
Relative-Rotations

Unknown Author

April 3, 2015

Things I will try the PASCAL/Imagenet Setting: 1. Take bounding boxes that are just rescaled to constant 256 x 256
2. Try cropping the boxes carefully. 3. On Picking run the experiment of trying with different layers - how well the network generalizes. Create a pose embedding. Do visual reasoning in this domain. ie. two buses subtracted from each other + car shifts the car in the right way. Initially I was making an error that mirror flipping was on while predicting the relative rotations. Even with this I got some reasonable weights while training from scratch. They are plotted below.

```
In [19]: %matplotlib inline
%load_ext autoreload
%autoreload 2
codeDir = '/work4/pulkitag-code/pkgs/caffe-v2-2/modelFiles/keypoints/code/'
expDir = '/work4/pulkitag-code/pkgs/caffe-v2-2/modelFiles/keypoints/exp/'
snapshotDir = '/data1/pulkitag/snapshots/keypoints/'
import caffe
import os
import my_pycaffe as mp
import matplotlib.pyplot as plt
#import plotly.plotly as ply
#plotly.tools.set_credentials_file(username='pulkit.audacious', api_key='9cqniif4ai',
#Experiment and snapshot paths.
#Load the module
currDir = os.getcwd()
os.chdir(codeDir)
import process3d as p3d
os.chdir(currDir)
expStr = 'rotObjs_128_kmedoids30_20'
defFile = os.path.join(expDir, expStr, 'caffenet_siamese.prototxt')

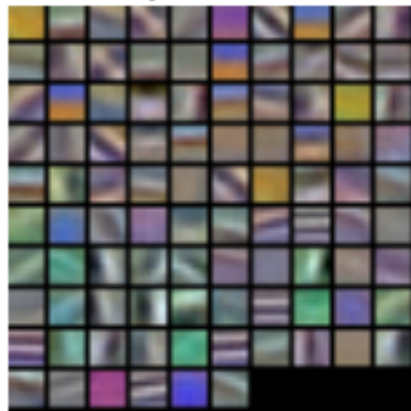
The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload
```

```
In [13]: #This network was trained by concatenating fc-6 features. Prior to concatenation t
numIterations = 60000
numIterations2 = 90000
imSz = 128
expStr = 'rotObjs_128_kmedoids30_20'
modelName = 'keypoints_siamese_scratch_iter_%d.caffemodel' % numIterations
defFile = os.path.join(expDir, expStr, 'keynet_siamese_deploy.prototxt')
modelFile = os.path.join(snapshotDir, 'exprotObjs_lblkmedoids30_20_imSz%d' % imSz,
net = mp.MyNet(defFile, modelFile)
#Network without dropouts on fc-6 features.
modelName2 = 'keypoints_siamese_scratch_nodrop_fc6_iter_%d.caffemodel' % numIterations
modelFile2 = os.path.join(snapshotDir, 'exprotObjs_lblkmedoids30_20_imSz%d' % imSz,
net2 = mp.MyNet(defFile, modelFile2)
fig = plt.figure()
ax1 = plt.subplot(1,2,1)
ax2 = plt.subplot(1,2,2)
net.vis_weights('conv1', ax=ax1, titleName='Dropout on fc6')
net2.vis_weights('conv1', ax=ax2, titleName='No Dropout on fc6')
ax1.axis('off')
ax2.axis('off')
```

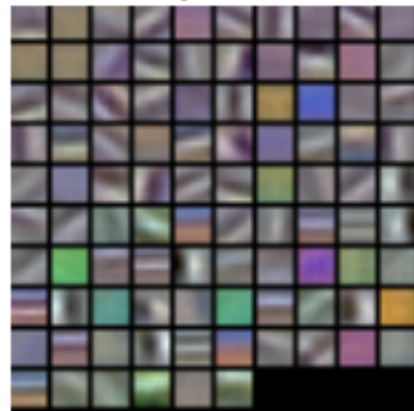
(-0.5, 79.5, 79.5, -0.5)

Out [13]:

