Final Project

최신기술프로젝트


Information from Image

Get Information!

- Watching directly

Information from Image

Get Information!

- Watching directly
- The method does not support a lot of data.

Information from Image

Get Information!

- Create a log file

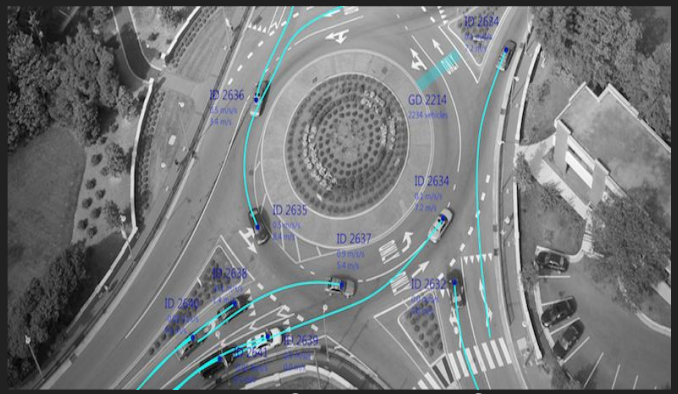
Information from Image

Get Information!

- Create a log file
- However, the method has a problem that missing necessary data or redundant data occurs.

Want Information From Image !

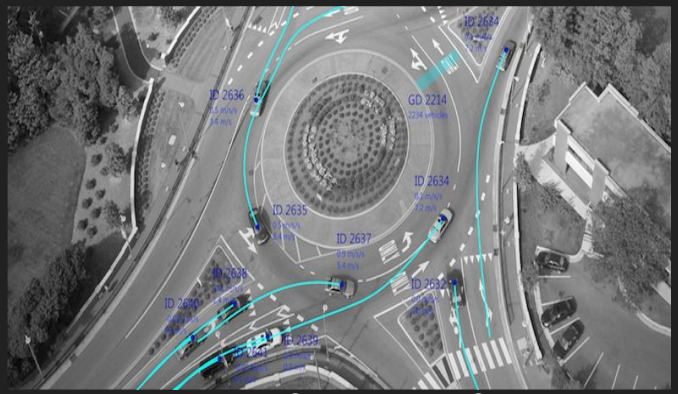
Get Information!



- Extract information from image

Want Information From Image !

Get Information!



- Extract information from image
- This method can extract only the information we need.

How to Get Information in Image ?

How to Get Information in Image ?

It is hard to adopt distribution algorithm, because image composed of a matrix

Ex) mapreduce(Key,Value) ,graph(node,edge)

How to Get Information in Image ?

Spark MLlib has image processing algorithm.
But spark performs navie image processing

Ex) random forest, Kmeans

Using Deep Learning

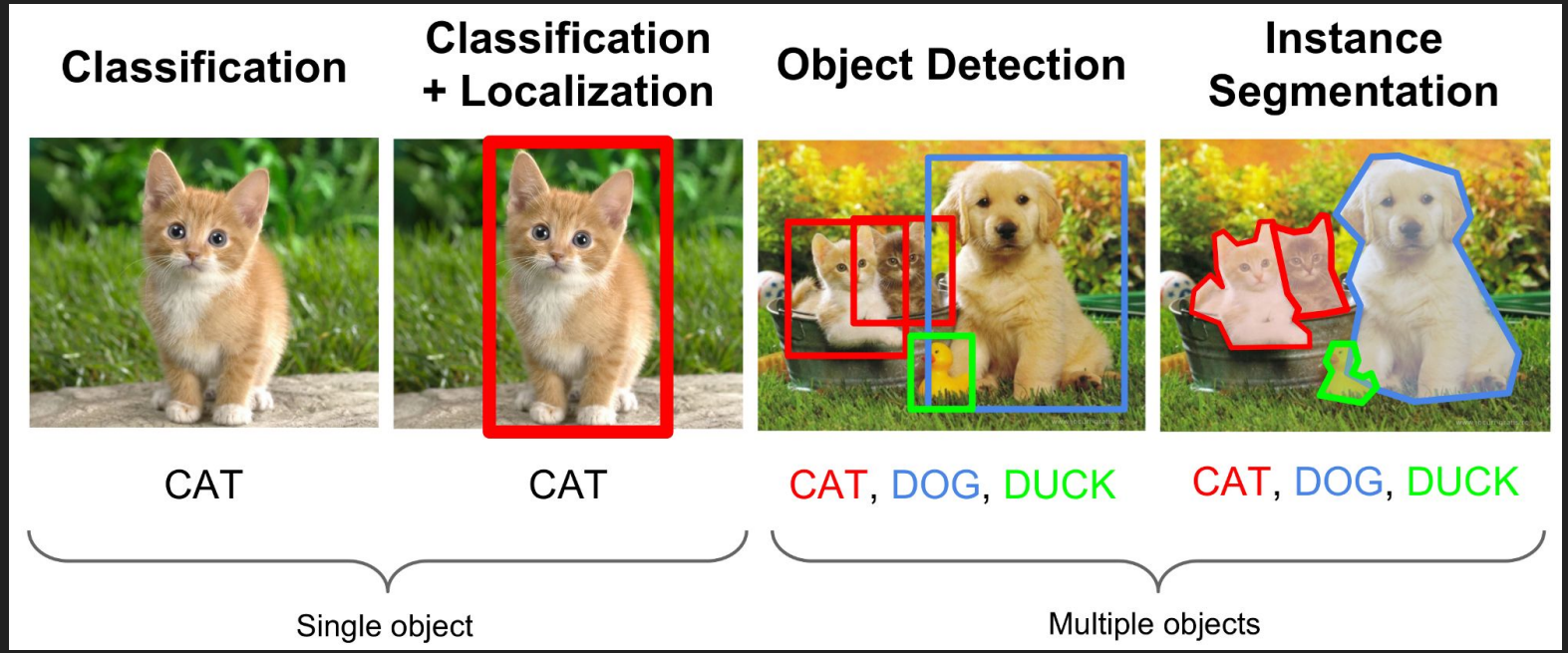
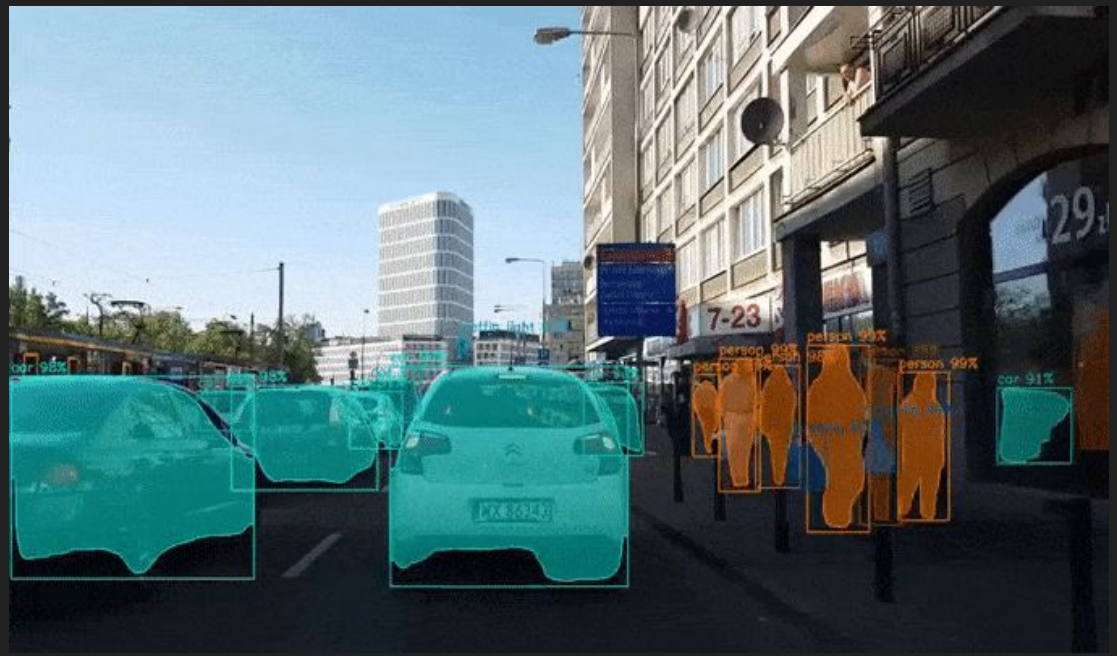


