# r/Place

## Community detection

Martin Bakoš

8.1.2024

# PRESENTATION STRUCTURE

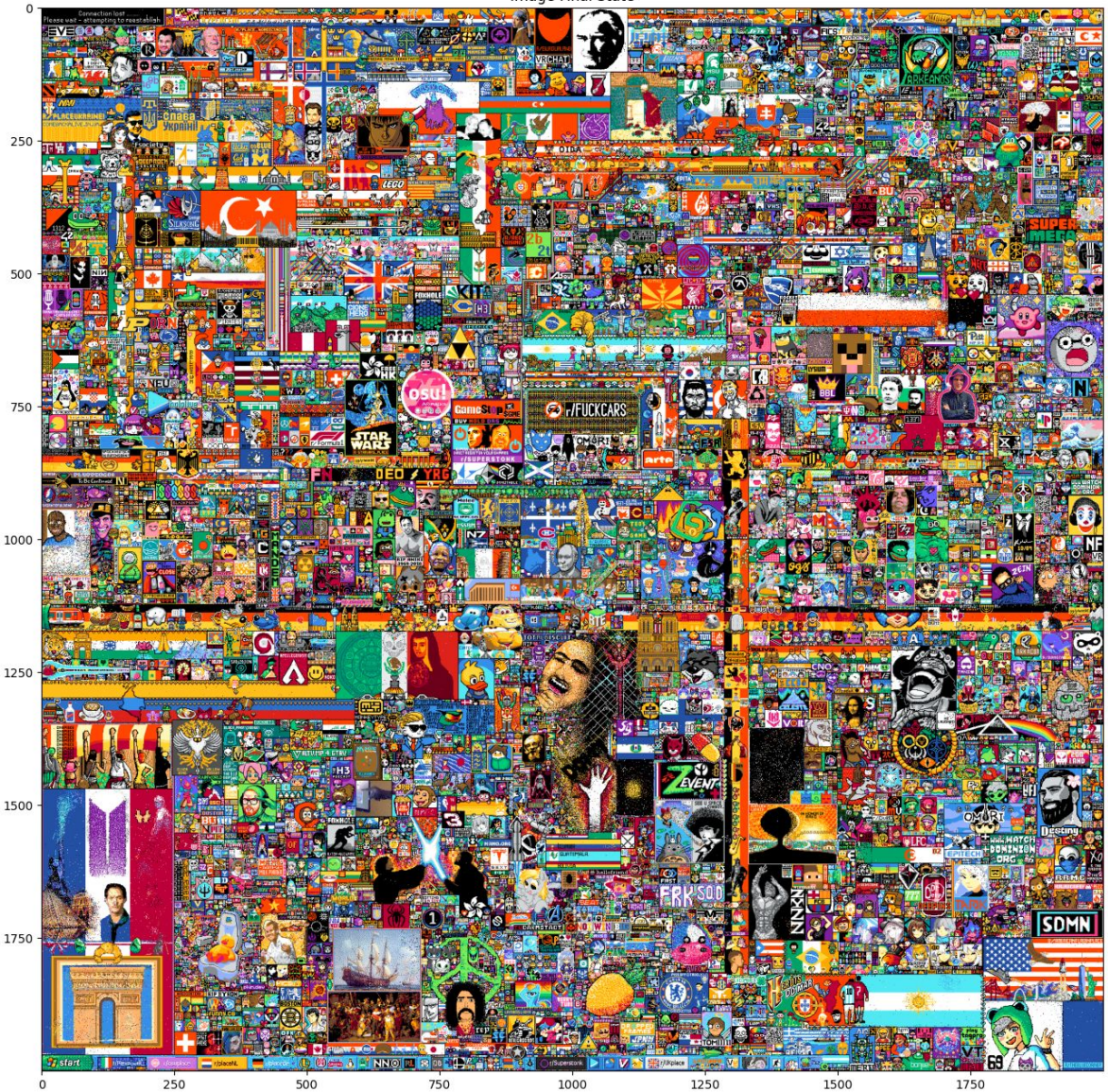
- r/Place
- Dataset
  - Preprocessing
  - Data Exploration
- Edge Computation
  - Edge
  - Color
  - Computation Speed
- Graph Analysis
- Communities
  - Results
- Summary

# WHAT IS r/Place

# r/Place

- A Reddit social experiment
- Users collaboratively contribute to a pixel art canvas.
- Users can place a pixel of their chosen color (32 options) every 5 minutes
- Users can influence and react to the evolving canvas
- It showcases the online collaboration and creativity of Reddit **communities**.

Image Final State



# DATASET

# DATASET INFO

- **Timestamp** -
  - the UTC time of the tile placement
  - 2022-04-03 17:38:22.252
- **User_id** -
  - a hashed identifier for each
  - UTC,yTrYCd4LUpBn4rIyNXkkW2+Fac5cQHK2lsDpNghkq0oPu9o//8oPZPlLM4CXQeEIId7l011MbHcAaLyqfhSRoA==
- **Pixel_color** -
  - the hex color code of the tile placed
  - #FF3881
- **Coordinate** -
  - the “x,y” coordinate of the tile placement.
  - 0,0

# DATASET CLEANING

- **Timestamp -**
  - UTC time => seconds after referenced time
- **User_id -**
  - hashed identifier => map to integers
- **Pixel_color -**
  - hex color => map to integers
  - save the mapping
- **Coordinate -**
  - “x,y” coordinate => separate to X and Y column

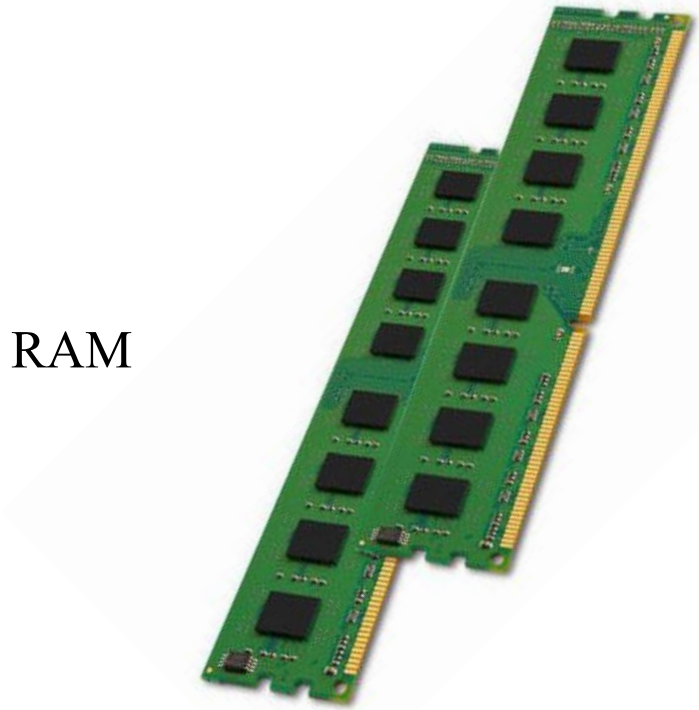
# DATASET REQUIREMENTS

**The dataset is really big:**

- ZIPED 12,1 GB
- UNZIPED 21,2 GB
- CLEANED 5,7 GB

=> Memory heavy

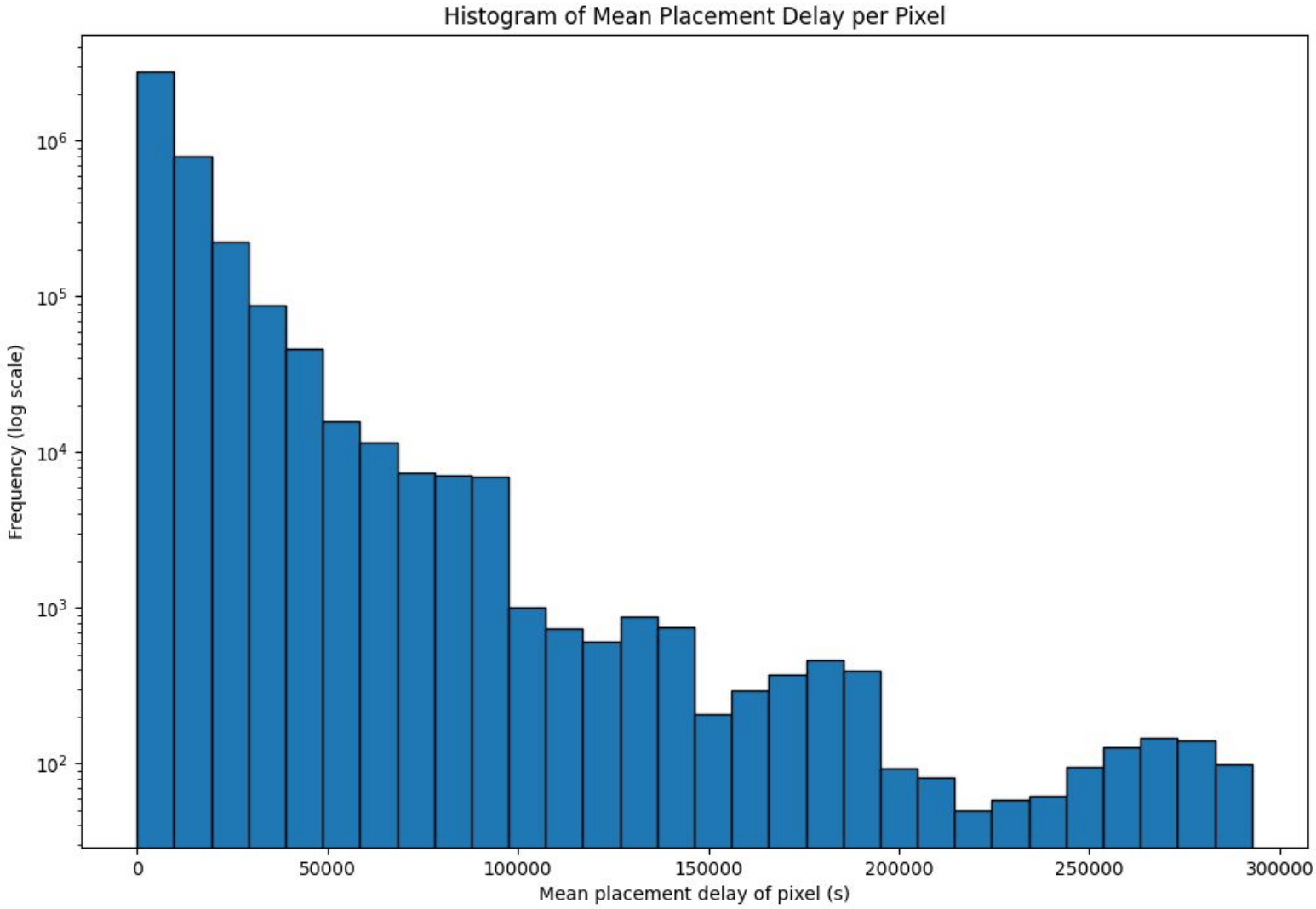
Running on PC with **48GB** RAM
Still kind of problematic



