

# PA5

November 25, 2016

## 1 Nima Aghli

### 1.1 PA5

#### 1.1.1 DagSeg

```
In [1]: %matplotlib inline
        #from pylab import *
        import matplotlib
        import matplotlib.pyplot as plt
        import numpy as np
        import cv2
        import matplotlib.image as mpimg
        from pylab import *
        import pandas as pd
        from scipy import misc
        import scipy.fftpack as fftim
        from scipy.misc.pilutil import Image
        from PIL import Image
        import cmath
        from joblib import Parallel, delayed
        import scipy.io as sio
        from pims import ImageSequence
        from skimage.segmentation import slic
        from skimage.color import rgb2lab
        from skimage import color
        from skimage.segmentation import mark_boundaries
        from skimage.util import img_as_float
        from sklearn.preprocessing import Normalizer
        from sklearn import metrics
        from sklearn.cluster import KMeans, MiniBatchKMeans
        from mpl_toolkits.mplot3d import Axes3D
        import matplotlib.pyplot as plt
        import warnings
        warnings.filterwarnings('ignore')
```

## 1.2 1-loading Frames And Masks Into Program

```
In [3]: images = ImageSequence('images//birdfall2_000*.png')
        masks = ImageSequence('masks//birdfall2_000*.png')

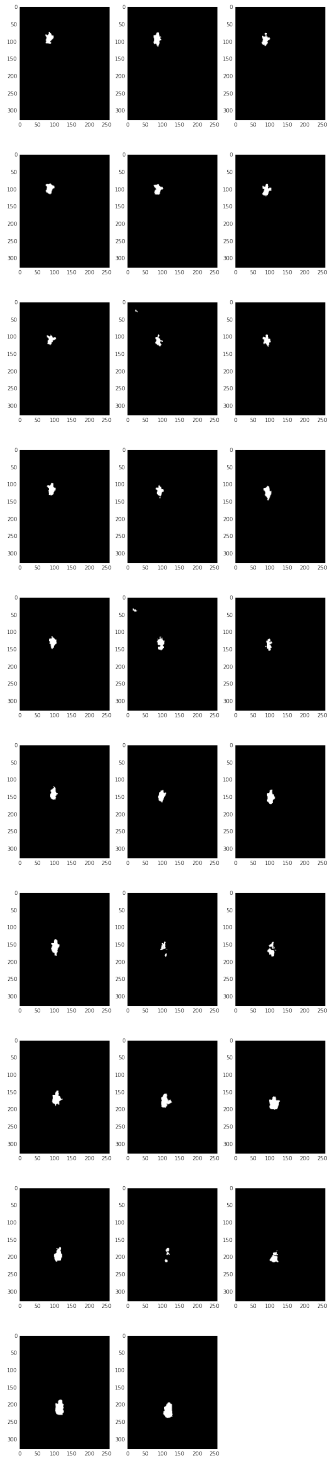
In [34]: framecount=29
         fig = plt.figure(figsize=(10, 140))
         fig.suptitle('20 Object Porposals For 20 Frames')
         for k in range(0,framecount):
             img=images[k]
             ax = fig.add_subplot(framecount, 3, k+1)
             ax.imshow(img)
```

20 Object Proposals for 20 Frames



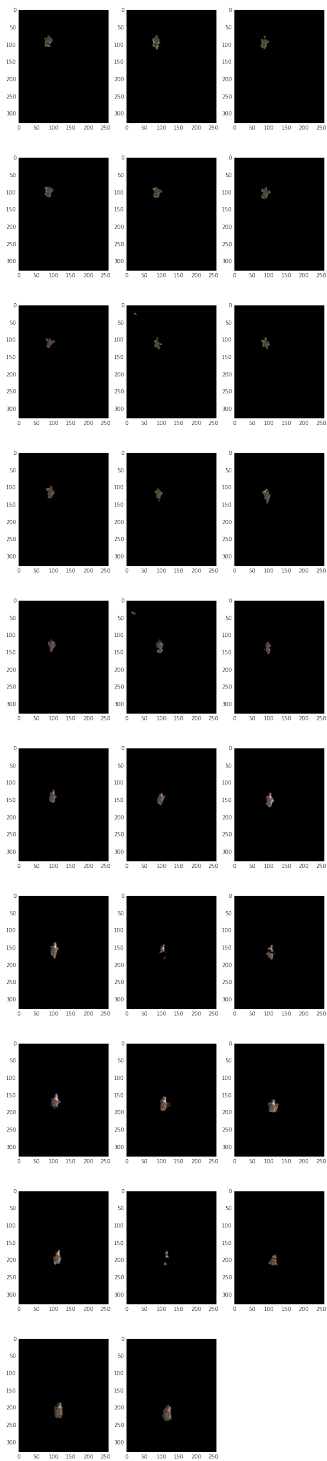
```
In [36]: fig = plt.figure(figsize=(10, 140))
fig.suptitle('Masks')
for k in range(0, framecount):
    img=masks[k]
    ax = fig.add_subplot(framecount, 3, k+1)
    ax.imshow(img, cmap = plt.get_cmap('gray'), vmin = 0, vmax = 255)
```

Masks



```
In [37]: fig = plt.figure(figsize=(10, 140))
fig.suptitle('Objects')
for k in range(0, framecount):
    img = cv2.bitwise_and(images[k], images[k], mask = masks[k])
    ax = fig.add_subplot(framecount, 3, k+1)
    ax.imshow(img)
```

Objects



```
In [ ]:
```