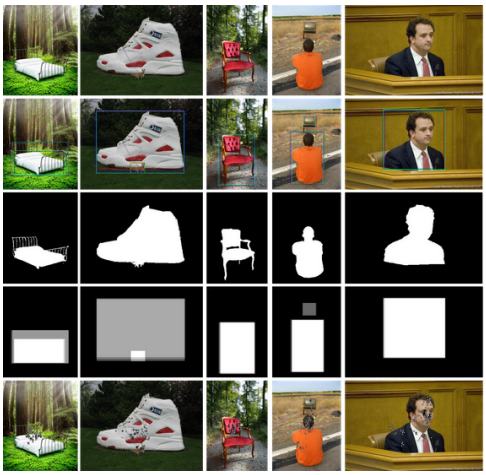


Figure 26: Samples of the DUT-OMRON dataset. From top to bottom: original image, bounding box ground truth, pixel-wise ground truth, average of the five binary masks and eye-fixation ground truth.

4.1.5 PASCAL-S

Another dataset, which aims at bridging the gap between fixations (human visual attention) and salient objects is the PASCAL-S dataset [51] provided by Georgia Tech, Caltech and UCLA. The dataset contains 850 images from the PASCAL 2010 with 12 subjects and 1296 object instances. The dataset is illustrated on Figure 27. The saliency segmentation method and the findings have been published at [50].

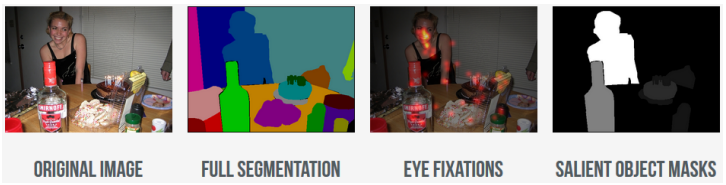


Figure 27: Examples of the PASCAL-S dataset.

4.2 Multimedia Datasets

4.2.1 MSRA-MM

In 2009, the researchers from Microsoft Research Asia have released 2 versions of large multimedia datasets- MSRA-MM [85]. MSRA-MM 1.0 consists of two sub-datasets, i.e., an image dataset and a video dataset that are collected from the image and video search engines. For image dataset, there are about 1000 images per query for 68 representative queries based on the log of search engines. There are 65443 images in total. For the video dataset, 165 representative queries have been selected from a log resulting in total of 10277 videos. Due to copyright issues, the raw image and video data are not available, but only features and annotations are provided. The dataset is explained in detail in a technical report [86].

4.3 Salient Regions Datasets

Surprisingly there are not many datasets available for testing salient region detection.

4.3.1 Oxford Dataset

For more than a decade the standard dataset was provided by Mikolajczyk et al. [57]. It is available under Test Data from the Oxford Vision Group page. The dataset is rather small, contains only 48 images, but real (not simulated). The dataset is illustrated Figure 28.

Five different changes in imaging conditions are represented: viewpoint changes (a) & (b); scale changes (c) & (d); image blur (e) & (f); JPEG compression (g); and illumination (h). The effect of changing the image conditions can be separated from the effect of changing the scene type. One scene type contains homogeneous regions with distinctive edge boundaries (e.g. graffiti, buildings), and the other contains repeated textures of different forms. The authors referred to these as *structured* versus *textured* scenes respectively. For example sequence (a) is of structured type, while (b) is example of textured sequence.

In the viewpoint change test the camera varies from a front-parallel view to one with significant foreshortening at approximately 60 degrees to the camera. For details of the other image transformations, the reader is referred to the publication [57] and the data description. All images are of small resolution for today's standards, but considered medium a decade ago (approximately 800 $\times$ 640 pixels). Since the images are either of planar scenes or the camera position is fixed during acquisition, the images are related by homographies (plane protective transformations). All homographies between the reference (leftmost) image and the other images in a particular dataset are provided.

Figure 28: Oxford vision group dataset: (a), (b) Viewpoint change, (c), (d) Zoom+rotation, (e), (f) Image blur, (g) JPEG compression, (h) Light change. In the case of viewpoint change, scale change and blur, the same change in imaging conditions is applied to two different scene types: structured and textured scenes. The left most image of each set is used as the reference image.

