PIBIC-PIBITI/CNPq/INPE
RELATÓRIO TÉCNICO DE ATIVIDADES

<v9>

Número do Processo Institucional:138924/2021-0

Número do Processo Individual: 111912/2022-9

Bolsista: Luís Henrique Ferreira Souza

Orientador: Claudio Clemente Faria Barbosa

Coorientador: (quando houver)

Área: Desenvolvimento de rotinas em ambiente PYTHON para o processamento de imagens ópticas na plataforma MAPAQUALI de monitoramento de sistemas aquáticos continentais por Sensoriamento Remoto.

Vigência original da bolsa: 01/05/2022 a 31/08/2022

Modalidade da bolsa: PIBIC

RELATÓRIO TÉCNICO

1) Resumo do Projeto

Iniciação científica no Laboratório de Instrumentação de Sistemas Aquáticos do INPE (http://www.dpi.inpe.br/labisa/ - labISA), visando trabalhar com a codificação de algoritmos e conversão de rotinas (R para Python) com finalidade de processar imagens de satélite na plataforma em desenvolvimento MAPAQUALI, com o objetivo de: 1) Desenvolver algoritmos para monitoramento de corpos d'água; 2) Converter algoritmos já existentes em linguagem R para Python; 3) Estudo da estrutura e arquitetura da plataforma MAPAQUALI (GeoServer, rotinas Python, algoritmos de Machine Learning (Random Forest)).

2) Objetivo

Implementar e converter rotinas em linguagem R para Python, no ambiente da plataforma de monitoramento e sistemas aquáticos por satélites MAPAQUALI. Implementar/Codificar algoritmos em ambiente Python para estimativa de indicadores de qualidade da água. Utilizar algoritmo de Random Forest de aprendizado de máquina, para estimar indicadores de qualidade de água.

3) Atividades Desenvolvidas durante o período da bolsa

No início da bolsa de iniciação foram propostas as seguintes atividades: 1) Estudo das rotinas e das bibliotecas e funções necessárias para a implementação das rotinas em Python; 2) Conversão de rotinas em R para Python; 3) Codificação em Python de rotinas de determinação de profundidade do sensor secchi e do coeficiente de atenuação difusa (KD).

Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

Figura 1 – Exemplo de rotina codificada em Python para recorte de um Corpo d'água usando um shapefile como referência

```python
def recortar(diretorio_imagem: str, shp: str, diretorio_destino: str):
    """Recorta uma número n de imagens com base num shapefile

    Args:
        diretorio_imagem (str): diretorio da imagem a ser recortada
        shp (str): shapefile correspondente
        diretorio_destino (str): path de destino da imagem recortada
    """

    bandas = ImagemProcessBatch.procurar_bandas(diretorio_imagem)
    for banda in bandas:

        nome_banda = Path(banda).stem
        recorte = ImagemProcess.recortar(banda, shp)
        ImagemProcess.salvar_imagem(recorte, diretorio_destino, nome_banda + '_recortada' + '.TIF')
```

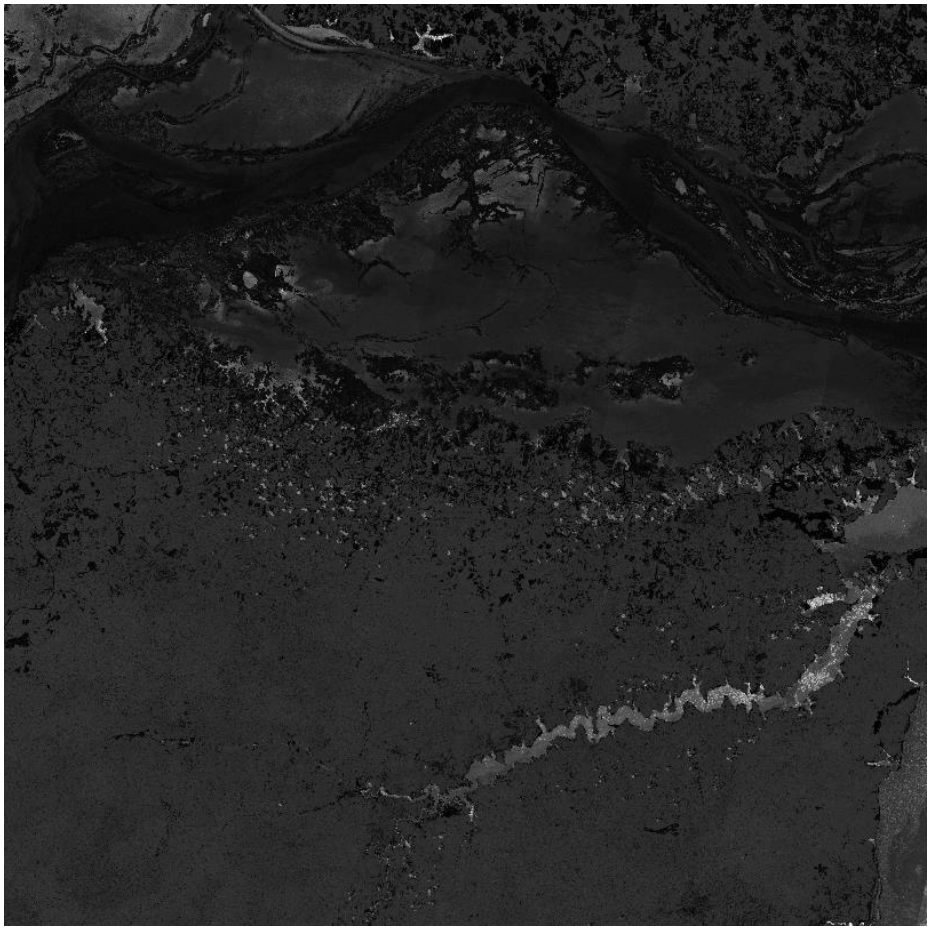
Figura 2.0 – Rotina em Python para determinação da profundidade Secch.

```python
class RF_secchi:
    def __init__(self):
        self.train_df = []
        self.rf_x = []
        self.rf_y = []
        self.sentinel2Bands = []
        self.img_metadata = {'rows':[],
                             'cols':[],
                             'bands':[],
                             'geo_transform':[],
                             'projection':[]
                             }
        self.x_data_nan = []

    def preload_imgs_data(self,path_list):
        self.sentinel2Bands = [gdal.Open(b, gdal.GA_ReadOnly) for b in path_list]
        self.img_metadata['rows'] = self.sentinel2Bands[0].RasterYSize
        self.img_metadata['cols'] = self.sentinel2Bands[0].RasterXSize
        self.img_metadata['bands'] = self.sentinel2Bands[0].RasterCount
        self.img_metadata['geo_transform'] = self.sentinel2Bands[0].GetGeoTransform()
        self.img_metadata['projection'] = self.sentinel2Bands[0].GetProjectionRef()

    def prepare_imgs_toRF(self):
        bands_dict = {}
        # Banda 2
        bands = ['B2','B3','B4','B5','B6']
        for band,i in zip(self.sentinel2Bands,bands):
            bands_dict[i] = (band.ReadAsArray() / np.pi) / 10000
```

```python
def Load_train_data(self,path,delimiter=','):
    if path.endswith(".csv"):
        self.train_df = pd.read_csv(path, delimiter=delimiter)
    if path.endswith(".xlmx"):
        self.train_df = pd.read_excel(path, delimiter=delimiter)
    self.rf_x = self.train_df.loc[:, ['B2', 'B3', 'B4', 'B5', 'B6', 'b3_b2', 'b3_b4', 'b3_b6', 'ndci']].values
    self.rf_y = self.train_df['secchi']

def save_rf_trees(self,path='./my_random_forest.joblib'):
    joblib.dump(self.rf_trees,path)

def load_rf_trees(self,path='./my_random_forest.joblib'):
    self.rf_trees = joblib.load(path)

def apply_RF_training(self,save=True,path='./my_random_forest.joblib',n_estimators=62, max_features = 6, random_state=0):
    self.rf_trees = RandomForestRegressor(n_estimators=n_estimators, max_features = max_features, random_state=random_state).fit
    (self.rf_x,self.rf_y)
    if save == True:
        self.save_rf_trees(path)

def apply_RF_prediction(self):
    prediction = self.rf_trees.predict(self.x_data_nan)
    self.img_prediction = prediction.reshape((self.img_metadata['rows'], self.img_metadata['cols']))
```


Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

Figura 3.2 – Mapeamento da profundidade Secchi a partir da rotina da Figura anterior.



Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

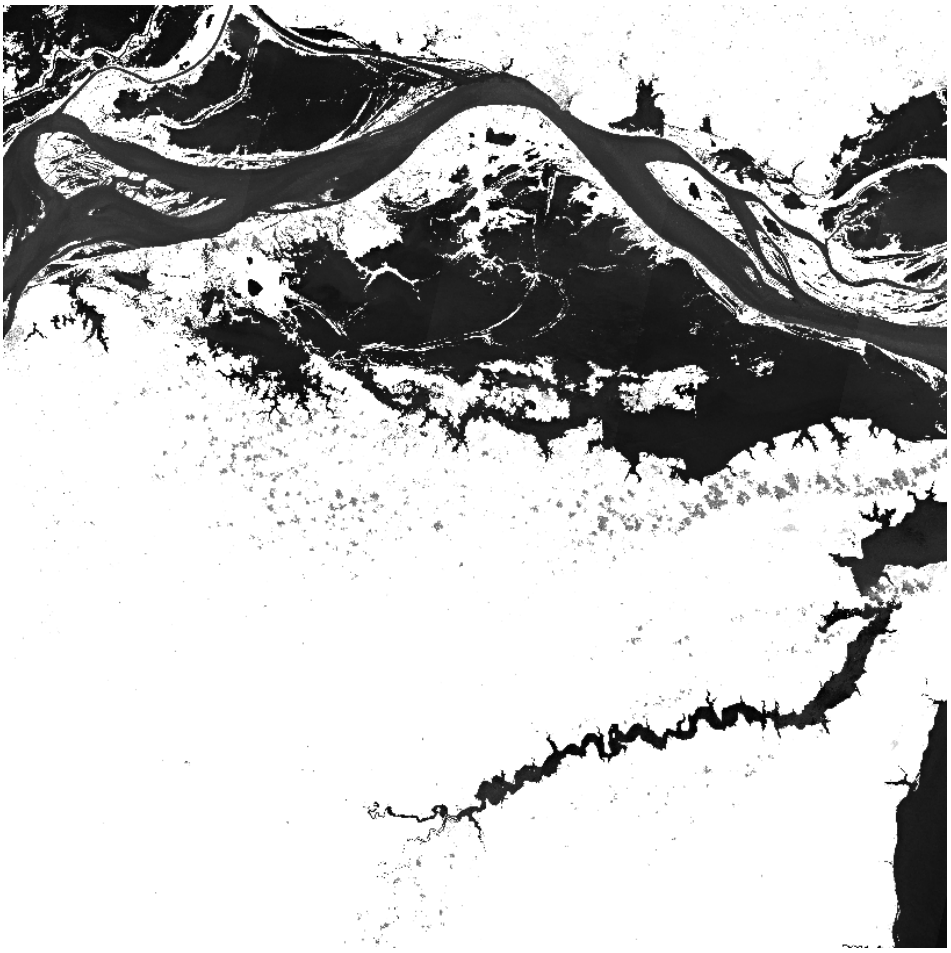
Figura 3.1 – Rotina convertida da Linguagem R para Python para estudo do campo de luz pelo KD.

```python
class KD_mapaquali:
    """Classe para realizar cálculos que compõem o KD
    """

    def __init__(self):
        """Inicia a classe criando as características das imagens
        """
        self.water_abs_bb = pd.read_csv(r"rotinas\kd\w_and_bb_s2a.csv")

        # Elements for equation
        self.w = self.water_abs_bb.iloc[2,3]
        self.j = 0.0604
        self.k = 0.0406
        self.l = 0.0402
        self.m = 0.1310

        self.sentinel2Bands = []
        self.img_metadata = {'rows':[],
                             'cols':[],
                             'bands':[],
                             'geo_transform':[],
                             'projection':[]
                             }
```

```python
#Kd 670

bb = self.bbp_670 + self.water_abs_bb.iloc[2,3]

ratio_bbw_bb = self.water_abs_bb.iloc[2,3]/bb

absorption = self.abs_660

# Semi-analitic model (KD)

kd_660 = (1+m0*teta_s)*absorption + (1-chi*ratio_bbw_bb) * m1 * (1 - m2 * np.exp(-m3*absorption))*(bb)

self.products_kd = {'490': kd_490, '560': kd_560, '670': kd_660}
```


Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

Figura 3.2 – Mapeamento do campo de luz subaquático pelo KD.



Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

4) Resultados Obtidos em função do Plano de Trabalho proposto

I) Aprendizado de uso do algoritmo de Random Forest de aprendizado de máquina; II) Aprendizado e uso dos diferentes tipos de dados de sensoriamento remoto; III) Aprendizado de codificação de dados científicos; IV) Desenvolvimento de rotinas para estimar indicadores de qualidade de água.

Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000

5) Conclusões Gerais

Apesar do pouco tempo de eságio foi possível ter contato básico sobre a área de sensoriamento remoto, tanto orbital quanto de coletas de campo em corpos d'água, além de aprimorar a parte técnica de programação em linguagem Python e em algoritmos computacionais e geração de produtos de qualidade de água.

22, de setembro de 2022

Luís Henrique F. Souza

Bolsista: (colocar nome completo e assinar)

Claudio Clemente Faria Barbosa

Orientador(a): (colocar nome completo e assinar)

Sede: Av. dos Astronautas, 1758, 12227-010 - São José dos Campos (SP) - Brasil Tel. + 55-12-32086000