Parallel compute *Wdist-8* for the selected multiple files

*path = handles.pushbutton2.file\_paths*

*num\_files =* Count the number of files

**Parallel.For** *g* = 1 to*num\_files*

*data =* Read the data of the *g*th *path*

*num\_rows =* Compute the number of rows of *data*

*num\_columns =* Compute the number of columns of *data*

*num\_pixels = num\_rows* times *num\_columns*

**If (**both*num\_rows* and *num\_columns* are greater than hundred**)**

Prompt user program is calculating, please wait

**End**

*num\_class\_data* = Count the number of class in *data*

*tabulate\_data* (1, *i) =* Count the number of pixels of the *i*th class in *data*

**For** *i* = 1 to *num\_class\_data*

*ln\_data +=* [(ln1), (ln2), ……, ln (*tabulate\_data* (1, *i*))]

**End**

*num\_class\_ln\_data* = Count the number of class in *ln\_data*

*tabulate\_ ln\_data* (1, *j*) *=* Count the number of the (ln*j*) in *ln\_data*

*fre\_ tabulate\_ ln\_data* (1, *j*) *=* Compute the frequency of the (ln*j*) in *tabulate\_ ln\_data*

**For** *j* = 1 to *num\_class\_ln\_data*

*w\_class +=* (ln*j*) times the frequency of the (ln*j*)

**End**

**For** *i* = 1 to *num\_class\_data*

*derive\_data =* Set the value of the *i*th class in *data* to 1 and those of other classes in *data* to 0

Search continuous space with a value of 1 in *derive\_data* according to eight-neighbor connectivity

**For** *p* = 1 to the number of continuous spaces with a value of 1

*tabulate\_ derive\_data* (1, *p) =* Count the number of pixels of the *p*th continuous space in *derive\_data*

*ln\_ derive\_data +=* [(ln1), (ln2), ……, ln (*tabulate\_ derive\_data* (1, *p*)]

**End**

**End**

*num\_class\_ln\_ derive\_data* = Count the number of class in *ln\_ derive\_data*

*tabulate\_ ln\_ derive\_data* (1, *q*) *=* Count the number of the (ln*q*) in *ln\_ derive\_data*

*fre\_ tabulate\_ ln\_data* (1, *q*) *=* Compute the frequency of the (ln*q*) in *tabulate\_ ln\_ derive\_data*

**For** *h* = 1 to *num\_class\_ln\_ derive\_data*

*w\_space +=* (ln*h*) times the frequency of the (ln*h*)

**End**

**For** *k* = 1 to *num\_pixels*

*w\_normal +=* (ln*k*) times (1/ *num\_pixels*)

**End**

*Wdist-4 =* (1- *w\_class*/ *w\_normal*) times (1- *w\_space*/ *w\_normal*)

*handles.listbox2 = Wdist-8*

**End**