# DATA EXPLORATION

# DATASET STATISTICS

- **Number of entries:** 155925983
- **Unique colors:** 32
- **Unique users:** 10257345
- **Duration:**
  - 295409.875s = 3d 10h 3m 30.0s
- **Placement delay:**
  - mean = 0.00189s
  - std = 0.04260s
- **Placement delay per pixel:**
  - mean = 2h 36m 2.8s
  - std = 3h 24m 0.31s


Histogram of Mean Placement Delay per Pixel
Frequency (log scale)
$10^6$
$10^5$
$10^4$
$10^3$
$10^2$
0
50000
100000
150000
200000
250000
300000
Mean placement delay of pixel (s)

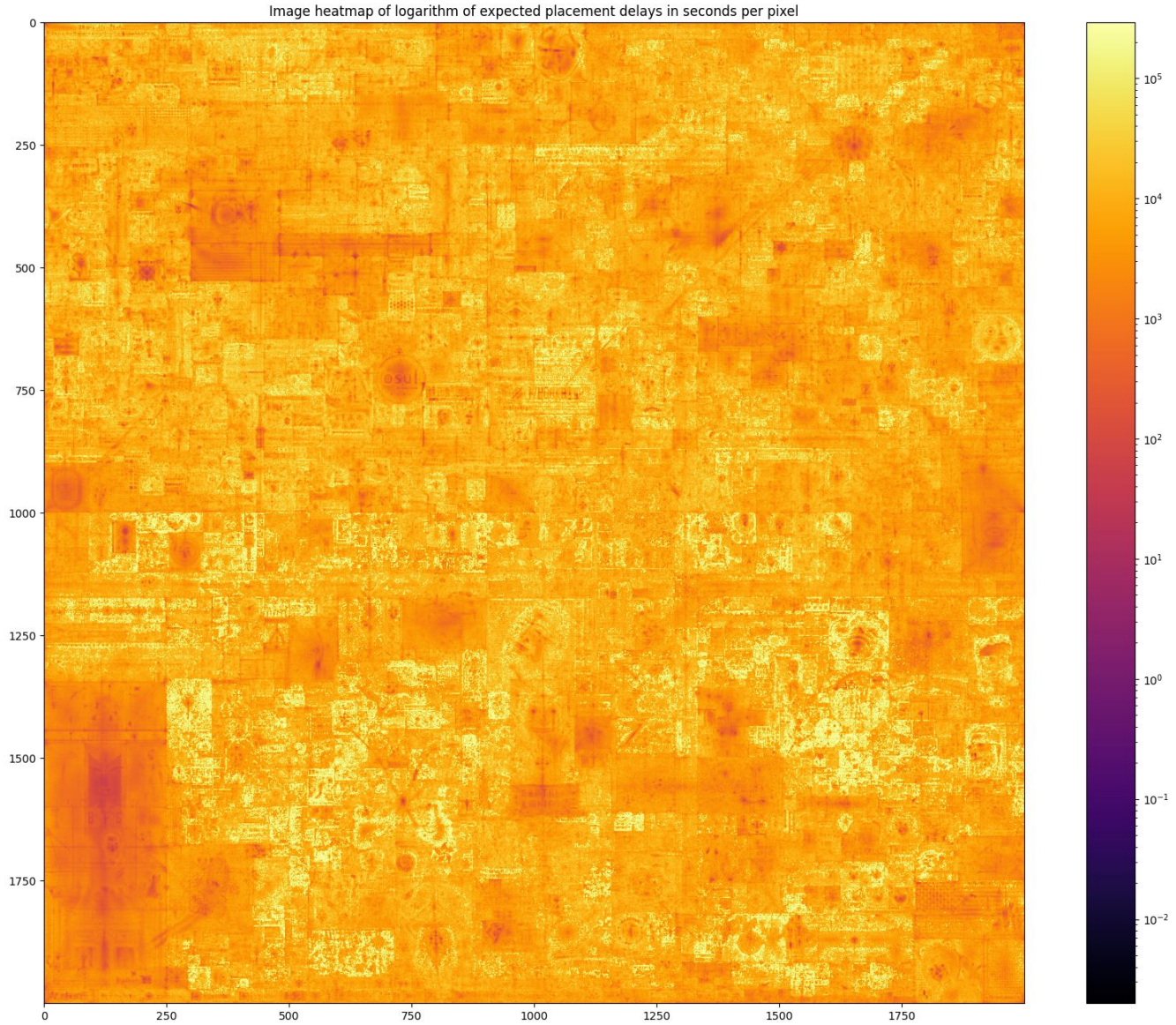


Image heatmap of logarithm of expected placement delays in seconds per pixel
0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750
$10^5$
$10^4$
$10^3$
$10^2$
$10^1$
$10^0$
$10^{-1}$
$10^{-2}$

