Label Detection and Discrepancy Analysis across Map Providers using Computer Vision

Adel Sabour

A thesis

submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND SYSTEMS

University of Washington

2021

Committee:

Mohamed Ali

Eyhab Al-Masri

Program Authorized to Offer Degree:

Computer Science and Systems

University of Washington

Abstract

Label Detection and Discrepancy Analysis across Map Providers using Computer Vision

ADEL ABDELSABOUR A ALY

Chair of the Supervisory Committee:
Mohamed Ali
Department of Computer Science and Systems

In this thesis we present a system that compares the text recognized from tiles images downloaded through the APIs provided by the different map providers, specifically in this thesis we compare Bing maps, Google maps, and OpenStreetMap. Then the system uses Computer Vision to understand the text from the tiles and help in reading it accurately through OCR.

The system highlights the discrepancies in the text, where we review the words that appear in a specific area and then determine whether they appear or not in each map provider, and the system determines if they appear completely mismatched due to some misspelling or any other reasons. It also compares the word size system between each map provider, the color of the word, and its different location in each map provider. The system is distinguished by displaying a large number of statistics about missing words in each map provider by country, as well as about mismatched

words, and displaying words that need revision to make their size larger or smaller compared to other map providers. This makes the tool valuable to the various geographic editorial teams for detecting, reviewing, and modifying text inconsistencies on the map providers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors Prof. Mohamed Ali and Prof. Eyhab El-Masri. Their resourceful knowledge and ideas always amaze and encourage me in course of this research. I am grateful for them continues guidance and enduring patience.

Last, but not the least, I would like to thank my parents for their constant support and words of encouragement at every step of the way, and thanks are due to my children and my wife, who sacrificed a lot of their time with me in support of me.

Chapter 1. INTRODUCTION

This thesis takes care of label detection through different tiles images captured from more than one map provider using computer vision for analyzing the discrepancy in the text shown on the maps between the different map providers.

Through tile systems followed by map providers, we have images that capture places on the surface of the Earth at a specific zoom level and whenever the zoom level varies, different information appears according to the importance of what is displayed, and the details shown on the map. Because there are no general rules that apply to each map provider and therefore internal rules to determine the importance of the names or the amount of details displayed based on the information available to each map provider. Accordingly, a lot of discrepancy appear between the images of the maps shown from one map provider to another. These images also reflect many challenges such as truncated words in the images, unclear italic words or words that depart with roads, different word locations from one map provider to another, and many other challenges that we will face during this research.

1.1 THESIS MOTIVATION

We all know that the use of electronic maps by ordinary users today has become almost essential in many countries, as the number of downloads of the Google maps application only within America in 2020 was 23.42 million, according to statista.com [2], as well as the importance of maps during business, as the number of sites that use Google maps alone in 2021, more than 3.7 million sites, according to SimilarTech.com [2]. The main motive is to disclose information that is missing from the Map provider and is found in another, to improve the information used by companies and individuals. And then reveal the best map provider according to the region and the text information displayed.

1.2 RESEARCH PROBLEM

The information that appears on each map provider differs from the other, some of them lose some information and some of them display the information at other levels of detail, and some of them display information that is not present at all with the rest of the map providers and we do not know whether it is correct information or not. In this thesis, we want to expose text discrepancy between different map providers for discrepancy detection to improve the maps people and companies use in their businesses, and because not every use of maps is simple and superficial. There are uses that care about data quality to make decisions, target audiences based on geographic information, improve relationships with customers, manage marketing campaigns, and others that depend on the presence of accuracy in information, and the information is complete, correct and consistent.

As an illustration of the problem, Figure 1.1 below shows a simple example of the differences in information between different map providers for one region and at the same zoom level as a tile captured for an area in Sri Lanka.

Bing maps | Google maps | OpenStreetMap

Figure 1.1. An example expressing the discrepancies between different map providers.

We notice from Figure 1.1 Inaccuracies among maps service providers, for example, the bing map in this area contains only the name of one area "Maho", on the contrary, google maps contains more information with a display of street names at the same zoom level. In contrast, OpenStreetMap displays more text information than google and Bing. But OSM does not display street names as Bing. We also notice differences in the position of the word Maho in the Bing map as well as the color is different from google maps and OSM. Although we note that this word is big in google map for OSM map which may reflect the importance of this area. We also note, for example, that on a google map the area appears “Galnewa” and in approximately the same place OSM the name appears differently “Bulnewa”. Thus, there are many differences that we must monitor to highlight to solve their problems.

1.3 THESIS OBJECTIVE

The main objective of this thesis is to do discrepancy analysis across different map providers to create through a concrete system that allows geographical editorial teams to detect, review and correct inconsistencies text in any map provider.

1.4 THESIS ORGANIZATION

The thesis is organized as follows. In addition to this introductory chapter, the thesis consists of six chapters. Chapter 2 Literature review. The related work in chapter 3. Chapter 4 discuss the proposed architecture and component stack. The developed MapsVision tool is presented in chapter 5. Chapter 7 presents the discussions and conclusion for the dissertation.

Chapter 2. LITERATURE REVIEW

In this chapter, we will explain the basic literature review on which the idea of thesis is based because the solution presented is based on more than one science. We will clarify the basic concepts of each topic, and then focus on the concepts frequently used in our solution, with an explanation of where it is used in our solution. In this thesis, we are concerned with showing the discrepancies between map providers, an area primarily associated with GIS, and of course addressing the textual differences in which we were exposed to using Text Analytics from Natural Language Processing (NLP) and for flexible comparison of texts we used Fuzzy logic. To detect texts from tile images, we deal with Optical character recognition OCR. Label detection is generally considered a branch of Computer Vision. Therefore, we will explain these sciences on which the solution presented in the research depends in the following points.

2.1 GEOGRAPHIC INFORMATION SYSTEM (GIS)

Geography is the main subject, so any system that works on the spatial basis is GIS. Therefore, GIS is a conceptual framework designed to deal with geographic information of all kinds, through capturing, storing, processing, querying, managing, deducing and sharing information, analyzing and comparing it to have an integrated geo-computing [2].

The GIS mapping is the basis in which we can visualize this type of information, it shows us what is where so that we can analyze with understanding and simplicity.

2.1.1 *Latitude, Longitude Map Coordinate System Grids*

The map image represents displayed area and its spatial information is determined by latitude and longitude in data tables as shown in Figure 2.1, as a spherical coordination system that determines positions on the Earth and is sometimes added to the elevation as it is known as Earth-centered, Earth-fixed (ECEF) Cartesian coordinates in 3 -space. But in the 2D map images we only use two coordinates. This coordinate system is used in MapsVision system when downloading tile images from Google maps.

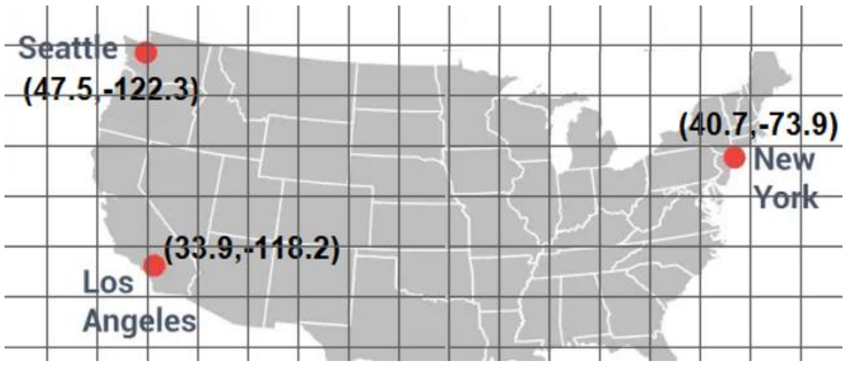
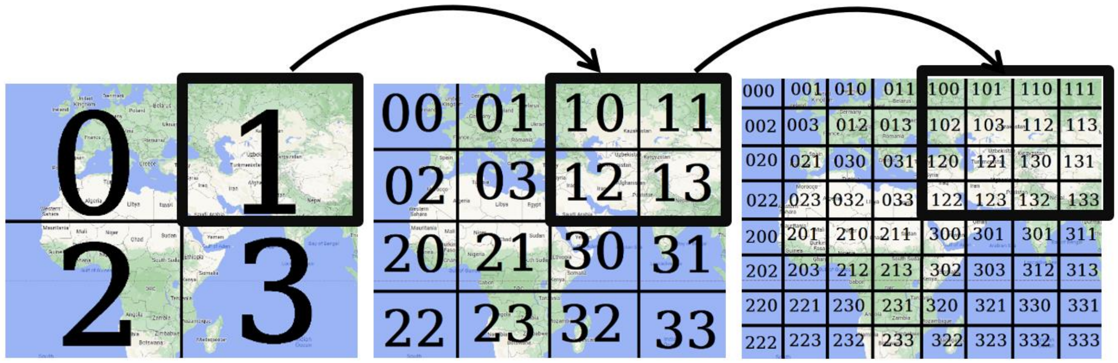


Figure 2.1. Longitude and latitude coordinate system on maps.

2.1.2 *Maps Tile System*

The tiles system in the maps is a contiguous square image that makes up the map of the globe at a specific Zoom Level. The next zoom Level of the current zoom level has other images that are square images that are stacked together to cover the globe, but with another level of detail. Each tile at a level is divided into four tiles at the next level of its level, and each of these four tiles has more details than the higher level, and at the same time it is divided into four tiles at the level following its level. And so, with each level and the next level. This is what the following figure 2.2 shows



And so on

Level 1) Initial zoom level

Level 2) The next zoom level divides the quad image into 4 tiles.

Level 3) The next zoom level divides the initial quad image into sixteen tiles.

Figure 2.2. Maps Tile System and zoom level.

Each tile covers a specific area of the globe with the same dimensions across all map providers, enabling us to compare these tiles.

2.1.3 *Pixel Coordinates*

There are also other systems of the basic geographical system based on latitude/longitude in map images, for example, we can use pixel left, top as X from left to right and Y from top to bottom to determine any point on the map image, so that it is always the upper left point (0 ,0). We can convert from latitude, longitude to Pixel (X, Y) and back.

2.1.4 *Tile Coordinates*

To simplify working with the tiles, and to speed up the retrieval of tiles images and display [2]. the Pixel (X, Y) coordinates on the map level are converted to X, Y, but only at the tile level. This is for the ease of moving each tile in the same order according to the resolution of the image. And we can only convert from Pixel (X, Y) to Tile (X, Y) and back. This system is used in our MapsVision system when downloading tiles from OpenStreetMap (OSM).

2.1.5 *Quadkey Coordinates*

The Quadkey system is short for quadtree keys and was created to improve storage and speed up indexing of tiles as database B-tree index. And its idea is to convert Tile (X, Y) to a single value in which the value of X is converted to its equivalent as binary number as well as with the Y. Then we make an interleave for Y and X and then it is converted to the base-4 system one so that we have a one-dimensional index key [2]. It is characteristic of this indexing that any Quadkey of any tile begins with the Quadkey of the parent tile as shown in Figure 2.2 above. So, the size of the Quadkey gets bigger (the number of digits) as the level of detail increases.

It is possible to convert the Quadkey number to Tile (X, Y) number and vice versa. This system is used in our MapsVision system when downloading tile images from Bing maps.

Table 2.1 shows the different coordinate systems, and the arrows between them, in order to show the possibility of converting to and from each of them. Below each system is the Map Provider, which uses that system to download its tiles images.

Table 2.1. Converting between coordinates, and which map provider uses them to download the tiles

Coordinates	Lat/lng	↔	Pixel(X,Y)	↔	Tile(X,Y)	↔	Quadkey
Used to download tiles from	Google Maps			OSM		Bing Maps	

From this table 2.1, we understand that when downloading tile images from OSM, for example, we convert the latitude/longitude to Pixel (X, Y), then we convert the Pixel (X, Y) to Tile (X, Y), then we can download the tiles images from OSM. And if we want to download images from Bing, we go one step further to the steps of downloading images from OSM, which is the conversion from Tile (X, Y) to Quadkey and vice versa if we want to convert vice versa to access downloading images from Google through the Quadkey.

2.2 COMPUTER VISION

Computer Vision (CV) is a field of artificial intelligence (AI) that attempts to simulate the highly capable human visual system. It is specialized in extracting meaningful information from digital images or videos, not only extracting texts from images or videos, but also locating and tracking any object and knowing what this thing is, and not only this, but also understanding the visual context, and after obtaining this information, decisions can be made and perform other processing on the data [2].

Optical character recognition or optical character reader (OCR) is a Computer Vision system that can see digital images or videos and can track typed text, numbers, and symbols and convert it to computer typed text. We use this software to read texts from tiles images and then perform processing on it after that [2].

Label detection Vision is an algorithm that can detect and extract information about objects inside an image or video [2]. We use that in our solution because we not only want to read the texts inside the images, but we want to extract information about them such as the color of their writing, the size of the font used, the location of the words and other properties of the words on the map images

Some image editing techniques are used for making adjustments that would improve the quality of the text in the image, such as separating the background or trying to eliminate overlapping lines with texts, canceling glossiness, highlighting letters, etc. Through which we can improve the accuracy of reading texts through OCR on map images.

