Bijikon server guideline

VMO Holdings .Jsc

20-01-2021

Mục lục

1 Account

1.1 Bunbu

- Google account:

```
username: duy.lai.khang@bunbusoft.com
password: sieu_123
```

- Github:

Github account contains repo for nowcasting code.

```
username: bunbuduylai
password: Sieu_2004
```

2 WNI script

2.1 Diagolization result script

2.1.1 Requirements

Given a table of prediction in different timestamps and localtion, export an diagolized table for each lclid (location name)

The columns in the given table include:

- context
- lclid
- t_0
- t_1
- …
- t_36

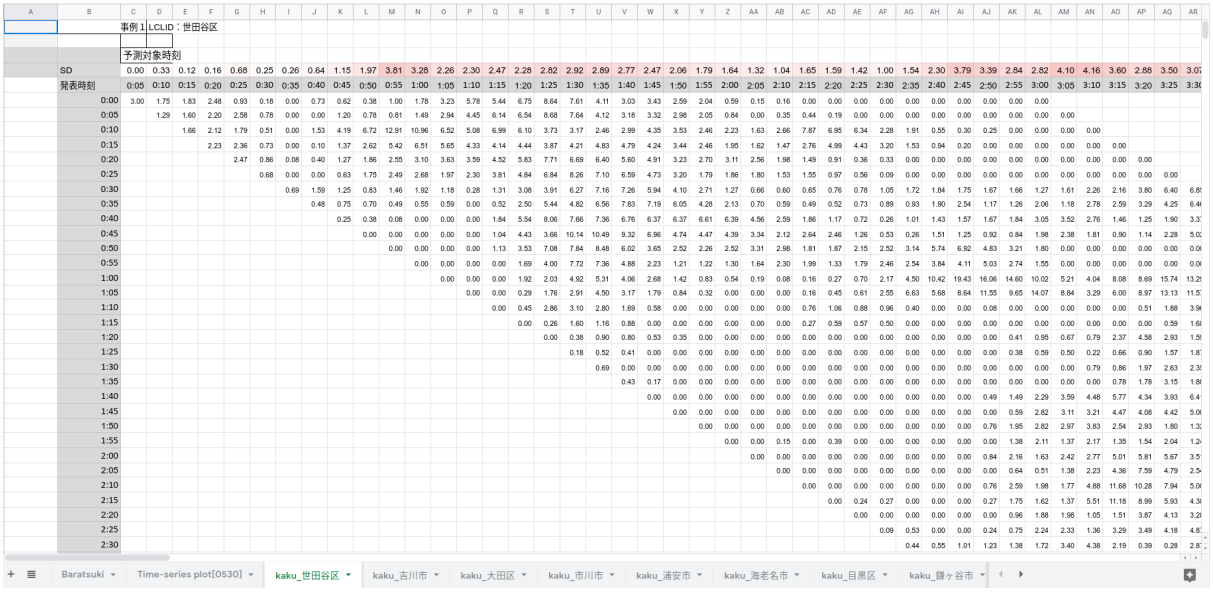
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	context	lclid	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_10	t_11	t_12	t_13
2	NONE	浦安市	20210313_0100	0.8416976	0.53675437	7.4792595	2.878951	2.4813795	4.852856	2.8326895	2.825183	2.2636557	2.8018532	0	1.7523365	0.76092803
3	NONE	市川市	20210313_0100	0.17039359	0	0	3.0846443	3.2580235	2.8057814	2.8995419	2.7939916	1.6734556	0.8089317	0.1199491	0.23601758	0
4	NONE	鎌ヶ谷市	20210313_0100	0.5306661	1.0029906	0.9374049	0	2.9784143	0.6724137	1.2594023	0.2043153	0.88425136	0.7927877	1.1912606	0.9426259	3.8039093
5	NONE	吉川市	20210313_0100	0	0	0	0	0	0	0	0	0	0	0	0	0
6	NONE	大田区	20210313_0100	3.4096198	5.5684195	3.456168	0.8586477	3.7581859	3.502962	3.502962	3.502962	3.4176857	1.7570533	1.1789975	0.4207549	0.81377363
7	NONE	世田谷区	20210313_0100	0	0	0	0	0	0.0147771835	0	0.26210797	0.49896812	0	1.1281085	0	0
8	NONE	目黒区	20210313_0100	0.02622199	0.025218844	0.025218844	0.17913544	0	0	0.2053914	0.025218844	0.025218844	0	0	0	0.007828474
9	NONE	海老名市	20210313_0100	0.5164372	0.5211414	0.5211414	0.5211414	0.5211414	0.5211414	0.5211414	0.5211414	0.5211414	0.3589965	0.7146013	7.785676	10.248099
10	NONE	浦安市	20210313_0105	1.1426318	1.8475721	1.2277472	1.9217184	0	1.8256123	6.5536394	8.675783	1.0716494	1.4031386	0.7680681	0.1514504	2.697336
11	NONE	市川市	20210313_0105	0.53592825	0.47093153	0.0524472	0	0	0	0.9095167	0.7925377	3.3291585	3.1892989	5.0072374	0.93567646	0
12	NONE	鎌ヶ谷市	20210313_0105	0.7154453	1.2792753	0.5948349	0.20497763	0.12538862	0.9528272	1.3242402	0.21546578	0.17980063	1.4948201	2.7359285	2.182465	0.93455493
13	NONE	吉川市	20210313_0105	0	0	0	0	0	0	0	0	0	0	0	0	0.47442734
14	NONE	大田区	20210313_0105	1.7780306	1.3480629	0.28648353	1.2878541	1.3058074	1.9810896	1.973481	2.2493162	2.3467374	2.3467374	1.9791487	1.9192443	2.5769649
