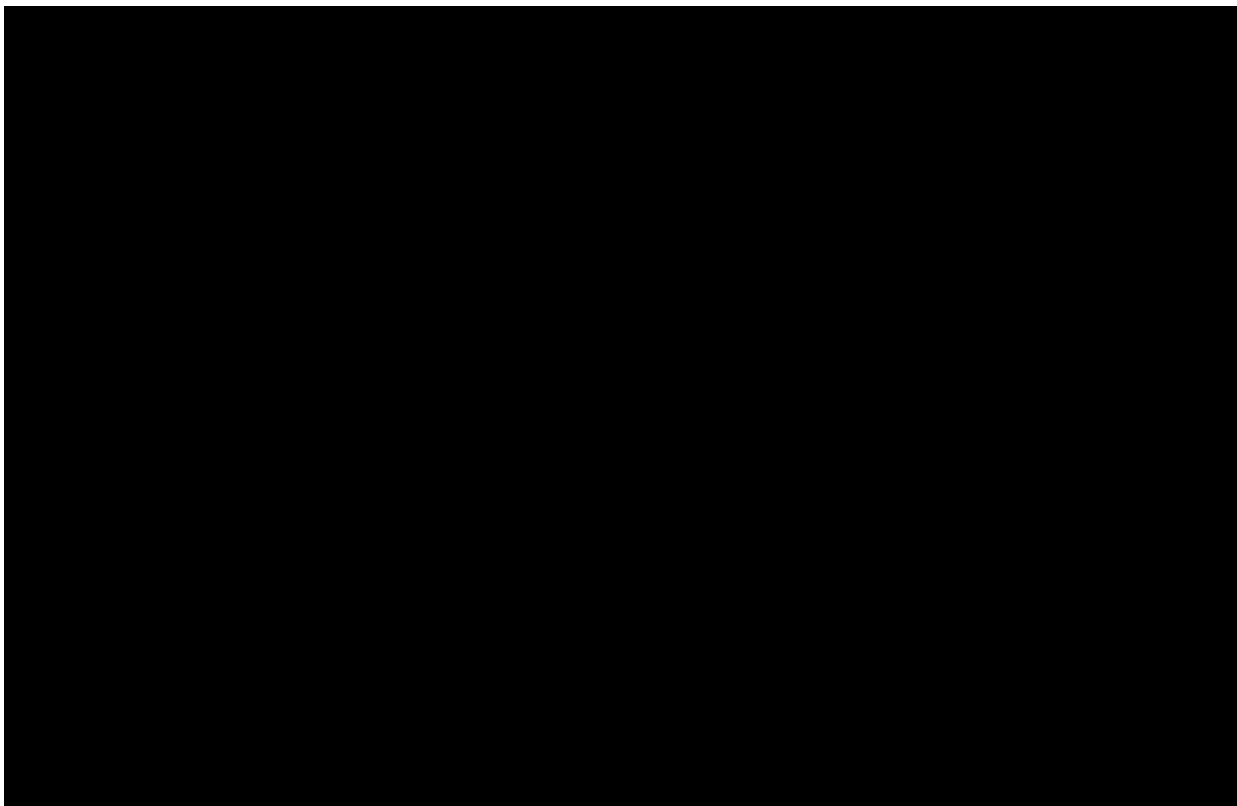


Image Track

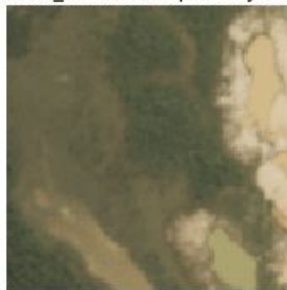


Subject	Site	Type	Volume_cc	Comment	individualID
10	3	0	15.11		MGH_WBMRI_patient_010
10	1	0	24.02		MGH_WBMRI_patient_010
10	1	0	31.64		MGH_WBMRI_patient_010
10	8	0	17.46		MGH_WBMRI_patient_010
10	1	0	132.69	> 5CM	MGH_WBMRI_patient_010
10	2	0	9.45		MGH_WBMRI_patient_010
10	2	0	46.97	> 5CM	MGH_WBMRI_patient_010
10	3	0	36.38		MGH_WBMRI_patient_010
10	3	0	2.25		MGH_WBMRI_patient_010
10	7	0	48.22	> 5CM	MGH_WBMRI_patient_010
19	0	1	35.6		MGH_WBMRI_patient_019
19	2	0	14.5		MGH_WBMRI_patient_019
19	3	0	2.34		MGH_WBMRI_patient_019

