1. Utworzenie tabeli

```
CREATE DATABASE cw7;

CREATE EXTENSION postgis;

CREATE EXTENSION postgis_raster;
```

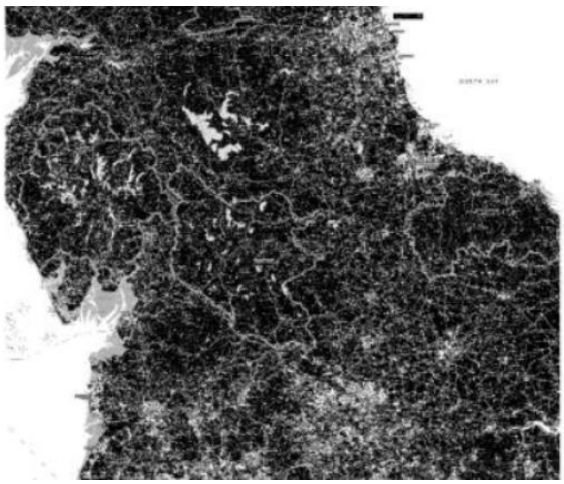
2. Załadowanie danych

Do zadania wybrano ograniczoną liczbę rastrów z powodu problemów obliczeniowych.

```
raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d
C:\Users\THINK\Downloads\ras250_gb\ras250_gb\data\data_cut\*.tif uk_250k | psql -d cw7 -h localhost -U
postgres -p 5432
```

```
CREATE INDEX idx_intersects_rast_gist ON uk_250k

USING gist(ST_ConvexHull(rast));
```

```
SELECT AddRasterConstraints('public'::name,

'uk_250k'::name,'rast'::name);
```

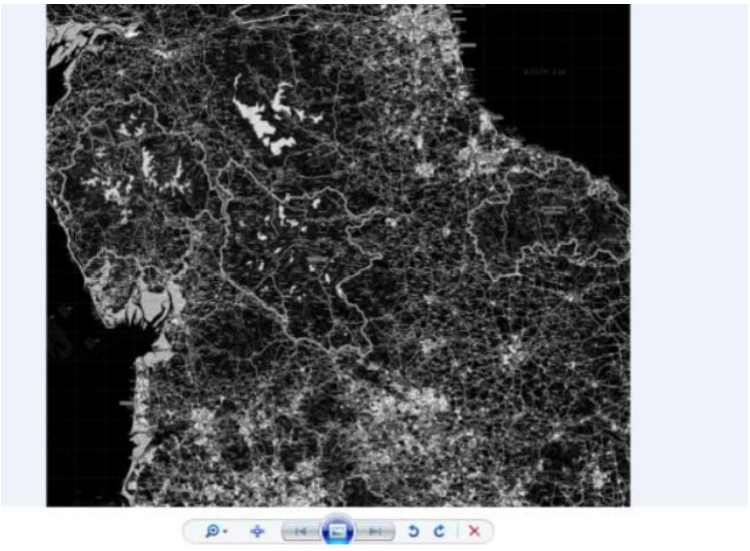


3. Połączenie danych I eksport

```
CREATE TABLE uk_250k_union AS

SELECT ST_UNION(rast)

FROM uk_250k
```

```
ALTER TABLE uk_250k_union
```

```
ADD COLUMN rid SERIAL PRIMARY KEY;

CREATE INDEX idx_intersects_rast_gist_union ON uk_250k_union

USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('public'::name,

'uk_250k_union'::name,'rast'::name);

SELECT ST_AsGDALRaster(rast, 'GTiff', ARRAY['COMPRESS=DEFLATE',

'PREDICTOR=2', 'PZLEVEL=9'])

FROM uk_250k_union;

CREATE TABLE tmp_out AS

SELECT lo_from_bytea(0, ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2',
'PZLEVEL=9'])  ) AS loid

FROM uk_250k_union;

SELECT lo_export(loid, 'C:\uk_250k.tiff')

FROM tmp_out;

SELECT lo_unlink(loid)

FROM tmp_out;
```

