Super-Resolution of SOHO/MDI Magnetograms Using SolarCNN

Chunhui Xu et al.
New Jersey Institute of Technology

1. Introduction

We introduce a deep learning method named SolarCNN for enhancing AR line-of-sight (LOS) magnetograms obtained by the Michelson Doppler Imager (MDI) onboard SOHO. The model is trained to super-resolve MDI magnetograms using references from the Helioseismic and Magnetic Imager (HMI) onboard SDO.

In this document, we describe the usage of the trained SolarCNN model to super-resolve the test data.

Since model training requires GPUs, it is omitted here.

2. Prediction Workflow

2.1 Load Model

The SolarCNN model has been saved using the model.save() function in TensorFlow. It includes custom loss and metric functions: mix_loss and ssim_metric, which must be specified when loading the model.

```
import tensorflow as tf
from model import build_model, mix_loss, ssim_metric
try:
    model = tf.keras.models.load_model("model/model_solarcnn",
                                        custom_objects={'mix_loss': mix_loss, 'ssim_metric': ssim_metric})
    print("model loaded")
except:
    print("model loading error")

```

✓ 4.4s

```
model loaded
```

2.2 Load Input Data

Test data are stored as FITS files in the directory data/image/test/. Each file is loaded and sorted by filename.

```python
from astropy.io import fits
import os

def load_data(path):
    data_row = [(name, fits.open(os.path.join(path, name))[0].data) for name in os.listdir(path)]
    data_sort = sorted(data_row, key=lambda x: x[0])
    data = [item[1] for item in data_sort]
    name = [item[0] for item in data_sort]
    return tf.Variable(initial_value=data, dtype=tf.float32), name

try:
    test_input, test_name = load_data("data/image/test/")
    print("data loaded, total sample:", len(test_name))
except:
    print("data loading error")
```

✓ 0.3s

```
data loaded, total sample: 3
```

2.3 Perform Super-Resolution

Each input magnetogram is passed to the SolarCNN model for prediction. The output is a super-resolved magnetogram saved in the FITS format with filenames of the form:

enh_mdi_YYYYMMDD_HHMMSS_SolarCNN.fits

Here's the prediction and saving loop:

```python
output_path = "data/pred/"
for i in range(test_input.shape[0]):
    print("Predict sample "+str(i+1))
    test_sample = tf.reshape(test_input[i], (-1, 256, 256, 1))
    prediction = model.predict(test_sample/2000.)
    pred_name = "enh_mdi_" + test_name[i][4:20] + "SolarCNN.fits"
    fits_data = fits.PrimaryHDU(prediction[0]*2000.)
    fits_data.writeto(os.path.join(output_path, pred_name), overwrite=True)
print("Finished")
```

✓ 1.9s

```
Predict sample 1
1/1 [==============================] - 1s 847ms/step
Predict sample 2
1/1 [==============================] - 0s 497ms/step
Predict sample 3
1/1 [==============================] - 0s 492ms/step
Finished
```

3. Conclusion

We present a deep learning model, SolarCNN, for enhancing AR LOS magnetograms of SOHO/MDI. This method can be applied to existing MDI data to produce high-resolution magnetic field maps.