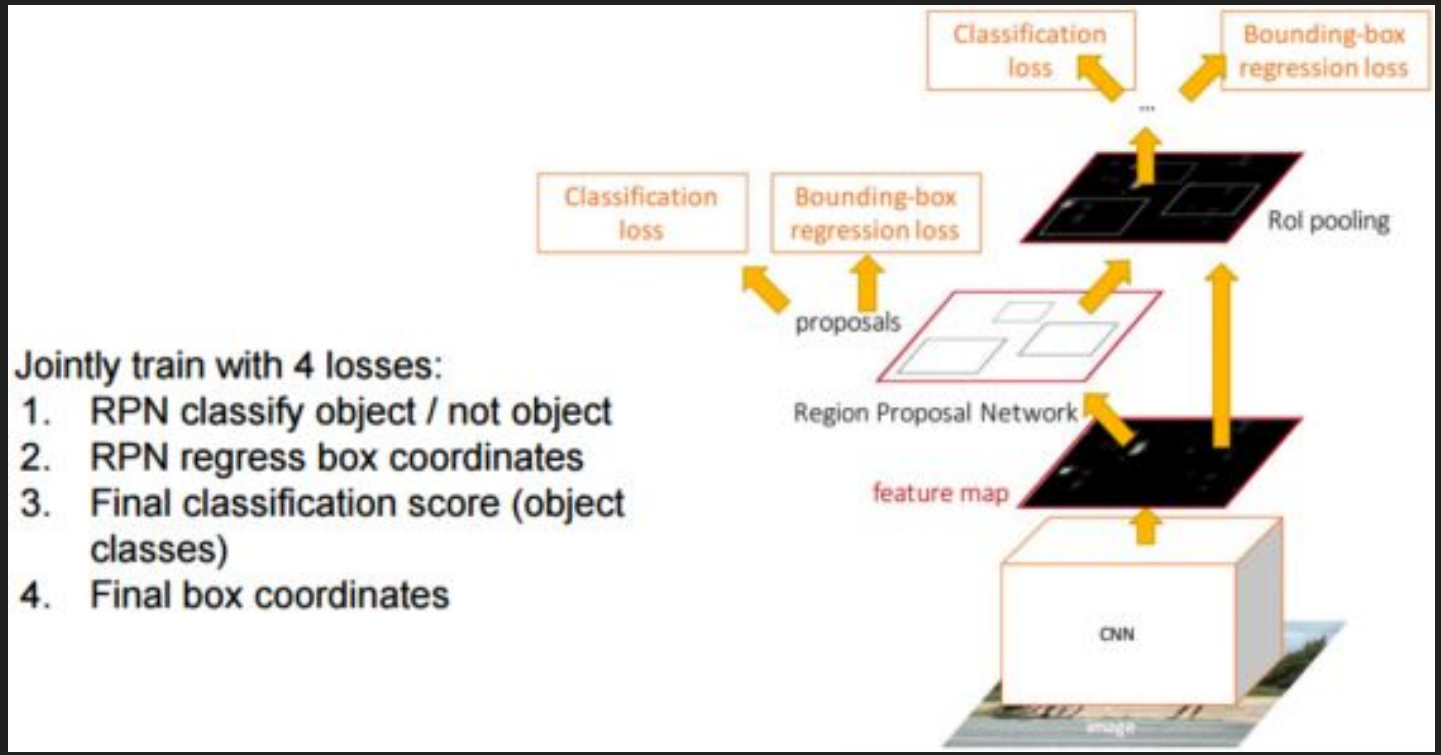
Image Segmentation!

Mask -RCNN

- Mask R-CNN is algorithm that performs image segmentation & detection
- It is composed of Faster RCNN & R-FCN



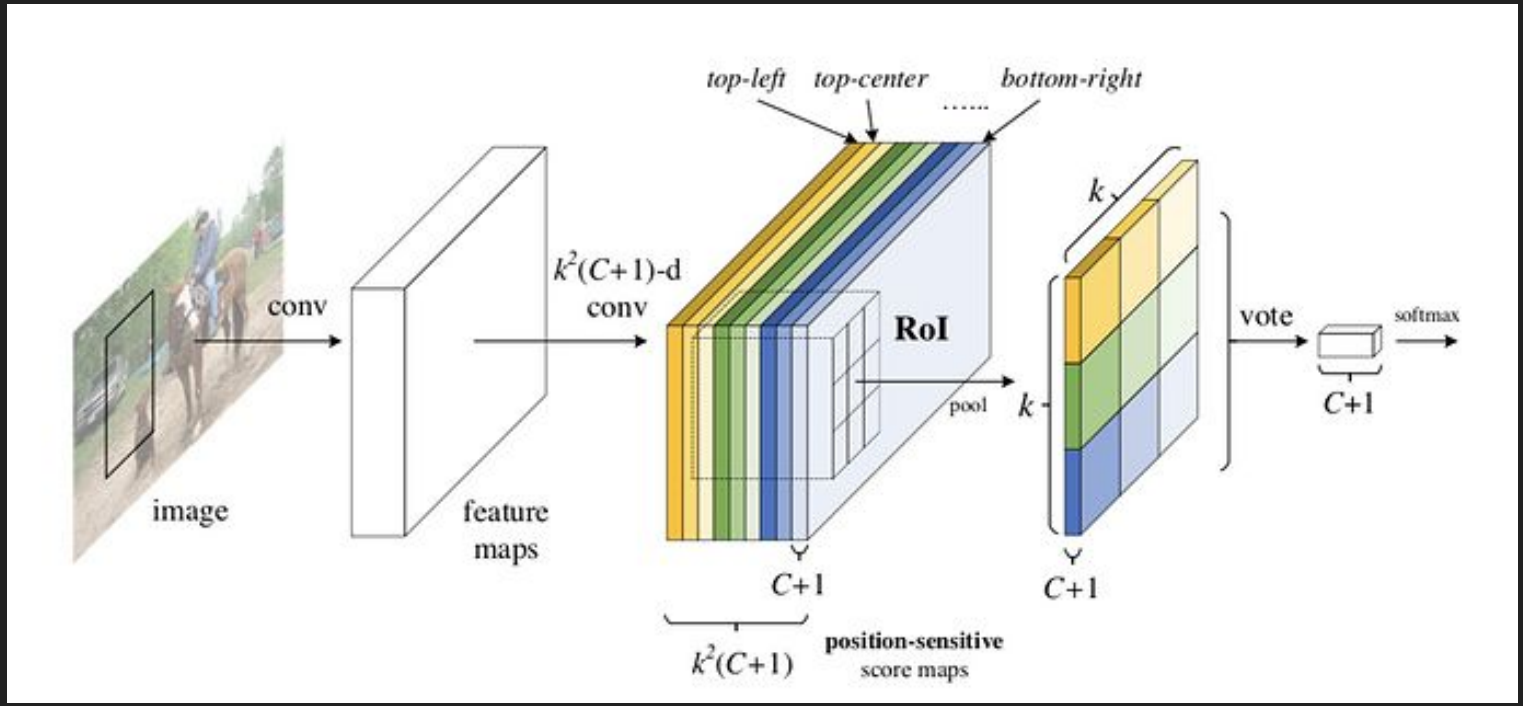
Faster R-CNN



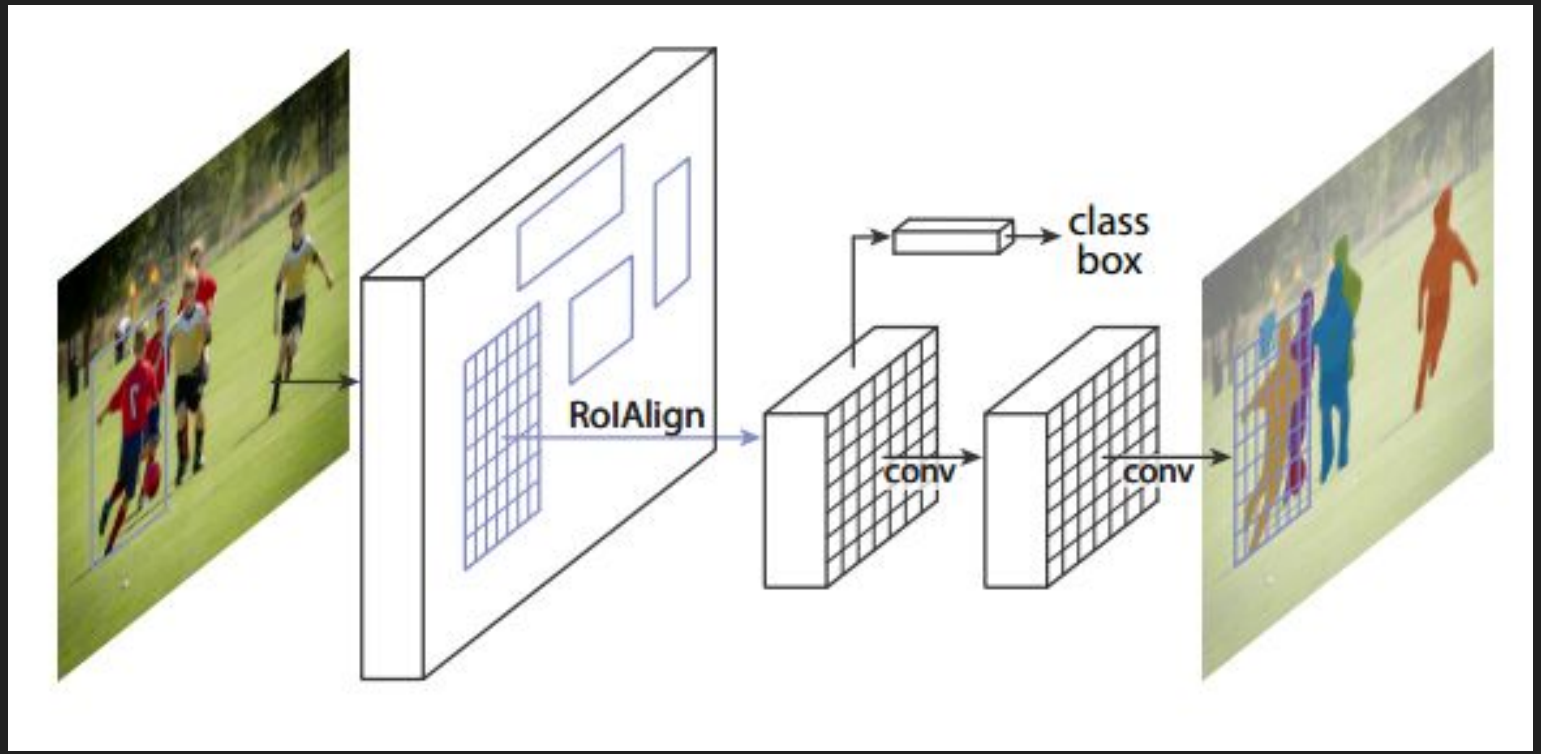
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

