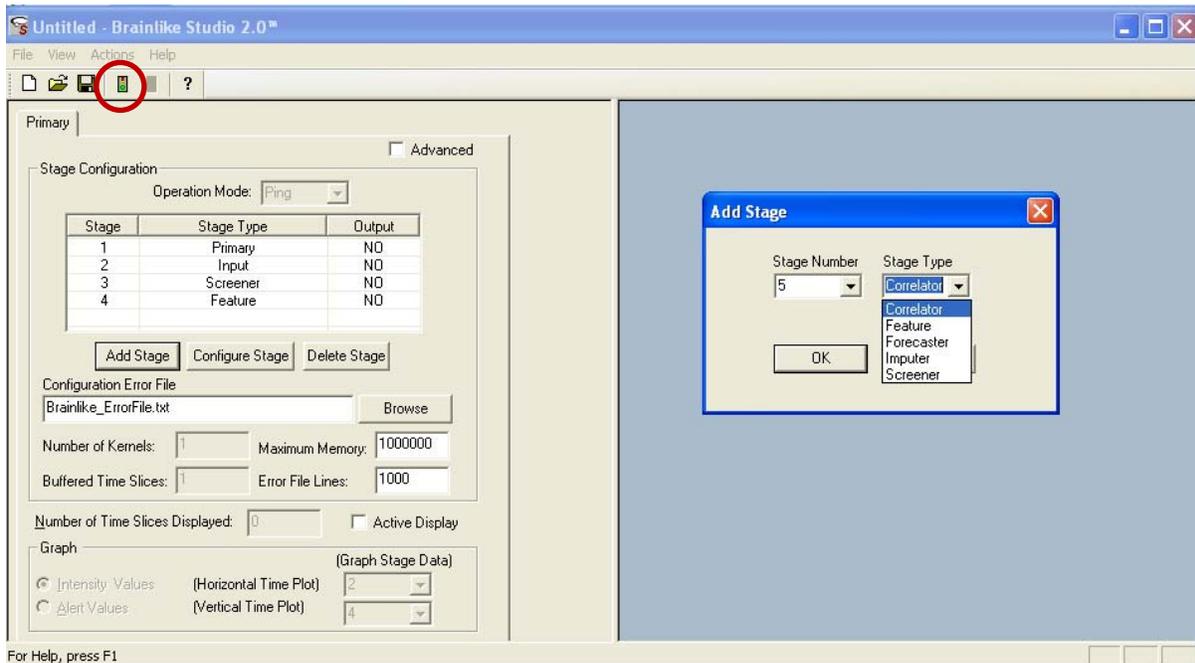


# A Brainlike Processing Illustration: Video Event Detection with General Application Utility

Brainlike, Inc.  
January, 2012

This report shows how Brainlike Studio™ toolkit and Brainlike Processor™ products can be configured to identify events of interest — automatically, adaptively, and in real-time. While the description below covers only one video application, it also outlines how Brainlike Studio™ and Brainlike Processor™ solutions can cover many other applications.

In the example below, a Brainlike Studio™ model has been configured to identify video change regions, based on pixels in a 240 by 320 video camera array. Similar Brainlike Studio™ models have been evaluated for video change detection and incorporated into [PixMin™ video reduction products](#).



**Figure 1. Brainlike Processing Example: Primary stage configuration.**

Figures 1 through 4 show four different views within the Brainlike Studio™ console. During configuration, analysts use the console to set metrics for a given problem and select historical data files for evaluation. Pressing the "run" button, shown inside the red circle in Figure 1, triggers three events. First, Brainlike Studio™ saves a configuration to a file. Second, it runs the configured model on historical data using a Brainlike Processor™ (BP) module. Third, it saves output data for testing and evaluation. After analysts have identified a satisfactory model based on the output data, they may integrate a corresponding BP module for real-time operation.

