Neural Networks Add Speed and Intelligence to Image Recognition

Imaging systems that can learn by example can result in much more robust recognition for inspections, security, defense and even consumer applications. They can also use uncertainty to enhance their learning and ultimately their accuracy.

by: Anne Menendez, Imaging Expert
General Vision

For those familiar with machine vision, it's 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 the calibration of such components. 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 the 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 that can learn by example. 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 the engine learns, the more expert it becomes. Sometimes it's even quite easy to automate the learning. In industrial inspection, for example, some parts can be pre-sorted for the purpose of learning them in batches of a selected category. The key to success is then shifted from the cost of equipment to a number of images necessary to train and build a robust engine.

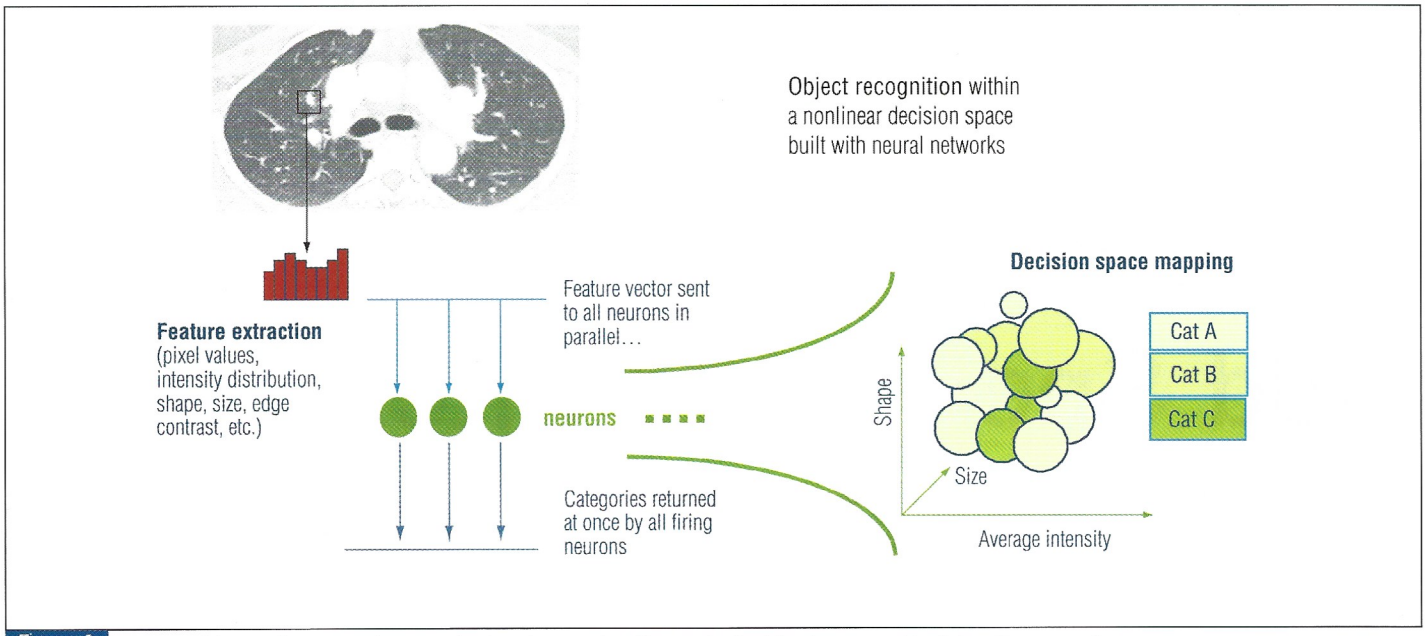

Figure 1 Recognition using neural networks involves extracting selected features and training the neural net, which learns the strengths of the parameters. When parameters from an object to be recognized are presented to the network and match the strengths of the learned parameters—even approximately—they can be mapped to decision space for classification to the degree of their similarity.

Identifying Patterns Impossible to Model Mathematically

For those familiar with image processing, it's no news that real-life imaging problems are nonlinear. 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. These remaining objects either don't fall within the standard threshold intervals or can't be processed with standard filters without significant loss of information. For some applications, the patterns to be observed are so fuzzy that they can't 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 can't express why with mathematical formulas. Another type of application 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 an event has to be more than a simple detection of motion. 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 on 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 (Figure 1).

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 overgeneralizes situations, it can be corrected at any time by teaching it counterexamples.

From a neural network standpoint, this operation consists of reducing the influence fields of the existing neurons to accommodate new examples that conflict with the existing decision space mapping. In the event that several features are used to identify accurately 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, the additional use of a fuzzy logic engine that can admit to uncertainty or even ignorance can be turned into a powerful tool for anomaly detection in critical applications such as medical diagnosis and prediction maintenance.

