Subject: Re: Newbie to IDL needs help :)
Posted by David Fanning on Thu, 09 May 2013 22:44:29 GMT
View Forum Message <> Reply to Message

alchemymetalworks@gmail.com writes:

> Ok, this is probably simple and I'm just missing something obvious.

Yeah, probably. ;-)

> I essentially have 4 maps that were created in ENVI (remote sensing software, if anyone's familiar with it). The maps are the exact same size/pixel count. Each pixel in each map is assigned a value from 1 - 15 based on the land cover that is (assumed)present. I want to create a composite map from these 4, using a set of rules:
>
> -if the value is the same in all the maps, keep that value in the final output map

OK.

> -if the value is 1, 2, 4-9, or 13 use the pixels with that value from map 1
> -if the value is 3, 8, 12, 14, or 15 use the pixels with that value
from map 2
> -if the value is 10 use the pixels with that value from map 3
> -if the value is 11 use the pixels with that value from map 4
> -if a pixel meets several of these criteria (for example, pixel P is value 1 in map 1 and value 10 in map 3)I want to designate which map should take priority.

What "value" are you talking about when you say "if the value"? Are you
talking about a value in a particular image?

I think you are having trouble writing the algorithm (assuming it CAN be
written, which seems doubtful to me at the moment) because you can't say
in words that are free from ambiguity what it is you actually want.

Your last criteria seems to me to contradict all that has come before,
but maybe I just don't know what "if the value" means.

Cheers,

David

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

Subject: Re: Newbie to IDL needs help :)
Posted by Jeremy Bailin on Thu, 09 May 2013 23:09:21 GMT
View Forum Message <> Reply to Message

On 5/9/13 5:25 PM, alchemymetalworks@gmail.com wrote:
> Ok, this is probably simple and I'm just missing something obvious. I'm basically trying to self-teach myself some IDL to complete a project at work, and since I've spent the last 2 days fighting the urge to throw my keyboard through my monitor I decided to put aside my pride and beg for help from the masses/experts... so here goes...
>
> I essentially have 4 maps that were created in ENVI (remote sensing software, if anyone's familiar with it). The maps are the exact same size/pixel count. Each pixel in each map is assigned a value from 1 - 15 based on the land cover that is (assumed)present. I want to create a composite map from these 4, using a set of rules:
>
> -if the value is the same in all the maps, keep that value in the final output map
> -if the value is 1, 2, 4-9, or 13 use the pixels with that value from map 1
> -if the value is 3, 8, 12, 14, or 15 use the pixels with that value from map 2
> -if the value is 10 use the pixels with that value from map 3
> -if the value is 11 use the pixels with that value from map 4
> -if a pixel meets several of these criteria (for example, pixel P is value 1 in map 1 and value 10 in map 3)I want to designate which map should take priority.
>
> The goal is to ultimately have a single output with all pixels classified (again) as 1-15 based on the preferred map(s), if that makes sense. And I'm stumped... I know what I want to do, but I can't figure out how to tell the computer to do it... Any help/guidance would be greatly appreciated.
>

I agree with David that I'm kind of getting lost, but I think this might
do what you're looking for. There's probably a more elegant way of
dealing with the masks, but this should work.

I am assuming that your maps are called map1...map4.

```
; create masks of which pixels you want from each map
mask_for_map1 = (map1 eq 1) or (map1 eq 2) or (map1 eq 4) or $
   (map1 eq 5) or (map1 eq 6) or (map1 eq 7) or (map1 eq 8) or $
   (map1 eq 9) or (map1 eq 13)
mask_for_map2 = (map2 eq 3) or (map2 eq 8) or (map2 eq 12) or $
   (map2 eq 14) or (map2 eq 15)
mask_for_map3 = (map3 eq 10)
mask_for_map4 = (map4 eq 11)

; clobber lower priority ones in the case where a pixel is part
; of the mask of a higher priority map. In this case, I am arbitrarily
; assuming map3 takes precedence, followed by maps 1, 4, 2 to show
; you the structure of the code.
mask_for_map1 *= (1-mask_for_map3)
mask_for_map4 *= (1-mask_for_map3)*(1-mask_for_map1)
```

```
mask_for_map2 *= (1-mask_for_map3)*(1-mask_for_map1)*(1-mask_for_map4)

combined_map = map1*mask_for_map1 + map2*mask_for_map2 + $
  map3*mask_for_map3 + map4*mask_for_map4
```

-Jeremy.