4.3.2 Freiburg Dataset

More recently, in 2014, Fissher et al. from Freiburg University proposed a new database for evaluation of salient regions descriptors performance [39]. The dataset contains 416 images. It is generated by applying 6 different types of transformations with varying strengths to 16 base images we obtained from Flickr Figure 29 shows some of them. The dataset is available online.

Figure 29: Example base images of the Freiburg dataset.

To each base image geometric transformations (rotation, zoom, perspective, and nonlinear deformation) have been applied in various magnitudes as well as changes to lighting and focus by adding blur. The transformations are shown in Figure 30.

Figure 30: Examples of extreme image transformations. From left to right: blur, lighting change, nonlinear deformation, perspective change, rotation, zoom.

While the Mikolajczyk et al. dataset [57] was not generated synthetically but contains photos taken from different viewpoints or with different camera settings, this dataset is artificially created. While the former reflects reality better than a synthetic dataset, the latter enables evaluation of the effect of each type of transformation independently of the image content.

4.4 Object and Scene Recognition Datasets

4.4.1 MIT-CSAIL

The goal of the MIT-CSAIL dataset [4] is to provide a large set of images of natural scenes (principally office and street scenes), together with manual segmentations/labelings of many types of objects, so that it becomes easier to work on general multi-object detection algorithms. The dataset contains indoor and outdoor objects in office and urban environments. There are

annotations for more than 30 objects in context in thousands of images and sequences with 2500 annotated frames. Examples of the dataset are shown in figure 31.

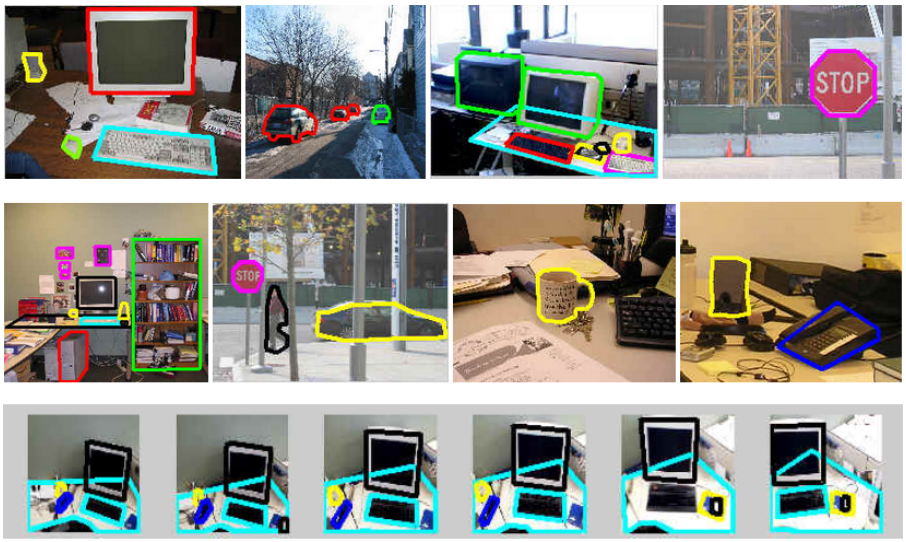


Figure 31: Examples of the MIT-CSAIL dataset.

4.4.2 LabelMe

An online annotated dataset incrementally filled up by users can be downloaded using the LabelME tool from [12]. The LabelMe3D dataset contains labeled images of many everyday scenes and object categories in absolute real world 3D coordinates. The toolboxes designed to work with these datasets are described in 3.3.7.

4.4.3 SUN

Another large dataset of annotated images covering large variety of scenes, places and objects within, provided also by the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT, is the SUN dataset, [34]. The SUN2012 contains 16873 images and SUN contains currently 131067 images, 908 Scene categories and 4479 object categories with more than $310k$ segmented objects. The SUN397 benchmark for scene classification can de used including code, pre-computed features etc. The SUN dataset can bee downloaded also with the LabelMe MATLAB Toolbox. The publication about the SUN dataset is [11].

4.4.4 Places

One of the largest annotated datasets for scene recognition is the Places dataset (also by CSAIL, MIT), [24]. It contains almost 2.5 million images in 205 scene categories. Along with the dataset, one can access the Places-CNNs, the convolutional neural networks trained on Places, DrawCNN- a visualization of the units' connections for the CNN, the online recognition demo and some sample MATLAB code for using the synthetic receptive field of unit to segment image and visualize the activated regions. The publications where the dataset is described are [27, 28].