Dropout on fc6



No Dropout on fc6



```
In []: #Define the experiment
In [17]: prms = p3d.get_exp_prms(imSz=128, lblType='uniform', numSamplesTrain=40000, numSamples
          azBin=30, elBin=10)

#Get the statistics of the training data.
#Training data is PASCAL 2012 + Entire Imagenet.
annCountTrain = p3d.get_class_statistics_exp(prms, 'train')
#Get the statistics of the val data
annCountVal = p3d.get_class_statistics_exp(prms, 'val')

In [18]: annCount = {'%s' % key: [val] for key, val in annCountTrain.iteritems()}
annCount = {'%s' % key: annCount[key] + [val] for key, val in annCountVal.iteritems()}
formatStr = "{:<12} {:<20} {:<20}"
print formatStr.format('Class', 'Train-BBox', 'Val-BBox')
print '-' * 100
for key, val in sorted(annCount.items()):
    bbTr, bbVl = val
    print ("%s" % formatStr).format(key, bbTr, bbVl)

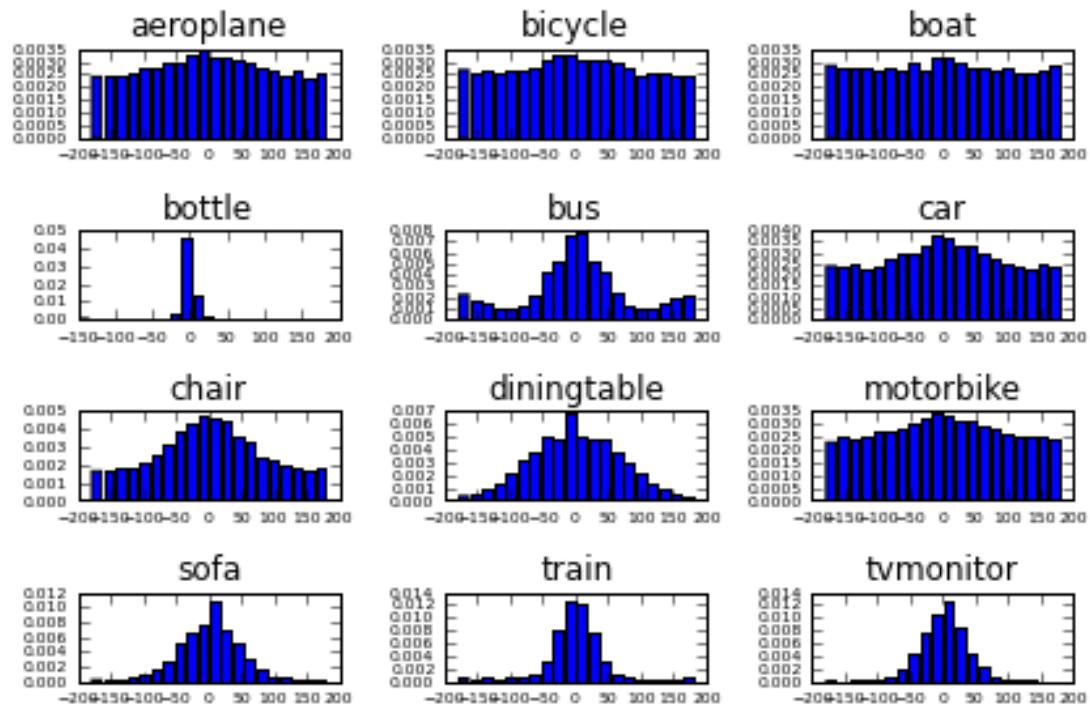
Class          Train-BBox          Val-BBox
-----
aeroplane      2314              361
bicycle         1656              340
boat            2489              265
bottle          1696              272
bus             1356              283
car             6194              594
chair           2059             1032
diningtable     2611              288
motorbike       1570              339
sofa            1749              294
train           1601              298
tvmonitor       1570              336

# Get statistics of pairwise labels
In [6]: lblType = 'diff'
numSamples = 20000
imSz = 128
cropType = 'contPad'
```

```
prms = p3d.get_exp_prms(imSz=imSz, lblType=lblType, numSamples=numSamples, cropT
pairLabels = p3d.get_pair_label_stats(prms, setName='train')
```

In [19]:

```
count = 1
for (key, val) in sorted(pairLabels.items()):
    ax = plt.subplot(4,3,count)
    azimuth = val[:,0]
    n, bins, patches = plt.hist(azimuth, 20, normed=1, histtype='bar', rwidth=0.8)
    plt.title(key)
    ax.tick_params(axis='x', labels=6)
    ax.tick_params(axis='y', labels=6)
    count += 1
plt.tight_layout()
```



In [30]:

```
#pairLabels = p3d.get_pair_label_stats(prms, setName='val')
count = 1
fig = plt.figure()
for (key, val) in sorted(pairLabels.items()):
    ax = plt.subplot(4,3,count)
    azimuth = val[:,1]
    n, bins, patches = plt.hist(azimuth, 20, normed=1, histtype='bar', rwidth=0.8)
    plt.title(key)
    ax.tick_params(axis='x', labels=6)
    ax.tick_params(axis='y', labels=6)
    count += 1
plt.tight_layout()
```



