

Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a umelej inteligencie

# **Modelovanie airglowu pomocou hlbokého učenia**

Bakalárska práca

## **AIRGLOW MODEL**

Systemová príručka

Vedúci bakalárskej práce:

Doc. Ing. Peter Butka, PhD.

Bakalár:

Michal Bencúr

Konzultant bakalárskej práce:

Ing. Viera Maslej Krešňáková, PhD.

Košice 2022

# Obsah

<b>Zoznam obrázkov</b>	<b>1</b>
<b>1 Funkcia programu</b>	<b>2</b>
<b>2 Inštalácia programu</b>	<b>3</b>
2.1 Požiadavky na programové prostriedky . . . . .	3
<b>3 Popis programu</b>	<b>4</b>

## Zoznam obrázkov

3–1 Importovanie potrebných knižníc a stiahnutie algoritmu hlbokého učenia . . . . .	4
3–2 Premenné sťahovania súborov . . . . .	5
3–3 Sťahovanie dát . . . . .	5
3–4 Skrátene názvu dát . . . . .	6
3–5 Spracovanie stiahnutých dát, príprava dát pre grafy a podklad Zemského povrchu . . . . .	6
3–6 Vykreslenie, uloženie a roztriedenie grafov určených pre trénovanie modelov . . . . .	7
3–7 Vytvorenie prierečinkov pre exploračné grafy . . . . .	7
3–8 Uloženie a vykreslenie pre grafy žiarení 1356 a LBH . . . . .	8
3–9 Uloženie a vykreslenie pre grafy SZA a žiarenia 1356 bez podkladu Zeme . . . . .	8
3–10 Vykreslenie prekrytia dát pomocou priehľadnosti . . . . .	9
3–11 Spustenie trénovanie modelu a vytváranie predikcií modelu . . . . .	9
3–12 Orezanie predikcií a očakávaných obrázkov na rovnaký formát . . . . .	10
3–13 Porovnanie predikcií s očakávanými obrázkami pomocou metrík MSE a PSNR . . . . .	11

# 1 Funkcia programu

Úlohou našich skriptov je stiahnuť dáta NASA misie GOLD, ktoré sú vo formáte `.nc`, a previesť ich do grafickej formy vo formáte `.png`. Následne ich roztriediť medzi množiny Severnej a Južnej pologule. Natrénovať model pomocou algoritmu Pix2Pix, orezať výsledné predikcie do formátu `.png` a porovnať ich s očakávanými obrázkami. Jednotlivé úlohy sú rozdelené do viacerých skriptov:

- Stiahnutie potrebných dát zo stránky misie, skrátiť ich názov a uložiť ich vo formáte `.nc`: `download.ipynb`
- Vytvorenie a uloženie grafických verzií dát pre rôzne zobrazenia vo formáte `.png`: `plots.ipynb`
- Grafické zobrazenie ukážka pokrytia dát za jeden deň:  
`all_plots_at_one.ipynb`
- Grafické zobrazenie ukážka prekrytia dát za jeden deň: `overlap_plots.ipynb`
- Vytvorenie a uloženie grafických verzií dát určených ako množina dát pre učenie modelu a roztriedenie podľa pologule vo formáte `.png`:  
`prediction_plots.ipynb`
- Natrénovanie modelu z pripravených dát pomocou algoritmu Pix2PixHD a vytvorenie predikcií z tohto modelu: `pix2pix.ipynb`
- Orezanie vytvorených predikcií aj pôvodných obrázkov aby ich bolo možné porovnať: `crop_images.ipynb`
- Porovnanie vytvorených predikcií a pôvodných snímok pomocou metrík:  
`compare.ipynb`

## 2 Inštalácia programu

### 2.1 Požiadavky na programové prostriedky

- Python 3.8.6
- Anaconda Navigator
- Jupyter Notebook
- Knižnice: `requests`, `bs4`, `urllib`, `os`, `glob`, `pathlib`, `netCDF4`, `matplotlib`, `numpy`, `datetime`, `dominate`, `opencv-python`, `Pillow`, `sewar`, `scipy`, `pandas`, `mpl_toolkits`