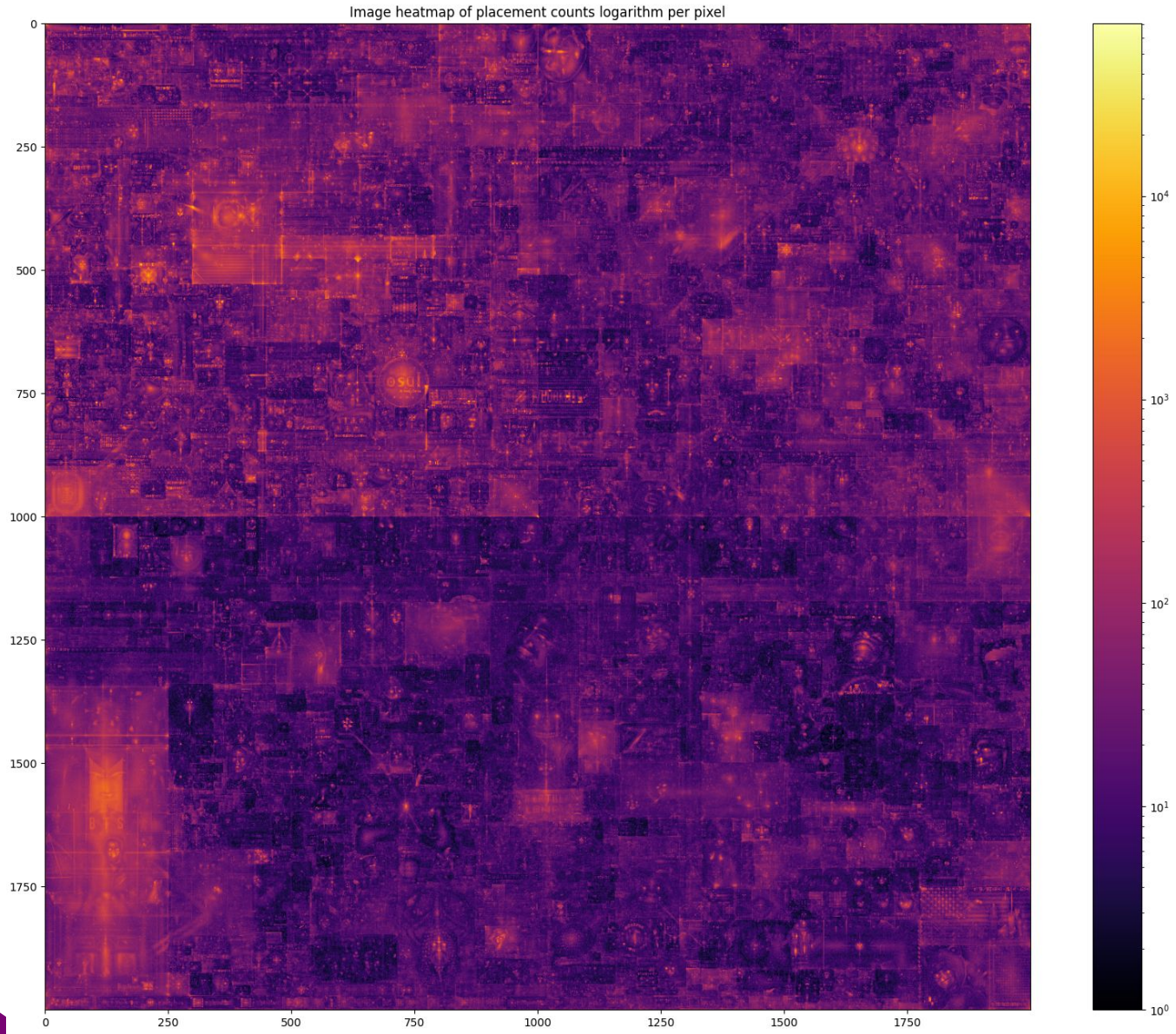


Image heatmap of placement counts logarithm per pixel
0
250
500
750
1000
1250
1500
1750
$10^4$
$10^3$
$10^2$
$10^1$
$10^0$

# EDGE COMPUTATION

# EDGE

**When are two user interacting?**

- If they are
  - Placing color on the same pixel
  - Placing color on neighbouring pixels

  in some time window

**When is the interaction positive?**

- Users are placing similar colors

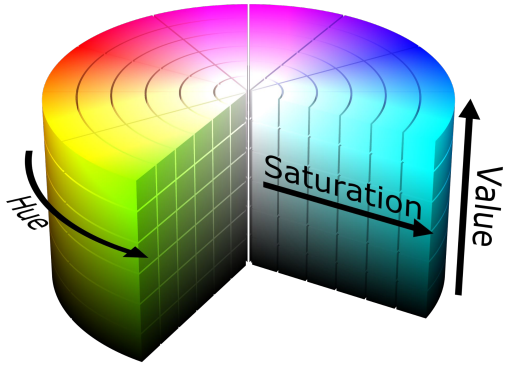
# SIMILAR COLORS

**What colors are “visualy”similar?**

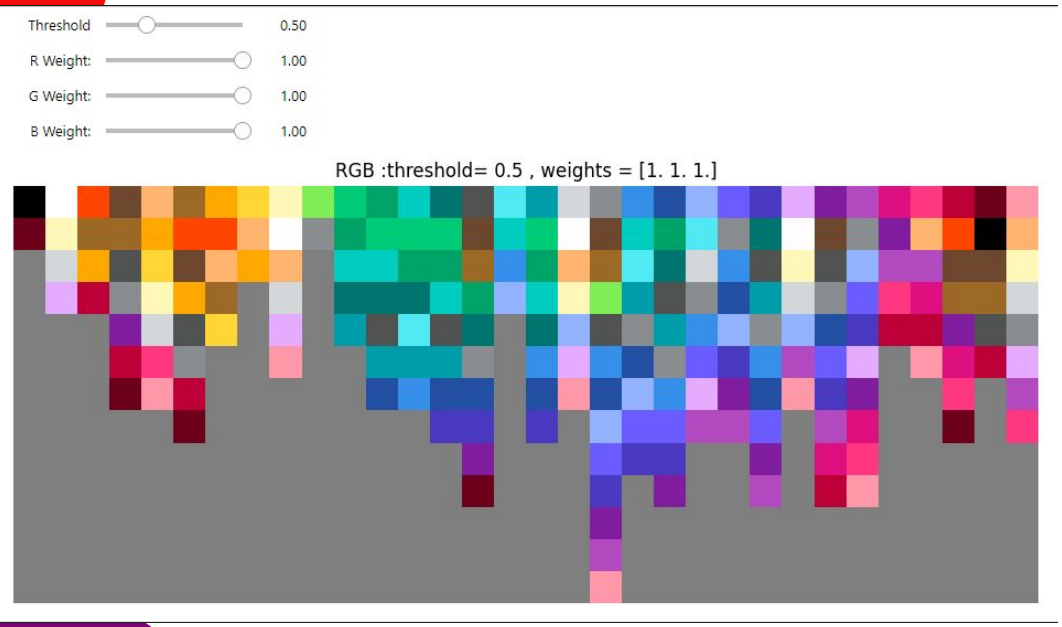
- No easy definition, highly subjective
- However we can measure a distance in some color space representation

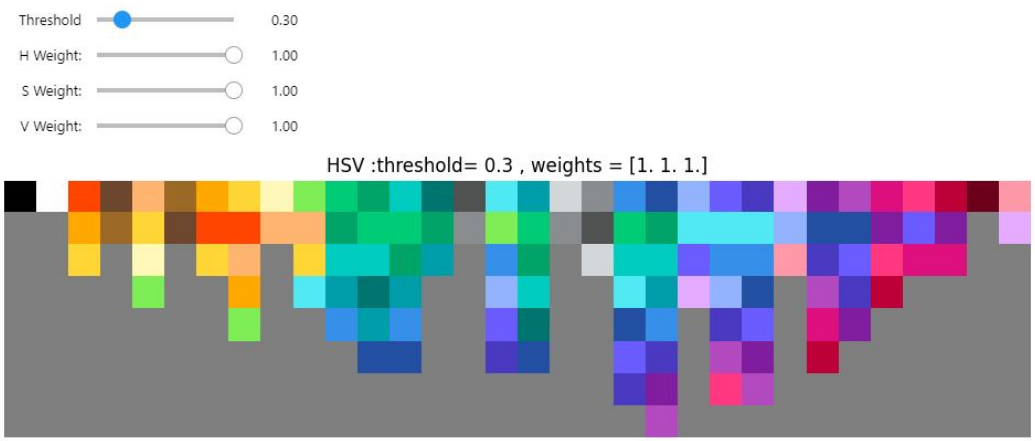
Mesuringing color similarity for representations:

- RGB
- HSV

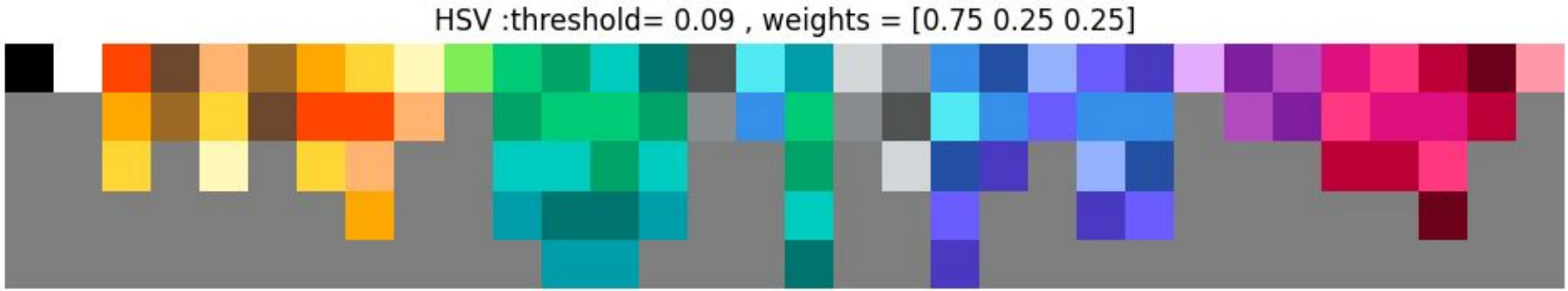


# SIMILAR COLORS - TOOL



# SIMILAR COLORS - RESULT




Based on this we can create matrix of color similarity

# GRAPH CREATION ALGORITHM

**Idea of edge reaction algorithm**:

For each entry:

For each interaction:

On positive interaction between users:

Add an edge with weight one or raise the weight by one.

We can do second pass through to subtract negative interaction and prune the graph.

# COMPUTATION

Sorting the data depending on approach by:

- X,Y,time
- time

Time for **1 000 000** entries:

- **Python** compare ~ (n[['X','Y']]==m[['X','Y']]).all()
  - Extremely slow
  - ≈ 35min
- **Python** grouped ~ df.groupby(['X','Y'])
  - Takes a long time
  - ≈2min
- **C++** Implementation
  - Memory efficient and quick
  - ≈ 3sec

# COMPUTED VARIANTS

**Same pixel** interacting with previous n users:

- 3 - good size
- 5 - on the edge of memory
- 10 - too large

**Neighborhood** interacting with n users:

- 4 - in Plus shape - on the edge of memory
- 8 - in Square shape - too large

# GRAPH ANALYSIS

Because of Networkx inefficiency for large graphs  igraph library was used.