2.3 NLP: Text Analytics

Text Analytics in natural language processing means performing multiple analyzes of text data and then extracting numerical data about these texts to conduct analyzes about them and create charts and graphs that reflect different points of view about the text data. And this is what we do inside the system after extracting the texts from the map images completely[2].

2.3.1 *The Text Analytics Process*

It contains many steps that we mention below and explain how the proposed system deals with each step [2].

a) Text cleansing

Remove unwanted information or "text normalization". In this step, we delete the special tags that are incorrectly read from OCR due to some icons and others. One- and two-letter texts are also excluded because in most cases, these letters are the remaining letters of incomplete/partial words.

b) Tokenization

Dividing texts into parts, for example, words or keywords, semantic symbols, and so on. In this stage, the system uses the text fragmentation for words, each word separately, the system also senses some of the words being read, such as restaurant, club, and so on.

c) Part-of-speech (PoS)

Considers the grammatical rules of the text, eg sentences with the same meaning in a different order or alternative words. In our solution we only consider that the words are arranged differently.

d) Data Analysis

Perform multiple analyzes of textual data and then deduce numerical data about these texts. In the system, we extract numerical data about texts, for example, the number of words in the map, how many words are correct and how many are wrong, and X, Y for word position, word font size and others, then compare these numbers with each Map provider and perform analyzes on these comparisons.

e) Visualization

Transforming analyzes and numbers into graphs, spreadsheets, and more. Finally, the system displays a set of graphs and figures that reflect the result of analyzing the textual data in each map provider.

2.4 FUZZY SETS AND FUZZY LOGIC

When writing Break.

2.4.1 *Crisp Sets*

A set-in classical set theory always has sharp boundaries because the characteristic (membership) function of that set is black-and-white or true-and-false, meaning that an object either belongs to the set or does not belong to the set at all. But we cannot track this type when comparing texts. For example, in the previous figure 1.1 if we make a special comparison of texts on the word *Bulnewa* from the OSM map provider and the word *Galnewa* from the Google map provider, the result will be completely uneven. And that situation is not acceptable. We want a way to find out that these two words are almost identical and have a slight difference.

2.4.2 *Fuzzy Sets*

Fuzzy set is a set with smooth boundaries. Fuzzy set theory generalizes classical set theory to allow partial membership to the set, and it was developed to cover areas that cannot be covered by classical set theory [2].

As an example, for simplification, when dividing the word “Bulnewa” into seven letters and then comparing it with the word “Galnewa” it will result that there are 5 letters that are 100% identical and there are only two letters that are different. If these two letters are far from the other, we calculate the percentage of their congruence with zero, so the result becomes a match of 71%.

Representation of Fuzzy Sets

If U is the universe of discourse and X is a particular element of U, then a fuzzy set A on U can be defined in two ways:

a) A collection of ordered pairs (x, μA (x)), each is called a singleton and has x first, followed by its membership in A, μA (x).

$$A = \{ (x, \mu_A(x)) | \, x \in U \} \text{ Eq(2.1)}$$

For example, the set A of medium integers, see figure 2.3. It is defined over the universe of discourse of positive integers in the interval [1, 10]. As shown, A can be represented by the collection of singletons

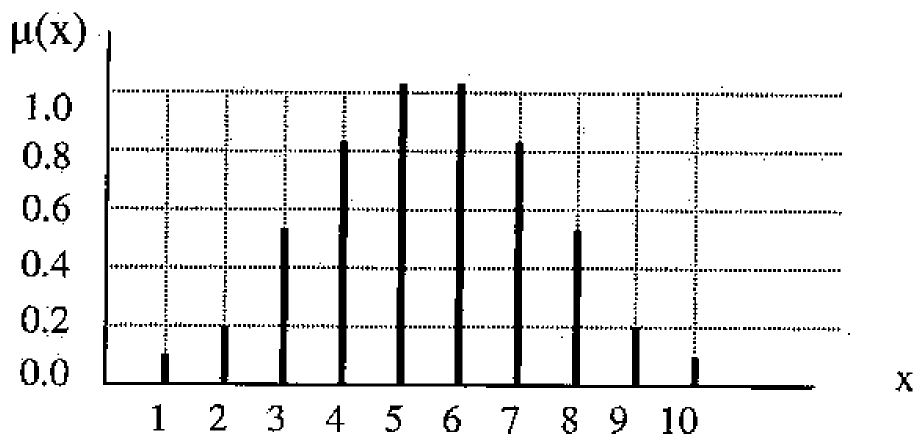


Figure 2.3. the fuzzy set of medium integers defined on the interval [1, 10]

By enumeration, we can define the fuzzy set of short integers defined on the integer numbers from 1 to 10, as follows:

A = {(2, 0.2), (3, 0.4), (4, 0.7), (5, 2.0), (6, 2.0), (7, 0.7), (8, 0.4), (9, 0.2)}

Singletons whose membership to a fuzzy set is zero can be omitted. Thus, the integer 2 belongs to the fuzzy set medium integers on the closed interval [1, 10] to a degree of 0.2, and the integer 7 belongs to A to a degree of 0.7.

A general notation can be used to describe the fuzzy set A through enumeration as follows:

$$\underset{\sim}{A} = \sum_{X_i \in U} \mu_{A\sim} \frac{X_i}{X_i} \qquad \text{If U is discrete Eq(2.2)}$$

$$\underset{\sim}{A} = \int_{x \in u} \mu_{A_\sim}(X)/X \quad \text{If U is continuous Eq(2.3)}$$

Both the summation and integral sign in the previous equations indicate the union operation of all singletons. Thus, we can write the fuzzy set A again as follows:

A = 0.2/2 + 0.4/3 + 0.7/4 + 2.0/5 + 2.0/6 + 0.7/7 + 0.4/8 + 0.2/9

It should be noted that enumerating each element in the universe of discourse with its membership value to define a fuzzy set is quite expensive in the case of continuous universes of discourse.

b) By defining a membership, function mathematically. That is, we can define a membership function, to cover the linguistic concept instead of enumerating all elements.

2.4.3 *Fuzzy Membership Functions*

Many types of the membership functions can be used to represent the fuzzy sets. Selection a specific membership function to represent a fuzzy set is a matter of designer experts and decisions. The efficiency here is how to select the most suitable membership function to the fuzzy set. Some of commonly used membership functions such as Triangular, Trapezoidal, Gaussian, Bell-Shaped, Sigmoidal, S, Π_1 and Π_2.

In our explanation, we will discuss the Triangular Membership Function because it is the one used in the system.

Triangular membership function as shown in EQ 2.4 and figure 2.4 has three parameters, first one says that, each predecessor element has zero membership to the fuzzy set, second determines the elements with a full membership, and the third parameter determines successor elements of it has a membership zero.

$$\text{Triangle}(x : a, b, c) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & x > c \end{cases}$$

$$= max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \qquad \text{Eq(2.4)}$$

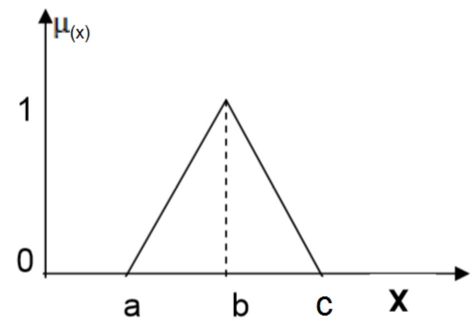


Figure 2.4. Triangular Membership Function

2.4.4 *Levenshtein Distance Algorithm*

This algorithm is a popular method of fuzzy string matching. The distance between two strings means the number of character replacements or character insert or character deletion required to transform one string to other. The fuzzy membership degree is calculated for each letter and then the average score for the whole word. This algorithm is applied in its standard form with minor differences in the result values. This algorithm produces a value that represents the difference between the two texts between 0 and 1. When a perfect match, the value is 1 and whenever the two words are separated due to any difference, the value decreases until it reaches 0, a complete mismatch.

The following figure 2.5 illustrates the idea in a simplified way about comparing two words. Each letter in the first word when matching represents a degree of 1, then compares each letter of the second word with the first word to take the matching degree. In the example shown, the first letter is 0.9, the second is 1 and the third is 0.75, and therefore the total matching degree for the second word is 0.88.

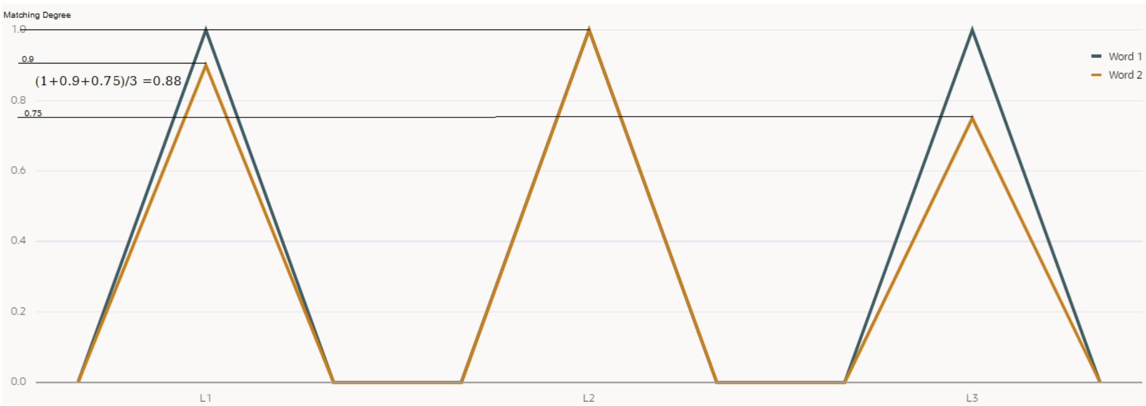


Figure 2.5. comparing two words using triangular membership function

Chapter 3. RELATED WORKS

This section presents several architectures and applications that have been proposed and developed for detecting discrepancy across map providers. Not all studies focus on label detection and not all of them use computer vision for comparison, but they use other methods for that, we will know about that in the following:

3.1 GeoDart: A System for Discovering Maps Discrepancies (2021) [4]

This paper is concerned with exposing discrepancies between map providers (Bing, Google, OSM), whether these discrepancies are in road network graphs, or routing discrepancies.

In our thesis, we deal with the contradictions of the text, and this is the first essential difference between what we present and what they present.

In this paper, they also create a system called GeoDart that is based on machine learning and classifies the discrepancies detected due to the characteristics of the road or part of it. Or because of the routing algorithm of each map provider, or because of routing data like time, distance, and route.

Their dataset was within the state of New Zealand, and the results presented in this research indicate that the system detects discrepancies with 80% accuracy, and classification up to 97% based on the machine learning model used.

In general, although they are interested in exposing the discrepancies between the same map providers, we differ in the type of discrepancies as well as the method of work [4].

3.2 OSMRunner : A System for Exploring and Fixing OSM Connectivity (2021) [4]

This paper is concerned with problems of incorrect connections between roads, misclassified road segments, and gaps in road networks. Based on these problems, they created a system called OSMRunner to handle these problems automatically, but only on OpenStreetMap.

They provided numerous statistics about incorrect connections discovered, number of roads, nodes and junction nodes, and their datasets were based in the Fiji Islands, Venezuela and New Zealand [4]. Of course, although this research is related work because it reveals inconsistencies in the maps, the researchers were interested in an aspect other than the text.

3.3 An Interactive System to Compare, Explore and Identify Discrepancies across Map Providers (2020) [4]

In this paper their system also exposes inconsistencies between map providers (Bing, Google, OSM) but in a different way through the Application Programming Interface API provided by each map provider, comparing the paths from the start of the travel point to the end of the travel point, then the comparison is done Depending on the geometry of the track and of course the duration and distance of the movement [4].

The system created in this paper can be divided into three components, the first component is Route Generator which deals with the API of each map provider to produce start and end points for movements in the map. The second component is Route Analysis, which is responsible for displaying the routes according to what the first component defines. It can also force the map provider to take another map provider's route to do conflict analysis, and it can also provide some statistics about those routes. The third component, Discrepancy Identification and Classification, is of course responsible for classifying, identifying and exposing these discrepancies.

Chapter 4. THE PROPOSED ARCHITECTURE AND COMPONENT STACK

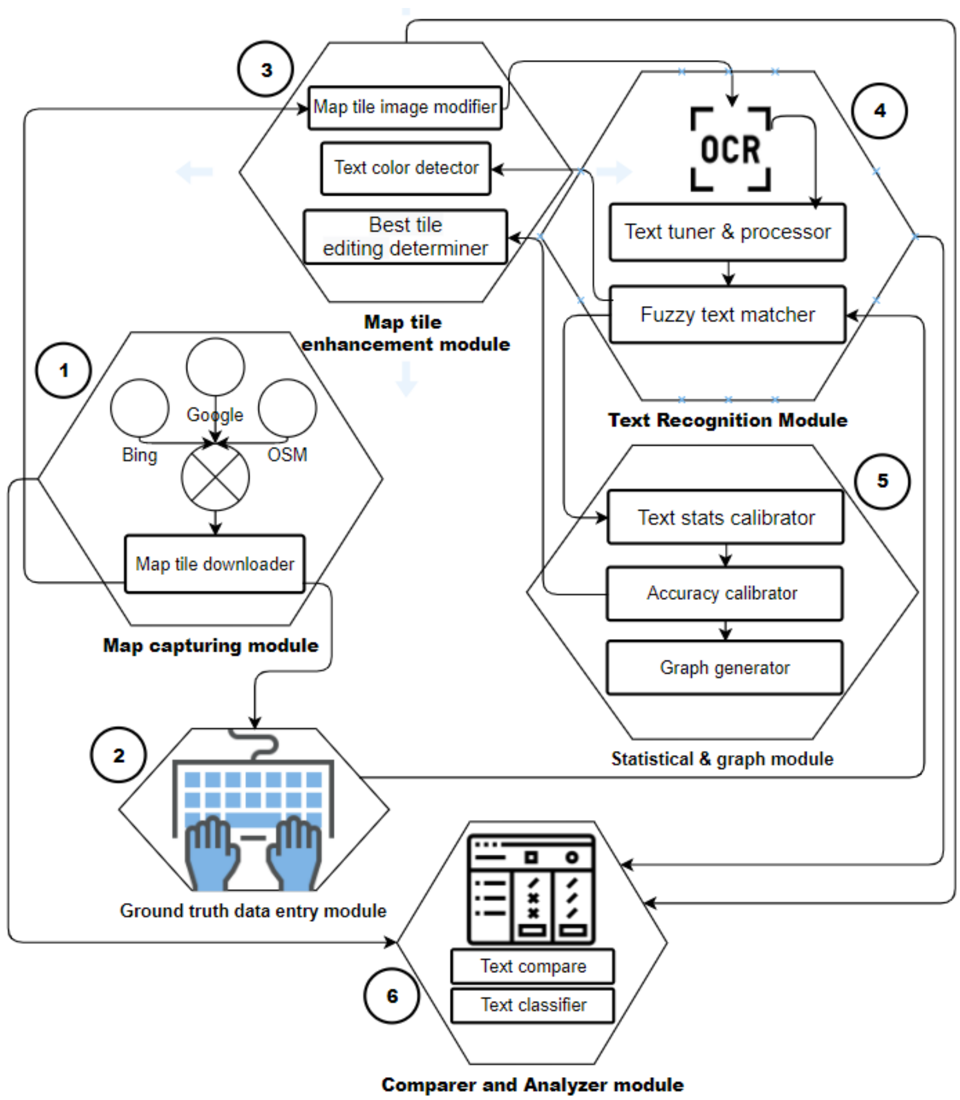


Figure 4.1. The proposed architecture and component stack.

4.1 MAP CAPTURING MODULE

- This is the module responsible for downloading map tiles images from different map providers (the system currently works with Bing maps, Google maps, OpenStreetMap).

- Any map tile can be downloaded from all the map providers by entering the longitude and latitude or by entering the quad key of any place where you want to download the map tiles.
- We can also upload a file with all the longitude and latitude or quad keys and then this module download all the tiles for those places.
- This module loads the tiles images with all possible resolutions according to the map provider. It can download resolutions (256x256, 512x512, 768x768, 1024x1024) from google, and download resolutions (256x256, 512x512) from Bing maps as well as from OSM.
- The tiles system in the maps is a contiguous square image that makes up the map of the globe at a specific Zoom Level.
- The next zoom Level of the current zoom level has other images that are square images that are stacked together to cover the globe, but with another level of detail.
- Each tile at a level is divided into four tiles at the next level of its level, and each of these four tiles has more details than the higher level, and at the same time it is divided into four tiles at the level following its level. And so, with each level and the next level. This is what the following figure shows

And so on

A) Initial zoom level

B) The next zoom level divides the image into 4 tiles.

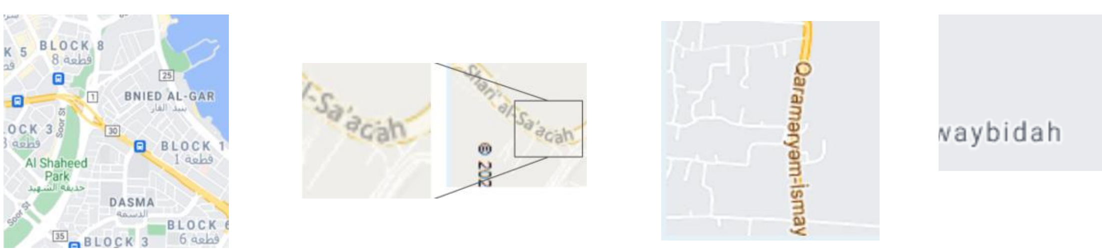
C) The next zoom level divides the initial image into eight tiles.

Figure 4.2. tile system and zoom level.

- Each tile covers a specific area of the globe with the same dimensions across all map providers, enabling us to compare these tiles.

4.2 GROUND TRUTH DATA ENTRY MODULE

- It is the module that data entry operators work with, they view each tile and manually write the texts shown on each tile. This is done for the dataset through which the accuracy of the system results is tested.
- The number of places in the dataset is 4,882 longitude and latitude repeated with each map provider and repeated with all available resolutions from each provider.
- The purpose of this module is to measure the accuracy of the texts extracted through OCR and their correspondence with the reality already shown on the image.
- This allows us to modify the system until we reach a high accuracy of the results, and after we adopt the accuracy of the system results, then this module is no longer important, and is not used.
- Despite this, we expect not to reach 100% accuracy due to many challenges such as the presence of many languages in the same image and sometimes the team cannot enter these languages, the presence of some unclear words that the team cannot read and therefore we do not expect how the OCR will read them, and the presence of some inverted letters although the language of the translation is English, the presence of some truncated characters that we might expect is one thing and the OCR expects another.



Many languages | unclear words | Words with inverted letters | truncated letters

Figure 4.3. Some problems with text appearing on maps.

4.2.1 *Map tile image modifier*

It is responsible for adjusting the tile images in order to be more readable through OCR. It consists of two levels of adjustments.

The first level is a set of simple filters on images such as:

- ***DetailEnhance:*** To make the image looks sharper
- ***Resizing the tile***, that filter change the width and height of the image with aspect ratio be preserved
- ***Grayscaling*** is the process of converting an image from other color spaces to shades of gray.
- ***Dilate:*** This procedure follows convolution with some kernel of a specific shape such as a square or a circle. This kernel has an anchor point, which denotes its center. This kernel is overlapped over the picture to compute maximum pixel value. After calculating, the picture is replaced with anchor at the center. With this procedure, the areas of bright regions grow in size and hence the image size increases.
- ***Erosion*** is quite a similar process as dilation. But the pixel value computed here is minimum rather than maximum in dilation. The image is replaced under the anchor point with that minimum pixel value.
- ***Adaptive thresholding*** is the method where the threshold value is calculated for smaller regions. This leads to different threshold values for different regions with respect to the change in lighting. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value.

The second level is a set of modifications to images such as:

- ***Contours of text technique:*** This a segmentation method, to differentiate the pixels we are interested in from the rest, we read the text from the OCR and then determine the position of all the words in the image. Then we separate these areas from the image. Once we have separated properly the important pixels, we can set them with a determined value to identify them. The following algorithm explains this technique.

//Contours of text algorithm

Input: tile image

Processing:

- **Reading** text from tile image using OCR
- **while** (word in text)
  - escape right and left spacing, special characters
  - create Rect(left, top, width, height) to rectangles
- Convert the tile image from color to grayscale
- Apply threshold binary invert to the tile image
- Get the tile image contours
- Create an empty text tile image
- **while** contours
  - remove small contour area, and large contour area
  - **if** (contour position in rectangles)
    - Draw contours in the text tile image

Output: text tile image

- Color-based Image Segmentation using K-Means clustering:

  K-Means clustering it maps an observation to one of the k clusters based on the squared (Euclidean) distance of the observation from the cluster centroids. The intuition here is to minimize the variance between points within the same cluster and maximize the variance amongst points from different clusters. Once the mapping is done, the observation point can simply be represented by the centroid of its respective cluster. Applying this on an image would imply reducing it to an image with k colors. This is an iterative refinement procedure that updates the cluster centroids in each iteration. The algorithm terminates when a certain convergence point is reached. The most commonly chosen to point of convergence is when the cluster centroids no longer move or get updated.

- **TextDilateBinaryInvert(bitwiseText):**

  It is a technique that makes four changes to the tile image, first converting the image to Grayscale, then selecting around the texts using Dilate, including making the Threshold application with type Binary, and finally applying Threshold with type BinaryInv to produce images as shown in the following images:

a) Original b) Grayscale c) Dilate d) Threshold, Binary e) Threshold, BinaryInv

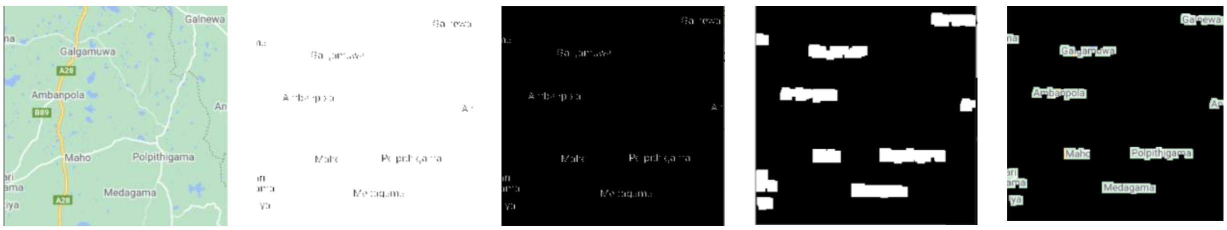
Figure 4.4. The stages of modifications to the tile through TextDilateBinaryInvert.

- Upscale with resize:

Enlarge an image four sizes larger depending on the base resolution of the image.

- BitwiseBackground

  It is a technique that takes many steps, but its main idea is to keep two copies of the first image, we apply a set of modifications to it to separate the text, and the most important of these steps is to make dilate in preparation to erase the texts from the screen and then convert the Threshold image into type Binary and then BinaryInv and then re-select the texts with dilate to produce a black image and places the texts are scanned in white. Then BitwiseAnd worked on the base image with the image on which the effects were applied, to return the image with text only and without a background.



a) Original b) Binary c) BinaryInv d) dilate e) BitwiseAnd

Figure 4.5. The stages of modifications to the tile through BitwiseBackground.

All modifications of the two levels change the ability of the OCR to read the texts in the tile, whether separately (each change or effect alone), combined (a group of changes or effects) or different order, as we will discuss this in the discussion of the results.

4.2.2 *Text Color Detector*

It is responsible for determining the color of each word in the tile to classify the words appearing in the tile based on the color used in writing the word.

This method takes advantage of the BitwiseBackground previously explained in the same module and then turns the background to white, then we use OCR to read the texts and to define a rectangle surrounding each word, then each rectangle containing a word is cut and sent to kmeans to know the color used - after deleting the color used in the background. And because the word is not one direct color, but it is gradients and the color of the frame is of a different color, and there may be colors from the background of the image, so we use the kmeans to know the color.

4.2.3 *Best tile editing determiner*

It is responsible for determining the best effects on tiles images that return the most accurate results, after several stages that will be mentioned in detail later, but the most important of which is comparing the results extracted from the OCR with the ground truth and more effects that give a value high match dependent, then generally applied later without ground truth. The measurement also varies according to the map provider and the accuracy of the images, and of course the results differ accordingly.

4.3 TEXT RECOGNITION MODULE

It is the module for reading texts with various OCR tools, word processing and text comparison.

4.3.1 *OCR*

A technology that allows us to electronically convert texts in image tiles into machine-readable text. The system uses more than one OCR tool to read texts such as Tesseract OCR and Microsoft Computer Vision, and the accuracy of the extracted texts is compared based on each tool separately.

4.3.2 *Text tuner & processor*

It is responsible for text processing in general. The most important of which is the processing of texts retrieved from OCR in a unified form on the system, processing of some partial words (words truncated at the edges of the image), processing and coordinating tables of correct and incorrect words and words that OCR could not read, and processing of different comparison tables.

4.3.3 *Fuzzy text matcher*

It is fuzzy string matching using Levenshtein distance algorithm. Its purpose is to compare two texts and measure letters that have been replaced or letters that have been added or letters that have been deleted. We also calculate the change of letter case from what increases the distance between the two texts.

4.4 STATISTICAL & GRAPH MODULE

Responsible for making aggregate statistics on text match and mismatch, statistics on the accuracy of extracted results, and graphic mapping.

4.4.1 *Text stats calibrator*

It is responsible for making aggregate statistics about the results stored in the database, from the number of correct words and their matching degree, the number of words read by the OCR and nothing similar to anything in the ground truth and the number of words in the ground Truth and OCR did not detect it, based on each map provider and on each resolution and based on each effect on the image - as each effect on the images makes OCR give different results.

Of course, the Matching degree value expresses the extent to which the texts are consistent and accurate compared to the manually entered ground truth. But there are other measurements that have been taken into account, such as

- precision is a fraction of relevant instances among the retrieved instances which represents the number of OCRs read that was integer divided by all OCRs read.
- The recall is a fraction of relevant instances that were retrieved that represents the number of correctly read OCRs divided by the audited ground truth.

