Decoding&Visualization of VIIRS Active Fire Product

Erdem ERDİ
Remote Sensing Division

6th SALGEE Workshop 14-17 October 2019, Darmstadt

Outline

- Active Fire Product
- Decoding Software
- To be added features

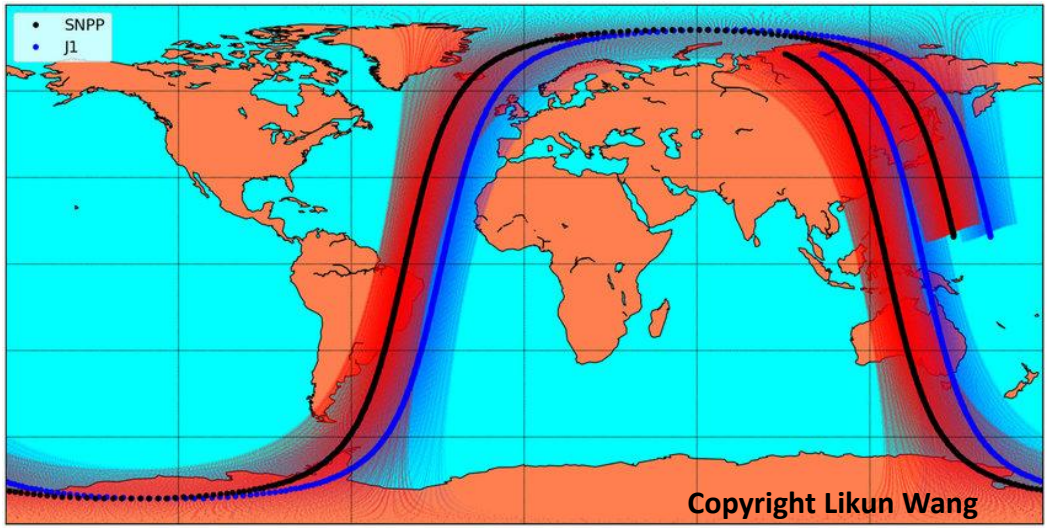
Active Fire Product

- VIIRS Active Fire EDR product (VNP14, not VNP14IMG):
  - Produced from both the SNPP and NOAA-20 satellites
  - Distribution through EUMETCast Europe & Africa on "EUMETSAT Data Channel 12" and "A1C-TPL-1" channels respectively.
  - Granules of approx. 1.5 minutes, 1000 files/satellite/day average
  - 750 meter resolution (not 375 meter)
  - Algorithm and performance are similar to MOD14 MODIS product, collection 6.

Active Fire Product

- SNPP and NOAA-20 are on same orbital plane

- NOAA-20 is about 50 min. ahead of SNPP



- For mid northern latitudes the ΔTime between the observation and product reception :
  - ~25 mins. Ascending and ~45 mins. Descending for NOAA-20
  - ~25 mins. Ascending and ~80 mins. Descending for SNPP

Decoding Software

- **Written in Python (3), cross platform**
- **Suitable for both the operational use or batch processing**
- **Easy configuration by the help of a self explaining ".ini file"**