15	NONE	世田谷区	20210313_0105	3.6009314	0	0	0	0	1.6388776	2.0943954	2.2019606	0	0	0	0	0.20337296
16	NONE	目黒区	20210313_0105	0	0	0	0	0	0	0.2807815	0	0	0	0	0	0
17	NONE	海老名市	20210313_0105	0.52149475	0.526199	0.526199	0.526199	0.526199	0.526199	0.526199	0.526199	0.526199	0.7964598	14.962672	9.566468	9.509917
18	NONE	浦安市	20210313_0110	2.7204843	6.309289	3.985684	0.9293511	0.9293511	4.5870123	0.70371723	0.6962106	0.6962106	0.6962106	0.66848946	0.5105978	0.029122353
19	NONE	市川市	20210313_0110	0.6822535	0	0	5.2587	0	0.3911605	0.07839024	1.7703631	0.015512109	0.015512109	0.015512109	0.015235305	0.015235305
20	NONE	鎌ヶ谷市	20210313_0110	0.27861607	1.4373021	2.8870316	5.899424	1.1432104	1.0359274	1.1538216	0.9587238	0.5256733	2.2800736	1.2987295	0.057228684	0.21388185
21	NONE	吉川市	20210313_0110	0	0	0	0	0	0	0	0	0	0.4104241	0	0	0
22	NONE	大田区	20210313_0110	0.5708853	0.6249007	2.171526	1.7955537	3.447507	3.447507	3.447507	3.447507	3.447507	3.447507	3.447507	3.4472308	3.4472308
23	NONE	世田谷区	20210313_0110	3.668211	0	0	0	0	2.7386932	1.6190407	1.6078012	0	0	0	0	0
24	NONE	目黒区	20210313_0110	0.003448248	0.0024451017	0.14466298	0	0.22130442	1.0358552	1.4078146	0.0024451017	0.0024451017	0.0024451017	0.0024451017	0.002169013	0.002169013
25	NONE	海老名市	20210313_0110	0.5307498	0.53545403	0.53545403	0.53545403	0.53545403	0.53545403	0.12836838	0.37330914	0.53545403	0.53545403	0.53545403	1.5632079	6.472422
26	NONE	浦安市	20210313_0115	1.3389595	0	1.6788411	2.026494	1.3925815	0.6032324	1.6941848	0.60153294	0.58333325	1.5411172	0.04414165	0.83697224	0.97138345
27	NONE	市川市	20210313_0115	0.36426938	0.04287207	0	0.23229432	1.1653247	0.16794169	0.669454	1.1653247	0.88794136	0.8770466	0	0	0
28	NONE	鎌ヶ谷市	20210313_0115	1.6893697	1.4861584	0.55176544	0.50384223	0.4203683	0.46151304	1.7012908	0.33194268	1.3682224	1.2731116	0.29008305	0.24419296	1.3381705
29	NONE	吉川市	20210313_0115	0	0	0	0	0	0	0	0	0	0.27560592	0	0.15234518	1.0903487
30	NONE	大田区	20210313_0115	2.3472018	0.7020707	1.4682348	0.31841183	3.1588287	2.3320935	2.3545356	1.8735502	1.9869472	2.3545356	2.3545356	2.3542595	2.3542595
31	NONE	世田谷区	20210313_0115	4.155401	0	0	0	0	1.8021315	1.8040745	1.6202956	0	0	0	0.76223624	0.05308163
32	NONE	目黒区	20210313_0115	0	0	0	0	0	0	0	0.15979743	0	0.09977496	0	0	0
33	NONE	海老名市	20210313_0115	0.5148039	0.5195081	0.5195081	0.5195081	0.5195081	0.5195081	0.5195081	0.5195081	0.5195081	1.1442214	6.465171	4.3794813	4.123646