- The F1 This measure is approximately the average of the two when they are close and is more generally the harmonic mean. Its value in the system is represented by the equation:

$$F1 = 2\left(\frac{precision \times recall}{precision + recall}\right) \qquad \text{Eq(4.1)}$$

4.4.2 *Accuracy calibrator*

Since all effects on tile images show different results from OCR and are measured statistically after comparing them with the manually entered facts, this Accuracy calibrator determines the best effect that gives the best results, but it is not the same effect for all types of tiles images, but according to each map provider as Effects may vary depending on the resolution of the tile images.

This calibrator identifies those results and sends them to the best tile editing determiner who identifies these effects by type and then uses his best effects when there is no ground truth. Therefore, when the effects that give the best results from the OCR are determined, the results are approved and there is no need to enter ground truth.

4.4.3 *Graph generator*

A tool that performs data visualization of the statistics and measurements that the system extracts, to make it easier to know data trends and allow decision makers to see the results visually. This tool produces graphs of the data that determine the best effects on the tiles images that give the best results from OCR in all its details. This tool also generates graphs of data for comparing texts in each map provider from the same place and at the same zoom level.

4.5 COMPARER AND ANALYZER MODULE

This is the module responsible for comparing the texts that appear on a tile for a specific location (lat, lng) with similar other map providers at the same zoom level. The comparison is in terms of the word matching with the map providers, the lack of the word appearing at any of the map providers, the difference in word size, the difference in the color of the word, and the difference in the position of the word in the tile image.

We know that each map provider changes the styles in which it displays its data every period, so the Analyzer analyzes the colors and sizes of the texts to know the significance and importance of the text, in other words we can know if this color represents the name of the county or street, is this the name of the city or the name of the province and so on .

Chapter 5. THE DEVELOPED MAPSVISION TOOL

This Chapter introduces a new developed MapsVision tool based on the proposed architecture presented in chapter 4.

5.1 MAP CAPTURING TOOL

This tool is responsible for downloading the tiles images from all the map providers in question namely Bing Maps, Google Maps, and OpenStreetMap.

This tool also takes care of matters such as the resolutions of the images of downloaded tiles, the locations, and the conversions between the different coordinates, which are referred to in Chapter 2.

5.1.1 *Download tiles for a specific location*

In the following figure 5.1 we can download images from the above three map providers with all available resolutions.

- For Bing/OpenStreetMap Maps: the resolutions available for public download are 256x256 and at some levels of detail 512 is available. As a single resolution feed available.
- For Google Maps: the resolutions available for download are 256x256, 512x512, 768x768, 1024x1024

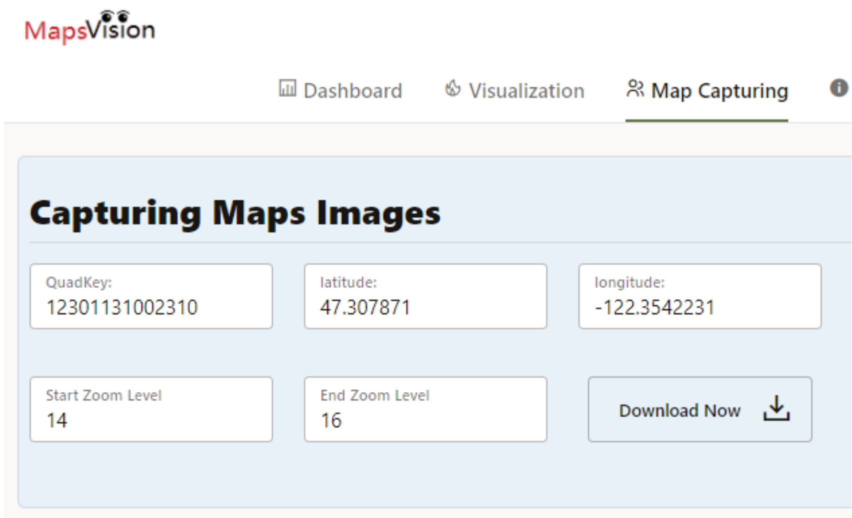


Figure 5.1.Download tiles for a specific location.

In the previous figure 5.1, we enter the Quadkey and it shows us the equivalent latitude, longitude and vice versa.

- Then we define the start zoom level, which here is 14, meaning that the initial tile image of the three map providers will be at detail level 14.
- The end zoom level is 16, which means that from level 15 we will download four detailed tiles for level 14 and then sixteen detailed images of the initial tile so that we will have 21 tiles from each map provider. This is as shown in Figure 2.2 above
- To find out the final number, 21 images are multiplied by 3 map provider and then multiplied by 6 (4 four resolutions from Google, 1 from Bing and 1 from OpenStreetMap) to get the result of the loaded tiles 378. According to the previous example.

5.1.2 *Download tiles images for multiple locations*

We can download a batch of images in one go. With a tsv file with Quadkeys in one column, or a tsv file with latitude, longitude in two columns. Then the tool downloads all the tiles as we explained in the previous part, but for each place, and the method is shown in Figure 5.2.

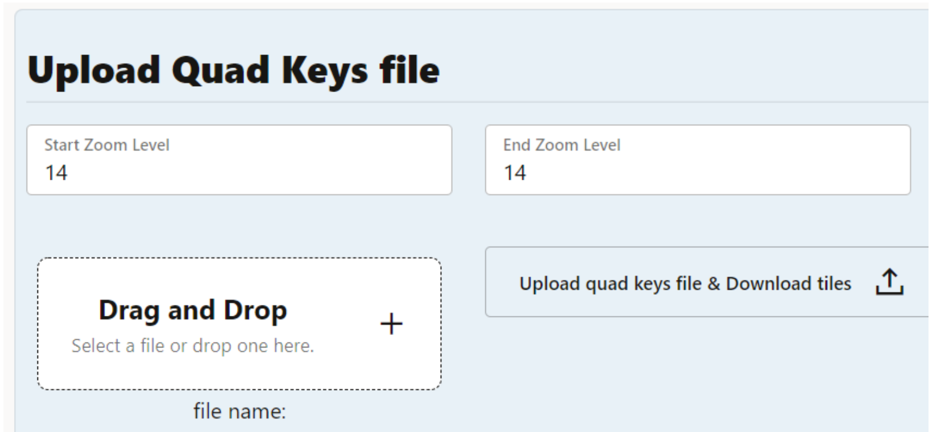


Figure 5.2. Download tiles images for multiple locations.

5.1 DATA ANALYSIS TOOL

It is a tool that enables us to navigate between maps and data for all areas downloaded through the Map Capturing tool, as well as view the words read by OCR and display accuracy measurements for OCR. It uses a set of computer visions techniques to influence the images and measure the accuracy afterwards. It also displays the properties of the texts displayed on the maps. As this tool is shown in the following figure 5.3.

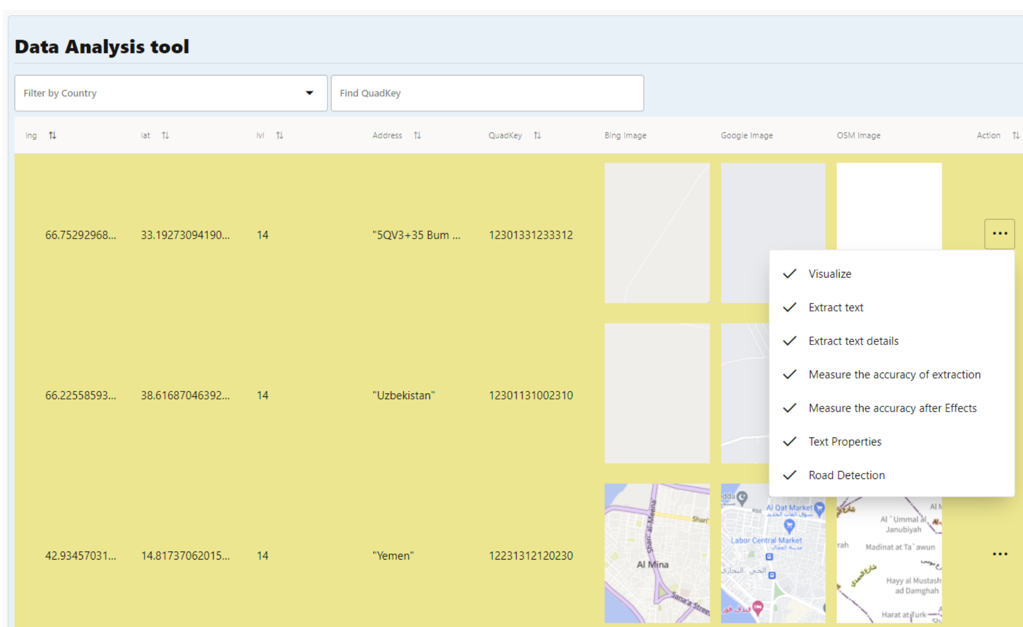


Figure 5.3. DATA ANALYSIS TOOL.

5.1.1 *Map Navigation*

The following figure 5.4 shows three tiles images of three Map providers for the same region, and in fact this image is divided into 4 quadrants when clicking on any quadrant. The zoom Level increases and a new image is displayed. For example, from zoom level 14 to 15 so that there are four images with a greater level of detail for each displayed image. To go back to a higher level, press the previous zoom level button

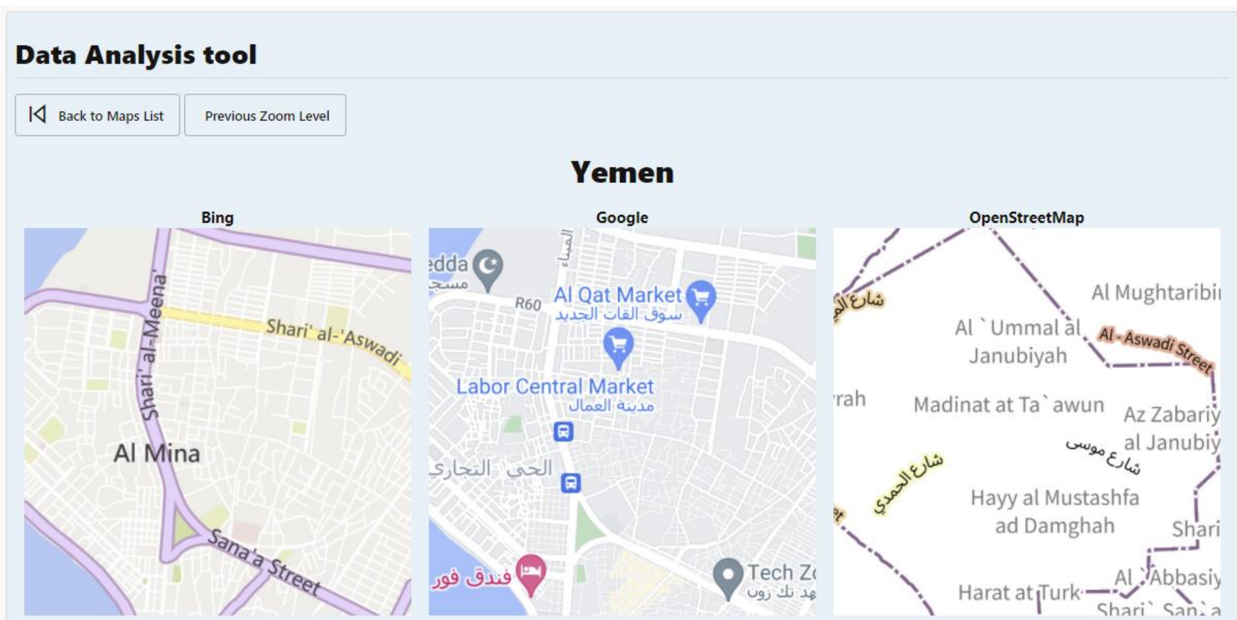


Figure 5.4. Map Navigation.

5.1.2 *Text extraction by OCR*

In the following figure 5.5, texts are read from different OCRs: Tesseract, Azure CV OCR, Google CV OCR, but the system currently only supports results from Tesseract. Jay update that.

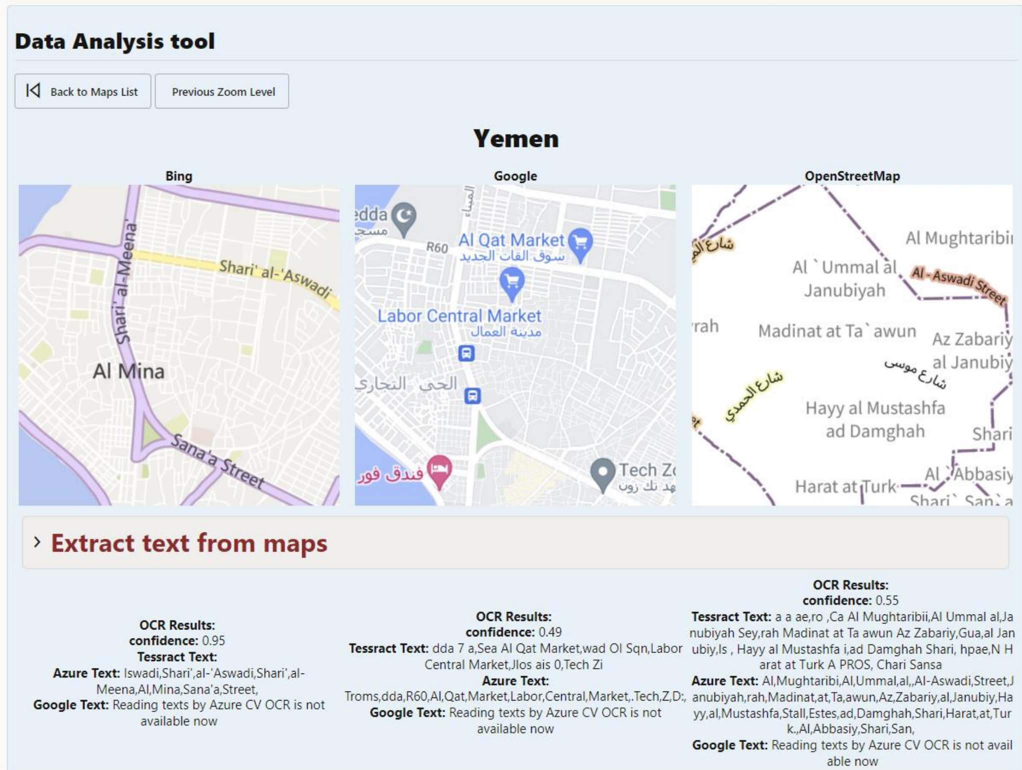


Figure 5.5. Text extraction by OCR.