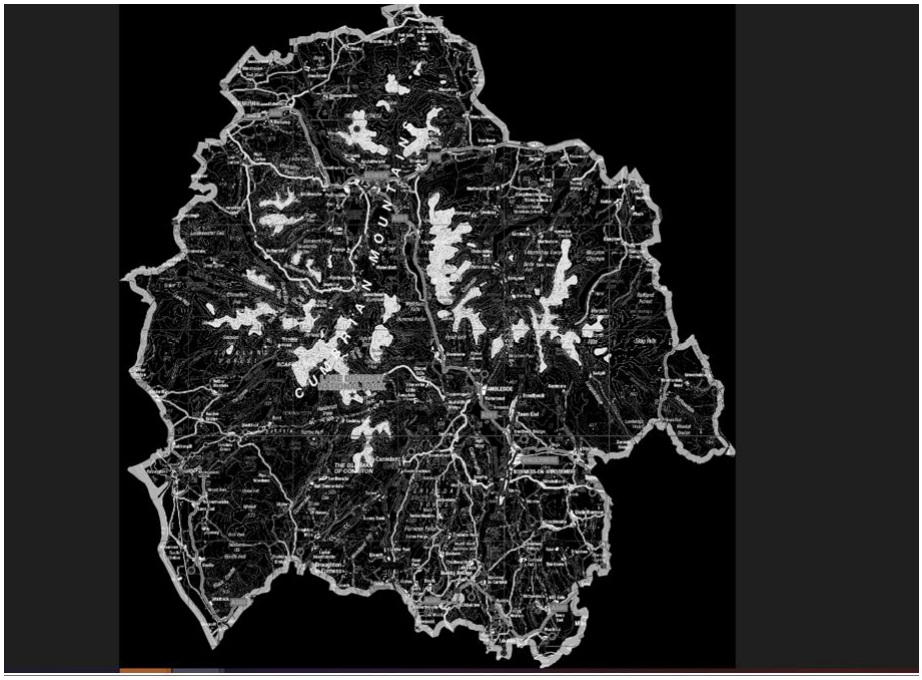


4/5/6. Pobranie danych, załadowanie do tabeli oraz utworzenie tabeli uk_lake_district

```
update national_parks
set geom = st_setsrid(geom, 27700);
create table uk_lake_district as (select st_union(st_clip(rast, geom)) as rast
                    from uk_250k u
                        inner join national_parks np on st_intersects(geom, rast)
                    where np.gid = 1);
```

7.

```
CREATE TABLE tmp_out_clipped AS
SELECT lo_from_bytea(0, ST_AsGDALRaster(ST_Union(st_union), 'GTiff', ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
 ) AS loid
FROM uk_lake_disctrict;

SELECT lo_export(loid, 'C:\uk_lake_district.tiff')
FROM tmp_out_clipped;

SELECT lo_unlink(loid)
FROM tmp_out_clipped;
```



8/9/10.

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d sentinel.tiff sentinel | psql -d cw7 -h localhost -U postgres -p 5432

```
create or replace function ndvi(
value double precision [] [] [],
pos integer [][],
VARIADIC userargs text []
)
RETURNS double precision AS
$$
BEGIN

RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value
[1][1][1]); --> NDVI calculation!
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
```

```
CREATE TABLE uk_lake_ndvi AS

WITH r AS (select st_clip(rast, st_transform(geom,32630)) as rast

                from sentinel u

                    inner join national_parks np on st_intersects(st_transform(geom,32630), rast)

                where np.gid = 1)

SELECT

    ST_MapAlgebra(

        r.rast, 1,

        r.rast, 4,

        '([rast2.val] - [rast1.val]) / ([rast2.val] +

        [rast1.val])::float', '32BF'

      ) AS rast

FROM r;
```

11.

```
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0,
                ST_AsGDALRaster(st_union(rast), 'GTiff', ARRAY ['COMPRESS=DEFLATE',
                  'PREDICTOR=2', 'PZLEVEL=9'])
        ) AS loid
FROM uk_lake_ndvi;



SELECT lo_export(loid, 'c:\raster.tiff')
FROM tmp_out;

SELECT lo_unlink(loid)
FROM tmp_out;
```

Niestety powyższa procedura nie działa z powodu brak zgodności układów współrzędnych.