ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2017. Vol. 6. No. 2. 79–84

Analysis of computer vision and image analysis technics

Z. Rybchak [1], O. Basystiuk [2]

[1]Lviv Polytechnic National University, e-mail: zoriana.rybchak@gmail.com
[2]Lviv Polytechnic National University, e-mail: obasystiuk@gmail.com

Received February 15.2017: accepted May 28.2017

Abstract. Computer vision and image recognition are one of the most popular theme nowadays. Moreover, this technology developing really fast, so filed of usage increased. The main aims of this article are explain basic principles of this field and overview some interesting technologies that nowadays are widely used in computer vision and image recognition.

Key words: computer vision, image recognition, object recognition, machine learning, computer with high-level understanding, digital images processing, scene reconstruction.

INTRODUCTION

Seems that our brains make visual recognise pretty easy. For humans it does not take any effort, to see the difference between a dog and a cat, or a car and a plane, read a sign, or recognize a human's face. However, what about computer vision and image recognition, is image recognition problem same easily solved for computer? Definitely not, there are actually hard problems, that need to be solve, to teach computer recognise images: they only for first view looks like easy, I think it happens because our brains are incredibly good at understanding images. Nevertheless, just imagine how many field of human life can be improved by using computer vision. Most common field of using is manufacturing, for example quality control, when you start manufacturing business you need quality control department, but what if replace this department using computers with computer vision, involve more people for creating something new, I think, this business will be more profitable . That`s why, last few years the field of machine learning has make huge progress in field of computer vision. Main point in this progress is creation of mathematical method for image recognition that will give us high accuracy result. Nowadays most popular are deep learning techniques for IR, especial convolutional neural nets, this method is much more advanced than earlier approaches like Fourier transforms. By involving deep learning method for this field was achieved significant increasing degrees of accuracy, with these techniques. Accuracy rate approaching to 95 percent. (Accuracy is usually measured against a how humans classified the data set.)

So, keep in mind that if you have not looked at Deep Learning based image recognition and object detection algorithms for your applications, you may be missing out on a huge opportunity to get better results.

Finally, we are ready to return to the main goal – understand main principles of image recognition methods using traditional computer vision techniques.

COMPUTER VISION MAIN TASKS

Computer vision is a part of Computer Science field that deals with how organise computer work for gaining high-level understanding information from digital images or videos. Moreover, computer vision unites the following areas image recognition, object recognition, object pose estimation, motion estimation, and scene reconstruction. There are huge volume of tasks, which computer vision can handle using methods for producing numerical or symbolic representation of information which was acquiring from real world, next steps are processing, analyzing and understanding digital images, and result of all this steps is high-dimensional data. Understanding in this context means the transformation of images into information which computer can understand. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and machine learning.

Some examples of computer vision applications and goals:

• automatic face recognition, and interpretation of expression;

• visual guidance of autonomous vehicles • automated medical image analysis, interpretation, and diagnosis;

• robotic manufacturing: manipulation, grading, and assembly of parts;

• OCR: recognition of printed or handwritten characters and words;

• agricultural robots: visual grading and harvesting of produce;

• smart offices: tracking of persons and objects; understanding gestures;

• biometric-based visual identification of persons • visually endowed robotic helpers;

• security monitoring and alerting; detection of anomaly;

• intelligent interpretive prostheses for the blind;

• tracking of moving objects; collision avoidance; stereoscopic depth;

• object-based (model-based) compression of video streams;

• general scene understanding.

In many respects, computer vision is an "AI-complete" problem: building general-purpose vision machines would entail, or require, solutions to most of the general goals of artificial intelligence. It would require finding ways of building flexible and robust visual representations of the world, maintaining and updating them, and interfacing them with attention, goals and plans.

Like other problems in AI, the challenge of vision can be described in terms of building a signal-to-symbol converter. The external world presents itself only as physical signals on sensory surfaces (such as videocamera, retina, microphone, etc), which explicitly express very little of the information required for intelligent understanding of the environment. These signals must be converted ultimately into symbolic representations whose manipulation allows the machine or organism to interact intelligently with the world.

Researchers that work in this field try to achieve automate tasks that the human visual system can do. So let`s take a look at the most popular method in field of Computer vision, and will examine how the works.