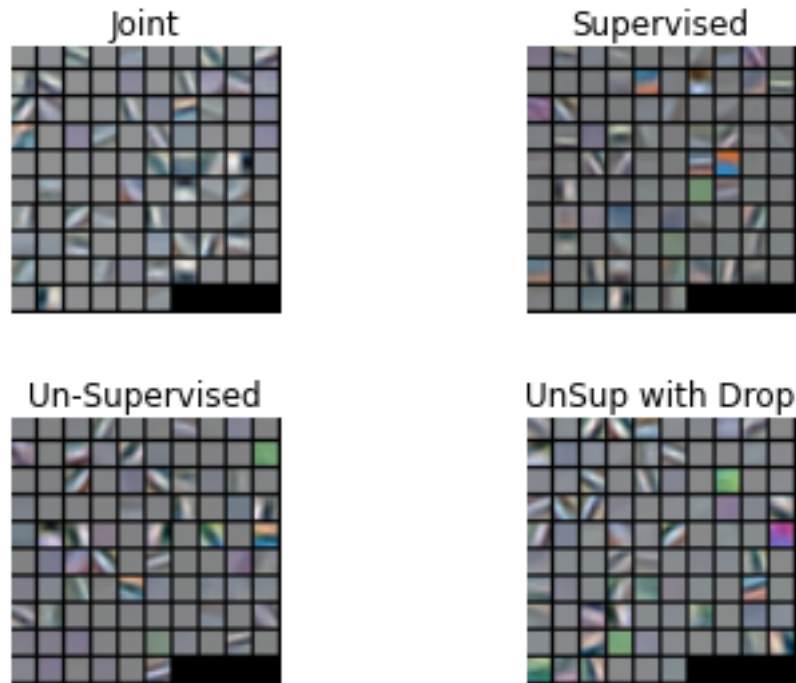
```

exp.init_from_self()
modelFile = exp.get_snapshot_name(numIter=numIter)
defFile = exp.files_['netdef']
netUnsup = mp.MyNet(defFile, modelFile)
print modelFile
netUnsup.vis_weights('conv1', ax=ax3, titleName='Un-Supervised')
ax3.axis('off')

#Get Unsupervised example.
cPrms = p3d.get_caffe_prms(isScratch=True, isClassLbl=False, concatLayer='fc6', noRot=
exp = p3d.get_experiment_object(prms, cPrms)
exp.init_from_self()
modelFile = exp.get_snapshot_name(numIter=numIter)
defFile = exp.files_['netdef']
netUnsup = mp.MyNet(defFile, modelFile)
print modelFile
netUnsup.vis_weights('conv1', ax=ax4, titleName='UnSup with Drop')
ax4.axis('off')
plt.tight_layout()

/data1/pulkitag/pascal3d/snapshots/imSz128_lbl-uni-az30e110_crp-
contPad16_ns4e+04_mb50/pascal3d_scratch_concat-
fc6_iter_50000.caffemodel
Ignoring line: # Autotmatically generated solver prototxt
/data1/pulkitag/pascal3d/snapshots/pascal3d_imSz128_lbl-uni-
az30e110_crp-contPad16_ns4e+04_mb50/caffenet_scratch_sup_noRot_fc6_ite
r_50000.caffemodel
Ignoring line: # Autotmatically generated solver prototxt
/data1/pulkitag/pascal3d/snapshots/pascal3d_imSz128_lbl-uni-
az30e110_crp-contPad16_ns4e+04_mb50/caffenet_scratch_unsup_fc6_iter_50
000.caffemodel
Ignoring line: # Autotmatically generated solver prototxt
/data1/pulkitag/pascal3d/snapshots/pascal3d_imSz128_lbl-uni-
az30e110_crp-contPad16_ns4e+04_mb50/caffenet_scratch_unsup_fc6_drop_it
er_50000.caffemodel

```



This is weird that the filters with k-Medoid clustering look so much better than filters obtained with the Euler angle uniform binning. The critical differences between the two approaches is as following:

1. First only considers small rotations (< 30 degrees) and bins all the other rotation into twenty categories.
2. First uses 0.005 weight std to initialize the FC layers, the latter uses 0.01 for initialization. This was done as with 0.005 init, the second case I was getting all zeros as output in the top layers.
3. First uses pascal only, whereas second used imagenet also.
4. There is a difference in classes that are being used. First uses a set of 15 manually chosen classes that are mostly rigid, 12 uses the 12 Rigid objects in pascal3d. The first includes the classes of person, sheep, bird, cow and horse that are ignored by the second. The second has bottle and diningtable which is not there in the first.

Possible things to do:

1. Compare accuracy on an auxiliary dataset - say Caltech101? Any suggestions?
2. The issue with using less data - should I use the same architecture as the one used by the unsupervised method - which has a lot more training data available? Is that fine?

In []: