



Author: Anne Menendez

Date: May 2001

Published by RTC Magazine, July 2001

Image recognition using neural networks

Imagine an image processor unaffected by lighting and optics...

For those of you familiar with machine vision, it is no news that the first key to success of an imaging application resides in its ability to obtain good quality images with the least possible lighting and contextual variations. If you take the example of a glossy object, the distribution of its color intensities will vary depending on the angle of incidence of the light and its wavelength. As a result, the color shades of the object will change as it moves or rotates. Coping with this problem implies the installation of optics, lighting and positioning equipment. The performance of the machine vision application depends on the accuracy and consistency of their calibration. The slightest change of hardware positioning affects the system. Nevertheless, once these settings are defined and frozen, one can start processing images to isolate regions of interest from background regions and measure objects for inspection or recognition. Repeatability and accuracy are achieved... until someone trips over a cable or moves your equipment.

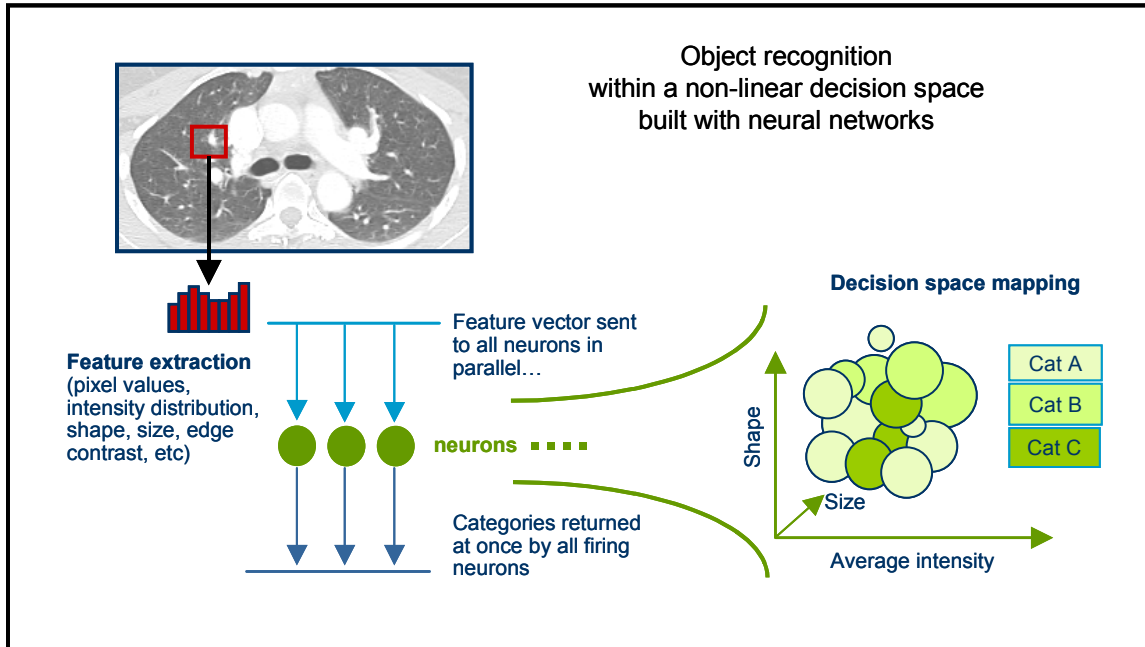
A neural network image processor can free applications from numerous constraints in terms of video acquisition, lighting and hardware settings. This degree of freedom is possible because a neural network lets you build an engine by learning examples. As a result, it can be trained to recognize good or bad parts under bright and dark lighting conditions, positioned in different orientations and so on. The more examples are learned, the more expert the engine and sometimes it is quite easy to automate the learning. In industrial inspection for example, some parts can be pre-sorted for the purpose of learning them per batches of a same category. The key to success is shifted from the cost of equipment to a number of images necessary to train and build a robust engine.

Imagine an image processor identifying patterns impossible to model mathematically...