TECHNIQUES AND TOOLS

The process of image recognition in general looks not really complicated. There are only three steps, that you need to go through, for achieving the result:

1. Preprocessing – at this step you take filters and try to make your image more adapted for recognition.
2. Feature Extraction – at this step you try to recognize useful information and extract it and throw away extraneous information.
3. Classification – analyzing and recognition the information that you fetched at feature extraction step.

PREPROCESSING METHODS

This step is important because quality of our preprocessing affect application outcome. Most common preprocessing steps are normalization of contrast, subtraction the mean of image intensities, brightness effects correction, and division of image by the standard deviation. Sometimes, gamma correction gives slightly better results. In case, when you are dealing with color images, a color space transformation may help get better results.

For achieving good results of preprocessing photos, you need to do the following steps:

- Firstly, need to make reasonable guesses based on type of images you need to recognize.
- Secondly, try a few different ones and some of them might give slightly better results.
- Finally, you need to resize image to a fixed size, because next step performed on a fixed sized images.

However, keep in mind that nobody knows in advance which combination of preprocessing methods will produce the best results. Comparing the methods of image recognition and classification leads to the conclusion that:

- for structural method of preprocessing is important the accuracy of the allocation boundaries and information within these boundaries;
- for statistical methods is important to know the internal contours and mutual distribution of brightness values;
- for feature extraction methods (filters) should allocate contour components, which is typical for some existing methods, and obtain information about the distribution of structural elements outside the boundaries of objects.

FEATURE EXTRACTION

Why is recognition so hard? The real world is made of a jumble of objects, which all occlude one another and appear in different poses. Furthermore, the variability intrinsic within a class (e.g., dogs), due to complex non-rigid articulation and extreme variations in shape and appearance (e.g., between different breeds), makes it unlikely that we can simply perform exhaustive matching against a database of exemplars.

The recognition problem can be broken down along several axes.

1. For example, if we know what we are looking for, the problem is one of object detection, which involves quickly scanning an image to determine where a match may occur.
2. If we have a specific rigid object we are trying to recognize, we can search for characteristic feature points and verify that they align in a geometrically plausible way.
3. The most challenging version of recognition is general category (or class) recognition, which may involve recognizing instances of extremely varied classes such as animals or furniture.

In many instances, recognition depends heavily on the context of surrounding objects and scene elements. Woven into all of these techniques is the topic of learning, since handcrafting specific object recognizers seems like a futile approach given the complexity of the problem. But, let`s investigate how the computer recognizes objects.

If you want find numbers located on paper, you will notice a significant variation in RGB pixel values (for example paper white, numbers black). By running an edge detector on an image, application will get the shape of the number, and by comparing the acquire shapes with default number shapes, will make conclusion about what numbers are at the paper. Edge detector will retains the essential information, and ignore non- essential information. The step is called feature extraction.

In computer vision design of these features are critical to the performance of the algorithm. Turns out, we can use already created solution for edge detection. There are huge volumes of systems, some well-known features used in computer vision are Haar-like features introduced by Viola and Jones, Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF).