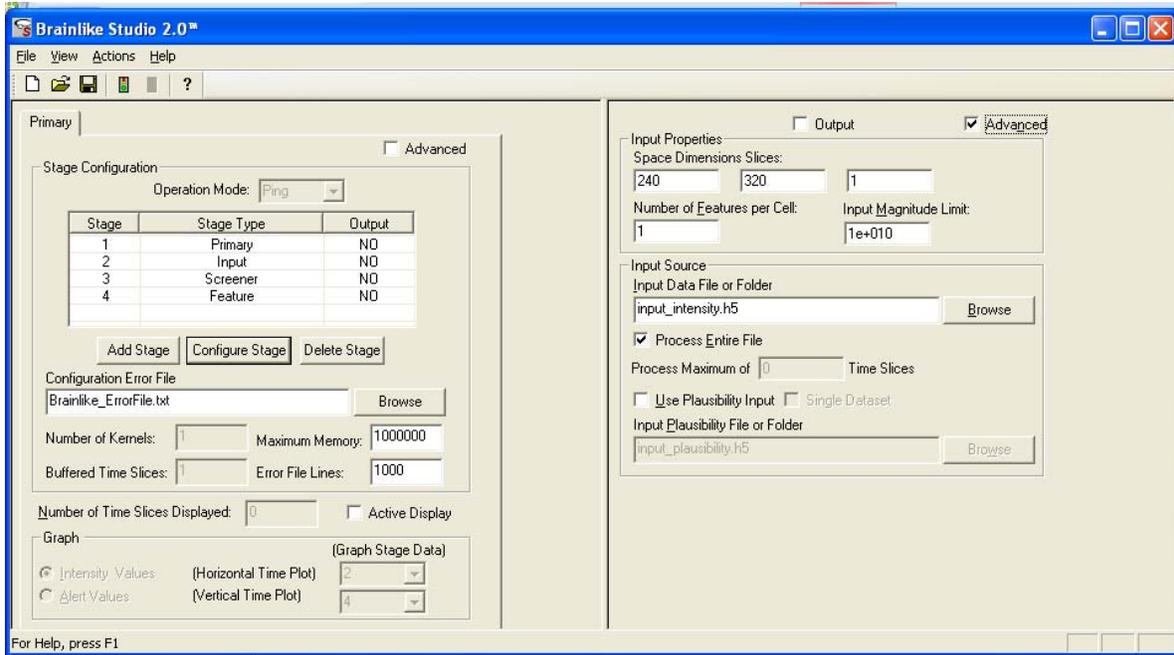


**Brainlike, Inc.**

[www.Brainlike.com](http://www.Brainlike.com)

Copyright 2005-2011 by Brainlike, Inc. All Rights Reserved

Figure 1 shows the "primary" stage view from the Brainlike Studio™ console. The table on the left shows that four stages have been selected. The pull-down menu on the right shows how these stages can be chosen during primary stage configuration. The next three figures show configuration details for each of the next three stages.



**Figure 2. Brainlike Processing Example: Input stage configuration.**

Figure 2 shows an input stage view from the Brainlike Studio™ console. As indicated, a 240 by 320 input data configuration has been selected. As a result, when the “run” button is pressed, Brainlike Processor™ will process all the 240 by 320 frames of data in the data input file, at the rate of one frame per “time slice.” As the figure shows, the configured input filename is “input\_intensity.h5.” The output intensity values from this stage are just like the input intensity values, but limited as necessary to prevent underflow/overflow.

Each Brainlike Processor™ operational stage may treat stage input and output values as missing or deviant. BP keeps track of missing or deviant values through the use of “plausibility” values. Each stage input value and output value has a corresponding plausibility value, between zero and one. Each BP input value may have input plausibility values that are either read from a file or set to zero by default. Since the “Use Plausibility Input” box in Figure 2 has not been checked, input plausibility values of zero have been configured. Output plausibility values from the input stage are identical to the stage input plausibility values, except when set to zero if its corresponding input value exceeds overflow/underflow bounds. In general, the intensity and plausibility output values from every stage are organized just like the input stage values.