### 3 Popis programu

V tejto príručke popisujeme prácu so všetkými skriptami ktoré sem v práci vytvorili vrátane explorácie dát. Všetky tieto skripty sú vo forme Jupyter notebookov dostupných a s celým projektom na Githube: [github.com/MichalBencur03/Airglow-modeling-by-deep-learning](https://github.com/MichalBencur03/Airglow-modeling-by-deep-learning)

1. Na Obrázku 3–1 sú zobrazené použité knižnice, ktoré sú potrebné pre naše skripty. Súčasťou je taktiež príkaz na stiahnutie potrebných súborov použitého algoritmu Pix2PixHD.

```
import requests
from bs4 import BeautifulSoup
import urllib
import os
from urllib.parse import urlparse
import glob
from pathlib import Path
from netCDF4 import Dataset
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime, date, timedelta
from mpl_toolkits.basemap import Basemap
from PIL import Image, ImageDraw
from sewar.full_ref import mse, psnr
import pandas as pd
!git clone -b video https://github.com/jctestud/pix2pixHD.git
```

**Obr. 3–1** Importovanie potrebných knižníc a stiahnutie algoritmu hlbokého učenia

2. Nasledujúca časť slúži na stahovanie potrebných dát misie GOLD zo stránky tejto misie čo zabezpečuje skript `download.ipynb`. Na Obrázku 3–2 je možné vidieť premenné pomocou ktorých môžeme nastaviť z akého obdobia sa majú dáta stiahnuť.
3. Dáta sme získali analyzovaním stránky s dátami a uložili odkazy na tie dáta ktoré obsahovali „NI1“ čo predstavovalo nočné dáta. Následne sme dáta z týchto odkazov uložili podľa ich roku a dňa (pozri obr.3–3).

```
iteration = 1
day = 1;
how_many = 365
year = 2021
```

**Obr. 3–2** Premenné stahovania súborov

```
while iteration <= how_many:

    def get_url_paths(url, ext='', params={}):
        response = requests.get(url, params=params)
        if response.ok:
            response_text = response.text
        else:
            return response.raise_for_status()
        soup = BeautifulSoup(response_text, 'html.parser')
        parent = [url + node.get('href') for node in soup.find_all('a') if node.get('href').endswith(ext)]
        return parent

    if day < 10:
        url = 'https://spdf.gsfc.nasa.gov/pub/data/gold/level1c/' + str(year) + '/00' + str(day) + "/"
    if day >= 10 and day < 100:
        url = 'https://spdf.gsfc.nasa.gov/pub/data/gold/level1c/' + str(year) + '/0' + str(day) + "/"
    if day >= 100:
        url = 'https://spdf.gsfc.nasa.gov/pub/data/gold/level1c/' + str(year) + '/' + str(day) + "/"
    ext = '.nc'
    result = get_url_paths(url, ext)

    result = [k for k in result if 'NI1' in k]

    for f in result:
        a = urlparse(f)
        name = os.path.basename(a.path)
        testfile = urllib.request.URLopener()
        testfile.retrieve(f, name)
    day = day + 1;
    iteration = iteration + 1;

    files = glob.glob(r'*.nc')

    for f in files:
        day = f[22:25]
        Path("data/" + str(year) + "/" + day).mkdir(parents=True, exist_ok=True)
        Path(f).rename("data/" + str(year) + "/" + day + "/" + f)
```

**Obr. 3–3** Stahovanie dát

4. Z názvov stiahnutých dát sme odstránili zbytočné informácie aby mali rovnaký formát (pozri obr.3–4).
5. Nasledujúca časť sa zaoberá spracovávaním dát vo formáte .nc do podoby grafov. Túto časť zabezpečuje skript `prediction_plots.ipynb`. Na Obrázku 3–5 je možné vidieť, že sme z dát dočasne ukladali hodnoty zemepisnej dĺžky, šírky,

```
files = glob.glob(r'data/' + str(year) + '/*/*.nc')

for f in files:
    os.rename(f, f[0:45] + ".nc")
```

**Obr. 3–4** Skrátene názvu dát

žiarenia, vlnovej dĺžky a uhlu slnečného žiarenia. Tieto premenné sme následne upravili do podoby ktorá sa dala v grafe interpretovať ako hodnota Airglowu. Nakoniec sme pripravili projekciu Zeme na ktorú budeme neskôr pokladať graf

```
files = glob.glob(r'data/2021/*/*.nc')

for f in files:
    nc = Dataset(f)
    plt.close()
    lon = np.transpose(nc.variables['REFERENCE_POINT_LON'][:])
    lat = np.transpose(nc.variables['REFERENCE_POINT_LAT'][:])
    mean = lat.data;
    mean = mean[~np.isnan(mean)];
    mean = mean.mean();
    radiance = nc.variables['RADIANCE'][:];
    wavelength = nc.variables['WAVELENGTH'][:];
    wave_len = len(wavelength) / 2
    wave_len = int(wave_len)
    wave_len2 = len(wavelength[0])/ 2
    wave_len2 = int(wave_len2)
    sza = np.transpose(nc.variables['SOLAR_ZENITH_ANGLE'][:])
    O5s_ids = np.argwhere((135.0 <= wavelength[wave_len, wave_len2, :] & (wavelength[wave_len, wave_len2, :] <= 137.0))
    O5s = np.nansum(radiance[:, :, O5s_ids], axis = 2)*.04
    O5s = np.transpose(O5s[:, :, 0])

    LBH_ids = np.argwhere((141.0 < wavelength[wave_len, wave_len2, :] & (wavelength[wave_len, wave_len2, :] < 153.0))
    LBHs = np.nansum(radiance[:, :, LBH_ids], axis = 2)*.04
    LBHs = np.transpose(LBHs[:, :, 0])

    mp = Basemap(projection='geos',
                  rsphere=(6378137.00, 6356752.3142),
                  resolution='c',
                  lon_0=-45)
```

**Obr. 3–5** Spracovanie stiahnutých dát, príprava dát pre grafy a podklad Zemského povrchu

6. Na zobrazenie dát sme použili typ grafu `Pcolor` ktorý sme následne vykreslili na mapu Zeme. Tento proces sme opakovali pre každý súbor a po roztriedení na Severnú a Južnú pologuľu ukladali do priečinnov ktoré budú použité pre tréovanie modelov (pozri obr.3–6).
7. Pri vytváraní dát na pre exploráciu začiatok skriptov `plots.ipynb`, `all_plots_at_one.ipynb` a `overlap_plots.ipynb` je rovnaký ako na Obrázku 3–5. Pokračovanie sa však líši podľa požadovaného cieľa. Ak sú vytvárané grafy



```

x,y = mp(lon,lat)
hour = f[40:42]
minutes = f[43:45]
day_num = f[36:39]
if hour != "00":
    day_num.rjust(13 + len(day_num), '0')
    year = 2020
    start_date = date(int(year), 1, 1)
    result_date = start_date + timedelta(days=int(day_num) - 1)
    result = result_date.strftime("%Y-%m-%d")
    if 'CHA' in f:
        channel = "CHA"

    if 'CHB' in f:
        channel = "CHB"

c_scheme = mp.pcolor(x,y,0.5s,cmap = 'jet',vmin=0,vmax=40)
mp.drawcoastlines()

if mean > 0:
    plt.savefig("pix2pixHD/datasets/north_year/test_frames/" + result + "_" + hour + "_" + minutes + "_" + channel + "_1356.png")
if mean < 0:
    plt.savefig("pix2pixHD/datasets/south_year/test_frames/" + result + "_" + hour + "_" + minutes + "_" + channel + "_1356.png")

plt.close()