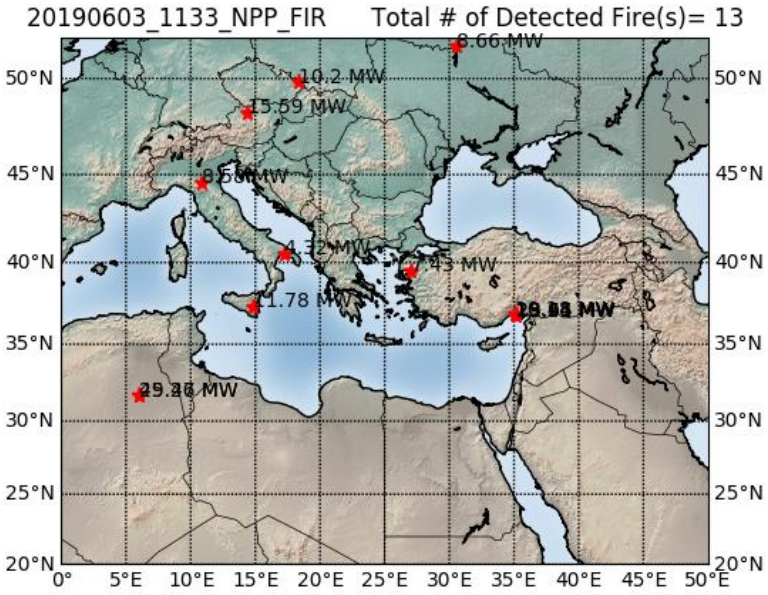
```
    swathLons.append(float(pair.split(' ')[0]))
    swathLats.append(float(pair.split(' ')[1]))


#print(swc)


# A simplified polygon of the swath is generated.
swath_coords = [(swath_ff_lat,swath_ff_lon),(swath_fl_lat,swath_fl_lon),(swath_lf_lat,swath_lf_lon),(swath_ll_lat,swath_ll_lon)]
swath_polygon = Polygon(swath_coords)

# If the polygon (swath) intersects the area of the interest...
if swath_polygon.intersects(aoi_polygon):
  print("Processing the granule: " + time_coverage_start.strftime('%Y-%m-%d %H:%M'))

  # Reading fire detection info from the product file
  readTables()
  # Number of fire detected pixels stored in the count variable
  count = len(latitude)

  if len(os.listdir(TemporaryFolder)) > 0:        # There is interim/temporary file(s) in the temporary folder
    interimFileTime = datetime.strptime(sorted(os.listdir(TemporaryFolder))[-1][:-18], '%Y%m%d_%H%M') # Finding the latest file
    diffMinutes = int(abs((time_coverage_start - interimFileTime).seconds / 60)) # Time diff between granule and latest interim file
    if diffMinutes < 60:                          # Interim file is from the same orbit (difference less than 60 mins.)
      with open(TemporaryFolder + '/' + sorted(os.listdir(TemporaryFolder))[-1], "a") as interimFile:
        for k in range(0,count):
          fireLocation = Point(latitude[k], longitude[k])
          if fireLocation.within(aoi_polygon): # Appending the fire detected pixel info if the pixel is in the aoi.
            writeToInterimFile()
    else:                                         # Interim file is from different/previous orbit
      endCurrentOrbit()                           # Ending the current orbit before starting the new orbit
      TemporaryFilename = TemporaryFolder + '/' + time_coverage_start.strftime('%Y%m%d_%H%M') + '_' + satellite_name + '_FIR.temporary'
      with open(TemporaryFilename, "a") as interimFile: # New interim file is created
        for k in range(0,count):
          fireLocation = Point(latitude[k], longitude[k])
          if fireLocation.within(aoi_polygon):
            writeToInterimFile()
  else:                                           #There is not any interim file in the temporary folder
    TemporaryFilename = TemporaryFolder + '/' + time_coverage_start.strftime('%Y%m%d_%H%M') + '_' + satellite_name + '_FIR.temporary'
    with open(TemporaryFilename, "a") as interimFile: # New interim file is created
      for k    in range(0,count):
```

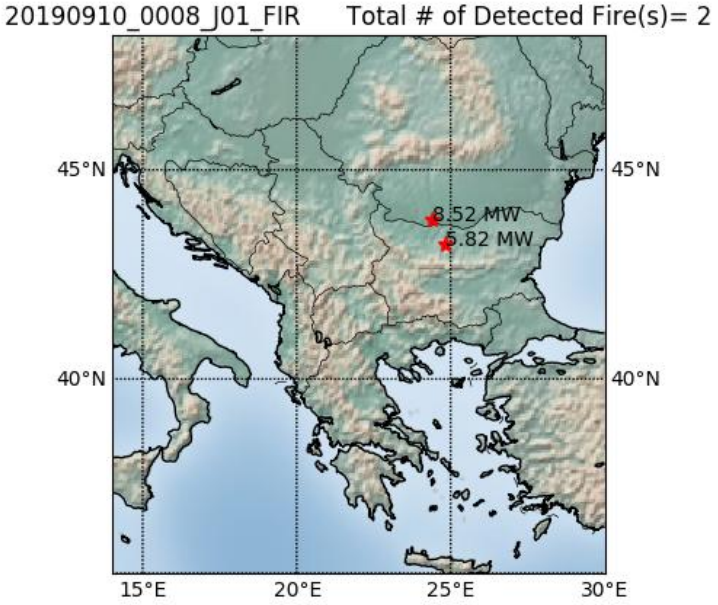
```
[FOLDERS]
InputFolder = ./input
OutputFolder = ./output
ArchiveFolder = ./archive

[AOI]
ll_lat = 40.00
ll_lon = 21.00
ur_lat = 46.00
ur_lon = 31.00

[ACTIONS]
CreateText = True
CreateImage = True
CreateShp = True
CreateGeotiff = True
DeleteIncomingFiles = True
ArchiveIncomingFiles = True
```