# GRAPH PRUNING

| **Same pixel** | 3 users | 5 users |
|---|---|---|
| Number of edges in original | 150 324 900 | 251 315 192 |
| Average weight difference per edge | 0.0225 | 0.0262 |
| Average weight difference per edge from edges with difference | 1.207 | 1.240 |
| Count of edges in original that are not in pruned graph | 2 168 981 | 4 049 196 |
| Percentage of edges in original that are not in pruned graph | 1.4429% | 1.6112% |

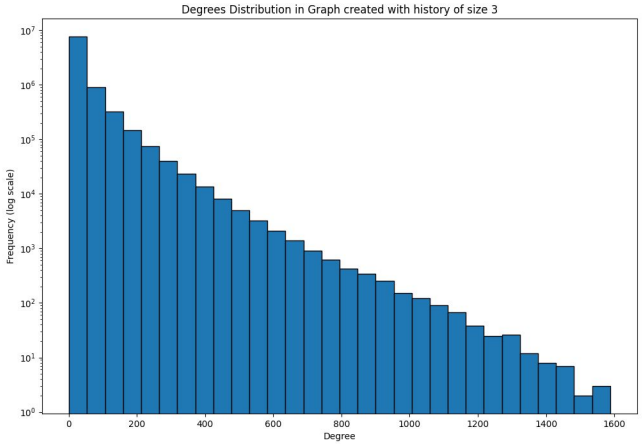
# INTERACTION DEFINITION

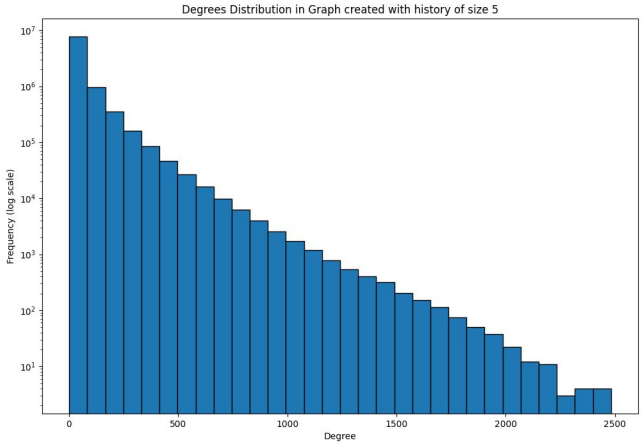
| **Same pixel** 5 users vs.<br>**Neighborhood** 4 users | Results |
|---|---|
| Average weight difference per edge | 0.0545 |
| Average weight difference per edge from edges with difference | 0.0553 |
| Percentage of edges in union that are not in **Same pixel** graph | 50.899% |
| Percentage of edges in union that are not in **Neighborhood** graph | 46.812% |
| Percentage of edges in union that are in both graphs | 2.289% |

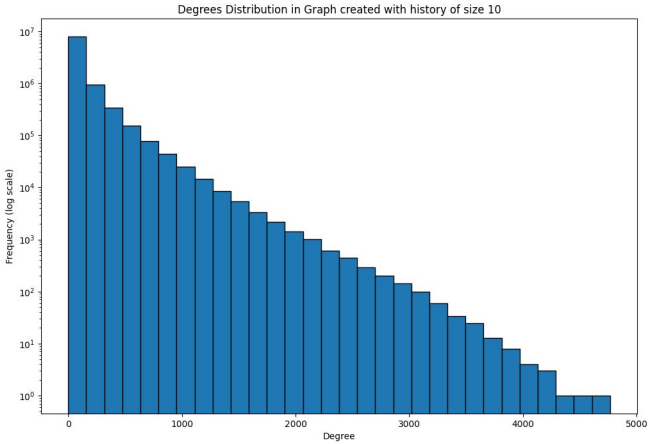
# SAME PIXEL - GRAPH COMPARISON

| **Same pixel** | 3 users | 5 users | 10 users |
|---|---|---|---|
| Isolated vertices count | 1025945<br>10% | 825975<br>8% | 625495<br>6% |
| Non-Isolated vertices count | 9231400<br>90% | 9431370<br>92% | 9631850<br>94% |
| Number of edges | 148 155 919 | 247 265 996 | 459 297 221 |
| Edge density | 2.816e-06 | 4.700e-06 | 8.731e-06 |