R-FCN


top-left
top-center
.....
bottom-right
conv
image
feature
maps
$k^2(C+1)$-d
conv
RoI
pool
k
k
vote
softmax
C+1
C+1
C+1
$k^2(C+1)$
position-sensitive
score maps

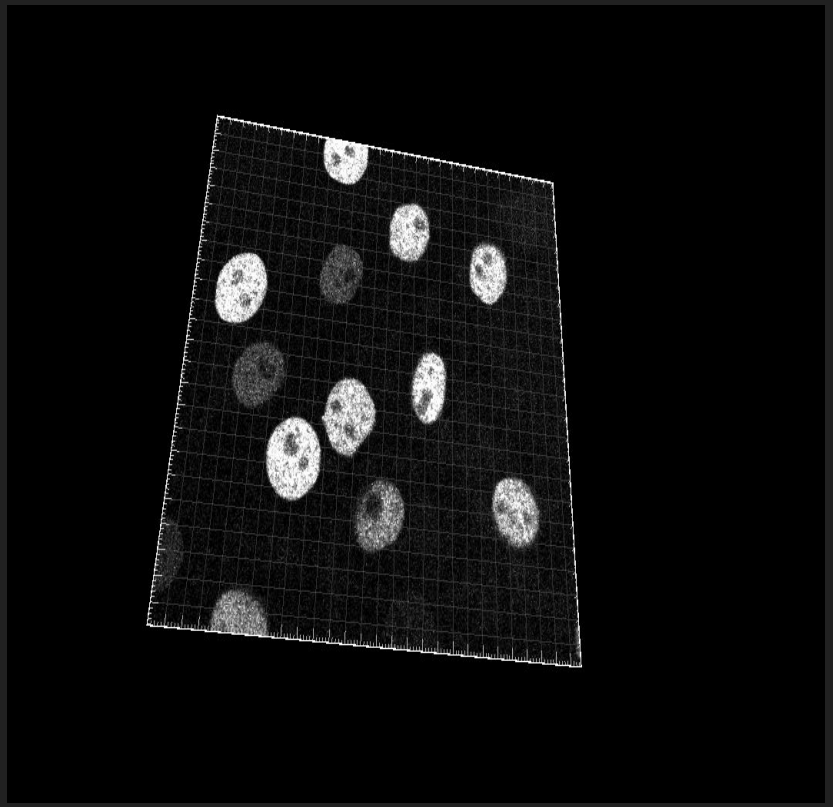
Mask R-CNN



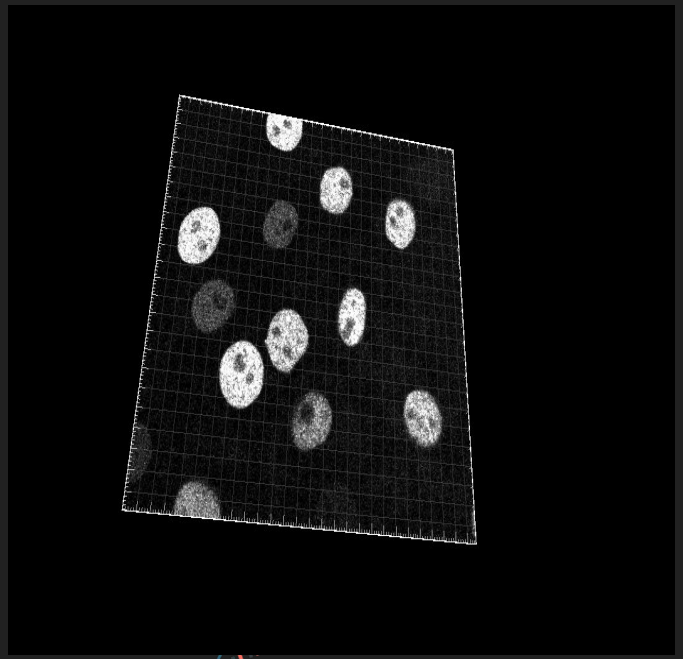
About Datasets

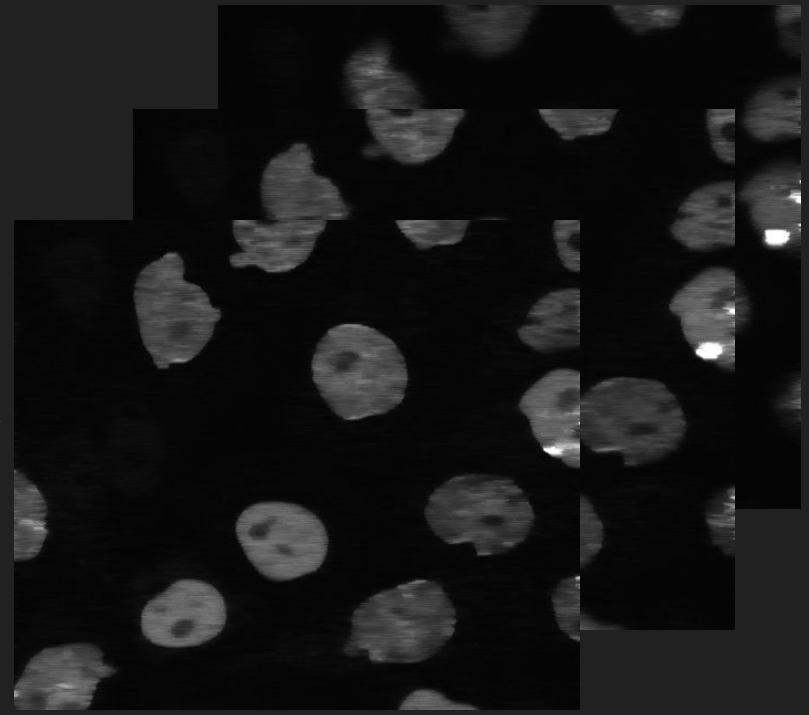
In Cell tracking Challenge

- kinds : 15 Class(2D or 3D)
- type: tiff files
- size: ~ 46.3Gb

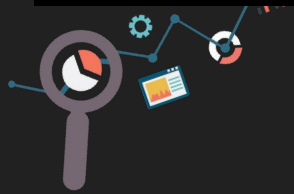


Data Preprocessing in local



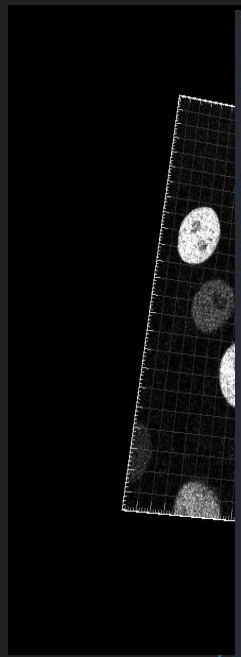


Frame Split



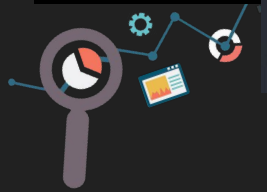
Total 102,989 images

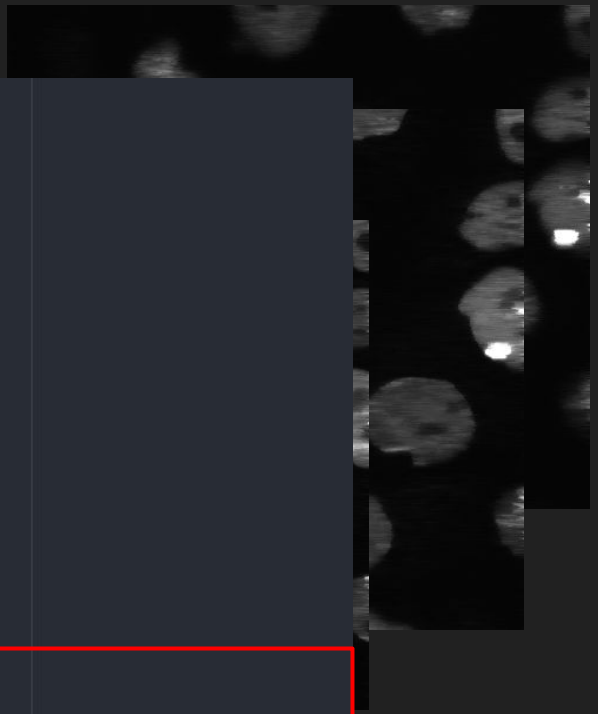
Data Preprocessing in local



```
%pylab
import cv2
import os
import glob
root=os.getcwd()
dataset=os.path.join(root,'dataset')
process=os.path.join(root,'process')
abspath=os.path.abspath("./")
for root, dirs, files in os.walk(dataset):
    if not root.find("./.") or 0<=len(files) <=1 :
        continue
    for file in files:
        abspath=os.path.abspath(root)
        os.path.join(abspath,file)
        abspath+='/'
        abspath+=file
        print(abspath)
        name=abspath[21:-4].replace("/",'')
        ret, images = cv2.imreadmulti(abspath)
        for idx ,img in enumerate(images):
            savedir=os.path.join(process,name)+'_'+str(idx)+'/images'
            os.makedirs(savedir)
            cv2.imwrite(savedir+'/'+name+'_'+str(idx)+'.png' ,img,params=[cv2.IMWRITE_PNG_COMPRESSION,0])
```

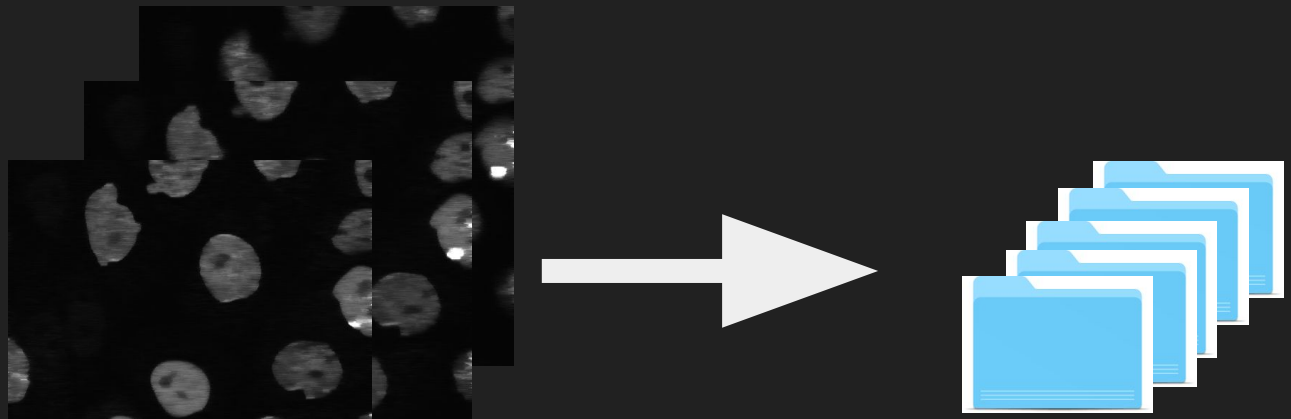
Read All Frame &
Saving All





Total 102,989 images

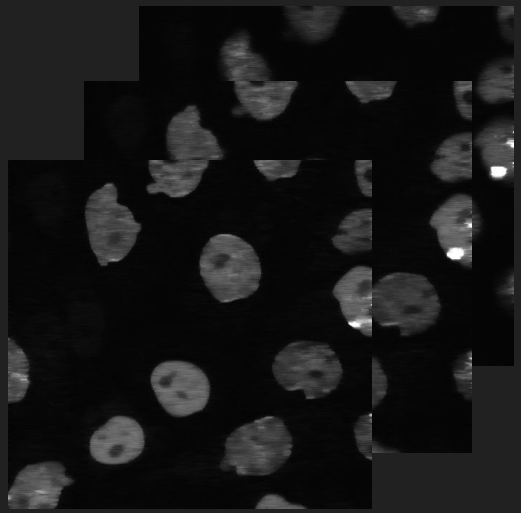
Data Preprocessing in local

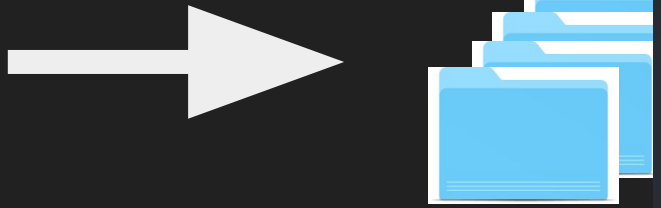


Split 103 Batch