5 Applications

In any scientific discipline, where for studying the object of research, the scientist need to process (large scale) image/video datasets, CV technology can be applied. For many decades, CV techniques have been applied widely and become standard tool in many scientific applications. Examples include remote sensing (imaging Earth or a planet), electrical resistivity imaging (geophysical method to image the underground), sonar and radar imaging, and one of the most important applications- (bio)medical imaging. Here, only a few examples of recently emerging application domains are given, even more new challenges will appear when scientists from different domains become aware of the capabilities and potential of the CV technology.

5.1 Animal biometrics

In [44], Kuehl and Burghardt give overview of the methodologies and trends in the emerging field of *animal biometrics*. It is an exciting field operating at the intersection between pattern recognition, ecology and information sciences. The subject of the field is to produce computerized systems for phenotypic measurement and interpretation. The main questions for which such systems helps to find the answers to are: how to profile species, individuals and animal behavior by representing phenotypic appearance. Figure 32 illustrates the main components of a biometric system.

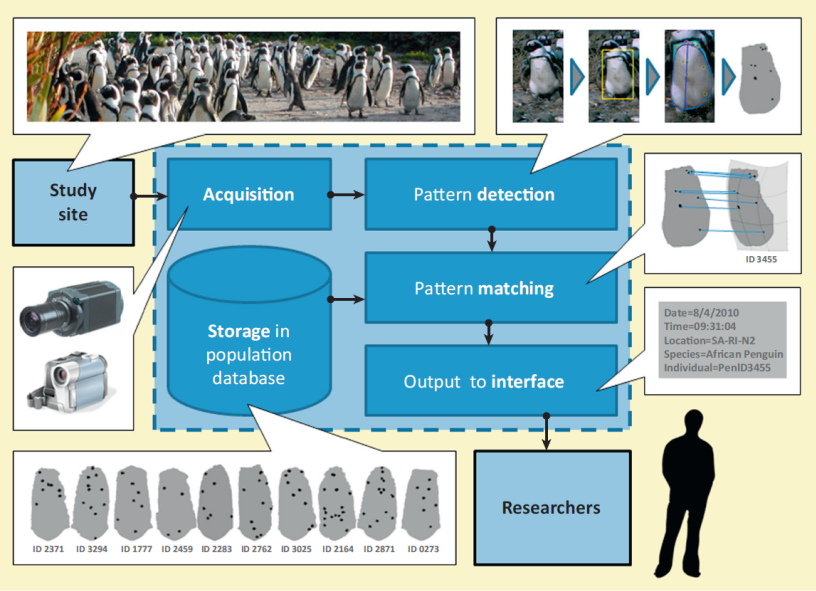


Figure 32: Main components of an animal biometric system. This flowchart summarizes how information from a study site is measured and interpreted for the researcher by an animal biometric system.

The system parts can either be connected directly on-site or remotely via networks. Each of the components is illustrated, using individual African penguin recognition by spot pattern as an example. Acquisition: automatic or semi-automatic collection of images or video from fixed field cameras, observers or the general public. Detection: the use of computer algorithms to search the images to find those that contain the biometric entity of interest and then to

extract relevant information about that entity (e.g., the chest spots of a penguin). Storage: the extracted data on the entity is reduced to a compact mathematical form that can be stored in a suitable database. Matching: the mathematical data on the entity are then compared with other data already stored in the database to find matches that enable the individual or the behavior to be identified, using methods akin to the matching of fingerprints to identify humans. Interfacing: presenting the output of the biometric system to a user or software system for further analysis.

Animal biometrics is important field not only for ecological researchers, but for the general public. For example, in [73], a biometric system for face recognition of pet animals (mainly dogs) have been developed.

5.2 Plant identification

Similar field is automatic plant identification. This is an example of the Classification task (What is my object?). Often the identification of trees and flowers is performed from images of the leaves of the plant. Nowadays, many mobile apps exist for the general public.

One such CV system is Leafsnap, [45]. Its goal is to assist botanists by automating the tedious and error-prone process of identifying existing plant species.

The recognition process, developed by the research groups from Columbia University and University of Maryland, consists of, [46]:

1. **Segmenting** the image to obtain a binary image separating the leaf from the background. This is implemented using an Expectation-Maximization framework, estimating foreground and background color distributions in the HSV color-space.

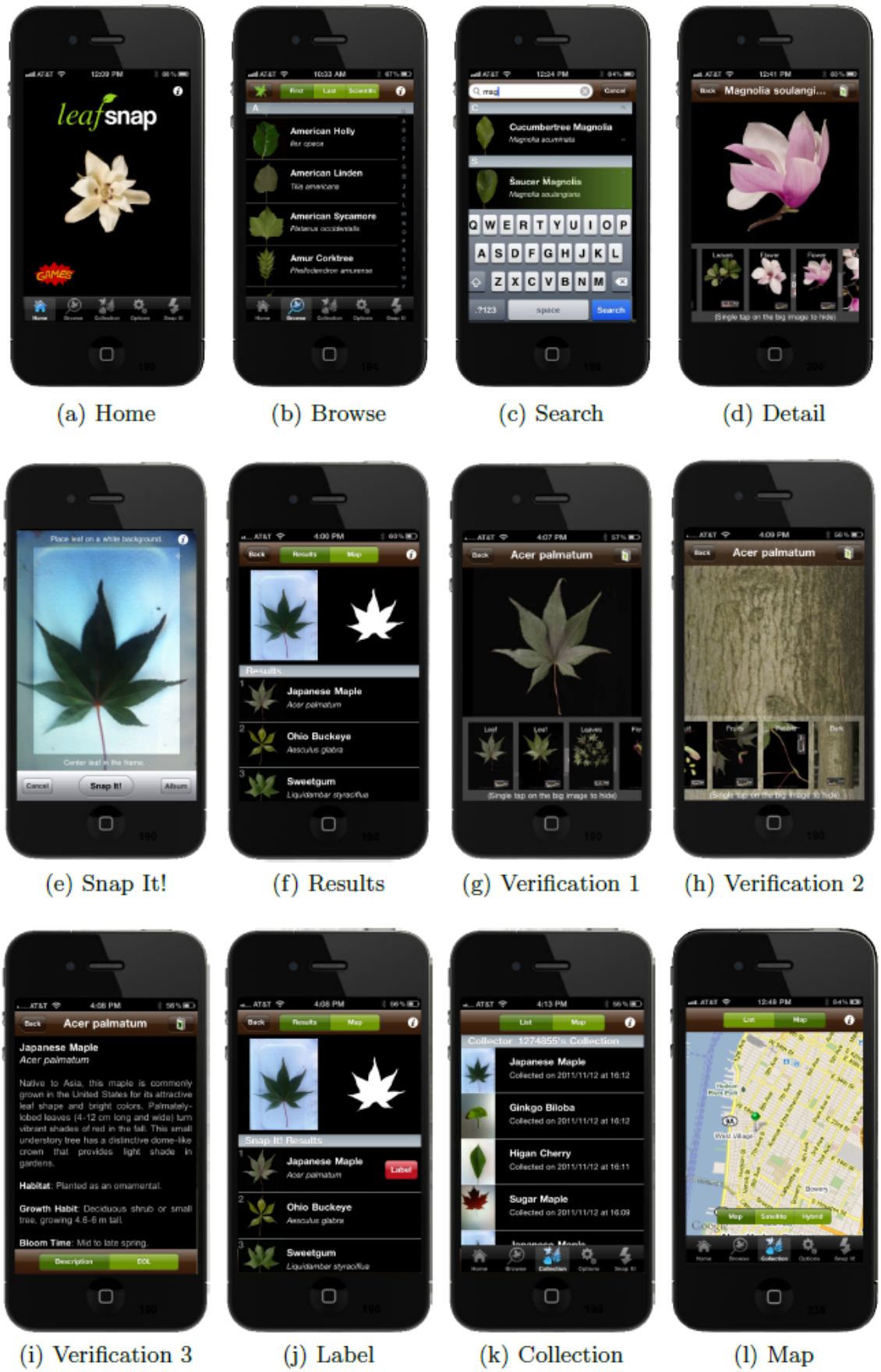
2. **Extracting features** from the binarized image for compactly and discriminatively representing the shape of the leaf. The features used are histograms of curvature over scale as the feature representation, robustly and efficiently implemented using integral measures of curvature.

3. **Comparing the features** to those from a labeled database of leaf images and returning the species with the closest matches. Due to the discriminative power of the features and the size of our labeled dataset, we use a simple nearest neighbor approach with the L_1-norm.

The system has the same components as that of an animal biometric system (Figure 32), which indicates, that plan biometrics is a very similar application domain to animal biometrics.