Subject: Re: Newbie to IDL needs help :)
Posted by alchemymetalworks on Fri, 10 May 2013 02:08:53 GMT

View Forum Message <> Reply to Message

Thanks for the ideas/suggestions. Let me give you a little more detail to see if that helps clear up the confusion. I get in the habit of not really going in-depth on what I do, people's eyes tend to glaze over (but I love my job). I am pretty sure what I'm looking for can be done as I've seen it referenced in several research papers as well as on the ENVI forums, but everyone just refers to 'building the IDL code' for it, nobody 'dumbs it down' so that I can figure the code out...

I am looking to do what could be called a hybrid classification. I use ENVI to evaluate satellite images and classify the land cover that is found in the images (based on vegetation spectral signatures and other factors). However, there are several different algorithms that can be used for the classification, each with certain strengths/weaknesses, so each provides a different final map. One algorithm may be great at one class of land (say, evergreen forest) and suck at another class (say, soybeans) and another algorithm may capture soybeans but not be great at forest, so neither map is as accurate as you would like overall. Hybrid classification processes look at ways to combine the good classes from the different maps generated by different algorithms into another 'final' map that has improved overall accuracy in all the classes. So I'm looking for a way to code the process so that if I tell it which map and which class is 'best' from all my (4) different maps, it builds the final map from those inputs. The final criteria probably isn't worded very well (and may not even be an issue) but since we are talking thousands of pixels each classified independent of the other maps, it is feasible that a pixel could fall in the 'best' class on 2 different maps and I'd need a 'tie breaker'.

I will definatly look into the suggested mask code when I get to work tomorrow, but if this (long winded) explaination helps and/or brings up any other suggestions please let me know.

Subject: Re: Newbie to IDL needs help :)
Posted by David Fanning on Fri, 10 May 2013 02:35:38 GMT

View Forum Message <> Reply to Message

alchemymetalworks@gmail.com writes:

> I am looking to do what could be called a hybrid classification.

I get the idea. I just don't get what you mean when you say

"if a value...". *What* value? In what image? How do you intend
to get started on this problem?

Cheers,

David

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

Subject: Re: Newbie to IDL needs help :)
Posted by alchemymetalworks on Fri, 10 May 2013 11:06:10 GMT
View Forum Message <> Reply to Message

On Thursday, May 9, 2013 9:35:38 PM UTC-5, David Fanning wrote:
> I get the idea. I just don't get what you mean when you say
>
> "if a value...". *What* value? In what image? How do you intend
>
> to get started on this problem?

Sorry, I misunderstood what part was confusing. Maybe I'm not using the right terms (I need IDL for Dummies). The 'values' I'm referring to are the class assignments (which are in numerical format in ENVI) that range from 1 through 15 (1 is soybeans, 2 is corn, 3 is idle grassland, etc.). The classes (values) are the same for all the classes in each map but the locations of the individual pixels in each map with those values/classes can vary.

So I already have 4 maps that are classified with the same 'number' scheme (I think of it kinda like a large array, but not sure if that's technically the correct term) and I want to 'merge' them by designating which map's class assignment (what I was calling the value) takes priority. In reading through what I can find on the internet in regards to IDL coding, 'if/then' and 'case' statements kinda sounded like what I needed, but I can't figure it out, and neither of them seems exactly right.