- **Easy and free to modify the source code, copyleft**

Decoding Software

- AOI is defined by user
- Consequent granule files which intersect with AOI are concatenated
- Images with spotted fire locations and calculated FRP values are generated

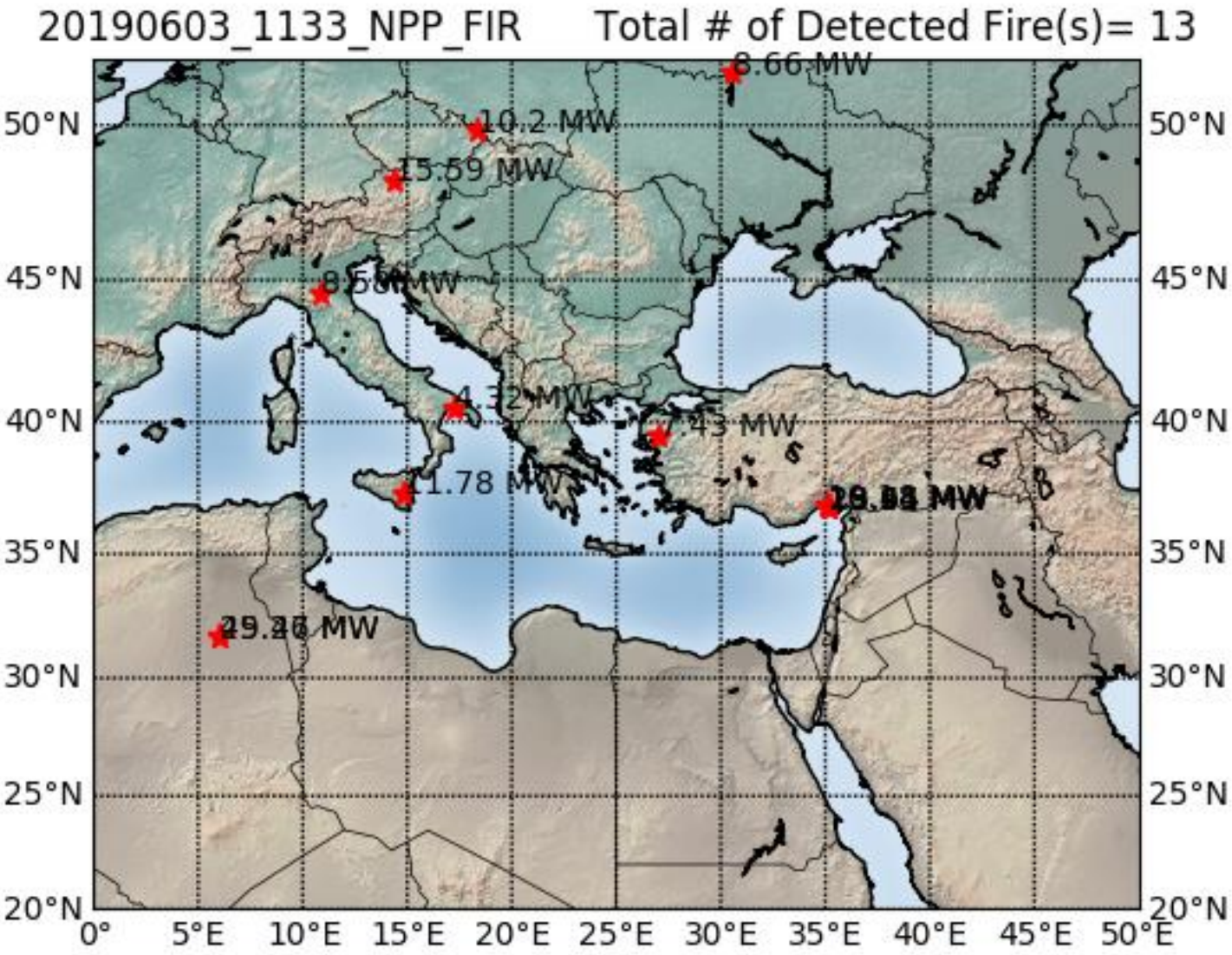




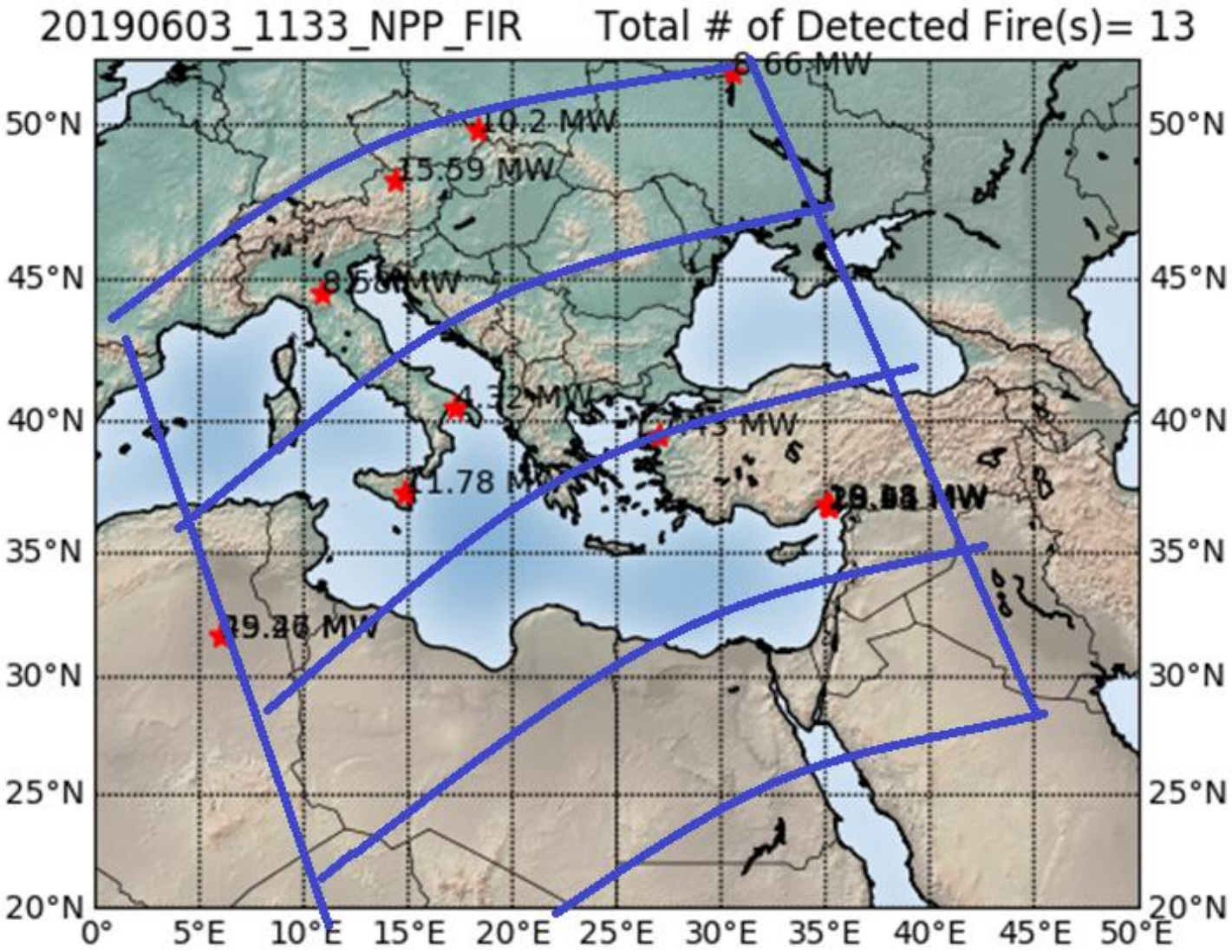
Decoding Software

- Regular text file outputs
- For easier ingestion of product into the GIS environment:
  - Shapefiles (point structure) along with associated .dbf and .prj files
  - GeoTIFF files with fire location pixels containing FRP value
  - Google Earth .kml files (to be implemented)
- CAP files (to be implemented)
- Only intersected granules are archived