If you process 100,000 at a time, the processor is killed..

Data Preprocessing in local





Split 103 Batch

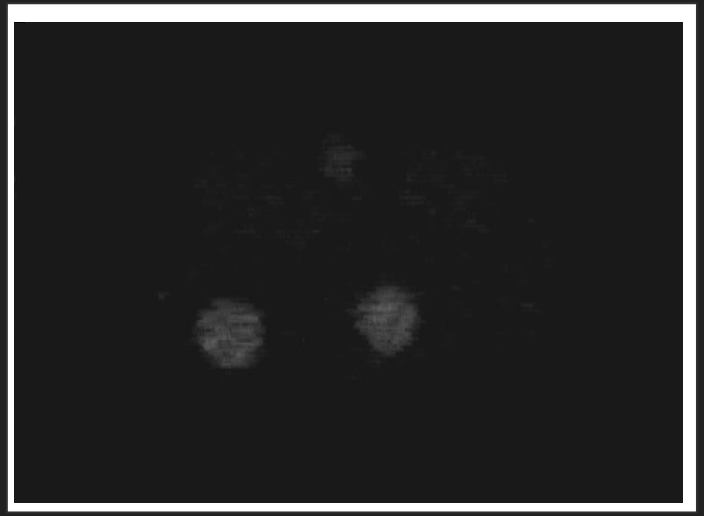
If you process 100,000 at a time, the processor is kille

```
#! /bin/bash
# make temp path
mkdir processTemp
cd processTemp
mkdir $user{1..103}
```

```
#! /usr/bin/python3
# moving batch
import os
import glob
import shutil
root=os.getcwd()
process=os.path.join(root,'process')
temp=os.path.join(root,'processTemp')
abspath=os.path.abspath("./")
sum_list=os.listdir(process)
sum_list=sorted(sum_list)
for idx,_list in enumerate(sum_list):
    buffer=(round)(idx/1000)
    path=os.path.join(process,_list)
    dest=os.path.join(temp,str(buffer))
    shutil.move(path,dest)
    print(path,"\n",dest,"\nmove complete")
```

```
#! /bin/bash
# starting script
for i in {0..103}
    do
      CUDA_VISIBLE_DEVICES=$((i%4)) python3 samples/nucleus/nucleus.py detect --dataset=../processTemp --subset=$i
    wait
done
```

Data Acquisition from Image



Mask R-CNN

Result Image

Metadata

```
['ImageId', 'EncodedPixels']
['1_PhC-C2DL-PSC01t198_0', ' 137369 4 137944 6 138519 8 139095 9 139671 9
5']
['1_PhC-C2DL-PSC01t198_0', ' 146293 2 146867 6 147442 8 148018 8 148594 8
3']
['1_PhC-C2DL-PSC01t198_0', ' 51379 6 51954 9 52530 10 53106 11 53682 11 54
```

HDFS Config

```
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.blocksize</name>
<value>128m</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/home/project/hadoop-2.7.5/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/project/hadoop-2.7.5/datanode</value>
</property>
```

Summary

Security is off.

Safemode is off.

416649 files and directories, 210249 blocks = 626898 total filesystem object(s).