I want to create a set of assignment 'rules' to establish the pixel values in the final (initially blank) output map. Something like 'if the target value is 1 (soybeans) refer to map 1 (good for soybeans), on the 'final' map give all pixels that share the same location information with soybeans (value 1) pixels from map 1 a value of 1 to indicate soybeans. And then continue this for each class, indicating the class number and the map that I want to use as the 'rule'.

I have been able to figure out how to write a bit of code that looks at all 4 maps, compares the pixel 'values' and indicates which ones match/don't match across all 4 maps, which is a nice little victory for me, but I'd really prefer to set the priority decision rules and create a merged map. Another idea down the road (if I could figure out IDL) would be to look at all 4 maps, if the values match across all 4 assign that value, if the values don't match across all 4 then use a set of priority rules, etc. But...baby steps...

Does that make it clearer or am I just making no sense whatsoever? I (thankfully) have other classification and/or software options I can explore if I can't figure this out in the allotted time, but I'd really like to compare a couple of different 'final' maps and see how the different techniques compare from an accuracy standpoint on the type of landscape I'm working on.

Subject: Re: Newbie to IDL needs help :)
Posted by David Fanning on Fri, 10 May 2013 15:09:10 GMT

View Forum Message <> Reply to Message

alchemymetalworks@gmail.com writes:

> -if the value is the same in all the maps, keep that value in the final output map

I wouldn't both with this. I think what you do below will solve this problem anyway, and this just adds complications.

> -if a pixel meets several of these criteria (for example, pixel P
> is value 1 in map 1 and value 10 in map 3)I want to designate
> which map should take priority.

I don't know what this means either. I think you are already making choices, so I don't know what other choices you would want to make. I'd leave it until you did everything else, then see if there are problem areas that need more attention.

> -if the value is 1, 2, 4-9, or 13 use the pixels with that value from map 1
> -if the value is 3, 8, 12, 14, or 15 use the pixels with that value from map 2
> -if the value is 10 use the pixels with that value from map 3
> -if the value is 11 use the pixels with that value from map 4

I am going to assume the images are name i1-i4, for simplicity. We will call the final image "f". Here is how I would construct a composite image.

```
f = i1 * 0
void = Histogram(i1,binsize=1,min=0,max=15, reverse_indices=ri1)
void = Histogram(i2,binsize=1,min=0,max=15, reverse_indices=ri2)
void = Histogram(i3,binsize=1,min=0,max=15, reverse_indices=ri3)
void = Histogram(i4,binsize=1,min=0,max=15, reverse_indices=ri4)

indices = cgReverseIndices(ri1,1)
f[indices] = i1[indices]
```

```
indices = cgReverseIndices(ri1,2)
f[indices] = i1[indices]
indices = cgReverseIndices(ri2,3)
f[indices] = i2[indices]
indices = cgReverseIndices(ri1,4)
f[indices] = i1[indices]
indices = cgReverseIndices(ri1,5)
f[indices] = i1[indices]
indices = cgReverseIndices(ri1,6)
f[indices] = i1[indices]
indices = cgReverseIndices(ri1,7)
f[indices] = i1[indices]
indices = cgReverseIndices(ri2,8)
f[indices] = i2[indices]
indices = cgReverseIndices(ri1,9)
f[indices] = i1[indices]
indices = cgReverseIndices(ri3,10)
f[indices] = i3[indices]
indices = cgReverseIndices(ri4,11)
f[indices] = i4[indices]
indices = cgReverseIndices(ri2,12)
f[indices] = i2[indices]
indices = cgReverseIndices(ri1,13)
f[indices] = i1[indices]
indices = cgReverseIndices(ri2,14)
f[indices] = i2[indices]
indices = cgReverseIndices(ri2,15)
f[indices] = i2[indices]
```

Then, I would look to see if there were any holes and decide what to do next. :-)

Cheers,

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

Subject: Re: Newbie to IDL needs help :)
Posted by sulsoftbr on Wed, 15 May 2013 12:36:37 GMT

View Forum Message <> Reply to Message

I sent you an email, but why not post here, it's an interesting problem ...