Figure 33 gives an impression of the functionality of Leafsnap.

The example of tree species identification (see the Introduction section) is another problem from the same domain. Also, in a the current NLeSC project candYgene image-based classification of different tomato species collected from different places in the world could be applied. Other examples of open-source platforms for biological-image analysis include Fiji, an image analysis tools for proteomics, or (plant) phenomics.



(a) Home (b) Browse (c) Search (d) Detail

(e) Snap It! (f) Results (g) Verification 1 (h) Verification 2

(i) Verification 3 (j) Label (k) Collection (l) Map

Figure 33: Tour of the iPhone version of Leafsnap.

5.3 Computer forensics

Forensic Science is the application of knowledge from several branches of science to answer questions relevant to a legal system. Due to the evolution of criminal activities, more specialized disciplines have been involved, such as Computer Science, Engineering and Economics. The research field that unites the fields of Forensic Science and Computer Science is called *Computer (Digital) Forensics* and encompasses the study of research methods, driven by hypothesis, of a specific problem, through the use of computers and computational methods.

Many questions in digital forensics can be answered (or assisted) by CV technologies, such as face detection and identification, same object/scene identification, image/video categorization, camera identification etc. for which the presented earlier CV methods are very highly applicable.

There are also specific problems for example, photogrammetry, 3D reconstruction of impressions, reconstruction of fragmented documents and images etc. Research is done in novel CV methods which help solve these problems, [3].

5.4 Social signal processing

In the last years, a multi-disciplinary area, called *Social Signal Processing* emerged where computer vision and social sciences converge. One of the research topics is the development of video surveillance algorithms to help studying social interactions. Technologically the problems are those of gesture (frequency) analysis, gait, pose and emotion recognition, geometric configuration of people, etc. (Figure 34). A survey of this domain with presentation of the main applications and research results can be found in [80].

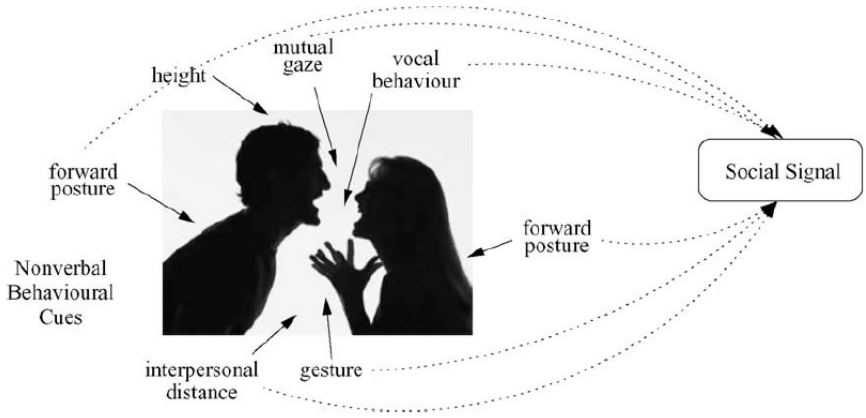


Figure 34: Behavioral cues and social signals. Multiple behavioral cues (vocal behavior, posture, mutual gaze, interpersonal distance, etc.) combine to produce a social signal (in this case aggressivity or disagreement).

Measuring behavior is the premier interdisciplinary event for scientists and practitioners concerned with the study of human or animal behavior. The biannual conference is held traditionally in the Netherlands. At the same time, the CV community becomes more aware of these applciaiton areas, for example one tutorial at the Computer Vision ad Pattern Recognition (CVPR) conference 2015 was on "Group Behavior Analysis and Its Applications".

6 Conclusions

To goal of this document was to present focused review of the state-of-the-art in (large-scale) computer vision research. The reader should have a general overview and find useful pointers (section Research) to where one might direct research and development efforts. It is also important to know the current trends in the field to be able to sustain some level of expertise in it. Based on researching the articles, software, the ever growing number of research image datasets and the (scientific) application domains few main concussions crystallize:

