Subject: Re: Getting ROI data from an image
Posted by Brian Daniel on Fri, 30 Sep 2011 18:42:55 GMT
View Forum Message <> Reply to Message

On Sep 30, 1:50 pm, Rebecca Brown <rab4...@gmail.com> wrote:
> I need to extract hyperspectral data from an image ROI :
>
> The project I'm working on requires me to code strictly with IDL, so I
> cannot use ENVI calls in the final product. I am running unsupervised
> classification (K-Means), which returns a classification image with
> each pixel assigned a number from [0,..nClasses]. I want to use the
> classification image to subset the image based on ROI and get the mean
> spectra from each class. The functionality I need is probably the same
> as ENVI_GET_ROI_DATA(). My exact question is, how can I use the one-
> dimensional calls to capture ROI data from the hypersepctral data?
>
> samples = 320
> lines = 1000
> bands = 300
> classes = 15
>
> result = KMEANS(img, ITERATIONS = 4, NCLASSES = classes)
>
> FOR i = 0, nClasses-1 DO BEGIN
> loc = WHERE(result EQ i, count)
> IF count EQ 0 THEN CONTINUE
>
> ; ?
>
> ; ?
>
> ENDFOR

Look at using the HISTOGRAM function on the result variable with
REVERSE_INDICES keyword.

http://www.idlcoyote.com/tips/histogram_tutorial.html

-Brian

Subject: Re: Getting ROI data from an image
Posted by David Fanning on Fri, 30 Sep 2011 19:19:21 GMT
View Forum Message <> Reply to Message

Brian J. Daniel writes:

> Look at using the HISTOGRAM function on the result variable with

> REVERSE_INDICES keyword.
>
> http://www.idlcoyote.com/tips/histogram_tutorial.html

Using the Reverse_Indices keyword can often seem complicated. You can use the ReverseIndices function from the Coyote Library to return the indices to you without having to learn the complicated vector in a vector syntax:

http://www.idlcoyote.com/programs/reverseindices.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Getting ROI data from an image
Posted by Rebecca on Fri, 30 Sep 2011 19:41:00 GMT
View Forum Message <> Reply to Message

OK, I can see how using the histogram w/ reverse indices will help quickly get the indices associated with each ROI. That's a great help, I wouldn't even need a FOR loop.
h = HISTOGRAM(result, MIN = 0, MAX = classes, NBINS = classes+1, REVERSE_INDICES=ri)

But this doesn't answer the problem I was having, which perhaps I didn't speak to as directly as I wanted- or perhaps you both are simply more versed at array indexing than I am! With HISTOGRAM or WHERE, it returns a 1D index of a 2D array (result), but I need to pull hyperspectral data from a 3D array using those indexes. My hyperspectral 'img' array might be 320 x 1000 x 300, for example. I cannot simply call
temp = img[ri[ri[1]:ri[2]-1], *]
And get the data I need. How would this be accomplished instead? I suppose I could use the 1D subscripts and, looping through each spectral band, get the data one band at a time but this seems tedious when I would hope an elegant solution would be possible.

As I said, I want the mean spectra for each class (ROI) identified by K-Means.

Thanks for the help so far,
Rebecca

On Sep 30, 3:19 pm, David Fanning <n...@dfanning.com> wrote:
> Brian J. Daniel writes:
>> Look at using the HISTOGRAM function on the result variable with
>> REVERSE_INDICES keyword.
>
>> http://www.idlcoyote.com/tips/histogram_tutorial.html
>
> Using the Reverse_Indices keyword can often seem
> complicated. You can use the ReverseIndices function
> from the Coyote Library to return the indices to you
> without having to learn the complicated vector in a
> vector syntax:
>
> http://www.idlcoyote.com/programs/reverseindices.pro
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Getting ROI data from an image
Posted by David Fanning on Fri, 30 Sep 2011 20:41:29 GMT
View Forum Message <> Reply to Message

Rebecca Brown writes:

> OK, I can see how using the histogram w/ reverse indices will help
> quickly get the indices associated with each ROI. That's a great help,
> I wouldn't even need a FOR loop.
> h = HISTOGRAM(result, MIN = 0, MAX = classes, NBINS = classes+1,
> REVERSE_INDICES=ri)
>
> But this doesn't answer the problem I was having, which perhaps I
> didn't speak to as directly as I wanted- or perhaps you both are
> simply more versed at array indexing than I am! With HISTOGRAM or
> WHERE, it returns a 1D index of a 2D array (result), but I need to
> pull hyperspectral data from a 3D array using those indexes. My
> hyperspectral 'img' array might be 320 x 1000 x 300, for example. I
> cannot simply call
> temp = img[ri[ri[1]:ri[2]-1], *]

> And get the data I need. How would this be accomplished instead?