clear;primary



artificial_mine;clear;primary;water



agriculture;clear;cultivation;habitation;primary;road



agriculture;clear;primary;road;water



cloudy



partly_cloudy;primary



agriculture;partly_cloudy;primary



agriculture;clear;road



haze;primary



What are the questions NF doctors want help on?

Volume, Volume, and Volume

3D-3D object detection needs 3D Convnet

```
: def conv3d(ni:int, nf:int, ks:int=3, stride:int=1, pad:int=1, norm='batch'):
    bias = not norm == 'batch'
    conv = faiv.init_default(nn.Conv3d(ni,nf,ks,stride,pad,bias=bias))
    conv = spectral_norm(conv) if norm == 'spectral' else \
        weight_norm(conv) if norm == 'weight' else conv
    layers = [conv]
    layers += [nn.ReLU(inplace=True)] # use inplace due to memory constraints
    layers += [nn.BatchNorm3d(nf)] if norm == 'batch' else []
    return nn.Sequential(*layers)

def res3d_block(ni, nf, ks=3, norm='batch', dense=False):
    """ 3d Resnet block of `nf` features """
    return faiv.SequentialEx(conv3d(ni, nf, ks, pad=ks//2, norm=norm),
                             conv3d(nf, nf, ks, pad=ks//2, norm=norm),
                             faiv.MergeLayer(dense))

: norm = 'batch'
layers = ([res3d_block(1,15,7,norm=norm,dense=True)] +
          [res3d_block(16,16,norm=norm) for _ in range(4)] +
          [conv3d(16,1,ks=1,pad=0,norm=None)])
model = nn.Sequential(*layers)
```

Converting stacks of 2D images into NII format

```
!dcm2niiX /home/jupyter/tutorials/image/WBMRI023/DICOM/
```

Chris Rorden's dcm2niiX version v1.0.20190902 GCC7.3.0 (64-bit Linux)

Found 40 DICOM file(s)

Warning: Siemens MoCo? Bogus slice timing (range -1..-1, TR=4190 seconds)

Convert 20 DICOM as /home/jupyter/tutorials/image/WBMRI023/DICOM/DICOM_SPINE_20070720114638_16 (322x1105x20x1)

Warning: Siemens MoCo? Bogus slice timing (range -1..-1, TR=454 seconds)

Convert 20 DICOM as /home/jupyter/tutorials/image/WBMRI023/DICOM/DICOM_SPINE_20070720114638_22 (322x1102x20x1)

Conversion required 0.081513 seconds (0.081448 for core code).

Get rid of missing masks/scans

```
[42]: print(len(img_list))
```

48

```
[15]: print(len(label_list))
```

51

```
[16]: def Diff(li1, li2):  
        return (list(set(li1) - set(li2)))  
        print(Diff(label_list, img_list))
```

['036.nii', '032.nii', '052.nii']

```
[17]: for element in Diff(label_list, img_list):  
        cleaned_label_list=label_list.remove(element)
```


Matching size of scans and masks

```
import SimpleITK as sitk
for element in img_list[1:]:
    file=os.path.join('/home/jupyter/tutorials/image/nii/b',element)
    nda = sitk.GetArrayFromImage(sitk.ReadImage(file))
    print(sitk.ReadImage(file).GetSize())
    print(nda.shape)
    nda = nda[1:18,0:894,1:322]
    img = sitk.GetImageFromArray(nda)
    writer = sitk.ImageFileWriter()
    writer.SetFileName(file)
    writer.Execute(img)
```