```

**Obr. 3–6** Vykreslenie, uloženie a roztriedenie grafov určených pre tréovanie modelov

aj pre žiarenie LBH, hodnoty SZA a grafu bez podkladu planéty je potrebné použiť podobný skript ako je možné vidieť na Obrázku 3–6 avšak s menšími zmenami a bez triedenia podľa pologúľ (pozri obr.3–7, 3–8, 3–9).

```

x,y = mp(lon,lat)
hour = f[40:42]
minutes = f[43:45]
day_num = f[36:39]
day_num.rjust(13 + len(day_num), '0')
year = 2021
start_date = date(int(year), 1, 1)
result_date = start_date + timedelta(days=int(day_num) - 1)
result = result_date.strftime("%Y-%m-%d")
Path("plots/" + result).mkdir(parents=True, exist_ok=True)
Path("plots/" + result + "/1356").mkdir(parents=True, exist_ok=True)
Path("plots/" + result + "/LBH").mkdir(parents=True, exist_ok=True)
Path("plots/" + result + "/SZA").mkdir(parents=True, exist_ok=True)
Path("plots/" + result + "/plot").mkdir(parents=True, exist_ok=True)
if 'CHA' in f:
    channel = "CHA"

if 'CHB' in f:
    channel = "CHB"

```

**Obr. 3–7** Vytvorenie priečinkov pre exploračné grafy

```

c_scheme = mp.pcolor(x,y,05s,cmap = 'jet',vmin=0,vmax=40)
plt.colorbar(c_scheme)
if 'CHA' in f:
    plt.title("Type: L1C CHA NI1 1356 \n Date: "+ result + " " + hour + ":" + minutes)

if 'CHB' in f:
    plt.title("Type: L1C CHB NI1 1356 \n Date: "+ result + " " + hour + ":" + minutes)
mp.drawcoastlines()

plt.savefig("plots/" + result + "/1356/" + result + "_" + hour + "_" + minutes + "_" + channel + "_1356.png")

plt.close()
c_scheme = mp.pcolor(x,y,LBHs,cmap = 'jet',vmin=0,vmax=80)
plt.colorbar(c_scheme)
if 'CHA' in f:
    plt.title("Type: L1C CHA NI1 LBH \n Date: "+ result + " " + hour + ":" + minutes)

if 'CHB' in f:
    plt.title("Type: L1C CHB NI1 LBH \n Date: "+ result + " " + hour + ":" + minutes)
mp.drawcoastlines()

plt.savefig("plots/" + result + "/LBH/" + hour + "_" + minutes + "_" + channel + "_LBH.png")

```

Obr. 3–8 Uloženie a vykreslenie pre grafy žiarení 1356 a LBH

```

plt.close()
c_scheme = mp.pcolor(x,y,sza,cmap = 'RdGy',vmin=0,vmax=180)
plt.colorbar(c_scheme)
if 'CHA' in f:
    plt.title("Type: L1C CHA NI1 SZA \n Date: "+ result + " " + hour + ":" + minutes)

if 'CHB' in f:
    plt.title("Type: L1C CHB NI1 SZA \n Date: "+ result + " " + hour + ":" + minutes)
mp.drawcoastlines()

plt.savefig("plots/" + result + "/SZA/" + hour + "_" + minutes + "_" + channel + "_SZA.png")

plt.close()
c_scheme = plt.pcolor(lon,lat,05s,cmap = 'jet',vmin=0,vmax=40)
plt.colorbar(c_scheme)
if 'CHA' in f:
    plt.title("Type: L1C CHA NI1 1356 \n Date: "+ result + " " + hour + ":" + minutes)

if 'CHB' in f:
    plt.title("Type: L1C CHB NI1 1356 \n Date: "+ result + " " + hour + ":" + minutes)
plt.savefig("plots/" + result + "/plot/" + hour + "_" + minutes + "_" + channel + "_1356_plot.png")

```

Obr. 3–9 Uloženie a vykreslenie pre grafy SZA a žiarenia 1356 bez podkladu Zeme

8. V prípade že je cieľom získanie grafov pokrytia je postup podobný ako v predošlých krokoch (pozri obr.3–5 a 3–6). Avšak je potrebné nakopírovať rovnaký kód pre každý súbor dňa. Jediná zmena je že všetky tieto grafy vykresľujeme na ten istý podklad Zeme. V prípade grafu prekrývania dát je taktiež treba upraviť graf Pcolor pridaním hodnoty `alpha` (pozri obr.3–10).

```
c_scheme2 = mp.pcolor(x2,y2,O5s2,cmap = 'binary',vmin=0,vmax=2,alpha = 0.05)
```