I think you need one more step:

```
s = Size(result, /DIMENSIONS)
h = Histogram(result, ....., REVERSE_INDICES=ri)
indices = ReverseIndices(ri, 1) ; As an example.
colrow = Array_Indices(s, indices, /DIMENSIONS)
x = Reform(colrow[0,*])
y = Reform(colrow[1,*])
temp = img[x,y,*]
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Getting ROI data from an image
Posted by Rebecca on Fri, 30 Sep 2011 21:16:01 GMT
View Forum Message <> Reply to Message

That's great, and it makes so much sense, but it doesn't seem like IDL obeys those laws of indexing. Using
temp = img[x,y,*]
produces an out of memory error. It's not hard to figure out why-
temp = img[x,y,0]
Produces a [npix, npix] array, where npix is the number of pixels referenced in 'indices'. What I was expecting to happen was a [npix] vector! IDL is playing by different rules here.

```
npix = N_ELEMENTS(indices)
z = INTARR(npix)
temp = img[x,y,z]
```

That produces the magical vector array I want. So, is there any way to play by these rules and grab 300 bands worth of data at once so I have a [npix, bands] array? Or should I give up the chase and just FOR loop it?

On Sep 30, 4:41 pm, David Fanning <n...@dfanning.com> wrote:

> Rebecca Brown writes:
>> OK, I can see how using the histogram w/ reverse indices will help
>> quickly get the indices associated with each ROI. That's a great help,
>> I wouldn't even need a FOR loop.
>> h = HISTOGRAM(result, MIN = 0, MAX = classes, NBINS = classes+1,
>> REVERSE_INDICES=ri)
>
>> But this doesn't answer the problem I was having, which perhaps I
>> didn't speak to as directly as I wanted- or perhaps you both are
>> simply more versed at array indexing than I am! With HISTOGRAM or
>> WHERE, it returns a 1D index of a 2D array (result), but I need to
>> pull hyperspectral data from a 3D array using those indexes. My
>> hyperspectral 'img' array might be 320 x 1000 x 300, for example. I
>> cannot simply call
>> temp = img[ri[ri[1]:ri[2]-1], *]
>> And get the data I need. How would this be accomplished instead?
>
> I think you need one more step:
>
> s = Size(result, /DIMENSIONS)
> h = Histogram(result, ..., REVERSE_INDICES=ri)
> indices = ReverseIndices(ri, 1) ; As an example.
> colrow = Array_Indices(s, indices, /DIMENSIONS)
> x = Reform(colrow[0,*])
> y = Reform(colrow[1,*])
> temp = img[x,y,*]
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Getting ROI data from an image
Posted by David Fanning on Fri, 30 Sep 2011 21:32:25 GMT
View Forum Message <> Reply to Message

Rebecca writes:

> That's great, and it makes so much sense, but it doesn't seem like IDL
> obeys those laws of indexing. Using
> temp = img[x,y,*]
> produces an out of memory error. It's not hard to figure out why-

> temp = img[x,y,0]
> Produces a [npix, npix] array, where npix is the number of pixels
> referenced in 'indices'. What I was expecting to happen was a [npix]
> vector! IDL is playing by different rules here.
>
> npix = N_ELEMENTS(indices)
> z = INTARR(npix)
> temp = img[x,y,z]
>
> That produces the magical vector array I want. So, is there any way to
> play by these rules and grab 300 bands worth of data at once so I have
> a [npix, bands] array? Or should I give up the chase and just FOR loop
> it?

Actually, you are doing this correctly by making
your own index vector. See the latter half of
this article:

http://www.idlcoyote.com/misc_tips/submemory.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Getting ROI data from an image
Posted by Rebecca on Mon, 03 Oct 2011 13:15:54 GMT
View Forum Message <> Reply to Message

OK, thanks for the help on understanding IDL's subscripting!

On Sep 30, 5:32 pm, David Fanning <n...@dfanning.com> wrote:
> Rebecca writes:
>> That's great, and it makes so much sense, but it doesn't seem like IDL
>> obeys those laws of indexing. Using
>> temp = img[x,y,*]
>> produces an out of memory error. It's not hard to figure out why-
>> temp = img[x,y,0]
>> Produces a [npix, npix] array, where npix is the number of pixels

>> referenced in 'indices'. What I was expecting to happen was a [npix]
>> vector! IDL is playing by different rules here.
>
>> npix = N_ELEMENTS(indices)
>> z = INTARR(npix)
>> temp = img[x,y,z]
>
>> That produces the magical vector array I want. So, is there any way to
>> play by these rules and grab 300 bands worth of data at once so I have
>> a [npix, bands] array? Or should I give up the chase and just FOR loop
>> it?
>
> Actually, you are doing this correctly by making
> your own index vector. See the latter half of
> this article:
>
> http://www.idlcoyote.com/misc_tips/submemory.html
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