For those of you familiar with image processing, it is no news that real-life imaging problems are non-linear. After spending the first 20% of a project on the separation of your main classes of objects, you are left with 80% of your efforts trying to accommodate the remaining population of objects. The latter either do not fall within the standard threshold intervals, or cannot be processed with standard filters without significant loss of information. For some applications, the patterns to be observed are so fuzzy that they cannot be modeled with conventional tools. A typical example would be a texture classifier for wood or paper grading where the samples have the same average chromaticity, no specific repetitive patterns but an overall consistency in graininess. A human eye can differentiate a dozen different textures but cannot express why with mathematical formulas. Another type of applications especially hard to model is scene monitoring where an alarm system has to detect a significant event in the field of view but where such event has to be more than a simple motion detection. For example, a surveillance camera at a bank teller could be trained to ring an alarm if a visitor appears with a mask on his head and/or a gun in his hands. On the other hand, the system should recognize if someone comes in with a hat and not sound the alarm.

A neural network image processor has the answer for such applications where diagnostics rely on the experience and expertise of an operator rather than models and algorithms from an engineer. The processor can build a recognition engine from simple image annotations made by the operator. It then extracts characteristics or feature vectors from the annotated objects and sends them to the neural network. Feature vectors characterizing visual objects can be as simple as raw pixel values, a histogram or distribution of intensities, intensity profiles along relevant axis or their gradients. More advanced features can include components from wavelet and FFT transforms. Once taught with examples, the neural network is capable of generalization and can consequently classify situations never seen before by associating them to similar learned situations. On the other hand, if an engine has the tendency to be too liberal and over-generalized situations, it can be corrected at any time by teaching him counter

examples. From a neural network standpoint, this operation consists of reducing the influence fields of the existing neurons to accommodate new examples which conflicts with the existing decision space mapping. In the event that several features are used to accurately identify a population of objects, each one of them contributes to the generation of a recognition engine or sub-engine. Their diagnostics can then be weighted and consolidated for final decision.



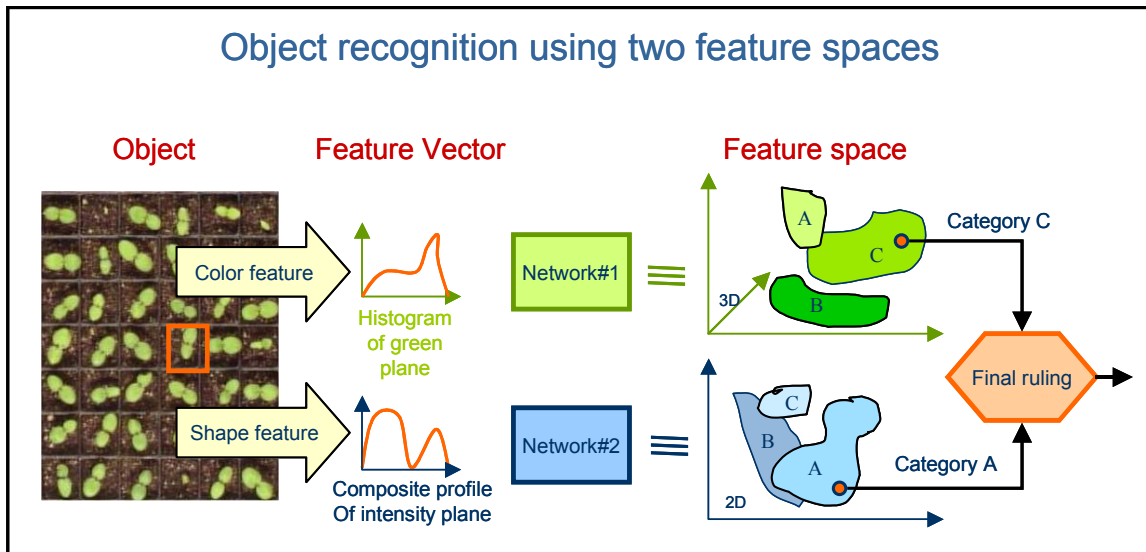
Imaging applications coping with contextual variations, fuzzy objects such as texture, surface inspection and anomaly detection can be implemented in a reasonable amount of time and with minimum programming thanks to neural network technology. Furthermore, if your fuzzy logic engine does not have too strong an ego and can admit to be uncertain or even ignorant, its humility can turn it into a powerful tool for anomaly detection and critical applications such as medical diagnosis and prediction maintenance. In such case, you are looking for signs of uncertainty to take upon a certain action (ring an alarm or else) or add the unknown situation to the engine if told to do so. The latter choice is appropriate for adaptive target tracking and allows the automatic re-learning of a target as it changes orientation and shape, and thus reinforces the accuracy of the engine as soon as it starts losing the target.