- CV field is very large and fast changing. More and more scientific disciplines pose new challenges to CV. To sustain some level of expertise in it, NLeSc should conduct (scientific-driven) CV research, for example within eStep.
- The research efforts should focus around the relevant research questions, presented in the document: localization, identification and classification. Given the current expertise, it is very reasonable to continue improving the MSSR salient region detector (see section 2.2 Salient regions). Also, some expertise should be build in Convolutional neural networks (section 2.3 Convolutional Neural Networks) by organizing seminars and obtaining practical knowledge for example within project Sherlock.
- The CV researchers in academia are focused mostly on large commercial applications like organizing large photo collections, autonomous driving, etc. There is still not enough effort directed towards the other domain sciences (except from the medical imaging) where NLeSC can contribute (section 5 Applications). Also, sustaining expertise in large scale frameworks for CV systems (section 2.4 Large Scale CV Systems) fits very well the mission and strategy of the center.
- Although not covered by this overview, at NLeSC we see the appearance of new modality, the Point clouds, by which the real world is captured in a way which removes some of the limitations of CV mentioned in section 1 Introduction. For example in Patty project, methods from CV have been applied to generate PC from images. There is a research in trying to identify objects directly from the PC and it is interesting to find out can the CV algorithms be applied directly on PC or new or adapted algorithms are needed?

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Ssstrunk. Frequency-tuned Salient Region Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597 – 1604, 2009.

[2] Harsh Agrawal, Clint Solomon Mathialagan, Yash Goyal, Neelima Chavali, Prakriti Banik, Akrit Mohapatra, Ahmed Osman, and Dhruv Batra. Cloudcv: Large scale distributed computer vision as a cloud service. *CoRR*, abs/1506.04130, 2015.

[3] Fernanda A. Andaló and Siome Goldenstein. Computer vision methods applicable to forensic science. In *Workshop of Theses and Dissertations, XXVI Conference on Graphics, Patterns and Images (WTD/SIBGRAPI '13)*, Arequipa, Peru, August 2013.

[4] Kevin Murphy Antonio Torralba and William Freeman. The mit-csail database of objects and scenes. `http://web.mit.edu/torralba/www/database.html`. [Online; accessed 18 June 2015].

[5] Apache. Storm. https://storm.apache.org/. [Online; accessed 5 August 2015].

[6] Itamar Arel, Derek C. Rose, and Thomas P. Karnowski. Research frontier: Deep machine learning–a new frontier in artificial intelligence research. *Comp. Intell. Mag.*, 5(4):13–18, November 2010.

[7] C. Sweeney at al. Hipi: A hadoop image processing interface for image-based mapreduce tasks. https://cs.ucsb.edu/~cmsweeney/papers/undergrad_thesis.pdf. [Online; accessed 5 August 2015].

[8] C. Sweeney at al. Hipi hadoop image processing interface. http://hipi.cs.virginia.edu. [Online; accessed 5 August 2015].

[9] H. Agarval at al. Cloudcv large-scale distributed computer vision as a cloud service. http://cloudcv.org/. [Online; accessed 5 August 2015].

[10] J. Schavemaker at al. Stormcv = apache storm +opencv= large-scale distributed image and video analysis. https://github.com/sensorstorm/StormCV. [Online; accessed 5 August 2015].

[11] J. Xiao at al. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.

[12] A. Torralba et al. B. Russell. Labelme. http://labelme2.csail.mit.edu/Release3.0/browserTools/php/dataset.php.

[13] Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Bold - binary online learned descriptor for efficient image matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[14] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[15] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[16] B.C.Russell and A. Torralba. Building a Database of 3D Scenes from User Annotations. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009.

[17] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[18] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[19] N. Butko. Nick's machine perception toolbox (nmpt). http://mplab.ucsd.edu/~nick/NMPT/main.html. [Online; accessed 4 August 2015].

[20] Nicholas J. Butko, Lingyun Zhang, Garrison W. Cottrell, and Javier R. Movellan. Visual saliency model for robot cameras. In *ICRA*, pages 2398–2403. IEEE, 2008.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.

[22] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[23] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[24] MIT CSAIL. Places, the scence recognition database. `http://places.csail.mit.edu/`.

[25] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[26] Michael Donoser and Horst Bischof. 3d segmentation by maximally stable volumes (msvs). In *ICPR (1)*, pages 63–66. IEEE Computer Society, 2006.

[27] B. Zhou et al. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

[28] B. Zhou et al. Object detectors emerge in deep scene cnns. *International Conference on Learning Representations (ICLR)*, 2015.

[29] J. Harel et al. Saliency map algorithm. `http://www.vision.caltech.edu/~harel/share/gbvs.php`. [Online; accessed 4 August 2015].