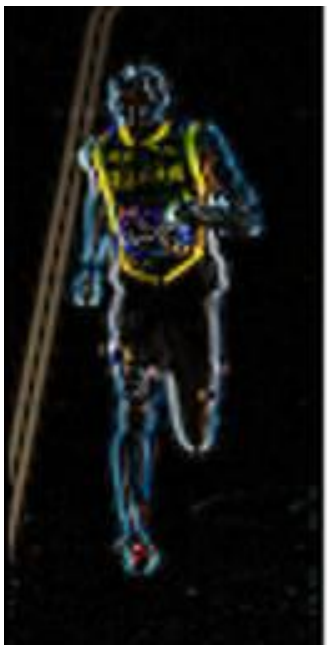

Fig. 1. Absolute value of x-gradient

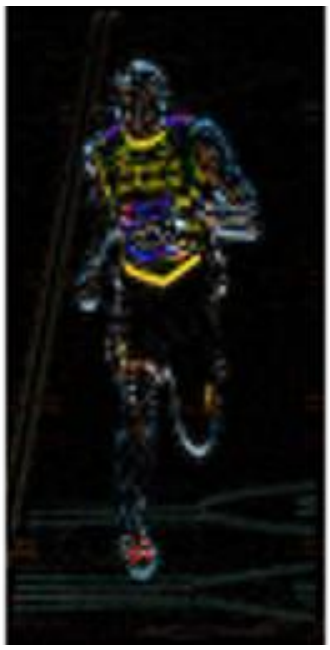

Fig. 2. Absolute value of y-gradient

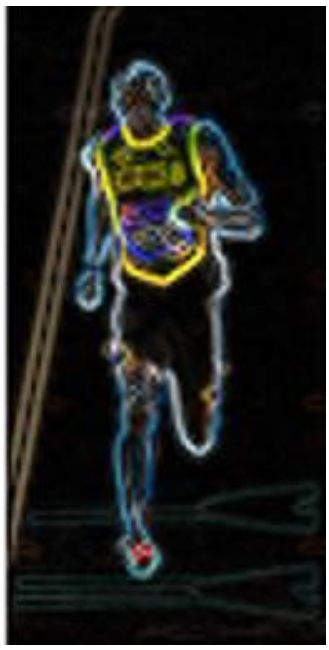

Fig. 3. Magnitude of gradient

Let's look crosier at algorithm implementation of Histogram of Oriented Gradients method of feature extraction. HOG is based on the idea that local object appearance can be effectively described by the distribution (histogram) of edge directions (oriented gradients). Algorithm consisted of six main steps:

- Image preprocessing – HOG feature descriptor used for pedestrian detection is calculated on a 64×128 patch of an image. Of course, an image may be of any size, but if image is bigger than 64×128 preprocessor will crop and scale image for this size.
- Calculate the Gradient Images – firstly, we need to calculate the horizontal and vertical gradients; This is easily achieved by filtering the image with the following kernels or by using ***Sobel*** operator in OpenCV with karnel size 1. Next step is finding of the magnitude and direction of gradient using the following formula:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

Or if you are using OpenCV, the calculation can be done using the function ***cartToPolar***.

The Figs. 1–3 below shows the gradients. Calculate Histogram of Gradients in 8×8 cells – In this step, the image is divided into 8×8 cells and a histogram of gradients is calculated for each 8×8 cells. But why 8×8 patch? It is a design choice informed by the scale of features we are looking for. HOG was firstly used for pedestrian detection initially and 8×8 cells in a photo of a pedestrian scaled to 64×128 are big enough to capture interesting features like face.

So, after creating of a cells(matrix in computer representation) we need to a histogram of gradients in these 8×8 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40–160 and than create the 9-bin histogram.

- Block Normalization – Gradients of an image are sensitive to overall lighting. If you make the image darker by dividing all pixel values by 2, the gradient magnitude will change by half, and therefore the histogram values will change by half. Ideally, we want our descriptor to be independent of lighting variations. In other words, we would like to "normalize" the histogram so they are not affected by lighting variations. Let's say we have an RGB color vector [128, 64, 32]. The length of this vector is:

$$\sqrt{128^2 + 64^2 + 32^2} = 146.64$$

This is also called the L2 norm of the vector. Dividing each element of this vector by 146.64 gives us a normalized vector [0.87, 0.43, 0.22].

Now consider another vector in which the elements are twice the value of the first vector 2 × [128, 64, 32] = = [256, 128, 6]. You can work it out yourself to see that normalizing [256, 128, 6] will result in [0.87, 0.43, 0.22], which is the same as the normalized version of the original RGB vector. You can see that normalizing a vector removes the scale.

Now that we know how to normalize a vector, you may be tempted to think that while calculating HOG you can simply normalize the 9×1 histogram the same way we normalized the 3×1 vector above. It is not a bad idea, but a better idea is to normalize over a bigger sized block of

16×16. A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector and it can be normalized just the way a 3×1 vector is normalized. The window is then moved by 8 pixels and a normalized 36×1 vector is calculated over this window and the process is repeated. Calculate the HOG feature vector – final feature vector is creating by appending all vectors into one huge vector.

How many memory we need to allocate for this vector? Each block is represented by 36×1 vector and each vector have 7 horizontal positions and 15 vertical positions. Let`s calculate:

$7 \times 15 = 105$;

$36 \times 105 = 3780$;

So we need **3780** dimensional vector.