In the PixMin™ case, a variety of other configurations have been used, including video configurations for three (RGB) input features per cell. Other models can be configured covering video and non-video applications, such three-dimensional (voxel) settings, as well as settings

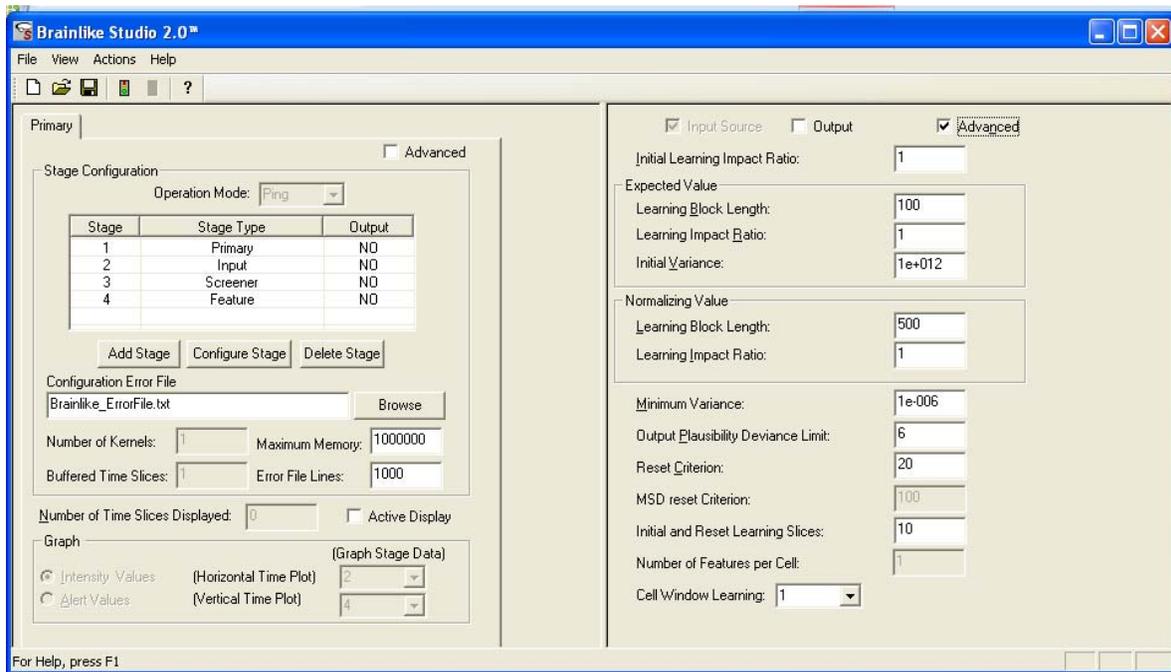


**Brainlike, Inc.**

[www.Brainlike.com](http://www.Brainlike.com)

Copyright 2005-2011 by Brainlike, Inc. All Rights Reserved

where many feature values for each pixel come from multiple sensors such as radar, infra-red, and other detectors.



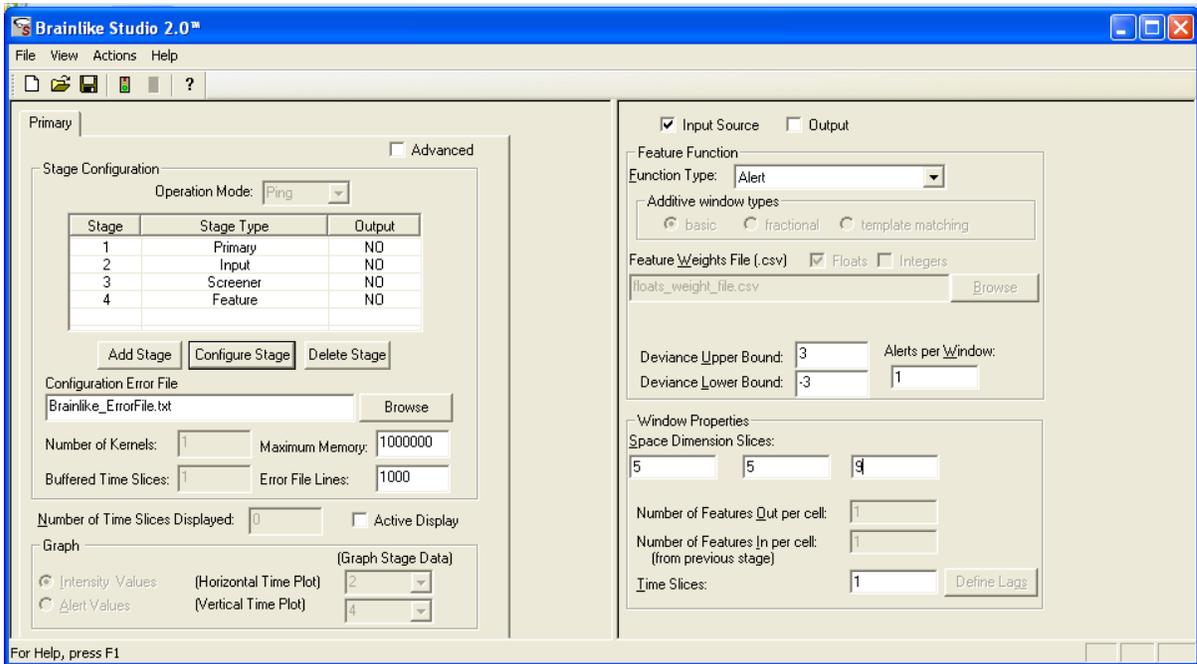
**Figure 3. Brainlike Processing Example: Screener stage configuration.**