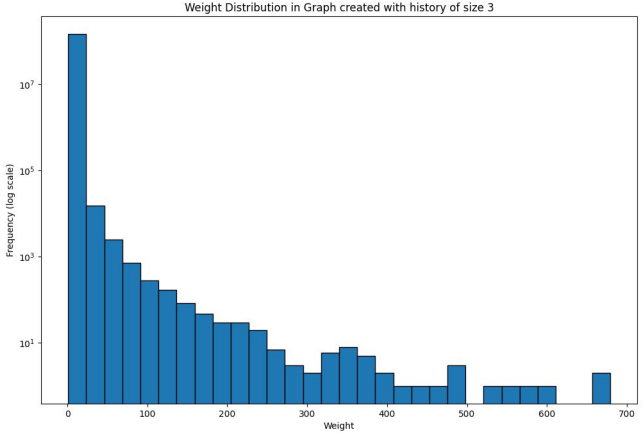
# Degrees

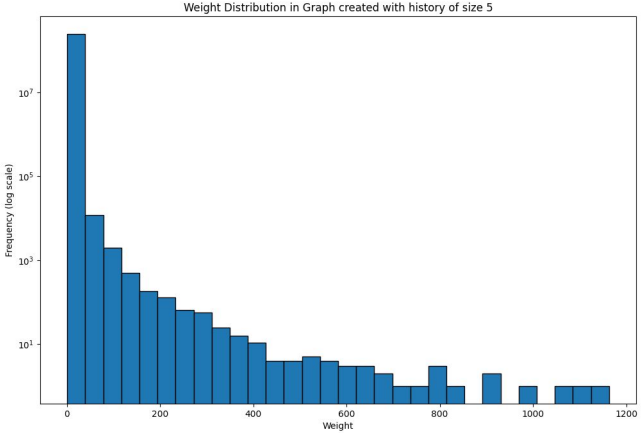


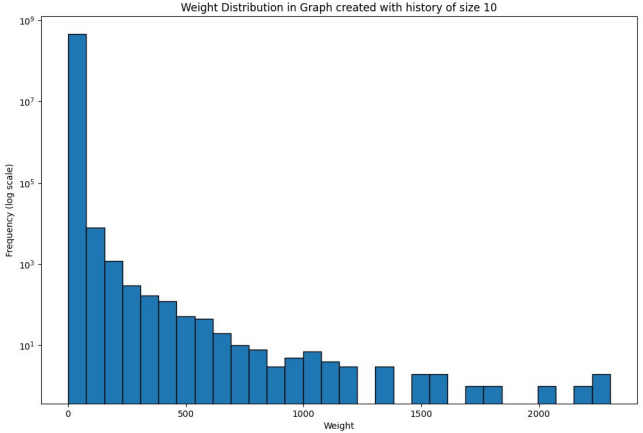




# Weights

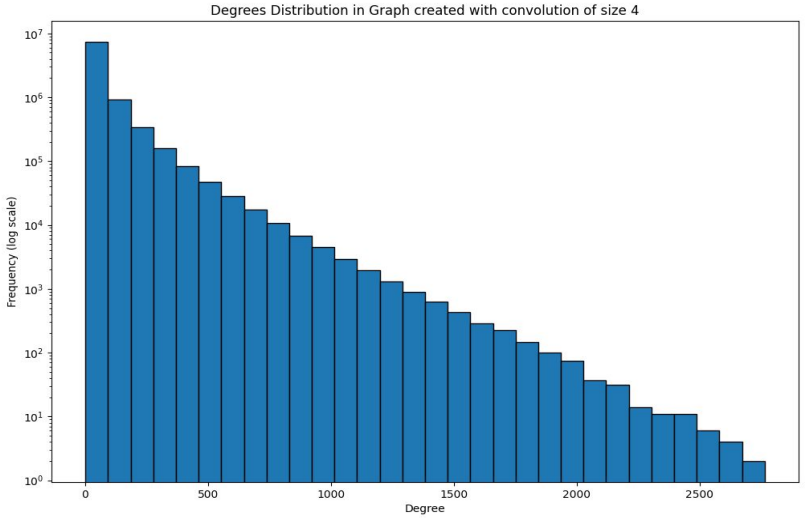


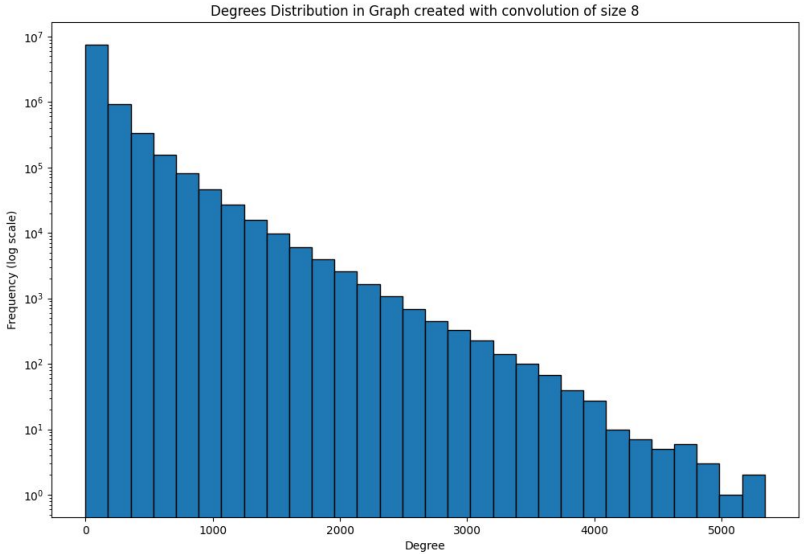



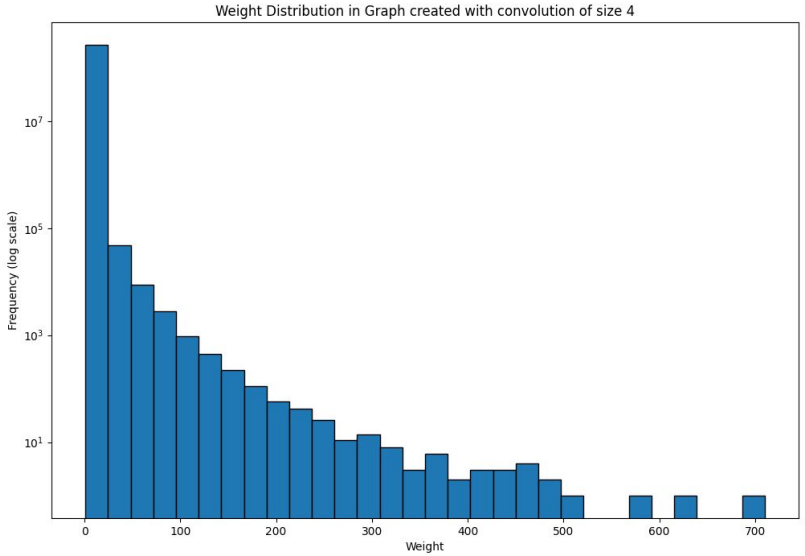



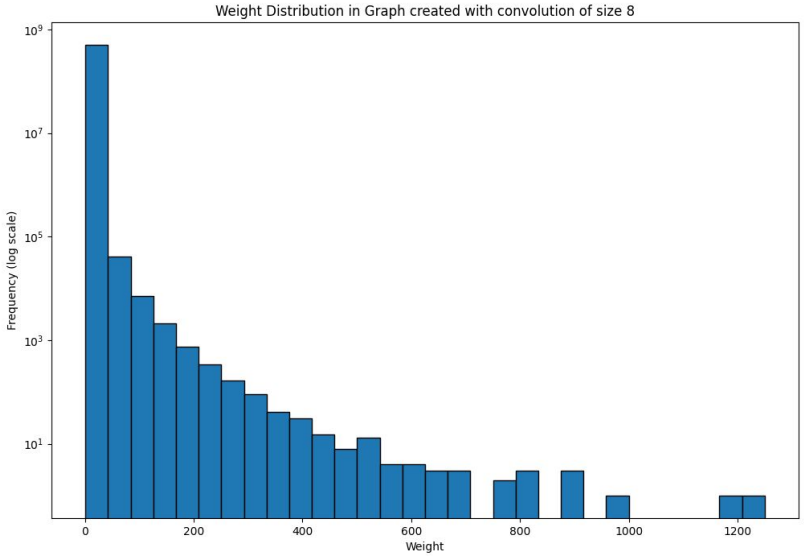
# NEIGHBOURHOOD - GRAPH COMPARISON

| **Neighbourhood** | 4 users | 8 users |
|---|---|---|
| Isolated vertices count | 1264558<br>12% | 1130168<br>11% |
| Non-Isolated vertices count | 8992787<br>88% | 9127177<br>89% |
| Number of edges | 267 851 035 | 505 284 112 |
| Edge density | 5.092e-06 | 9.605e-06 |


Degrees Distribution in Graph created with convolution of size 4
Frequency (log scale)
Degree


Degrees Distribution in Graph created with convolution of size 8
Frequency (log scale)
Degree


Weight Distribution in Graph created with convolution of size 4
Frequency (log scale)
Weight


Weight Distribution in Graph created with convolution of size 8
Frequency (log scale)
Weight