5.1.3 *Details text extraction*

In the following figure 5.6 the texts from the OCR are read in detail for each word, with each word's location, width, length in pixels and confidence from the OCR and the word. This is for you map provider separately.

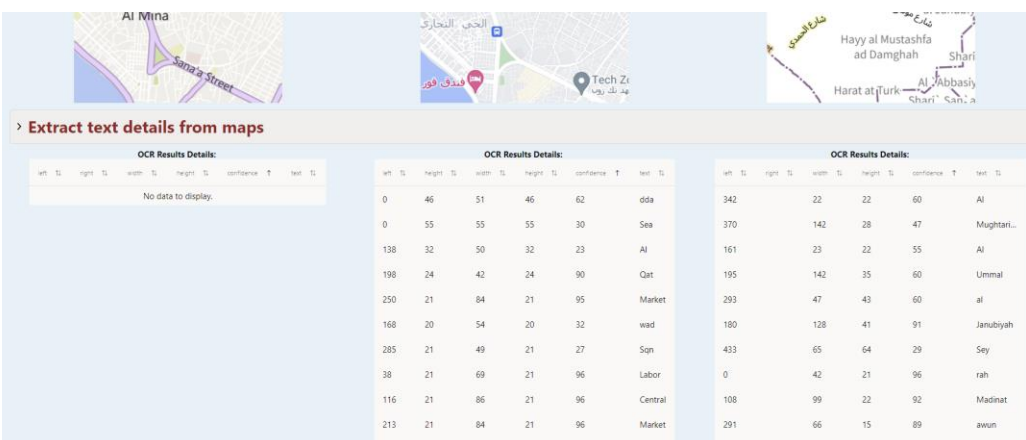


Figure 5.6. Details text extraction.

5.1.4 *Ground truth*

In the following figure 5.7 we can enter the ground truth (which is the text that actually appears on the image). We enter this information manually only to compare the results extracted from the OCR with the correct results, to measure the accuracy of the extracted results, to know whether we need to make improvements to the images so that we can read better or not. And to know the quality of the numbers that are extracted based on the OCR used.

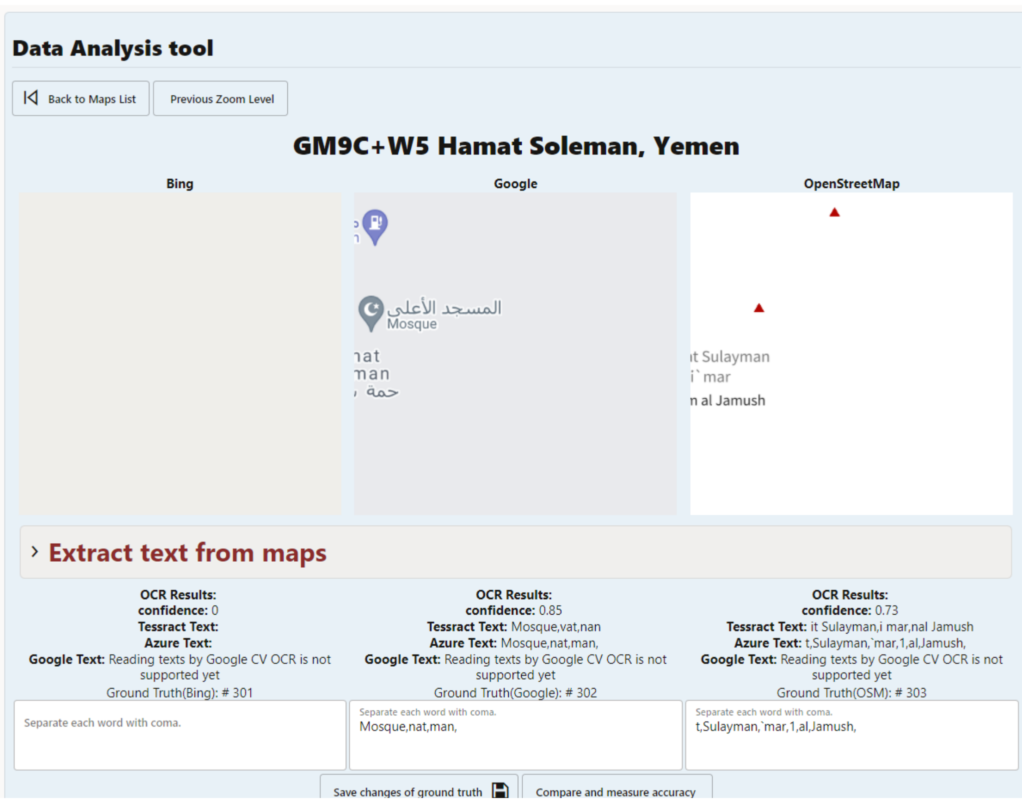


Figure 5.7. Ground truth.

5.1.5 *Compare and measure the accuracy*

After entering the ground truth, we can read the text extracted from the OCR and perform some statistics to know the accuracy of the OCR as shown in Figure 5.8

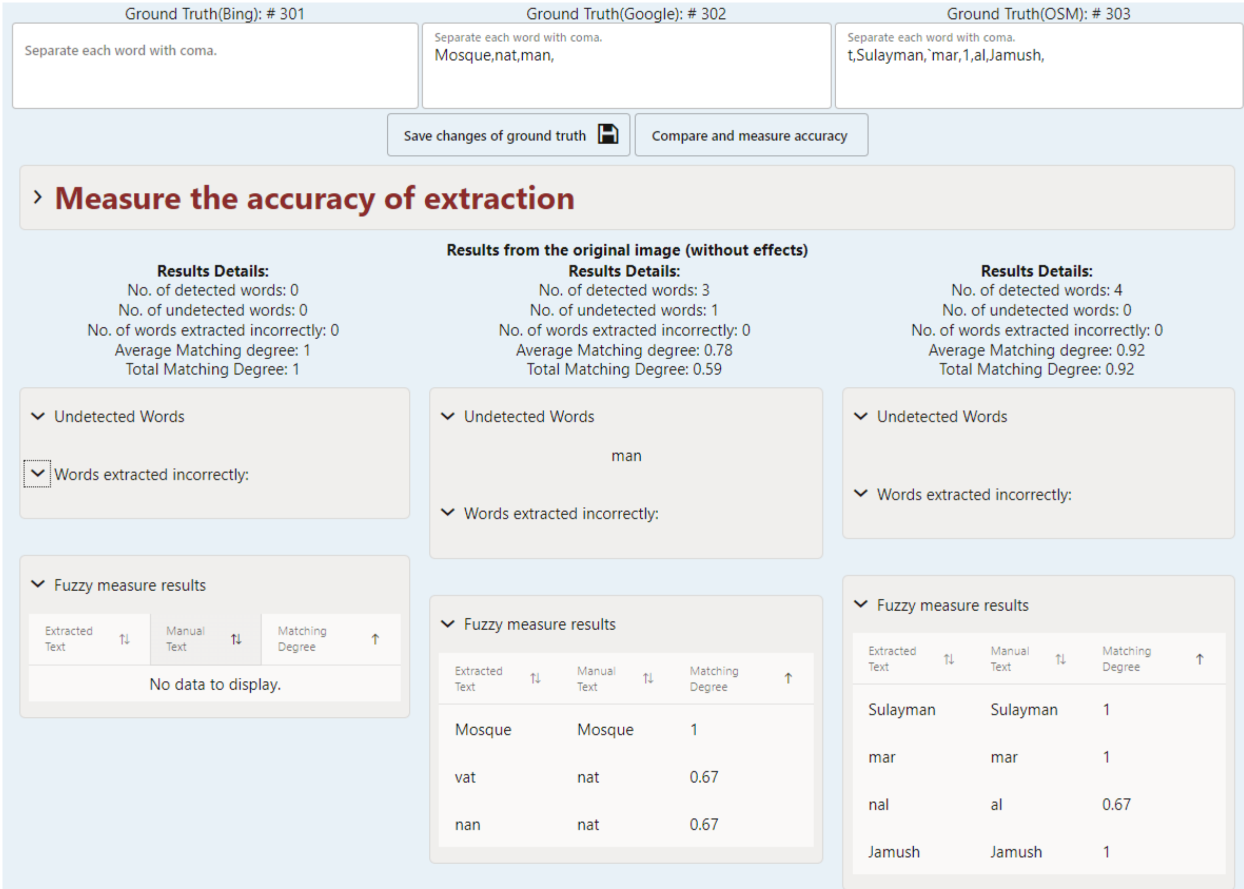


Figure 5.8. Compare and measure the accuracy.

Figure 5.8 shows that Bing in this image has no data, and in Google Map 3 words were extracted and 1 word was lost with a match ratio of 0.78. In the OSM map, 4 words were extracted and no words were not read from the OCR, with a matching ratio of 0.92. The tables below the images show a Fuzzy comparison between text manually entered at ground truth and text automatically extracted from OCR.

5.1.6 *Map tile image modifier*

This tool makes modifications to the image to get a better result than OCR. For example, in Figure 5.9, a group of modifications that will be made to the image has been selected, such as separating the image in the background from the words, making the background transparent, and when pressing “Apply Effects” new images appear after the effect.

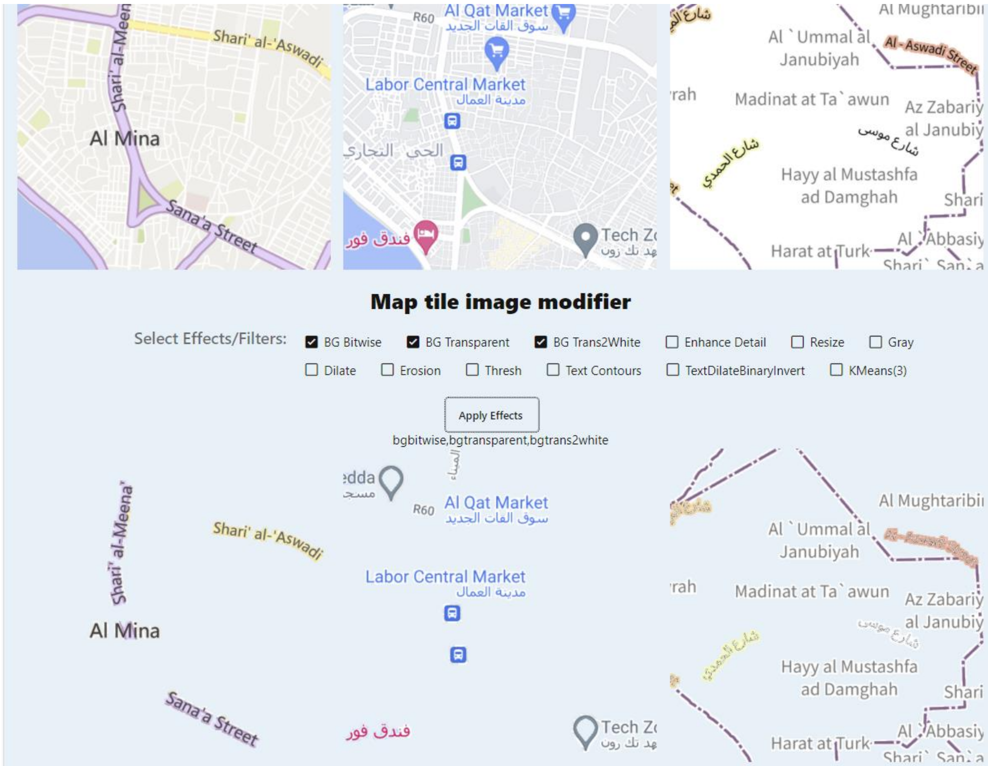


Figure 5.9. Map tile image modifier (Background effects).

Figure 5.10 shows another example using K-means as illustrated in chapter 4 with the number of clusters 3.