Figure 3 shows configuration for a “screener” stage. Screener stages convert input feature values to standardized values, based on continuously and recursively updated metrics. In the process, the screener allows BP to continuously adapt to changing conditions. Resulting event detection is far more robust, with fewer false alerts and higher event detection rates than would otherwise be possible.

The screener receives input values from the stage before it, which in this case is the input stage. The screener stage supplies output “deviance” values to the next stage in one-to-one correspondence to the screener input values. The magnitude of each deviance value indicates how unusual its corresponding input value was, relative to recently learned trends. Each output plausibility value is the same as its corresponding input plausibility value, except it is zero if its corresponding deviance magnitude is higher than the configured "Output Plausibility Deviance Limit." The screener stage uses its plausibility values to flag anomalous input values as deviant. Screener plausibility values also determine if and how corresponding input values will influence learning.

During each time slice, the screener normalizes each input intensity value based on learned metrics, which are updated before the end of the time slice. Updating rates depend on “learning blocks” as shown in Figure 3. For example, the configured expected value learning block value is set to 100, as shown in figure 3. As a result, expected values are updated so that at any given frame or time slice, half of all prior learning occurred during the last 100 times slices; half of the remaining prior learning occurred during 100 the time slices before then; and so on.





**Figure 4. Brainlike Processing Example: Feature stage configuration.**

Figure 4 shows a feature stage configuration view. The pull-down menu on the right shows that the alert stage is one of several feature stage types that BP can run. In this case, stage input values, that is screener deviance values from the last stage, which deviate by more than plus or minus 3 are counted in a five by five window, centered at each of the pixel locations (except those on the edges). If the number of deviant values exceeds 9, its corresponding output value is returned as a one. Otherwise, it is returned as a zero. Alternative input layouts can be configured along these lines for other feature and alert applications.

At the end of each time slice, BP receives a vector containing output values from each stage, along with pointers to the beginning of each stage's output. In the PixMin™ case, change detection call function simply uses the alert stage output intensity values to identify change regions. This simple alert configuration works well in general as a basis for identifying pixels that change under non-stationary background conditions. It works well mainly because of the screener stage, which continuously and robustly learns what to expect.

In summary, this report has illustrated how the Brainlike Studio™ tool kit can be used to configure Brainlike Processor™ models. The example has shown how resulting applications such as [PixMin™ data reduction products](#) may be configured, tested, and evaluated. While it has outlined only one special case, the Brainlike Studio™ and Brainlike Processor™ may be applied to a broad variety of other cases. Key Brainlike Studio™ and Brainlike Processor™ advantages include its generality, ease of use, and robust performance. Underlying [Brainlike technology](#) has been developed over many years to add significant value, through its “auto-adaptive, enabling technology. [Patented Brainlike technology](#) has been delivered as valuable products, carefully designed to cover a broad variety of applications, as described at [www.Brainlike.com](http://www.Brainlike.com).