# COMMUNITIES

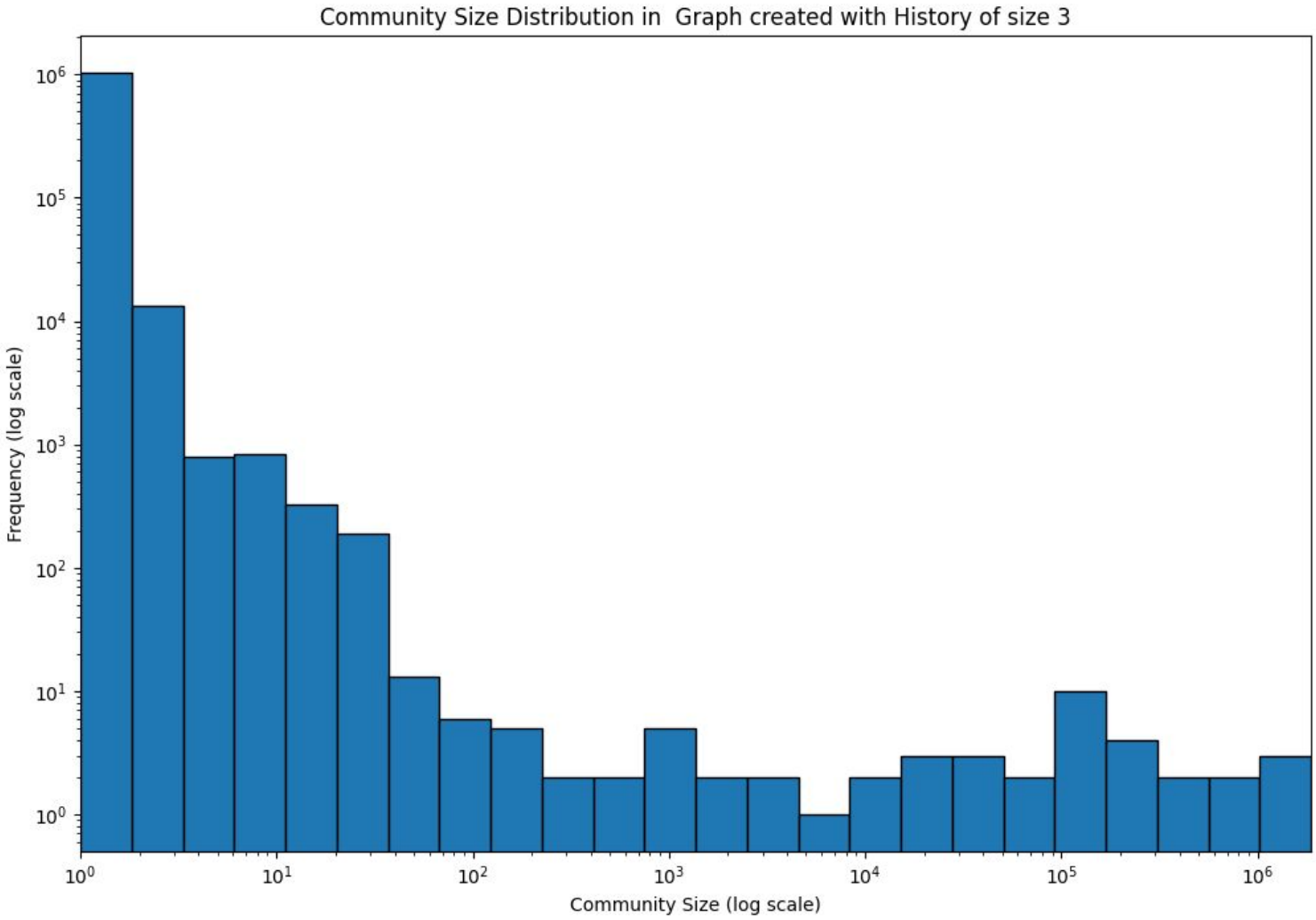
# COMMUNITIES

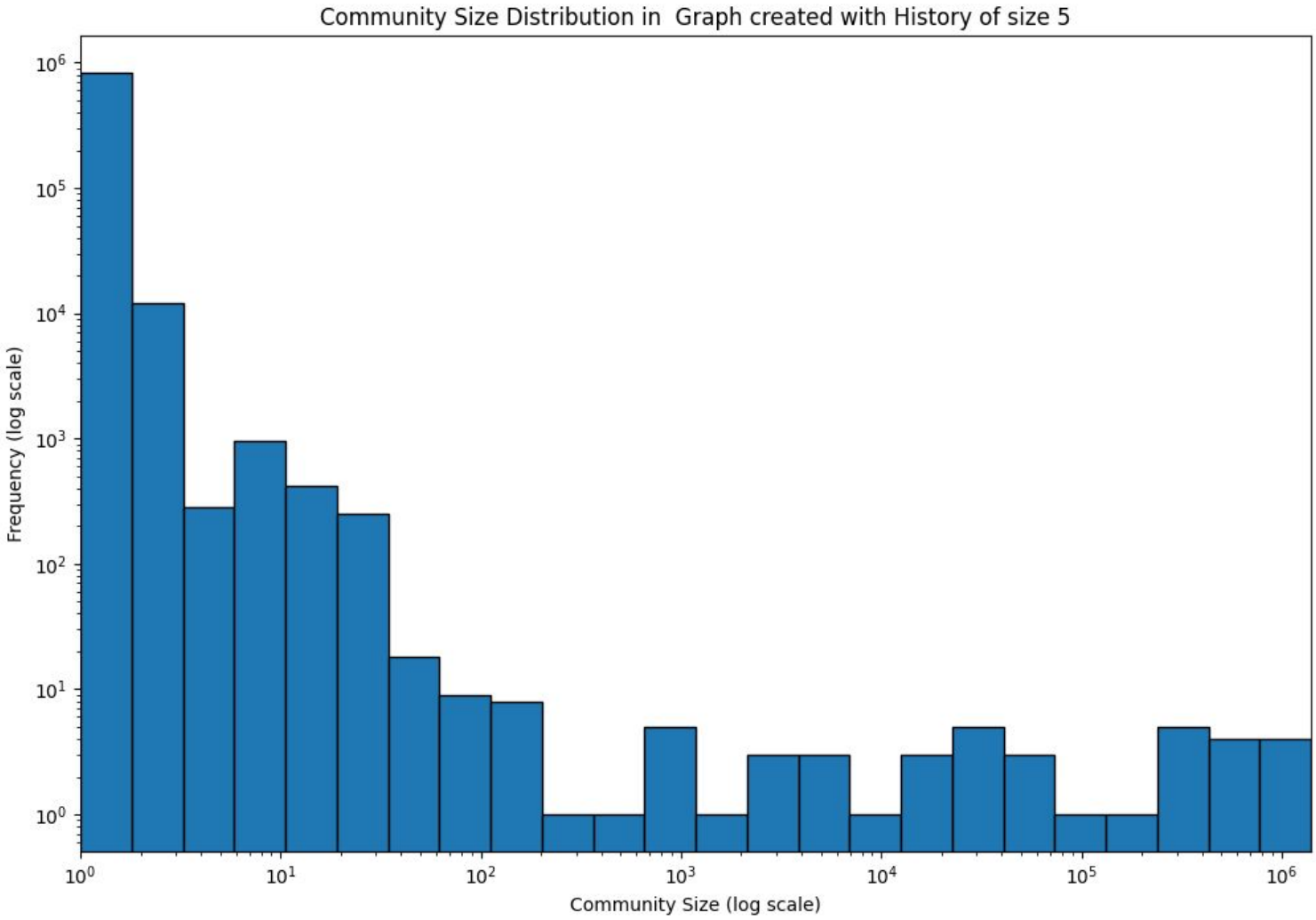
**Louvain** method for community detection was used

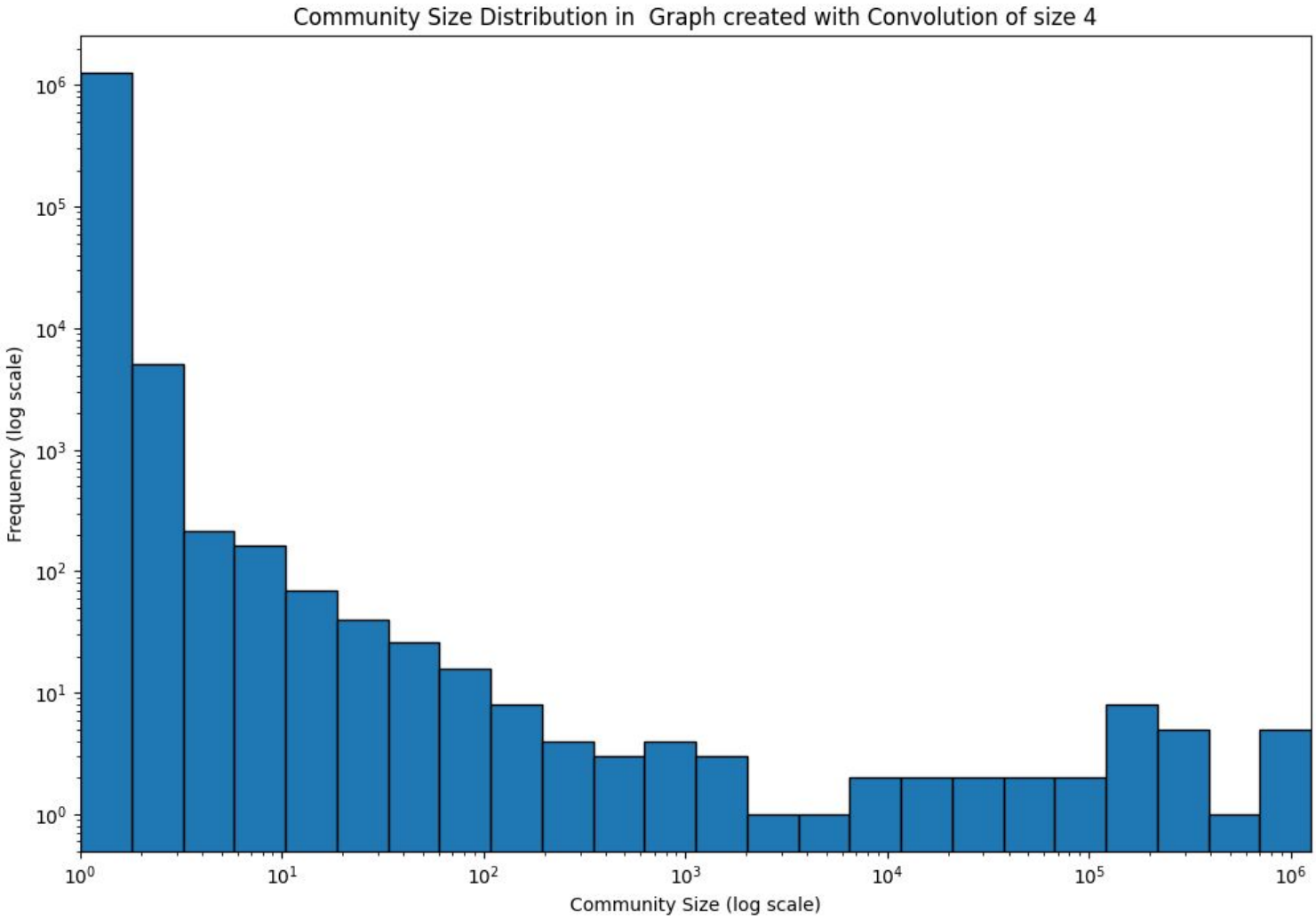
Communities were computed for:

- **Same pixel** interacting with previous 3 users
- **Same pixel** interacting with previous 5 users
- **Neighborhood** interacting with 4 users

Without using the weights


Community Size Distribution in  Graph created with History of size 3
Frequency (log scale)
Community Size (log scale)
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
$10^5$
$10^6$


Community Size Distribution in  Graph created with History of size 5
Frequency (log scale)
Community Size (log scale)
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
$10^5$
$10^6$


Community Size Distribution in  Graph created with Convolution of size 4
Frequency (log scale)
Community Size (log scale)
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
$10^5$
$10^6$

# RESULTS

# RESULTS OVERVIEW

Despite using different graphs, a lot of communities were not that different. However the are some noticeable differences in the biggest one.

In next slides There will be some community artwork showcases, where:

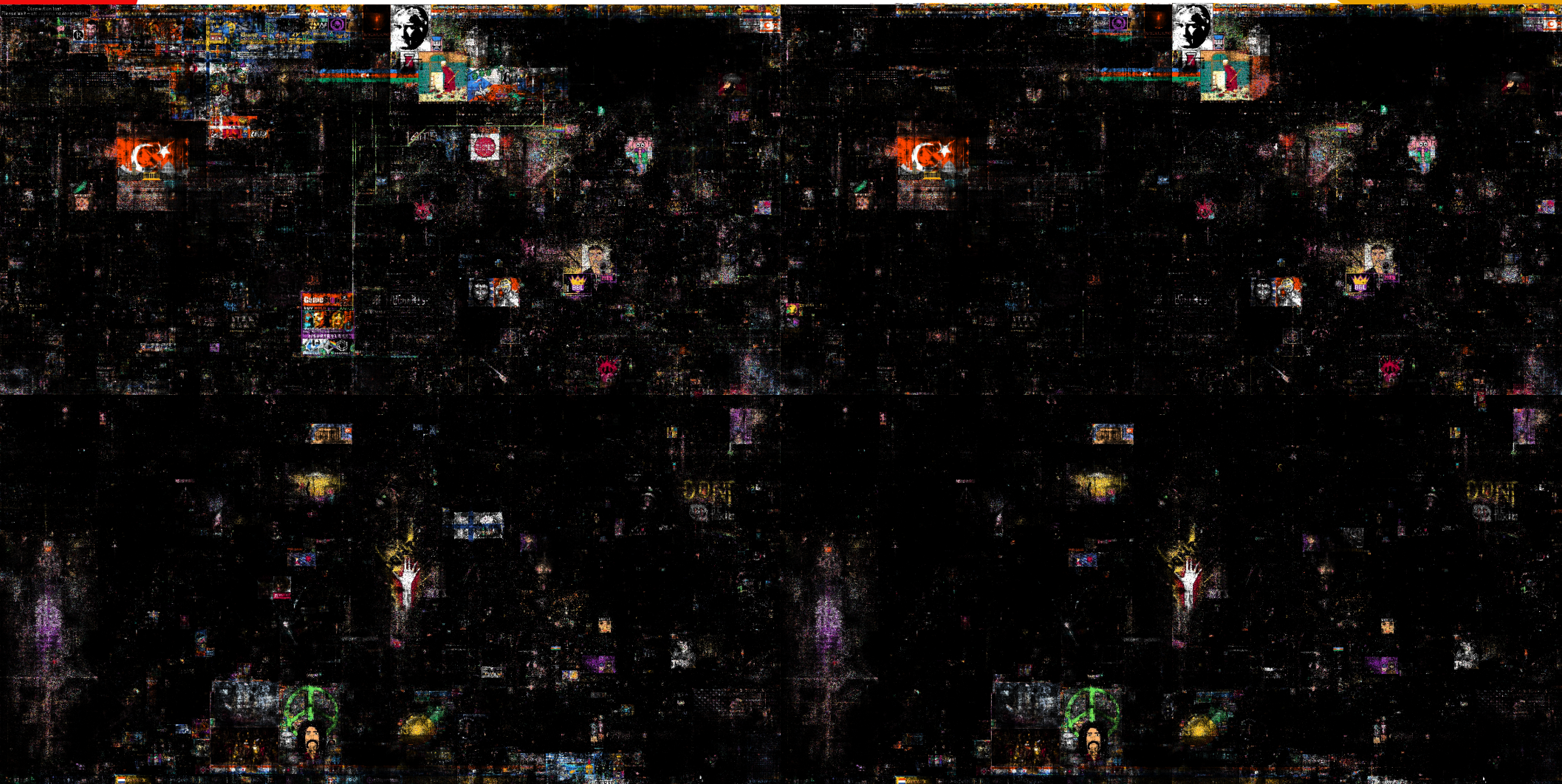
On the **left**

- **Same pixel** interacting with previous 5 users

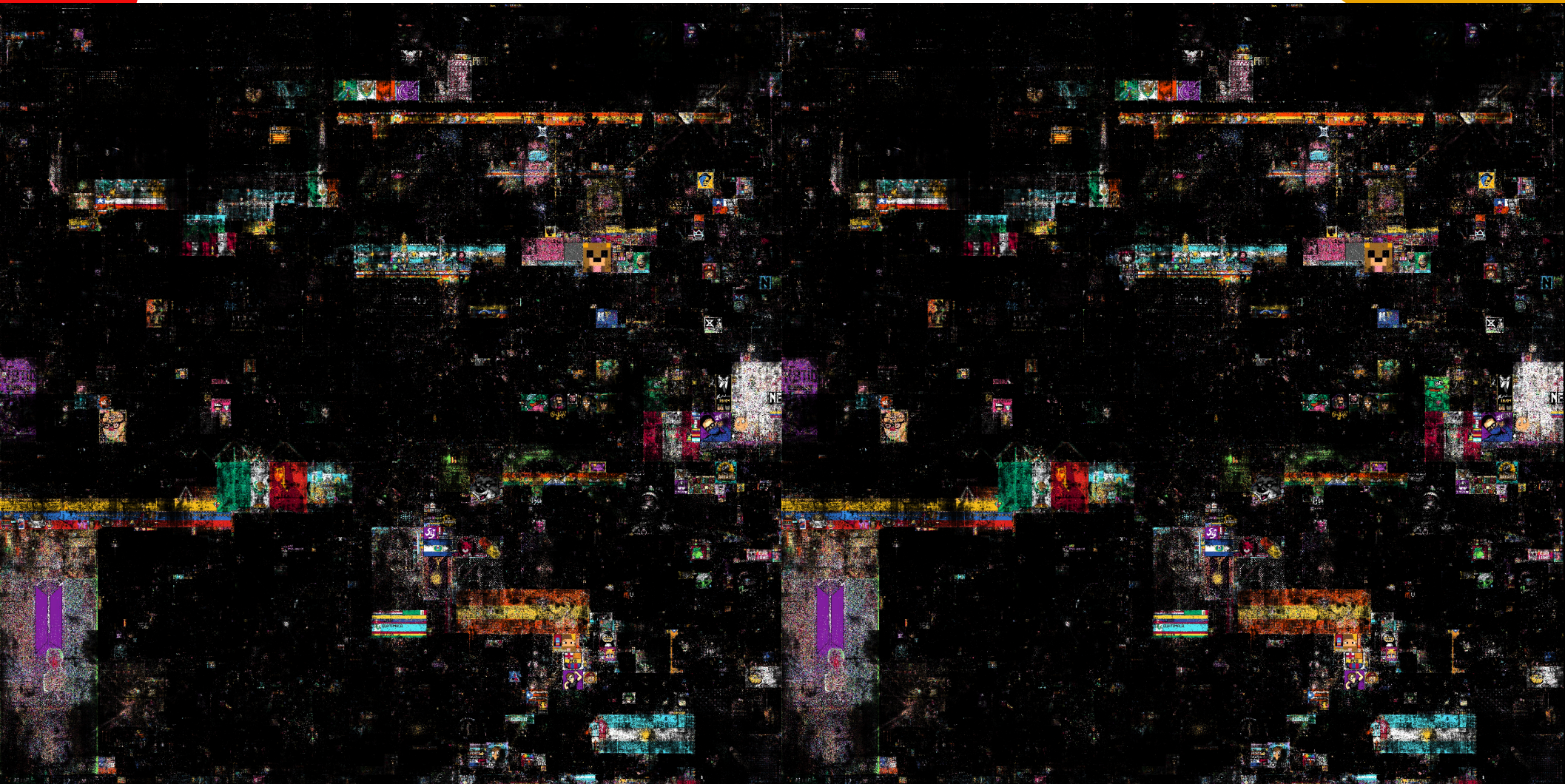
On the **right**

- **Neighborhood** interacting with 4 users

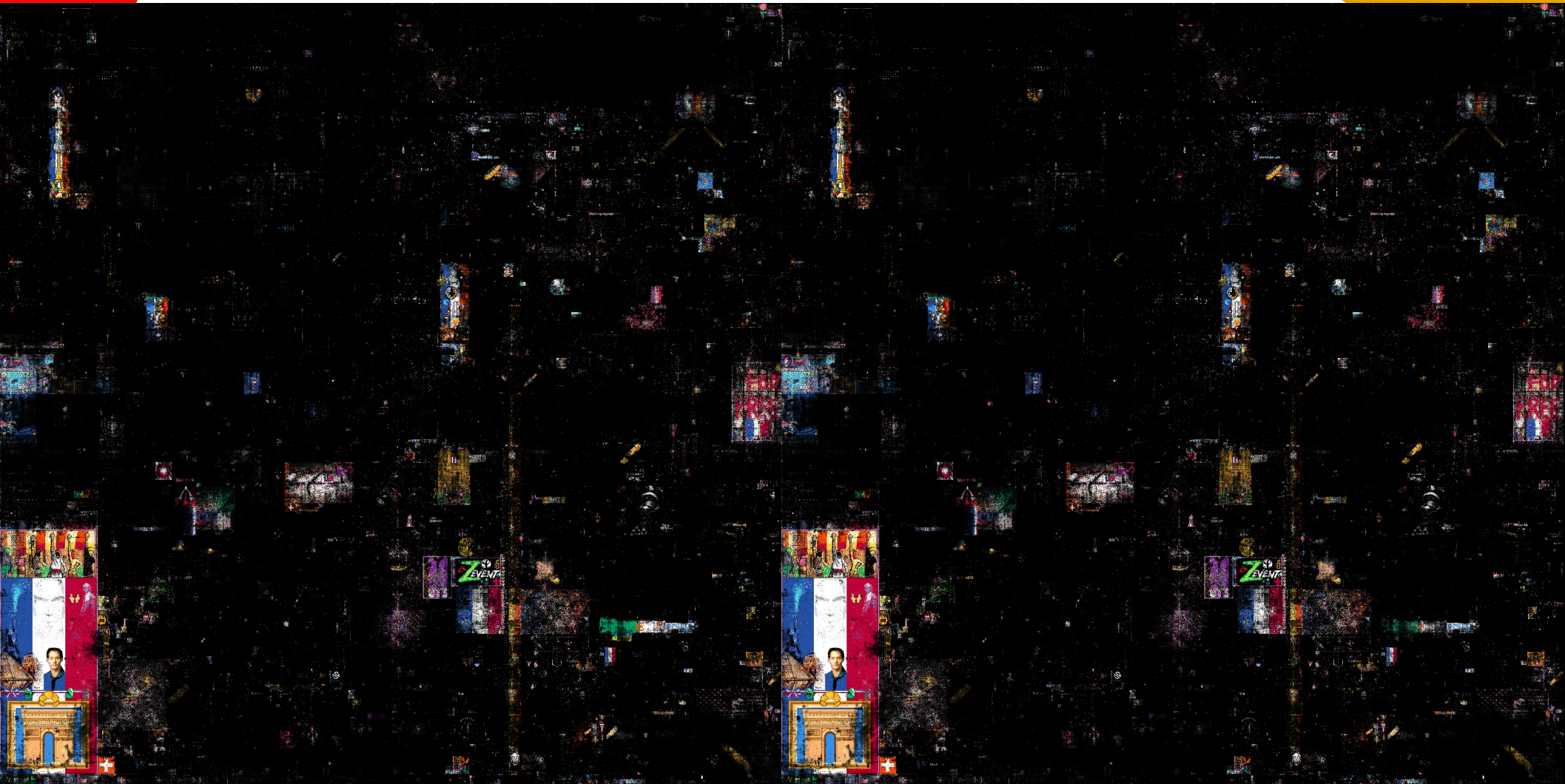
# TURKEY (SOME NORTHERN EU?)