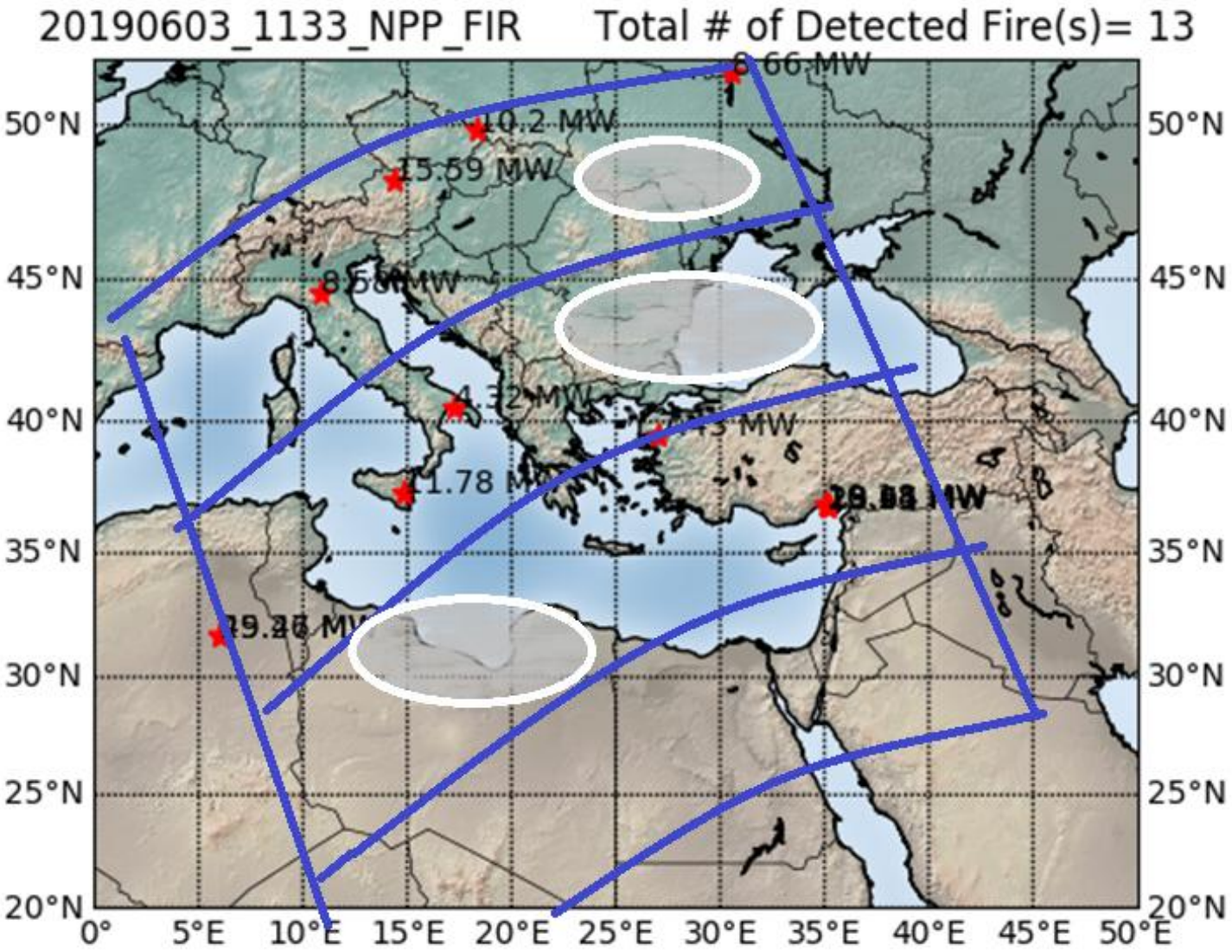
To be added features



To be added features



To be added features



To be added features

- **Extension of the code to cover other fire detection products distributed on EUMETCast:**
  - **MPEF FIR product**
  - **LandSAF FRP product**
  - **LandSAF FD&M product**

THANK YOU.