Imagine an image processor so smart and small that it can be embedded into sensors and devices...

For those of you familiar with imaging products, their evolution over the past decade has been to become faster, then more powerful, now smaller and soon very much cheaper. Speed increase came from faster CPU clocks and DSP technology. The power of imaging applications has risen with the introduction of advanced software tools and libraries. Today the trend is the embedded market so cameras can be installed everywhere starting on your desktop or hand-held computer, and spreading to cellular phones, toys and many more appliances. This evolution is possible thanks to the finer geometric resolution achieved in the semiconductor industry. However applications for embedded cameras are often limited to image capture and transmission. Neural network recognition engines can help miniature cameras go beyond the scope of a movie recorder and give them the intelligence that will attract large consumer markets.

For these markets, the key to acceptance is an unsupervised and adaptive learning. This means that the device must be capable of learning an object with a minimum of, or no, intervention from an operator. The dolls of the future will have the ability to learn the face of the child unwrapping them from the box and to ask her for her name and possibly her surrounding friends' names. An unsupervised learning for a cellular phone will consist of learning the fingerprint of its first-time owner. The owner's authentication can also be reinforced by combining face, fingerprint and voice recognition on the same device.

In the context of unsupervised learning, the device has to build on its own the recognition engine that will work best for its operating environment. For example, the intelligent doll has to recognize its first-time owner whatever her skin and hair color, the location and season of the year. At first the engine must use all the feature extraction techniques it knows. It will generate a series of sub-engines, each intended for the identification of the same categories of objects, but based on the observation of different features (color, graininess, contrast, edge density, etc). The global engine can then evaluate the sub-engines that give the best throughput and/or accuracy and consolidate their diagnostics according to the priorities of the application. The throughput and accuracy of an engine can be quantified by its ratio of positive identifications versus uncertain identifications and unknown identifications. If the cost of error of an application is low, the engine might be tuned to privilege the sub-engines with the highest throughput. If, on the other hand, accuracy is the most important, the sub-engines can be used in a very conservative manner such that the global engine only takes for an answer the one where all sub-engines are in agreement. Any disagreement between sub-engines would result in a negative identification.



Real-life implementation of an unsupervised and adaptive learning requires that the engine can clear and free neurons when they become irrelevant after a while. Neurons that were allocated at the beginning of the learning curve may end up being unused if the situation they described no longer occurs or if they belong to a sub-network that has poor performances. This cleaning process can be efficient under the condition that the active neurons remain unchanged and that the learning can just go on without starting from scratch. Adding high speed performance to that and real-time learning reinforcement is achieved.

Too good to be true?

Neural networks are the key to smart and complex vision systems for research and industrial applications. Hardware-based neural networks are the key to smart cameras and appliances for commercial and consumer products. Their ideal description is to behave like the human brain which is to learn whatever is new and to accept to be corrected if wrong. Both of these actions must not jeopardize the safety of the entire memory and the knowledge accumulated over time. There is no limit to the knowledge that can be accumulated as long as the instructor is consistent with what is being taught. The patterns to recognize can propagate in parallel through the entire network so the response time does not depend on the diversity of the problem. All these requirements are achievable today using silicon neural networks with parallel architecture. This last specification delivers ultra fast recognition rates, and allows adding neuronal chips to a network when needed without losing or deteriorating the existing knowledge. Among the neural network models implemented on such neuronal chips, the RBF (Radial basis Function) model is highly adaptive and ideal for non linear applications, while the KNN (K-Nearest neighbor) is more suitable for exact pattern matching. The price reduction of vision sensors and neuronal chips also favors the trend of replacing sophisticated imaging systems with several smart cameras which can share

General Vision News Release

the work and reduce the complexity of an application. All this considered, Big Brother can be replaced by an army of miniature intelligent vision sensors embedded in all kinds of devices and making your work and life easier and safer. The silicon neural network technology is now mature enough to stimulate a revolution in the imaging marketplace.