Heap Memory used 142.06 MB of 257.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 66.04 MB of 67.19 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	141.78 GB
DFS Used:	26.77 GB (18.88%)
Non DFS Used:	31.2 GB
DFS Remaining:	77.46 GB (54.63%)
Block Pool Used:	26.77 GB (18.88%)
DataNodes usages% (Min/Median/Max/stdDev):	18.85% / 18.85% / 18.94% / 0.04%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2018. 6. 10. 오전 12:31:56

HDFS Loading

```
#!/bin/bash
${HADOOP_HOME}/bin/hdfs dfs -mkdir -p /usr/${USERNAME}/source/processed
${HADOOP_HOME}/bin/hdfs dfs -ls /usr/${USERNAME}/source
path=~/processed/nucleus/*
for i in ${path}
do
   ${HADOOP_HOME}/bin/hdfs dfs -put $i /usr/$USER/source/processed
done
```

Spark Config

Executors

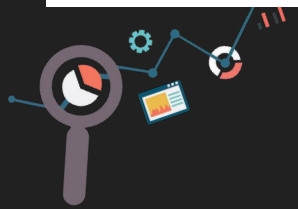
Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write
Active(4)	0	0.0 B / 1.5 GB	0.0 B	6	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B
Total(4)	0	0.0 B / 1.5 GB	0.0 B	6	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B

Executors

Show 20 entries Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs	Thre Dum
driver	192.168.0.156:45118	Active	0	0.0 B / 384.1 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B		Threa Dump
0	192.168.0.152:43073	Active	0	0.0 B / 384.1 MB	0.0 B	2	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Threa Dump
1	192.168.0.163:38744	Active	0	0.0 B / 384.1 MB	0.0 B	2	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Threa Dump
2	192.168.0.151:35693	Active	0	0.0 B / 384.1 MB	0.0 B	2	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Threa Dump



<spark web ui>

1 master ,3 worker

standalone

cluster mode

Spark Processing - Image preprocessing

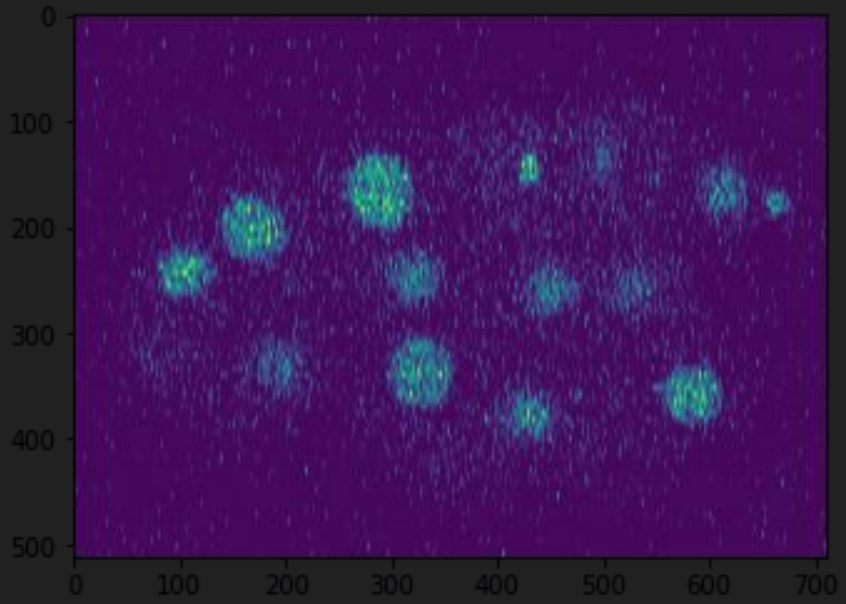
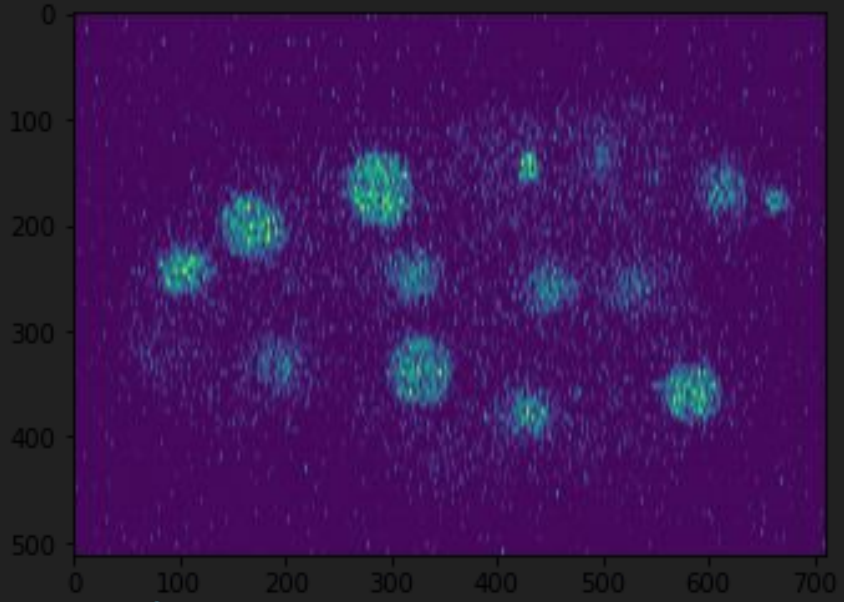


Image has a **noise**!

Spark Processing - Image preprocessing



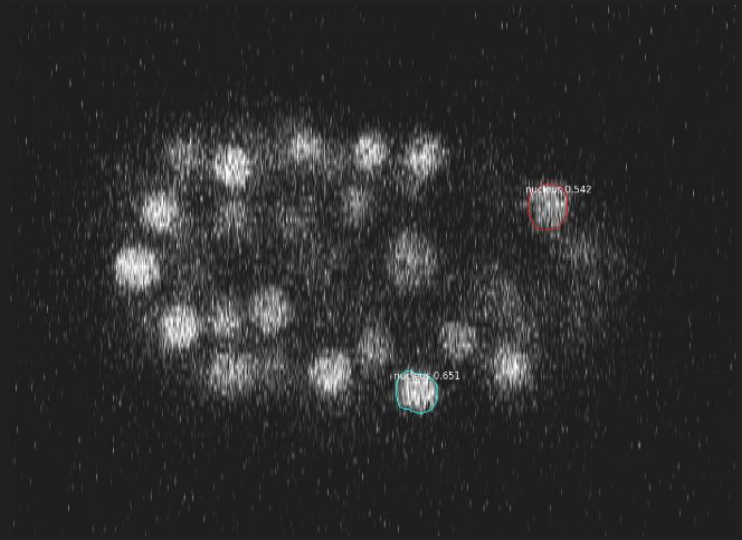
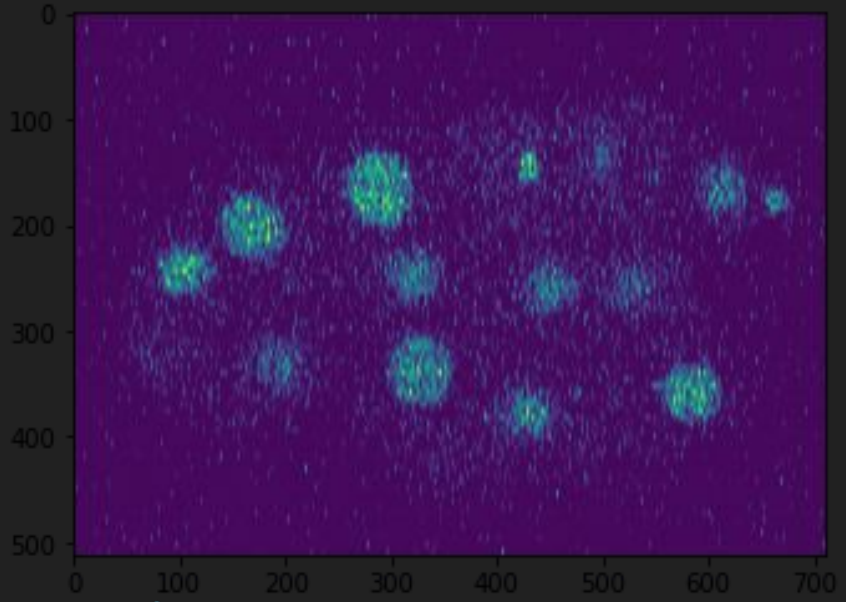
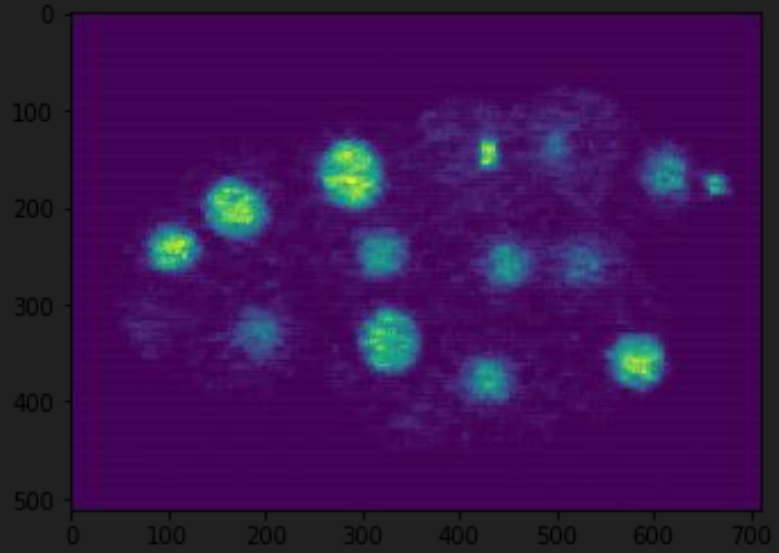


Image has a **noise**!
So algorithm does not work well!