Hình 1: Data from customer

Important columns:

1. lclid: place name
2. t_1 -> t_36: accuracy

For each lclid, rearrange data as follow:



Hình 2: Rearranged data

The script has been finished and can use immediately.

2.2 Transpose accuracy result script

2.3 Storm and map drawing script using matplotlib

3 Improve accuracy of WNI nowcasting using deep learning instead of traditional machine learning method

3.1 Applied deep learning on local optical flow

Local optical flow right now is using Hornchunk as a prediction method. Accuracy can be improve using deep learning. Implement the following method on the nowcasing code.

PWCNet

3.2 Applied deep learning on global optical flow

3.3 Applied deep learning on radar image prediction

4 New task for Harupy-san

4.1 Data

The data we use is 3 types of weather, which is in these dates. Please apply the algorithms on all of these dates, and compare them with the existing algorithms.

The data is stored in `/media/hdd0/D2_high_res_jma/raw/` in the server.

1. Squall line

- 2018/6/20
- 2018/6/29
- 2019/6/30
- 2019/7/19
- 2020/7/3~8

2. Typhoon

- 2018/6/15~16
- 2018/9/4
- 2018/9/30

- 2019/8/9
- 2019/9/30
- 2019/10/12

3. Scatter Rain

- 2018/07/08
- 2018/09/02
- 2018/10/18
- 2018/10/19
- 2019/07/16
- 2019/09/10
- 2020/05/11

These data could be cut to match the storm map.

4.2 Give harupy sans the definition of .flo files and pwcnet model.

.flo is a global extension of opticalflow. Research and give harupy sans the documents for flo files.

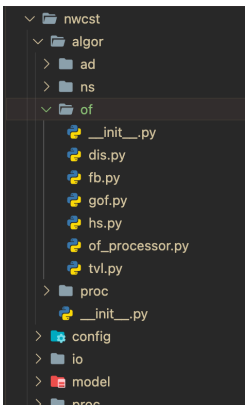
Give harupy sans the flownet2 and pwcnet model for the environment setup.

4.3 Add PWCNet and FlowNet2 on the nowcasting code

Right now, PWCNet (pwc) and FlowNet2 (fn2) are running in the docker, you can find that more in the github repository. Link of PWCNet and FlowNet are below.

1. FLowNet2
2. PWCNet

Customer want to merge these optical flow prediction method to the nowcasting code. We have to add new separated file for the algorithms in the following directory of the nwcst repo.



Hình 3: place to add pwc and fn2

Add new file `nwcst/algor/of/pwc.py` and `nwcst/algor/of/fn2.py`.

In `nwcst/algor/of/__init.py__`, define pwc and fn2.

```
class OpticalFlow:

    model_class_map = {
        "hs": HS,
        "fb": FB,
        "dis": DIS,
        "tvl": TVL,
        "gof": GOF
    }

    @staticmethod
    def create(model,*args,**kwargs):
        return OpticalFlow.model_class_map[model](*args,**kwargs)

    @staticmethod
    def HS(*args, **kwargs):
        return HS(*args, **kwargs)

    @staticmethod
    def FB(*args, **kwargs):
        return FB(*args, **kwargs)

    @staticmethod
    def DIS(*args, **kwargs):
        return DIS(*args, **kwargs)

    @staticmethod
    def TVL(*args, **kwargs):
        return TVL(*args, **kwargs)

    @staticmethod
```

```
    def GOF(*args, **kwargs):
        return GOF(*args, **kwargs)
```

for example, in `model_class_map` add `"pwc": PWC`.

Then, sefine with decorator `@staticmethod`.

```
    @staticmethod
    def PWC(*args, **kwargs):
        return PWC(*args, **kwargs)
```

The new file has to have the structure like this.

```
import numpy as np
import optflow
import cv2
from nwcst.algor.proc import func
'''==========Config Parameters============
        fb:
            scale_image: True
            numLevels: 5
            pyrScale: 0.5
            fastPyramids: False
            winSize: 13
            numIters: 10
            polyN: 5
            polySigma: 1.5
            flags: 0
'''


class FB:
    def __init__(self, conf=None):
        self.conf = conf['fb']

    def calc(self, imgs):
        if self.conf['scale_image']:
            print("scale")
            im_scaled_0, c0_1, c0_2 = func.scaler(imgs[-2])
            im_scaled_1, c1_1, c1_2 = func.scaler(imgs[-1])
        else:
            im_scaled_0 = imgs[-2].copy()
            im_scaled_1 = imgs[-1].copy()
        im0 = im_scaled_0.astype(np.uint8)
        im1 = im_scaled_1.astype(np.uint8)

        of_instance = cv2.FarnebackOpticalFlow_create(numLevels=self.
            conf['numLevels'],
                                                      pyrScale=self.
                                                          conf['pyrScale
                                                          '],
```

```
                                                                fastPyramids=self
                                                                    .conf['
                                                                    fastPyramids'
                                                                    ],
                                                                winSize=self.conf
                                                                    ['winSize'],
                                                                numIters=self.
                                                                    conf['numIters
                                                                    '],
                                                                polyN=self.conf['
                                                                    polyN'],
                                                                polySigma=self.
                                                                    conf['
                                                                    polySigma'],
                                                                flags=self.conf['
                                                                    flags'])
        delta = of_instance.calc(im0, im1, None)
        u = delta[..., 0]
        v = delta[..., 1]
        return u, v
```

Where as class name = PWC or FN2 upper case.

Replace FB lowercase and uppercase as pwc and fn2 corresponding to it.

Define fn2 and pwc algorithms in the `calc` function.

The goal is to compare the algorithm with the original one, and see if it improves accuracy.

Create new branch and push it the the repo.