In such a case, you are looking for signs of uncertainty to trigger a certain action, such as ringing an alarm. Alternatively, encountering uncertainty might add the unknown situation to the engine if told to do so in order to refine learning. The latter choice is appropriate for adaptive target tracking because it allows the

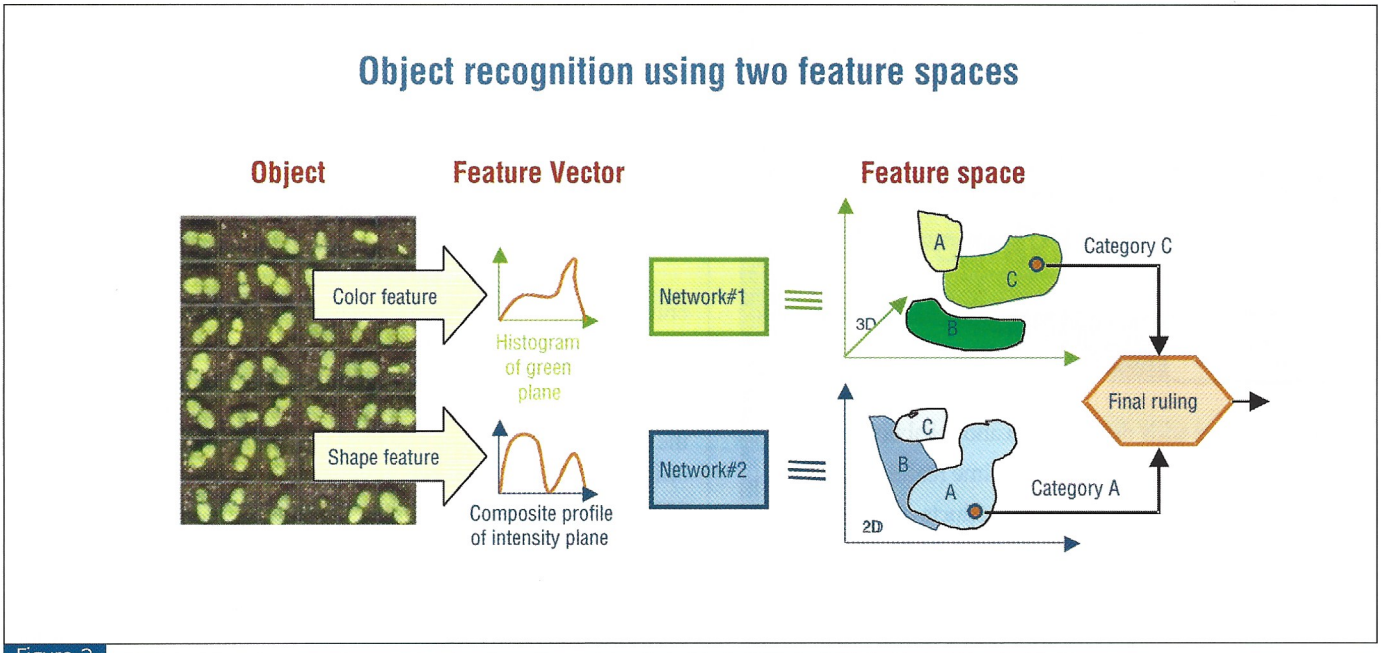

Figure 2 Training two (or more) neural subnets with parameters from different feature spaces such as color and shape allows a global neural engine to compare the most similar parameters of each feature space and make a final ruling.

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.

Smart Imaging Embedded Into Sensors and Devices

The evolution of imaging products over the past decade has become faster, then more powerful, now smaller and soon will be very much cheaper. Speed increase has come from faster CPU clocks and DSP technology. The power of imaging applications has also 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 handheld 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 such markets, the key to acceptance is unsupervised and adaptive learning. This means that the device must be capable of learning an object with minimal 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 such as color, graininess, contrast and edge density. 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 (Figure 2).

Training two (or more) neural subnets with parameters from different feature spaces such as color and shape allows a global neural engine to compare the most similar parameters of each feature space and make a final ruling.

The throughput and accuracy of an engine can be quantified by its ratio of positive identifications to 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 the active neurons remain unchanged and the learning can just go on without starting from scratch.

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 means to learn whatever is new and to accept correction 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 doesn't depend on the diversity of the problem. All these requirements are achievable today using silicon neural networks with parallel architecture.

This last specification delivers ultrafast 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 Radial Basis Function (RBF) model is highly adaptive and ideal for non-linear applications, while the K-Nearest Neighbor (KNN) 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 that can share 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 can make your work and life easier and safer. The silicon neural network technology is now mature enough to stimulate a revolution in the imaging marketplace.

General Vision
Petaluma, CA.
(707) 765-6150.
[www.general-vision.com].