Spark Processing - Image preprocessing





Apply median filter in spark

Spark Processing - Image preprocessing

```
import findspark
import csv,os
findspark.init()
from pyspark import SparkContext
from pyspark.sql import *
import numpy as np
root=os.getcwd()
join=os.path.join
import cv2
import imageio
sc = SparkContext()
from io import StringIO
URI           = sc._gateway.jvm.java.net.URI
Path          = sc._gateway.jvm.org.apache.hadoop.fs.Path
FileSystem    = sc._gateway.jvm.org.apache.hadoop.fs.FileSystem
Configuration = sc._gateway.jvm.org.apache.hadoop.conf.Configuration
fs = FileSystem.get(URI("hdfs://master:9000"), Configuration())
status = fs.listStatus(Path('/usr/project/source/processed'))
```

Read hdfs path in python

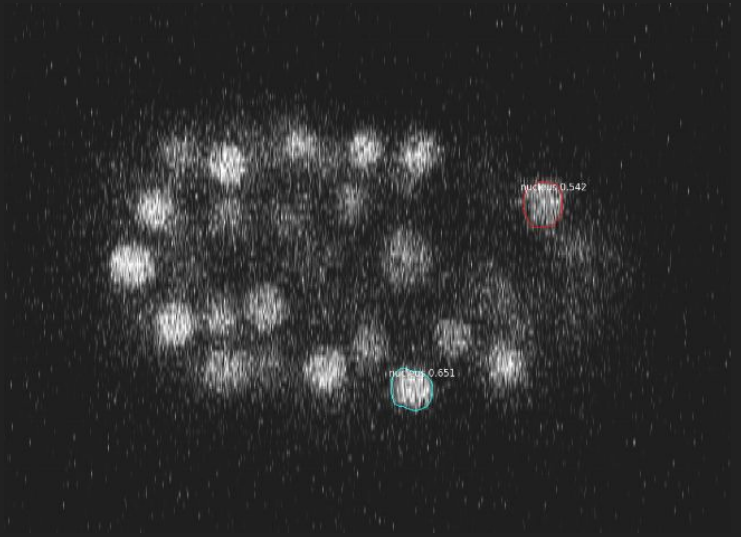
```
def fun(img):
    img=cv2.medianBlur(img,15)
    return img
```

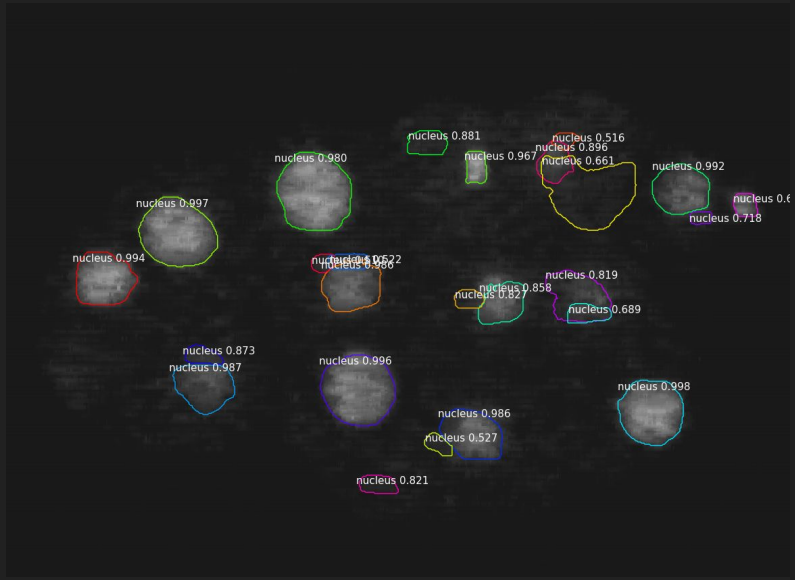
Define medianBlur

```
for fileStatus in status:
    df = imageio.imread(join(str(fileStatus.getPath()),'*.png')
    images = sc.parallelize(df)
        cv2.imwrite(j[:-4]+str(df.shape)+'.png',np.array(images.map(fun).collect()).squeeze()
```

Apply & saving

Spark Processing - Image preprocessing





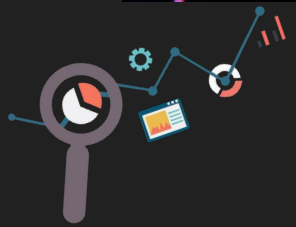
Spark Processing - csv processing

```
submit_20180609T154534  submit_20180609T174540
submit_20180609T154539  submit_20180609T174831
submit_20180609T154543  submit_20180609T175220
submit_20180609T154547  submit_20180609T175710
submit_20180609T154725  submit_20180609T180154
submit_20180609T155554  submit_20180609T180619
submit_20180609T155727  submit_20180609T180710
submit_20180609T160209  submit_20180609T181522
submit_20180609T160422  submit_20180609T181727
submit_20180609T160831  submit_20180609T182353
submit_20180609T160939  submit_20180609T182358
submit_20180609T161248  submit_20180609T182443
submit_20180609T161915  submit_20180609T183352
submit_20180609T162004  submit_20180609T183942
submit_20180609T162115  submit_20180609T184123
submit_20180609T162629  submit_20180609T184628
submit_20180609T162939  submit_20180609T184711
submit_20180609T163135  submit_20180609T185157
```

- 103 folders
- png files & one csv in a folder

Spark Processing - csv processing

```
1_Fluo-C2DL-MSC01t000_0(832, 992).png
1_Fluo-C2DL-MSC01t001_0(832, 992).png
1_Fluo-C2DL-MSC01t002_0(832, 992).png
1_Fluo-C2DL-MSC01t003_0(832, 992).png
1_Fluo-C2DL-MSC01t004_0(832, 992).png
1_Fluo-C2DL-MSC01t005_0(832, 992).png
1_Fluo-C2DL-MSC01t006_0(832, 992).png
1_Fluo-C2DL-MSC01t007_0(832, 992).png
1_Fluo-C2DL-MSC01t008_0(832, 992).png
1_Fluo-C2DL-MSC01t009_0(832, 992).png
1_Fluo-C2DL-MSC01t010_0(832, 992).png
1_Fluo-C2DL-MSC01t011_0(832, 992).png
1_Fluo-C2DL-MSC01t012_0(832, 992).png
1_Fluo-C2DL-MSC01t013_0(832, 992).png
```



For decode,

need to know pixel size (shape)

Spark Processing - csv processing

ImageId	EncodedPixels
1_Fluo-C3DH-H15701t009_24	
1_Fluo-C3DH-H15701t008_22	
1_Fluo-C3DH-H15701t001_20	
1_Fluo-C3DH-H15701t000_22	
1_Fluo-C3DH-H15701t003_11	
1_Fluo-C2[	189463 14 190294 17 191125 20 191956 22
1_Fluo-C2[	668687 1 669505 21 670337 22 671168 24 6
1_Fluo-C2[	263198 17 264029 22 264861 25 265693 35
1_Fluo-C2[	669630 3 670461 5 671293 6 672125 6 6729
1_Fluo-C2[	466431 2 467262 5 468094 6 468927 5 4697
1_Fluo-C2[	279880 9 280711 12 281542 15 282373 20 2
1_Fluo-C2[	572877 2 573708 5 574538 9 574554 8 5753
1_Fluo-C3DH-H15701t008_9	
1_Fluo-C2[	297259 3 298030 18 298808 24 299588 27 3

ImageId	EncodedPix	Size
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C2[	189463 1	832, 992
1_Fluo-C2[	668687 1	832, 992
1_Fluo-C2[	263198 1	832, 992
1_Fluo-C2[	669630 3	832, 992
1_Fluo-C2[	466431 2	832, 992
1_Fluo-C2[	279880 9	832, 992
1_Fluo-C2[	572877 2	832, 992

Before size process

After size process

Spark Processing - csv processing (Decoding)

ex) 189463 14 190294 17 ...

RLE Encoded Pixels

ex) 832 x 992 = 825,344

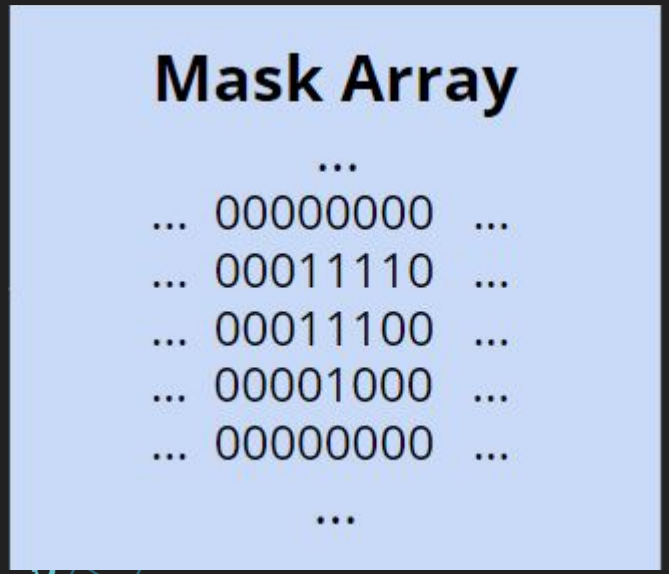
Shape Value

Decoding Function

Mask Array

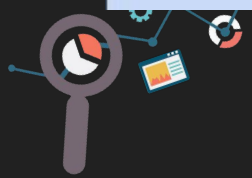
...
... 00000000 ...
... 00011110 ...
... 00011100 ...
... 00001000 ...
... 00000000 ...
...