[30] Ming-Ming Cheng et al. Global contrast based salient region detection. `http://mmcheng.net/zh/salobj`. [Online; accessed 13 April 2015].

[31] Radhakrishna Achanta et al. Frequency-tuned salient region detection. `http://ivrgwww.epfl.ch/supplementary_material/RK_CVPR09/index.html`. [Online; accessed 13 April 2015].

[32] R.Collobert et al. Theano. `http://deeplearning.net/software/theano/`. [Online; accessed 30 June 2015].

[33] R.Collobert et al. Torch. `torch.ch`. [Online; accessed 30 June 2015].

[34] Xiao et al. Sun database. `http://groups.csail.mit.edu/vision/SUN/`.

[35] Yangqing Jia et al. Caffe. `http://caffe.berkeleyvision.org/`. [Online; accessed 29 June 2015].

[36] Mark Everingham. The pascal visual object classes homepage. `http://pascallin.ecs.soton.ac.uk/challenges/VOC/`. [Online; accessed 14 April 2015].

[37] Bin Fan, Fuchao Wu, and Zhanyi Hu. Rotationally invariant descriptors using intensity order pooling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):2031–2045, 2012.

[38] Shufei Fan and Frank Ferrie. Structure guided salient region detector. In *In Proceedings of British Machine Vision Conference*, pages 423–432, 2008.

[39] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014.

[40] Per-Erik Forssen. Maximally stable colour regions. `http://www.cs.ubc.ca/~perfo/mscr/`. [Online; accessed 21 April 2015].

[41] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *CVPR*, 2007.

[42] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 19*, pages 545–552. MIT Press, 2007.

[43] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.

[44] H. Kuehl and T. Burghardt. Animal biometrics: quantifying and detecting phenotypic appearance. *Trends in Ecology & Evolution*, 28:432–441, 2013.

[45] N. Kumar. Leafsnap: An electronic field guide. `http://neerajkumar.org/projects/leafsnap/`. [Online; accessed 7 August 2015].

[46] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida Lopez, and Joo V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *The 12th European Conference on Computer Vision (ECCV)*, October 2012.

[47] Tai S. Lee and David Mumford. Hierarchical bayesian inference in the visual cortex. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 20(7):1434–1448, 2003.

[48] Tai Sing Lee, David Mumford, Richard Romero, and Victor A.F. Lamme. The role of the primary visual cortex in higher level vision. *Vision Research*, 38:2429 – 2454, 1998.

[49] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society.

[50] Jian Li, M. D. Levine, Xiangjing An, Xin Xu, and Hangen He. Visual saliency based on scale-space analysis in the frequency domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):996–1010, 2013.

[51] Yin Li. The pascal-s dataset. `http://cbi.gatech.edu/salobj/`. [Online; accessed 14 April 2015].

[52] Tie Liu, Jian Sun, Nan-Ning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1–8, 2007.

[53] David Lowe. Sift keypoint detector. `http://www.cs.ubc.ca/~lowe/keypoints/`. [Online; accessed 4 August 2015].

[54] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[55] David Martin. Bsd300/500: The berkeley segmentation dataset and benchmark. `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`. [Online; accessed 14 April 2015].

[56] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 36.1–36.10, 2002.

[57] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, November 2005.

[58] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[59] Ondrej Miksik and Krystian Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *ICPR*, pages 2681–2684. IEEE, 2012.

[60] MIT. Keras. `http://keras.io/`. [Online; accessed 12 August 2015].

[61] M.M.Cheng. Msra10k: Pixel accurate salient object labeling for 10 000 images from msra dataset. `http://mmcheng.net/msra10k/`. [Online; accessed 13 April 2015].

[62] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.

[63] NVIDIA. Nvidia digits interactive deep learning gpu training system. `https://developer.nvidia.com/digits`. [Online; accessed 5 August 2015].

[64] Rasmus Berg Palm. Deeplearntoolbox. `https://github.com/rasmusbergpalm/DeepLearnToolbox`. [Online; accessed 1 July 2015].

[65] Wolfgang Ponweiser, Gerald Umgeher, and Markus Vincze. *A reusable dynamic framework for cognitive vision systems*. na, 2003.

[66] E. B. Ranguelova and E. J. Pauwels. Morphology-Based Stable Salient Regions Detector. In *Proceedings of International Conference on Image and Vision Computing New Zealand 2006*, pages 97 – 102, 2006.