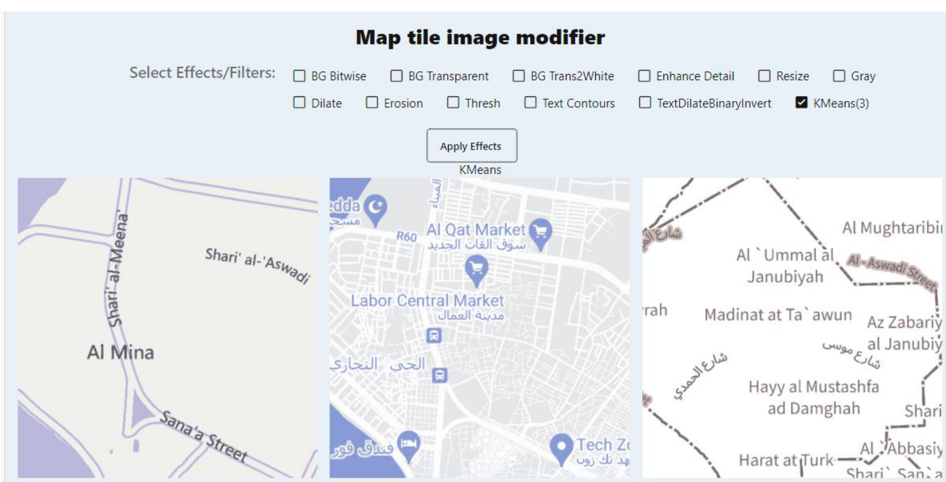


Figure 5.10. Map tile image modifier (K-Means clustering).

Figure 5.11 shows another example using a different set of effects, most notably TextDilateBinaryInvert explained in Chapter 4.

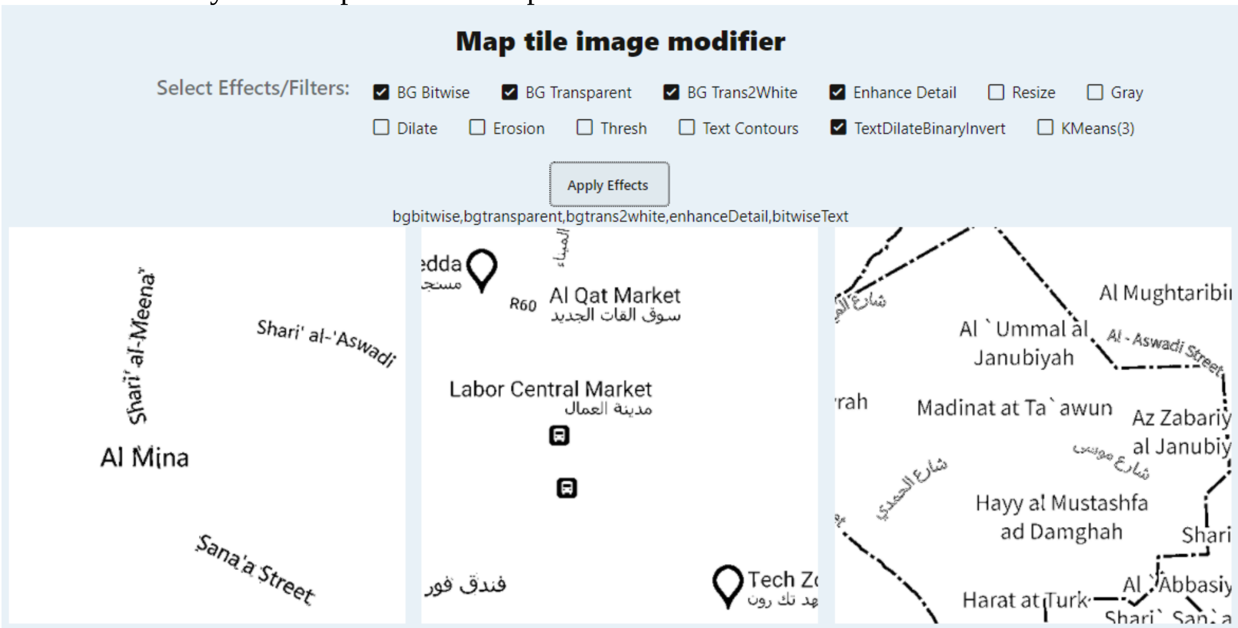


Figure 5.11. Map tile image modifier (group of effects).

After making any modifications to the map images, we can measure the results before and after adding any effects to the images, as shown in the following figure 5.12.

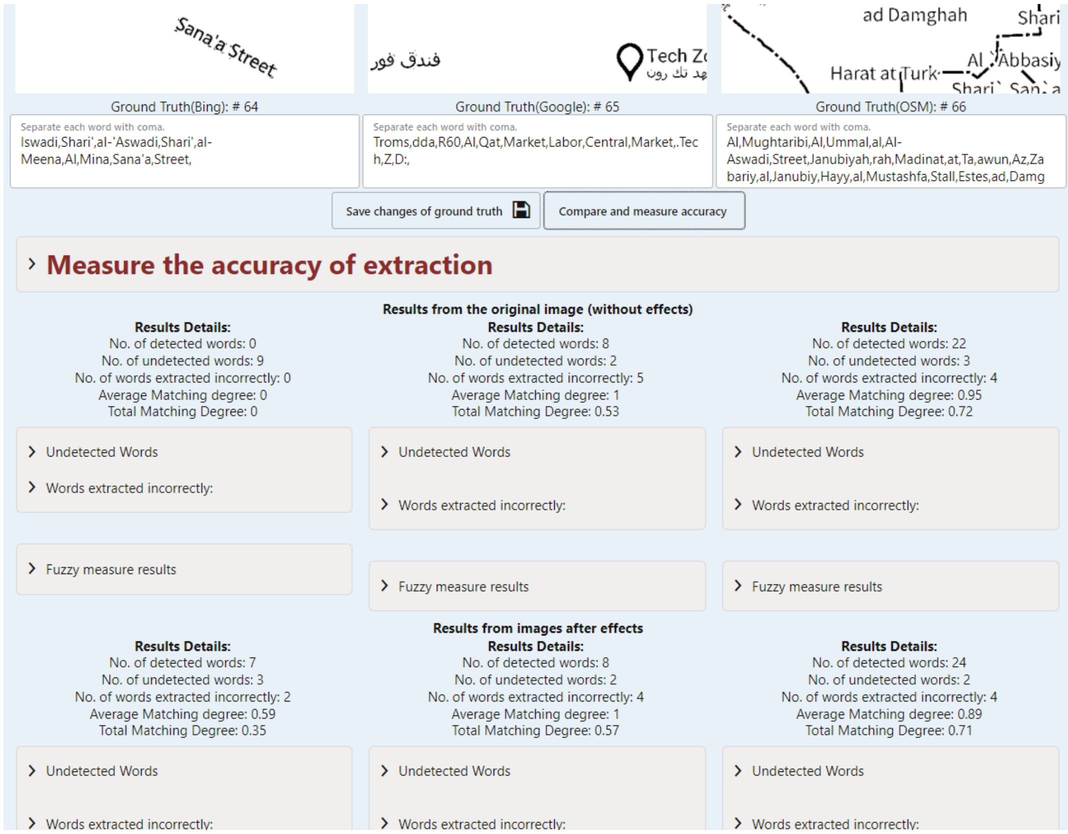


Figure 5.12. Results before and after adding modifications to the map images.

Commenting on the results of the previous figure 5.11

- It appears in the image of Bing Map that after applying the modifications to the image, we were able to read 7 words and before the modifications to the image, the OCR was unable to read any words.
- In the image of a Google Map, there were no significant changes in the OCR result, as the result read 8 words and lost 2 with a percentage that corresponds to 100% of the read words. The only change is that the number of words extracted by mistake from the OCR before modifications to the image is 5 words and after modifications 4.
- In the image of the OSM map, the words extracted before the modifications to the image were 22 and after the modifications 24. The words that the OCR did not read were 3 before the modifications and after the modifications. The percentage of congruence between the

data extracted by the OCR before the modifications is 0.95 and after the modifications is 0.89 because the OCR reads new words that are close but not completely identical to the ground truth.

- As a summary, making modifications to the images using computer vision before reading with OCR will change the result. But the question is what are the best modifications that give us the best results.

5.1.7 *Text Color detector*

This tool specializes in discovering the color of the text used, and it depends on the modifications for separating the background and converting the background color to white, then counting the colors in the image, as shown in Figure 5.13. which represent the colors of the text written in each image after excluding the white color as shown in Figure 5.14 for the entire tile image in general, as This tool also calculates the colors for each separate word.

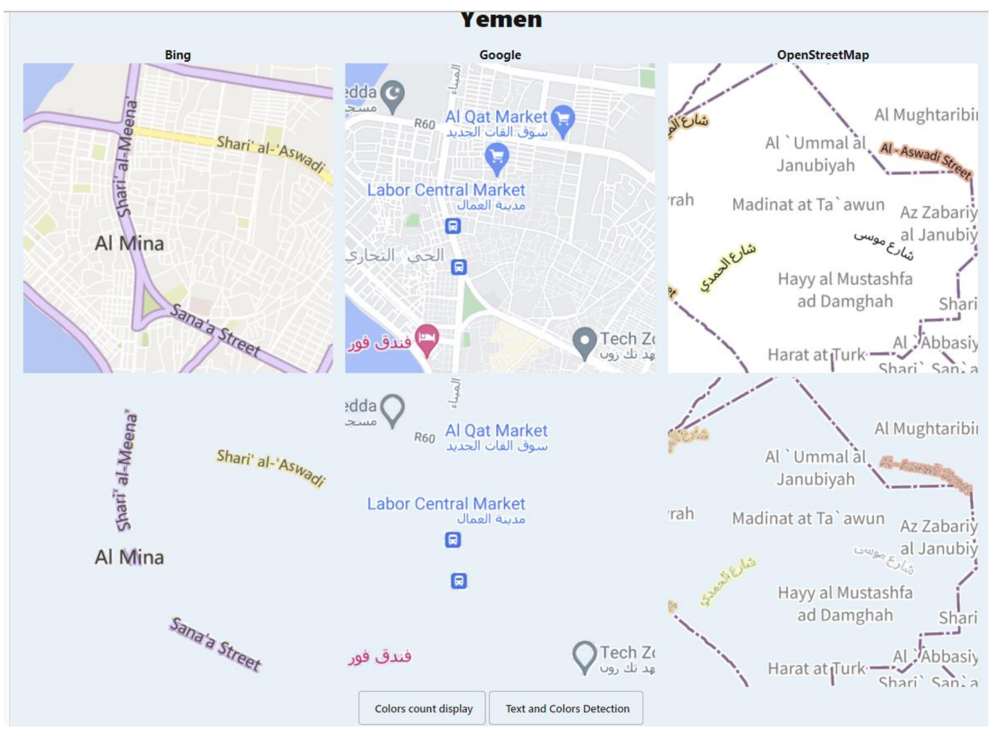


Figure 5.13. Modifications that the text color detector depends on.

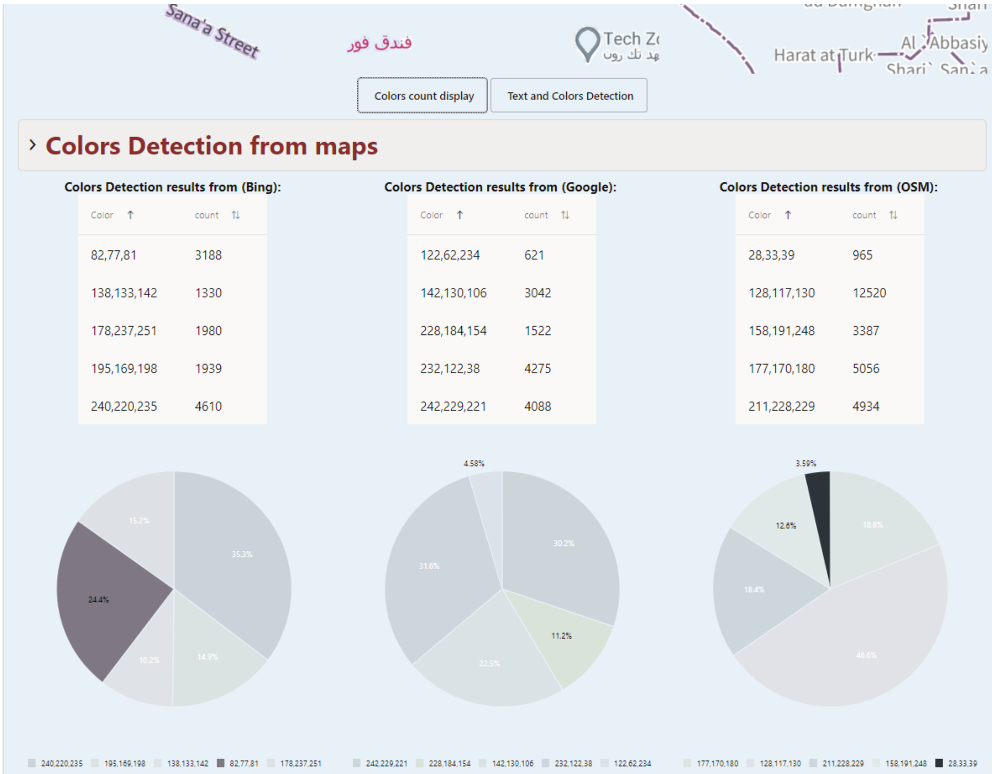


Figure 5.14. Statistics of colors used in the text in the tiles.