**Obr. 3–10** Vykreslenie prekrytia dát pomocou priehľadnosti

9. Nasledujúca časť sa zaoberá tréновaním modelu a vytváraním predikcií po natréновaní modelu. O túto časť sa stará skript `pix2pix.ipynb`. Trénovanie a predikovanie prebieha zavolaním skriptu ktorý je súčasťou stiahnutého algoritmu Pix2Pix a uvedením požadovaných parametrov a nastavení (pozri obr.3–11).

```
!python pix2pixHD/train_video.py --name year_south --dataroot pix2pixHD/datasets/south_year/ --save_epoch_freq 5 #--continue_train
...
!python pix2pixHD/generate_video.py --name year_north --dataroot pix2pixHD/datasets/north_year/ --how_many 6 --which_epoch 200
...
```

**Obr. 3–11** Spustenie tréновania modelu a vytváranie predikcií modelu

10. Nasledujúca časť sa zaoberá hodnotením presnosti predikcií. Prvým krokom je orezanie časti predikcií a originálnych obrázkov mimo povrchu Zeme, čím ich taktiež dostaneme na rovnaký formát a veľkosť. O tento krok sa stará skript `crop_images.ipynb` a je možné ho vidieť na obrázku 3–12.
11. Druhým krokom je použitie samotných metrík na orezané obrázky. O tento krok sa stará skript `compare.ipynb` za použitia knižnice `sewar` ktorá obsahuje výpočty metrík MSE a PSNR (pozri obr.3–13).

```
#files = glob.glob(r'pix2pixHD/datasets/south_year/for testing/*.png')
files = glob.glob(r'checkpoints/year_north/frames/*.jpg')

for filename in files:

    img = Image.open(filename)

    width, height = img.size
    x = (width - height)//2
    img_cropped = img.crop((x, 0, x+height, height))

    mask = Image.new('L', img_cropped.size)
    mask_draw = ImageDraw.Draw(mask)
    width, height = img_cropped.size
    mask_draw.ellipse((40, 35, width-31, height-37), fill=255)
    #mask.show()

    img_cropped.putalpha(mask)

    img_cropped.save(os.path.splitext(filename)[0] + ".png")
    #img_cropped.save(os.path.splitext(filename)[0] + "_2.png")

    img_cropped.show()
```

**Obr. 3 – 12** Orezanie predikcií a očakávaných obrázkov na rovnaký formát

```
#2020-01-10
im0 = Image.open("checkpoints/year_south/2020-01-10/frame-00005.png")
im0 = np.array(im0)
im1 = Image.open("checkpoints/year_south/2020-01-10/frame-00006.png")
im1 = np.array(im1)
im2 = Image.open("checkpoints/year_south/2020-01-10/frame-00007.png")
im2 = np.array(im2)
im3 = Image.open("checkpoints/year_south/2020-01-10/frame-00008.png")
im3 = np.array(im3)
im4 = Image.open("checkpoints/year_south/2020-01-10/frame-00009.png")
im4 = np.array(im4)
im5 = Image.open("checkpoints/year_south/2020-01-10/frame-00010.png")
im5 = np.array(im5)
org0 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_22_25_CHB_1356_2.png")
org0 = np.array(org0)
org1 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_22_55_CHB_1356_2.png")
org1 = np.array(org1)
org2 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_23_10_CHB_1356_2.png")
org2 = np.array(org2)
org3 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_23_25_CHB_1356_2.png")
org3 = np.array(org3)
org4 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_23_40_CHB_1356_2.png")
org4 = np.array(org4)
org5 = Image.open("pix2pixHD/datasets/south_year/for_testing/2020-01-10_23_55_CHB_1356_2.png")
org5 = np.array(org5)

data = [[mse(im0,org0), psnr(im0, org0)],
        [mse(im1,org1), psnr(im1, org1)],
        [mse(im2,org2), psnr(im2, org2)],
        [mse(im3,org3), psnr(im3, org3)],
        [mse(im4,org4), psnr(im4, org4)],
        [mse(im5,org5), psnr(im5, org5)]]
print(pd.DataFrame(data, columns=["MSE", "PSNR"]))
```

Obr. 3–13 Porovnanie predikcií s očakávanými obrázkami pomocou metrík MSE a PSNR