Spark Processing - csv processing (Decoding)



Count number of 1

cell ratio - 1’s count / shape



Spark Processing - csv processing

ImageId	EncodedPix	Size
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C3DH-H15701t		832, 992
1_Fluo-C2D	189463 14	832, 992
1_Fluo-C2D	668687 1	832, 992
1_Fluo-C2D	263198 17	832, 992
1_Fluo-C2D	669630 3	832, 992
1_Fluo-C2D	466431 2	832, 992
1_Fluo-C2D	279880 9	832, 992
1_Fluo-C2D	572877 2	832, 992

Before decoding

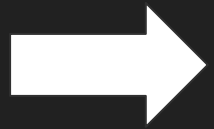
ImageId	EncodedPixe	Size	total_area	cell_area	cell_ratio
1_Fluo-C3DH-H15701t0(		832, 992	0	0	0
1_Fluo-C3DH-H15701t0(		832, 992	1	1	1
1_Fluo-C3DH-H15701t0(		832, 992	2	2	2
1_Fluo-C3DH-H15701t0(		832, 992	3	3	3
1_Fluo-C3DH-H15701t0(		832, 992	4	4	4
1_Fluo-C2D	189463 14	832, 992	825344	2196	0.00266071
1_Fluo-C2D	668687 1 6	832, 992	825344	3449	0.00417886
1_Fluo-C2D	263198 17	832, 992	825344	1284	0.00155572
1_Fluo-C2D	669630 3 6	832, 992	825344	63	7.63E-05
1_Fluo-C2D	466431 2 4	832, 992	825344	52	6.30E-05
1_Fluo-C2D	279880 9 2	832, 992	825344	1050	0.0012722
1_Fluo-C2D	572877 2 5	832, 992	825344	790	0.00095718
1_Fluo-C3DH-H15701t0(		832, 992	12	12	12
1_Fluo-C2D	297259 3 2	782, 1200	938400	2033	0.00216645
1_Fluo-C2D	800299 4 8	782, 1200	938400	118	0.00012575
1_Fluo-C2D	614706 4 6	782, 1200	938400	1498	0.00159633

After decoding

Data Visualization

Data to Mysql

```
mysql> SHOW COLUMNS FROM BD;
ERROR 1146 (42S02): Table 'bd.BD' doesn't exist
mysql> SHOW COLUMNS FROM BIGDB;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| Index2  | int(11)      | NO   | PRI | 0       |       |
| ImageId | varchar(30)  | NO   |     | NULL    |       |
| ENPIXEL | varchar(1000)| YES  |     | NULL    |       |
| SIZE    | int(11)      | YES  |     | 0       |       |
| total   | int(11)      | YES  |     | 0       |       |
| CELL    | int(11)      | YES  |     | 0       |       |
| Ratio   | double       | YES  |     | 0       |       |
+---------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

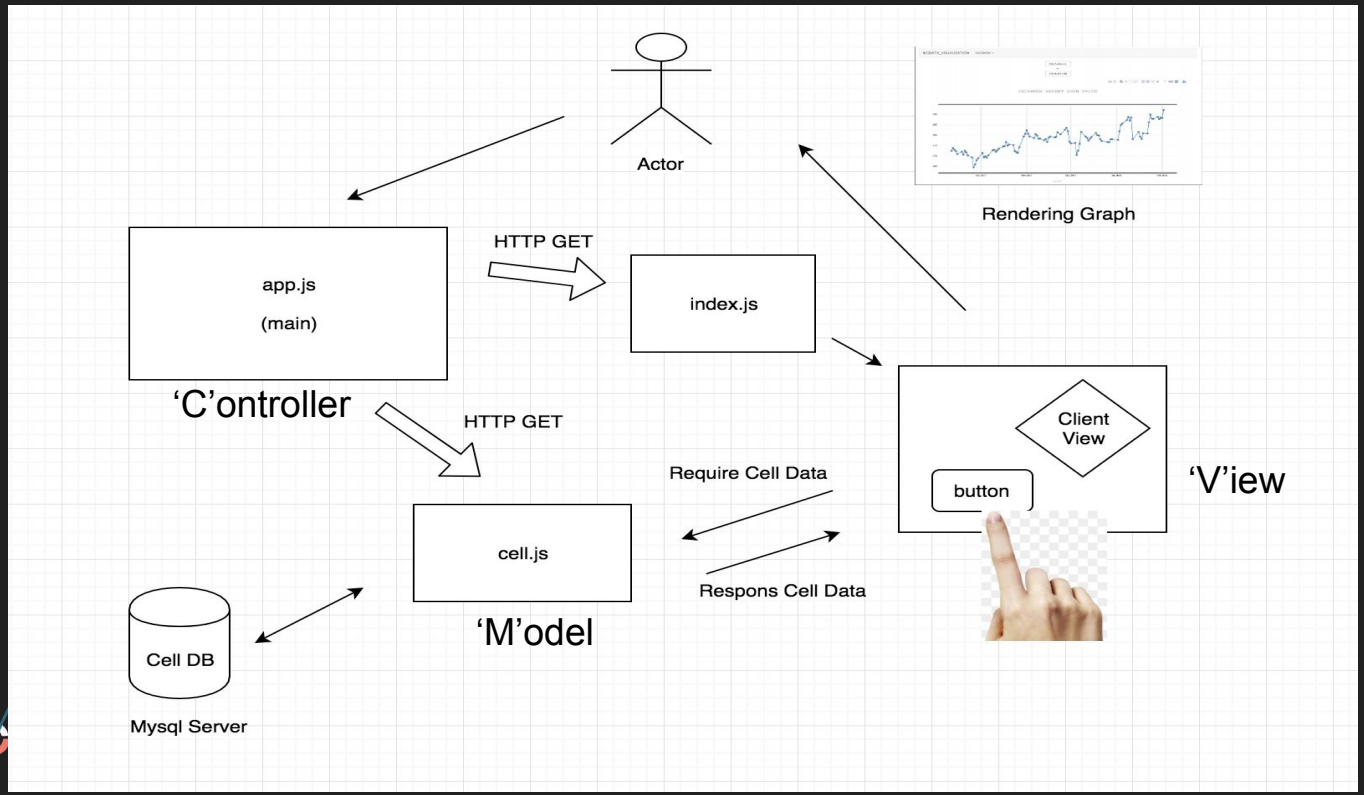


```
                    |   0 |   992 |      4 |     4 |
|      5 | 1_Fluo-C2DL-MSC01t030_0   |  189463 14 190294 17 191125 20 1919
70 38 203602 40 204435 39 205267 40 206099 40 206931 41 207763 42 208595 4
2 220244 42 221076 42 221908 42 222740 41 223572 41 224404 40 225236 40 22
6886 27 237719 25 238552 23 239385 21 240218 19 241053 15 241887 11 242722

                    |   0 |   992 | 825344 |  2196 |
|      6 | 1_Fluo-C2DL-MSC01t030_0   |  668687 1 669505 21 670337 22 67116
3 48 682815 50 683647 50 684479 51 685311 52 686143 53 686975 53 687807 54
 699456 55 700288 55 701120 55 701952 55 702784 55 703615 55 704447 55 705
094 51 716926 50 717758 50 718590 49 719422 48 720254 47 721086 47 721918