[67] E. B. Ranguelova and E. J. Pauwels. Saliency Detection And Matching For Photo-Identification Of Humpback Whales. *International Journal on Graphics, Vision and Image Processing*, 2006.

[68] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[69] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[70] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[71] B.C. Russell, A. Torralba, K.P. Murhy, and W.T.Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

[72] C. Scharfenberger, A. Wong, K. Fergani, J.S. Zelek, and D.A. Clausi. Statistical textural distinctiveness for salient region detection in natural images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 979–986, June 2013.

[73] Santosh Kumarand Sanjay Singh. Biometric recognition for pet animal. *Journal of Software Engineering and Applications*, 7:470–482, 2014.

[74] Sudipta N. Sinha, Jan michael Frahm, Marc Pollefeys, and Yakup Genc. Gpu-based video feature tracking and matching. Technical report, In Workshop on Edge Computing Using New Commodity Architectures, 2006.

[75] Skymind. Dl4j deep learning for java. `http://deeplearning4j.org/`. [Online; accessed 5 August 2015].

[76] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Pub., 1999.

[77] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[78] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.

[79] Veldadi and Fulkerson. Matconvnet. `http://www.vlfeat.org/matconvnet/`. [Online; accessed 1 July 2015].

[80] Alessandro Vinciarelli, Maja Pantic, and Hervé Bourlard. Social signal processing: Survey of an emerging domain. *Image Vision Comput.*, 27(12):1743–1759, November 2009.

[81] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[82] Microsoft Research Visual Computing Group. Msra salient object database. `http://research.microsoft.com/en-us/um/people/jiansun/salientobject/salient_object.htm`. [Online; accessed 13 April 2015].

[83] Oxford Visual Geometry Group. Affine covariant regions. `http://www.robots.ox.ac.uk/~vgg/research/affine/`. [Online; accessed 21 April 2015].

[84] Dirk Walther and Christof Koch. Saliencytoolbox. `http://saliencytoolbox.net/`. [Online; accessed 4 August 2015].

[85] Meng Wang. Msra-mm - msr asia internet multimedia dataset 1.0 and 2.0. `http://research.microsoft.com/en-us/projects/msrammdata/`. [Online; accessed 14 April 2015].

[86] Meng Wang, Linjun Yang, and Xian-Sheng Hua. Msra-mm: Bridging research and industrial societies for multimedia information retrieval. Technical Report MSR-TR-2009-30, Microsoft, March 2009.

[87] Sh. Wang, W. Wang, D Liu, F. Gu, and B.B.Dickson. Enhanced maximally stable extremal regions with canny detector and applicationin image classification. *Journal of Computational Information Systems*, 10(14):6093–6100, 2014.

[88] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 603–610, Washington, DC, USA, 2011. IEEE Computer Society.

[89] B. White. hadoop_vision. `https://github.com/bwhite/hadoop_vision`. [Online; accessed 5 August 2015].

[90] Brandyn White, Tom Yeh, Jimmy Lin, and Larry Davis. Web-scale computer vision using mapreduce for multimedia data mining. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, MDMKDD '10, pages 9:1–9:10, New York, NY, USA, 2010. ACM.

[91] S. Wrede, J. Fritsch, C. Bauckhage, and G. Sagerer. An XML Based Framework for Cognitive Vision Architectures. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 757–760, 2004.

[92] Sebastian Wrede, Christian Bauckhage, Gerhard Sagerer, Wolfgang Ponweiser, and Markus Vincze. Integration frameworks for large scale cognitive vision systems - an evaluative study. In *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004.*, pages 761–764, 2004.

[93] Changchang Wu. iftgpu: A gpu implementation of scale invariant feature transform (sift). `http://cs.unc.edu/~ccwu/siftgpu/`. [Online; accessed 4 August 2015].

[94] Qiong Yan. Cssd: Complex scene saliency dataset. `http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html`. [Online; accessed 14 April 2015].

[95] Qiong Yan. Ecssd: Extended complex scene saliency dataset. `http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html`. [Online; accessed 13 April 2015].

[96] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical Saliency Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 1155 – 1162, 2013.

[97] Chuan Yang. The dut-omron image dataset. `http://202.118.75.4/lu/DUT-OMRON/index.htm`. [Online; accessed 14 April 2015].

[98] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3166–3173. IEEE, 2013.

[99] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.