5.1.8 *Text Properties*

This tool displays every word read from the OCR and displays it in pixels, its position in the tile image, and the color used in this word as shown in the following figure 5.15

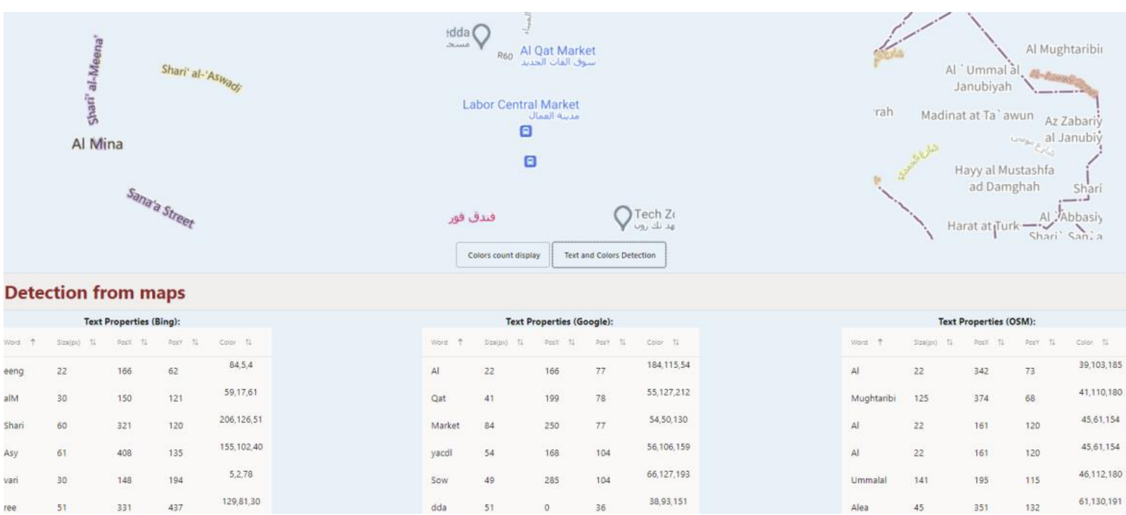
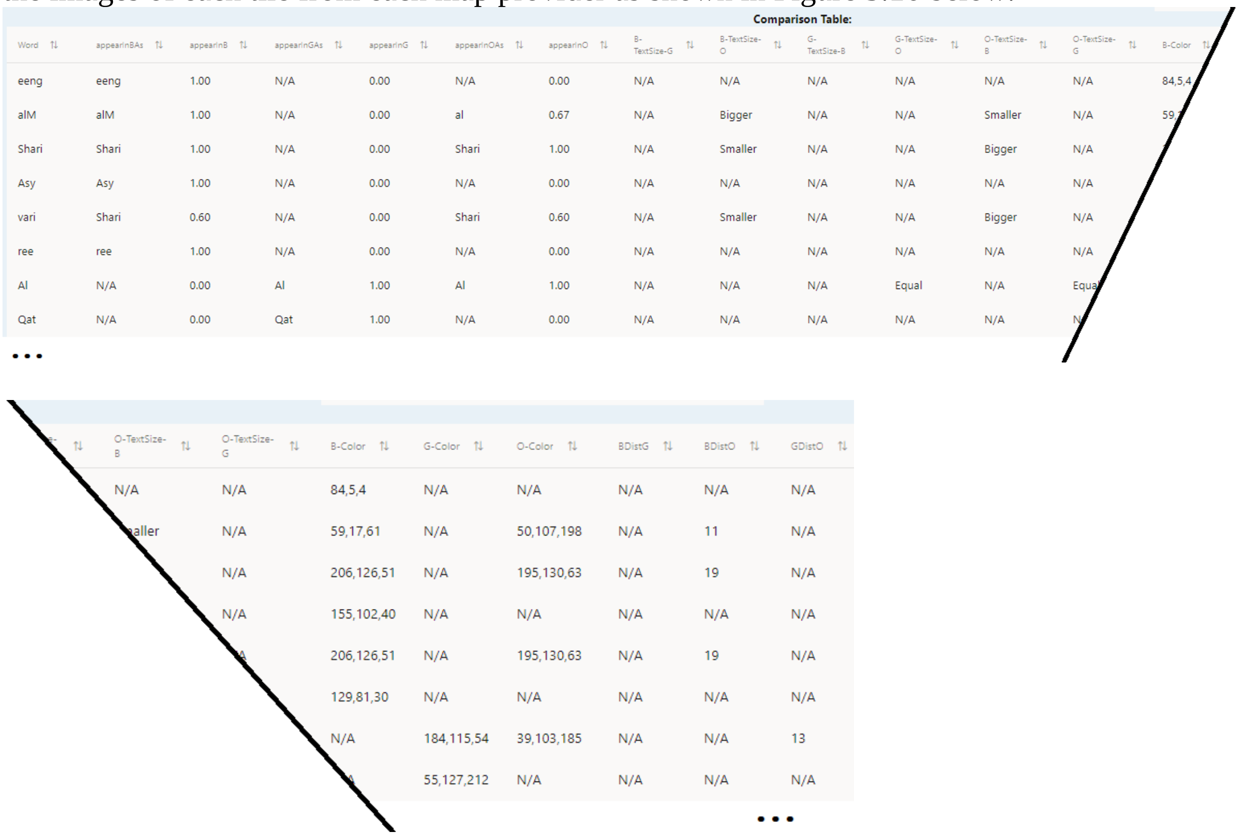


Figure 5.15. Properties of the text displayed in the tile's images.

5.1.9 *Comparison of text properties*

This tool takes advantage of all the previous services to be able to display a comparison between the images of each tile from each map provider as shown in Figure 5.16 below.


Comparison Table:

Word	appearinBAs	appearinB	appearinGAs	appearinG	appearinOAs	appearinO	B-TextSize-G	B-TextSize-O	G-TextSize-B	G-TextSize-O	O-TextSize-B	O-TextSize-G	B-Color
eeng	eeng	1.00	N/A	0.00	N/A	0.00	N/A	N/A	N/A	N/A	N/A	N/A	84,5,4
alM	alM	1.00	N/A	0.00	al	0.67	N/A	Bigger	N/A	N/A	Smaller	N/A	59,
Shari	Shari	1.00	N/A	0.00	Shari	1.00	N/A	Smaller	N/A	N/A	Bigger	N/A	
Asy	Asy	1.00	N/A	0.00	N/A	0.00	N/A	N/A	N/A	N/A	N/A	N/A	
vari	Shari	0.60	N/A	0.00	Shari	0.60	N/A	Smaller	N/A	N/A	Bigger	N/A	
ree	ree	1.00	N/A	0.00	N/A	0.00	N/A	N/A	N/A	N/A	N/A	N/A	
Al	N/A	0.00	Al	1.00	Al	1.00	N/A	N/A	N/A	Equal	N/A	Equa	
Qat	N/A	0.00	Qat	1.00	N/A	0.00	N/A	N/A	N/A	N/A	N/A	N	

• • •

O-TextSize-B	O-TextSize-G	B-Color	G-Color	O-Color	BDistG	BDistO	GDistO
N/A	N/A	84,5,4	N/A	N/A	N/A	N/A	N/A
aller	N/A	59,17,61	N/A	50,107,198	N/A	11	N/A
	N/A	206,126,51	N/A	195,130,63	N/A	19	N/A
	N/A	155,102,40	N/A	N/A	N/A	N/A	N/A
	A	206,126,51	N/A	195,130,63	N/A	19	N/A
		129,81,30	N/A	N/A	N/A	N/A	N/A
		N/A	184,115,54	39,103,185	N/A	N/A	13
		A	55,127,212	N/A	N/A	N/A	N/A

• • •

Figure 5.16. Part of the result of comparing text between the three map providers.

As an illustration of the comparison shown in Figure 5.16 above, the following table 5.1 shows each data column in the comparison result and what it contains. Note that we use **B** for Bing Maps, **G** for Google Maps, and **O** for OpenStreetMap.

Table 5.2. Describe the data in the text comparison table between the map providers

Column(s)	Description
Word	A list of all the words that appear in the map providers images without repetition and using the fuzzy matching degree function.

appearIn**B**As, appearIn**G**As, appearIn**O**As	Each word we have we show in the same row how it appeared in each map provider, whether it matches exactly or matches partially, or N/A to indicate that the word did not appear in this map provider.
appearIn**B**, appearIn**G**, appearIn**B**	Each word we have we show in the same row the percentage of its matches with the word that appeared in each map provider. If it matches exactly, it takes 1 and if it does not exist, it takes 0, and what is between that expresses the percentage of match between the two words
B-TextSize-G	Comparison of the text size of the word between Bing compared to the word shown in Google (takes the value bigger, smaller, same, or N/A if the word is not shown in the two map providers)
B-TextSize-O	Comparison of the text size of the word between Bing compared to the word shown in OpenStreetMap (takes the value bigger, smaller, same, or N/A if the word is not shown in the two map providers)
G-TextSize-B	Comparison of the text size of the word between Google compared to the word shown in Bing (takes the value bigger, smaller, same, or N/A if the word is not shown in the two map providers)
O-TextSize-B	Comparison of the text size of the word between OpenStreetMap compared to the word shown in Bing (takes the value bigger, smaller, same, or N/A if the word is not shown in the two map providers)
O-TextSize-G	Comparison of the text size of the word between OpenStreetMap compared to the word shown in Google (takes the value bigger, smaller, same, or N/A if the word is not shown in the two map providers)

B-Color, **G**-Color, **O**-Color	Displays the color used for the word according to the map provider (B/G/O) or N/A if the word does not appear in that map provider.
BDist**G,** **B**Dist**O,** **G**Dist**O,**	The distance between the word position in the first and second map provider in pixels. Or take the value N/A if the word does not appear in one of the map providers.

Chapter 6. DISCUSSIONS AND CONCLUSION

6.1 OCR STATISTICS

Initially we show a result set about the accuracy of word extraction from OCR used "Tesseract" according to each map provider. As well as statistics about the effects or modifications made to the images to make OCR extract more accurate words.

6.1.1 *Sample size used to measure the accuracy of OCR results:*

The size of the random sample that the ground truth is entered for the OCR test is 198 places for three map providers equal to 594. One resolution from Bing, another from OSM and four Resolutions from Google is handled, so the result is 3,564 and then applies 20 sets of adjustments and effects to the images next to the image the original, without any modifications, so that the size of the readings with which we test the OCR is 74,844 different readings, the results of which we test with the entered ground truth.

6.1.2 *The best effect to extracting the correct results (low resolution)*

Figure 6.1 shows a graph showing the best modifications to the tiles images from each map provider and the accuracy of the extracted words compared to the manually entered ones at resolution 256x256 only because it is available on all map providers.

Table 6.3. Effects on top-ranked images in matching degree among Map providers (low resolutions)

Effects	MapProvider	MatchDegree
enhanceDetail,bitwiseText	Google	89.45
enhanceDetail,bitwiseText	Bing	86.87
bgbitwise,bgtransparent,bgtrans2white,enhanceDetail,resize,bitwiseText	OSM	82.87

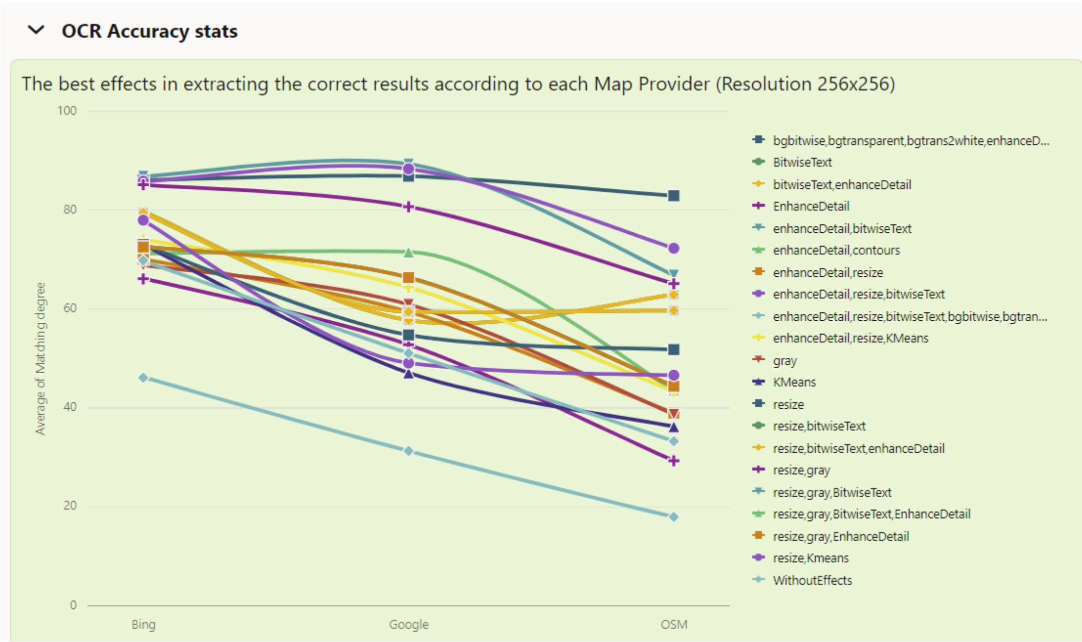


Figure 6.1. The best effect to extracting the correct results (low resolution).

As a comment on the result shown in Figure 6.1 and Table 6.3.