LEARNING ALGIRITHMS FOR CLASSIFICATION

In the paragraph about Feature Extraction, we learned how to convert an image to a feature vector. In this paragraph, we will learn how a classification algorithm handle recognition process.

Before a classification algorithm can do recognition process, we need to trained him by labeled "training" data. Different learning algorithms learn differently, but the general principle is that learning algorithms treat feature vectors as points in higher dimensional space, and try to find similarity between all vectors of same class of images. Nevertheless, they need massive amounts of data to do it. There are massive and free-to-anyone databases contain millions of images labeled by keywords about what's inside the pictures. In case, if you have enough data for training you algorithm, it's time to build a machine that can learn from it. There are a lot of framework and open-source libraries, which can help you, build your recognition machine:

- Google TensorFlow – one of the better-known library, for machine learning.
- UC Berkeley's Caffe – library created by Berkeley University, become popular because of its ease of customizability and large community of innovators.
- Torch – is also popular, owing to its use by Facebook AI Research (FAIR), which open sourced some of its modules in early 2015.

Most common way nowadays to learn algorithm for recognition is deep learning. Deep learning algorithms are based on distributed representations. The underlying assumption behind distributed representations is that observed data are generated by the interactions of factors organized in layers. Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can be used to provide different amounts of abstraction.

Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones. These architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features are useful for learning.

For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures which remove redundancy in representation.

Many deep learning algorithms are applied to unsupervised learning tasks. This is an important benefit because unlabeled data are usually more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief network.

LEARINING AND VISION

When we recover the geometry of the world, or recognize objects, we are fitting models to the data provided by our eyes. These models are formed from our experience in two ways. In supervised learning, a teacher specifies class labels "this is a tiger" for example, and images. In unsupervised learning, which is the norm in biological systems, the process has to be driven by an attempt to fin1d good internal representations given the statistics of natural images.

Finding the best model is a compromise between fitting the data and minimizing the complexity of the model. In science the preference for simple theories over complex ones is known as Occam's razor and often is treated as a matter of aesthetics. However, for a visual organism that is constantly engaged in model construction, constructing better theories, i.e., ones with greater predictive power, is a matter of life and death!

A mathematical justification for preferring simple models or theories in accordance with available data arises from statistical learning theory. Flexible models with many degrees of freedom adapt to stochastic fluctuations in the data, whereas overly simple models cannot represent essential aspects of the signal. Vapnik and Chervonenkis estimate how much the expected performance risk of a selected solution, i.e., the best solution on the available data, deviates from the optimal solution in the model class.

This optimal solution with minimal expected risk most often does not minimize the costs on the available data. Uniform convergence of empirical risk to expected risk is a necessary and sufficient mathematical condition for learning. Image analysis is inherently multiscale; segmentation, grouping, or classification have to be performed at the appropriate scales of resolution. The foliage of trees in a photograph of a forest might appear

homogeneous at low resolution but a closer look at high resolution reveals differences in leaf shapes that generate tree-specific foliage textures. Statistical learning theory relates the complexity of models to the amount of available data, i.e., the appropriate image scale. In image segmentation these scales denote the spatial resolution of segments, the fuzziness of segment boundaries, and the number of segments, respectively. These scales have to be coupled by an underlying inference principle. The well-known stochastic optimization algorithm simulated annealing or its deterministic variants provide a computational temperature as a control parameter to couple these scales. These algorithms can be tuned for real-time applications and for just-in-time processing with limited resources as they occur in robotics, vision-based surveillance, and inspection.

Last but not the least, there are a lot of visual recognition challenges. Best known is ImageNet was launched by computer scientists at Stanford and Princeton in 2009 with 80,000 tagged images. It has since grown to include more than 14 million tagged images, any of which are up for grabs at any time for machine training purposes.

CONCLUSION

Nowadays, computer vision became one of the most popular sphere of machine learning. In my opinion, main reason for this is high spectrum of usage for computer vision. There are many frameworks for creation your recognition system, by them you can create app in a few minutes. Application based on computer vision technologies can solve not only trivial tasks, such as recognise, but complicated one like scene reconstruction or motion analyses. Some companies use combinations of open data and open-source frameworks, as long as they have a team of engineers, or they might just use hosted APIs if computer vision is not something on which they are staking their entire business.