                    |   0 |   992 | 825344 |  3449 |
|      7 | 1_Fluo-C2DL-MSC01t030_0   |  263198 17 264029 22 264861 25 2656
10 41 277342 40 278175 39 279007 39 279839 39 280671 39 281503 39 282336 3
6 294006 9 294841 1
```

Overall Server & Client Structure



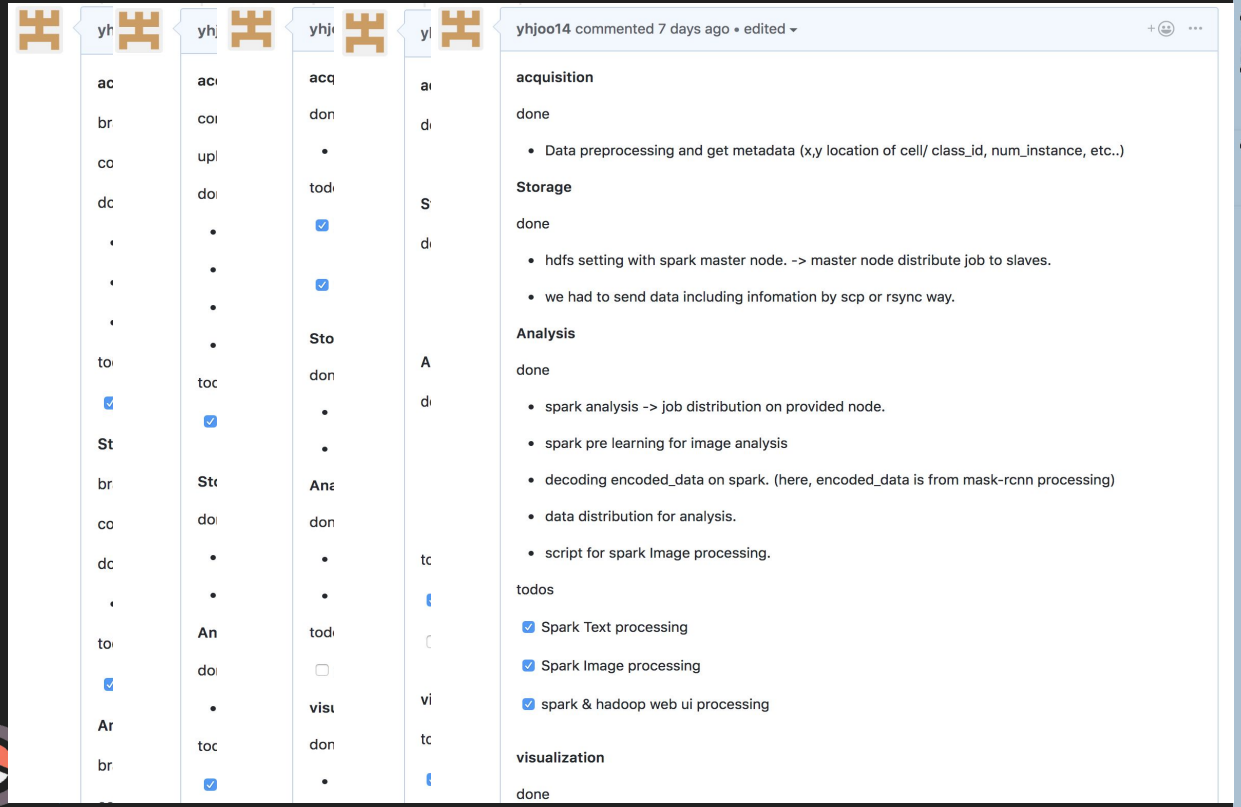
Review

How to follow the milestone

Milestone

Level	Task	5월				6월	
		1주차	2주차	3주차	4주차	5주차	12주차
Data Acquisition	*Contact the professor abroad.*						
	Response from prof.						
Data Preprocessing	*Processing Data on Mask_RCNN*						
	Export Meta data and Image						
Data Strorage	*Hdfs multi node Setting*						
	Data to Hdfs Storage						
Data Analysis	*Spark Multi node Setting*						
	Do the Job on Spark env.						
Data Visualization	*Script Programming on Node.js*						

How to share Weekly Progress - 5 Times



yhjoo14 commented 7 days ago • edited

acquisition

done

- Data preprocessing and get metadata (x,y location of cell/ class_id, num_instance, etc..)

Storage

done

- hdfs setting with spark master node. -> master node distribute job to slaves.
- we had to send data including infomation by scp or rsync way.

Analysis

done

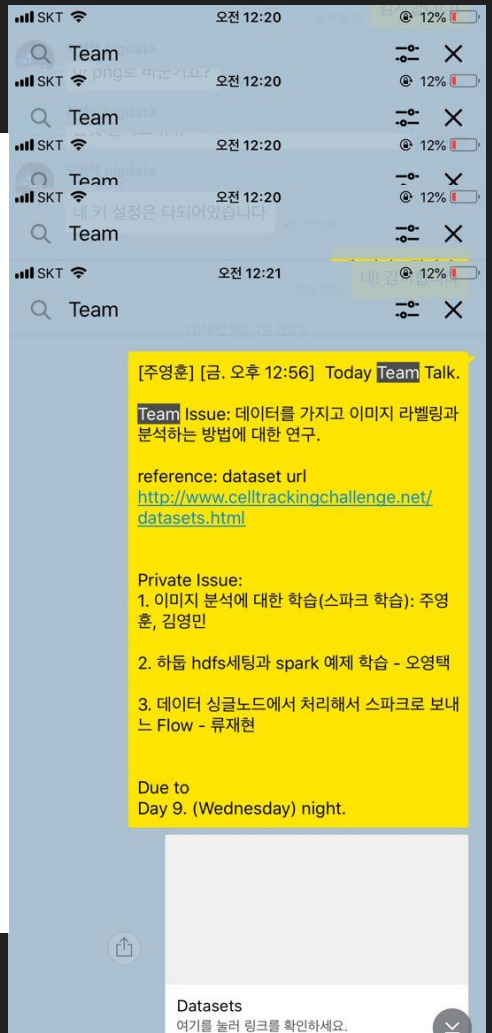
- spark analysis -> job distribution on provided node.
- spark pre learning for image analysis
- decoding encoded_data on spark. (here, encoded_data is from mask-rcnn processing)
- data distribution for analysis.
- script for spark Image processing.

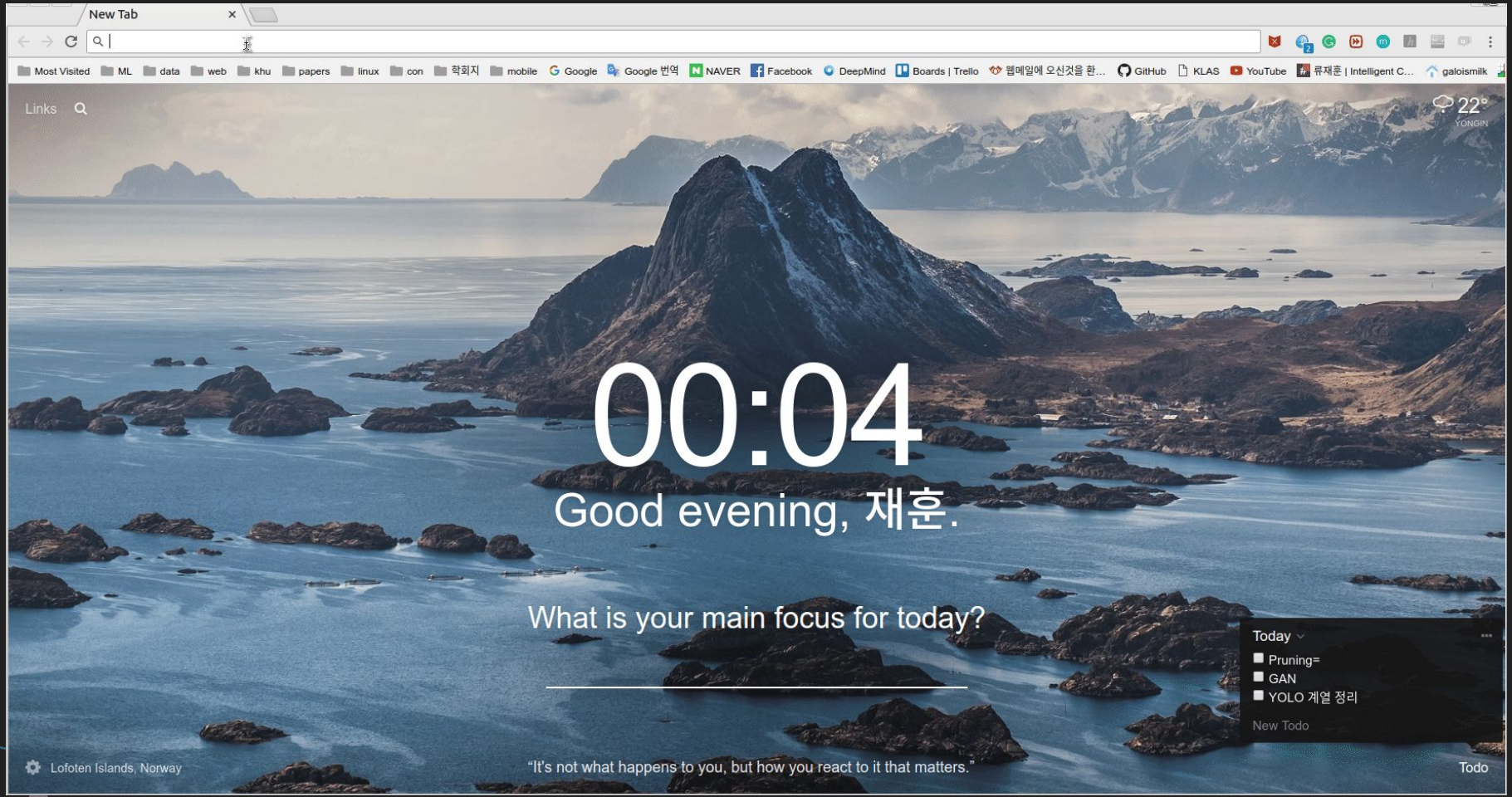
todos

- [x] Spark Text processing
- [x] Spark Image processing
- [x] spark & hadoop web ui processing

visualization

done




New Tab
Most Visited
ML
data
web
khu
papers
linux
con
학회지
mobile
Google
Google 번역
NAVER
Facebook
DeepMind
Boards | Trello
웹메일에 오신것을 환...
GitHub
KLAS
YouTube
류재훈 | Intelligent C...
galoismilk
Links
22°
YONGIN
00:04
Good evening, 재훈.
What is your main focus for today?
Today
Pruning=
GAN
YOLO 계열 정리
New Todo
Lofoten Islands, Norway
"It's not what happens to you, but how you react to it that matters."
Todo

Thank you

http://133.186.134.34:8060