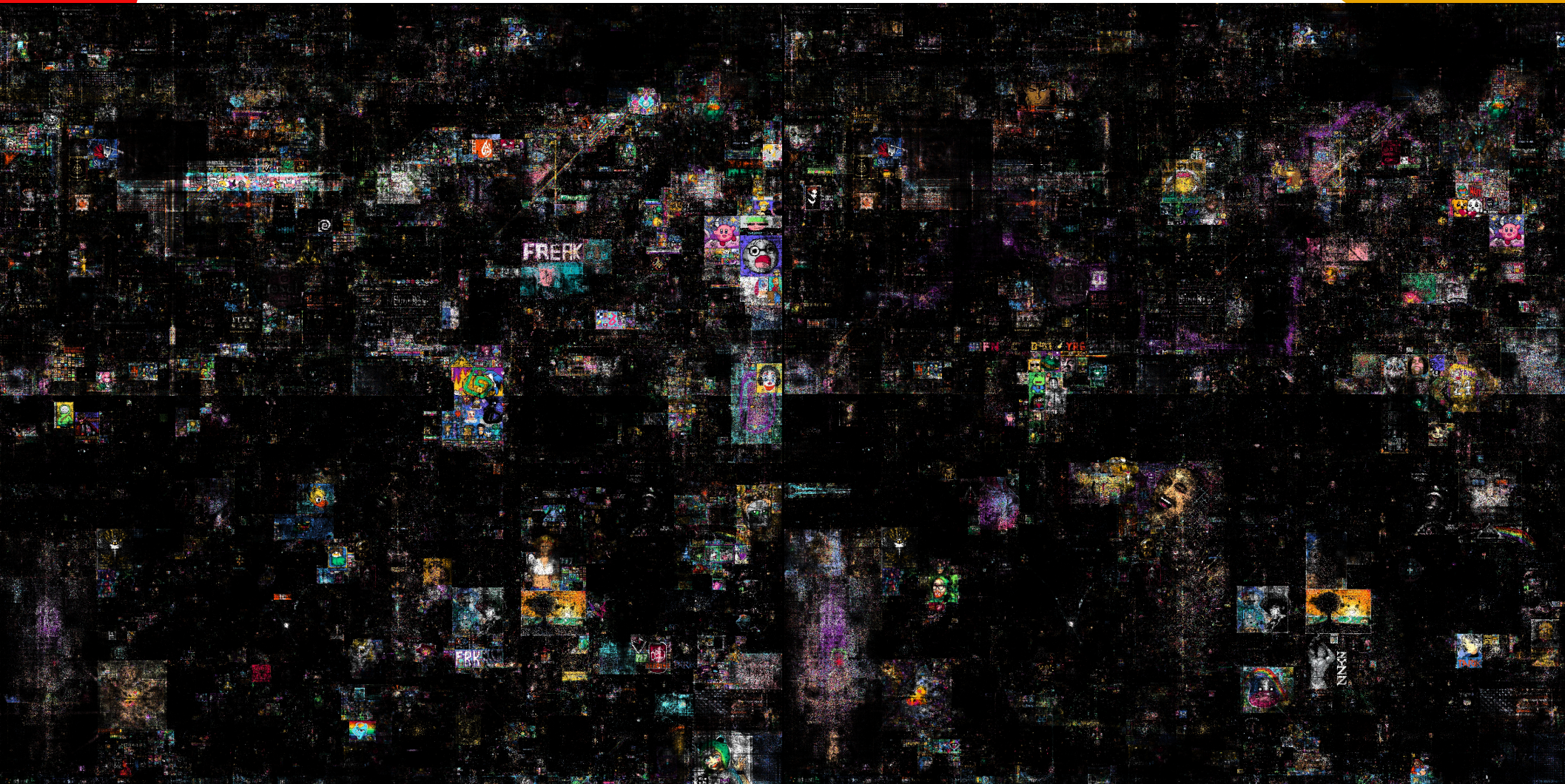
# SPANISH SPEAKING



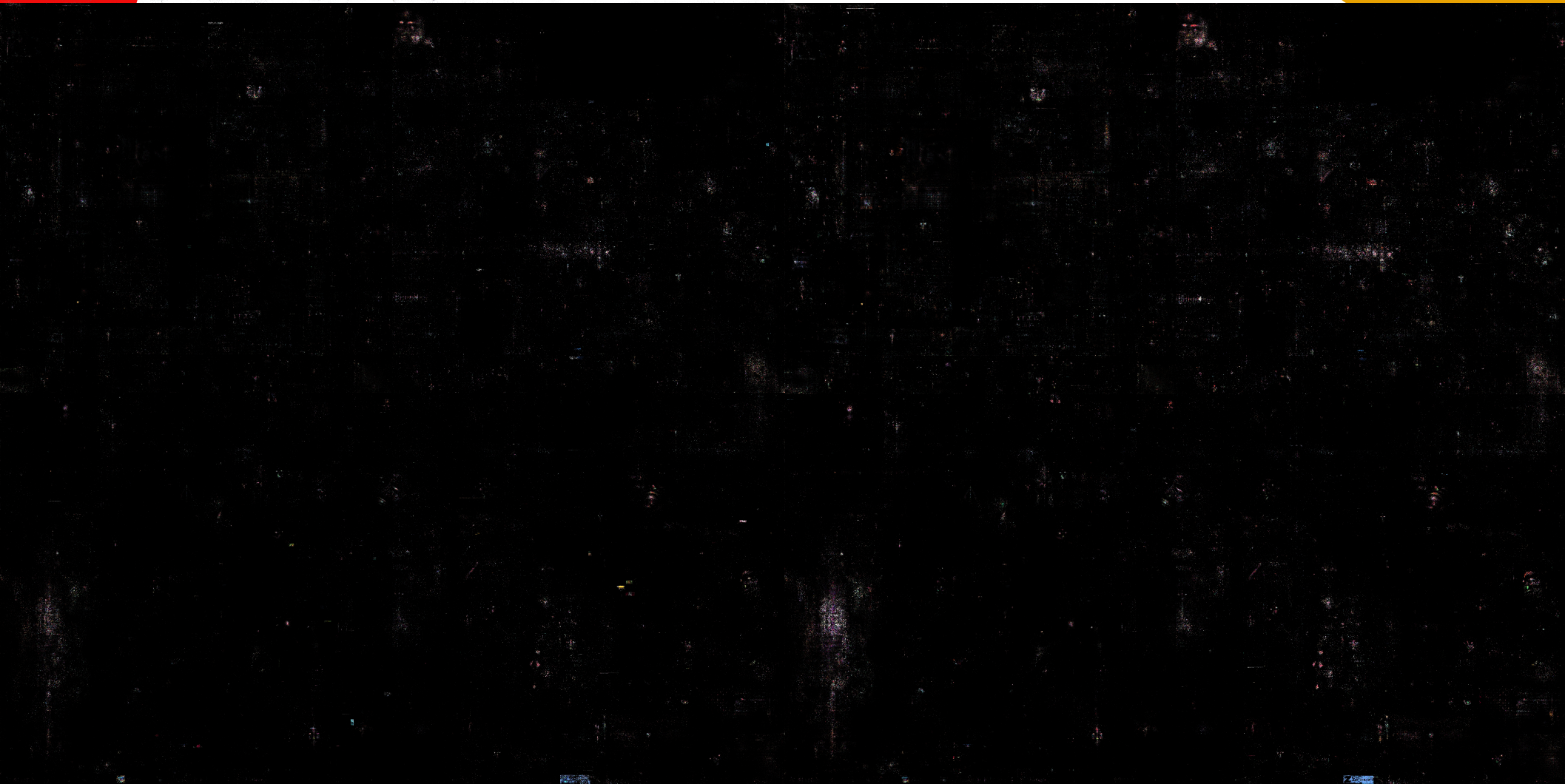
# FRENCH



# TWITCH STREAMERS



# REST (COMMUNITIES WITH SIZE SMALLER THAN 200)



# SUMMARY

# SUMMARY

During the project we achieved:

- Working with large dataset
- Finding interesting communities
- Compared few different approaches

The are a lot of ways how to further proceed with this project:

- Analyze union graph of **Neighborhood** and **Single pixel** approaches. (More RAM needed 64GB)
- Use project **r/Place Atlas** to label communities or compute communities of subreddits.
- More detailed found Community comparison from different approaches

# REFERENCES


https://www.reddit.com/r/place/

https://place-atlas.stefanocoding.me/

https://igraph.org/python/doc/api/igraph.Graph.html

https://networkx.org/documentation/stable/index.html

https://en.wikipedia.org/wiki/Color_difference

THANK YOU FOR ATTENTION