And for companies with a wide range of very specific needs, there are custom solutions. No matter how it's approached, though, it's clear that image recognition rarely exists in isolation; it's made stronger by access to more and more pictures, real-time big data, unique applications and speed. The businesses that make the most of these connections are the ones that will be best poised for success. Huge amounts of different software tools, like: free-to-anyone databases, open-source libraries, frameworks, API, etc.; provided by top IT companies of the world create really good opportunities for studying fast. Many different worldwide competitions at computer vision. In my opinion, computer vision become trivial thing in next few years, and will be one of highly use technology in future.

REFERENCES

1. **Richard Szeliski. 2011.** Computer Vision: Algorithms and Applications. – United Kingdom: Springer London, 812 p.
2. **Richard Szeliski. 2014.** Concise Computer Vision: An Introduction into Theory and Algorithms. – United Kingdom: Springer London, 429 p.
3. **Brytik V., Grebinnik O., Kobziev V. 2016.** Research the possibilities of different filters and their application to image recognition problems. – Poland: ECONTECHMOD. An international quarterly journal, Vol. 5, No. 4, pp. 21–27.
4. **Ethem Alpaydin. 2010.** Introduction to Machine Learning. London: The MIT Press, 584p.
5. **Satya Mallick. 2016.** Image Recognition and Object Detection. Available online at: http://www. learnopencv.com/image-recognition-and-object-detection-part1/
6. **Ken Weiner. 2016.** Why image recognition is about to transform business. Available online at: https://techcrunch.com/2016/04/30/why-image-recognition-is-about-to-transform-business/
7. **John C. Russ, F. Brent Neal. 2015.** The Image Processing Handbook. United States of America: Florida CRC Press, 1035 p.
8. **Venmathi E. Ganesh, N. Kumaratharan. 2016.** Kirsch Compass Kernel Edge Detection Algorithm for Micro Calcification Clusters in Mammograms. Middle-East Journal of Scientific Research, 24 (4), pp. 1530–1535.
9. **Brytik V., Zhilina E., 2014.** Investigation possibilities of various filters which used in pattern recognition problems Bionica Intellecta, 2(83), pp. 88–95.
10. **Semenets V., Natalukha Yu., O. Taranukha, Tokarev V., 2014.** About One Method of Mathematical Modelling of Human Vision Functions. ECONTECHMOD. An international quarterly journal, Vol. 3, No. 3, pp. 51–59.
11. **Nick McClure. 2017.** TensorFlow Machine Learning Cookbook. Packt Publishing, 370 p.
12. **Tensorflow.** Image Recognition. Available online at: https://www.tensorflow.org/tutorials/image_recognition
13. **Michael Nielsen. 2017.** Using neural nets to recognize handwritten digits. Available online at: http://neuralnetworksanddeeplearning.com/chap1.html
14. **Michael Nielsen. 2017.** How the backpropagation algorithm works. Available online at: http://neuralnetworksanddeeplearning.com/chap2.html
15. **Michael Nielsen. 2017.** Improving the way neural networks learn. Available online at: http://neuralnetworksanddeeplearning.com/chap3.html

16. **Michael Nielsen. 2017.** Why are deep neural networks hard to train? Available online at: http://neuralnetworksanddeeplearning.com/chap5.html
17. The British Machine Vision Association and Society for Pattern Recognition. **2017**. What is computer vision? Available online at: http://www.bmva.org/visionoverview
18. **Gary Bradski, Adrian Kaehler. 2016.** Learning OpenCV 3 Computer Vision in C++ with the OpenCV Library. O'Reilly Media, 1024 p.
19. **Parker J. R. 2011**. Algorithms for Image Processing and Computer Vision. Wiley, 504 p.
20. **Simon J. D.** Prince. **2014**. Computer Vision: Models, Learning, and Inference. Cambridge University Press, 505 p.
21. **Giovanni Maria Farinella, Sebastiano Battiato, Roberto Cipolla. 2015**. Advanced Topics in Computer Vision. Springer Science & Business Media, 433 p.