There are 5 basic rules you defined, I numbered them from 1 to 5. First you have to create a mask for each of the rules:

Rule 1 : mask1 = (map1 eq map2) and (map1 eq map3) and (map1 eq map4)

Rule 2: mask2 = map1 eq 1 or map1 eq 2 or (map1 ge 4 and map1 le 9) or map1 eq 13

Rule 3: mask3 = map2 eq 3 or map2 eq 8 or map2 eq 12 or map2 eq 14 or map2 eq 14

Rule 4 : mask4 = map3 eq 10

Rule 5 : mask5 = map4 eq 11

Now,

```
  mixed = mask1 +2b*mask2 + 4b*mask3 + 8b*mask4 + 16b*mask5
  mixed = mixed *  ((mask1+mask2+mask3+mask4+mask5) ge 2)
```

is an image where non-zero pixel contain a bit code telling you for which classified pixels you met more than one criteria, for instance

```
  mixed[x,y] = 10
```

would tell you that pixel [x,y] has either value 1,2,4,5,6,7,8,9 or 13 in map1, and value 10 in mask4 -   how you go about that is now your decision. For all pixels with value 0 and 1 in "mixed" you can already fill your output classification result:

```
   result = (map1*mask1 + map1*mask2 + map2*mask3 + map3*mask4 + map4*mask5 )   *
(mixed le 1)
```

The "several criteria met" pixels in result will be zero , and you will have to fill these in according to the priority you've choosen.

(Obs: this all assumes you don't have unclassified (value 0) pixels in map 1 through 4)

cheers,

Michael

Em sexta-feira, 10 de maio de 2013 08h06min10s UTC-3, alchemym...@gmail.com escreveu:
> On Thursday, May 9, 2013 9:35:38 PM UTC-5, David Fanning wrote:
>
>> I get the idea. I just don't get what you mean when you say
>

>>

>

>> "if a value...". *What* value? In what image? How do you intend

>

>>

>

>> to get started on this problem?

>

>

>

>

>

> Sorry, I misunderstood what part was confusing. Maybe I'm not using the right terms (I need IDL for Dummies). The 'values' I'm referring to are the class assignments (which are in numerical format in ENVI) that range from 1 through 15 (1 is soybeans, 2 is corn, 3 is idle grassland, etc.). The classes (values) are the same for all the classes in each map but the locations of the individual pixels in each map with those values/classes can vary.

>

>

>

> So I already have 4 maps that are classified with the same 'number' scheme (I think of it kinda like a large array, but not sure if that's technically the correct term) and I want to 'merge' them by designating which map's class assignment (what I was calling the value) takes priority. In reading through what I can find on the internet in regards to IDL coding, 'if/then' and 'case' statements kinda sounded like what I needed, but I can't figure it out, and neither of them seems exactly right.

>

>

>

> I want to create a set of assignment 'rules' to establish the pixel values in the final (initially blank) output map. Something like 'if the target value is 1 (soybeans) refer to map 1 (good for soybeans), on the 'final' map give all pixels that share the same location information with soybeans (value 1) pixels from map 1 a value of 1 to indicate soybeans. And then continue this for each class, indicating the class number and the map that I want to use as the 'rule'.

>

>

>

> I have been able to figure out how to write a bit of code that looks at all 4 maps, compares the pixel 'values' and indicates which ones match/don't match across all 4 maps, which is a nice little victory for me, but I'd really prefer to set the priority decision rules and create a merged map. Another idea down the road (if I could figure out IDL) would be to look at all 4 maps, if the values match across all 4 assign that value, if the values don't match across all 4 then use a set of priority rules, etc. But...baby steps...

>

>

>

> Does that make it clearer or am I just making no sense whatsoever? I (thankfully) have other classification and/or software options I can explore if I can't figure this out in the allotted time, but I'd really like to compare a couple of different 'final' maps and see how the different techniques

compare from an accuracy standpoint on the type of landscape I'm working on.