For Bing and Google, the effect of enhanceDetail, bitwiseText gives the best matching accuracy between data extracted from OCR compared to data entered manually, up to 89.5 in Google and 87 in Bing. For OSM, the set of image modifications bgbitwise, bgtransparent, bgtran2white, enhanceDetail, resize, bitwiseText is the most accurate, meaning that one modification does not have to be the best of all the map providers due to the different style of each map provider

6.1.3 *Evaluation of effects according to incorrectly extract/ non-extracted words*

Table 6.4 presents the top 10 effects that give the highest result in text reading according to each map provider, as well as the number of unread words from OCR and the number of misspelled words (the OCR-readers' words that are not found in ground truth).

This data reflects Figure 6.2 on the x-axis the number of words the OCR did not read and on the y-axis the number of words that the OCR had incorrectly read. The size of the circle expresses the number of words that the OCR read correctly. Each circle expresses a set of specific effects. The larger the circle and the closer to zero on the two axes, the better.

Table 6.4. presents the top 10 effects that give the highest result based on detected/undetected/wrong words (low resolutions)

Effects	MapProvider	Detected	Undetected	Wrong
bgbitwise,bgtransparent,bgtrans2white,enhanceDetail,resize,bitwiseText	OSM	328	106	129
enhanceDetail,resize,bitwiseText	Google	321	75	82
enhanceDetail,resize,bitwiseText	OSM	308	121	135
enhanceDetail,bitwiseText	Google	285	123	33
bgbitwise,bgtransparent,bgtrans2white,enhanceDetail,resize,bitwiseText	Google	285	126	51
enhanceDetail,bitwiseText	OSM	207	225	66
enhanceDetail,resize,KMeans	Google	191	216	72
bgbitwise,bgtransparent,bgtrans2white,enhanceDetail,resize,bitwiseText	Bing	185	89	53
enhanceDetail,resize,bitwiseText	Bing	175	93	70
resize,Kmeans	OSM	175	248	53

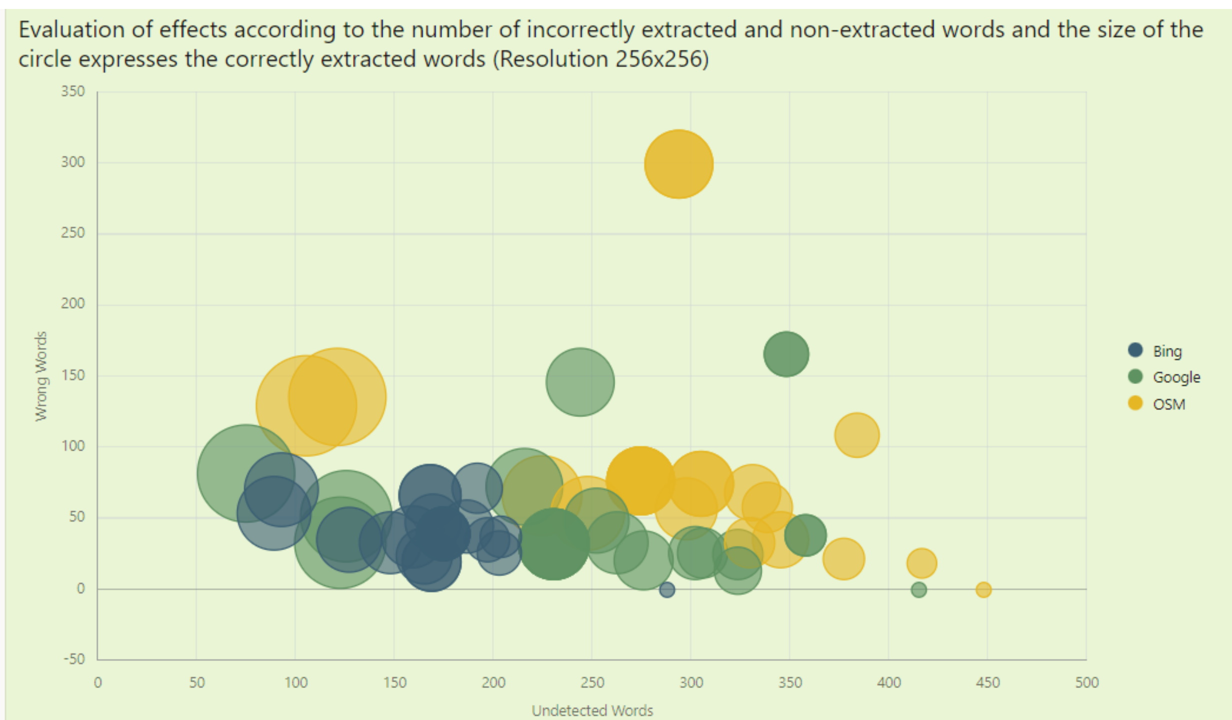


Figure 6.2. Evaluation effects based on detected/undetected/wrong words (low resolution).

6.1.4 *Statistics on the effect of resolutions on the accuracy of the retrieved results*

Since we were able to get four different resolutions from google, we want to know how the resolutions affect the accuracy of the information retrieved from the OCR, considering the different modifications to the images. The following figure 6.3 appears, which shows that the resolution is high after the low resolution of 256 x 256. It also shows that there are effects that give high results with low resolution and vice versa with high resolution and vice versa.

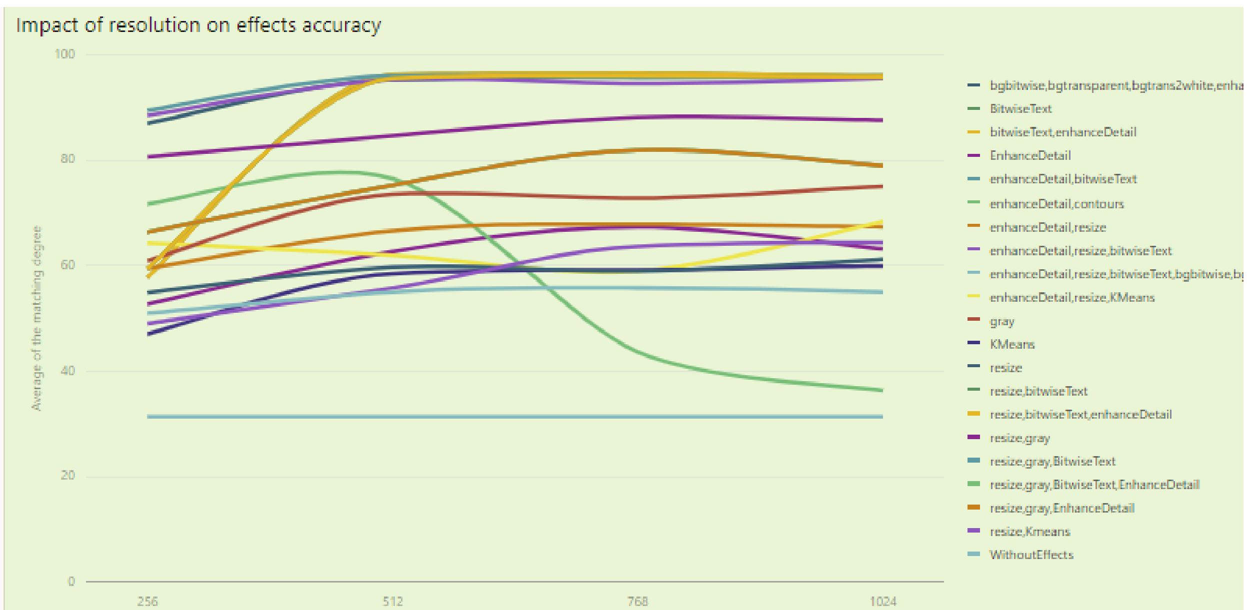


Figure 6.3. Impact of resolutions on the accuracy

Table 6.5 shows the best image modifications that give the highest matching degree according to different resolutions on Google Maps.

Table 6.5. Best image modifications with highest matching degree per resolutions on Google Maps

effects	resolution	MatchDegree
BitwiseText	768	96.52
bitwiseText,enhanceDetail	512	96.25
BitwiseText	1024	96.01
enhanceDetail,bitwiseText	256	89.45

Figure 6.4 displays the same results as before, but in a different way, each group reflects images' modifications and their results during different images resolutions from Google.

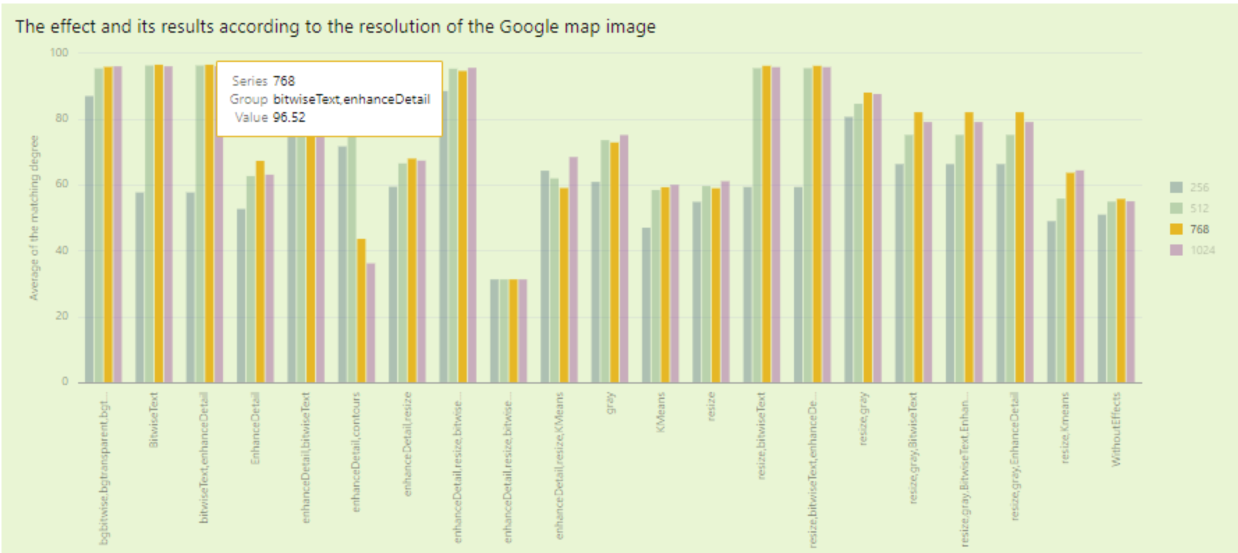


Figure 6.4. The effects and its results according to the different resolutions

6.1.5 *Different measurements show the best "effects" on the tile's images*

Table 6.6 presents the best "effects" that achieve the most harmonic mean between precision and recall "f1". It also shows the difference between the highest percentage obtained on the system and the image of the tile in its normal without using any "effects" on it. Effects improved the "f1" by 38% on Google Maps. It is improved by 27% on Bing Maps. and improved by 51% on OSM maps.

Table 6.6. Best image modifications with highest matching degree per map provider

mapProvider	effects	Matching	precision1	recall	f1
Google	enhanceDetail,bitwiseText	0.89	0.91	0.85	0.86
Bing	bgbitwise,bgtransparent,bgtrans2white,enh	0.86	0.89	0.85	0.85
OSM	bgbitwise,bgtransparent,bgtrans2white,enh	0.83	0.81	0.80	0.78
Bing	WithoutEffects	0.70	0.95	0.56	0.58
Google	WithoutEffects	0.51	0.95	0.46	0.48
OSM	WithoutEffects	0.33	0.96	0.25	0.27

Figure 6.5 shows the matching degree for each set of image modifications, for each map provider. The results are the same as Table 6.6, but the chart is for all the "effects", and of course that's only for the lower resolution.

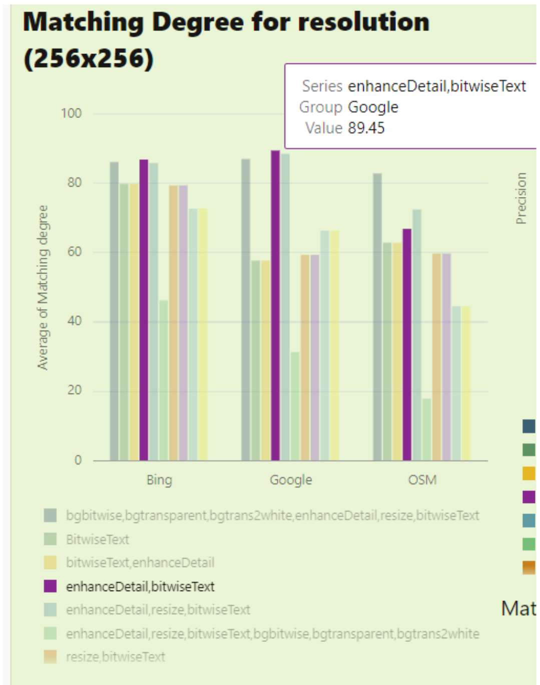


Figure 6.5. Matching degree for resolution (256x256) for each “effects” per map provider

Figure 6.6 shows the precision for each set of image modifications, for each map provider. The results are the same as Table 6.6, but the chart is for all the "effects", and of course that's only for the lower resolution.

Figure 6.7 shows the recall for each set of image modifications, for each map provider. The results are the same as Table 6.6, but the chart is for all the "effects", and of course that's only for the lower resolution.

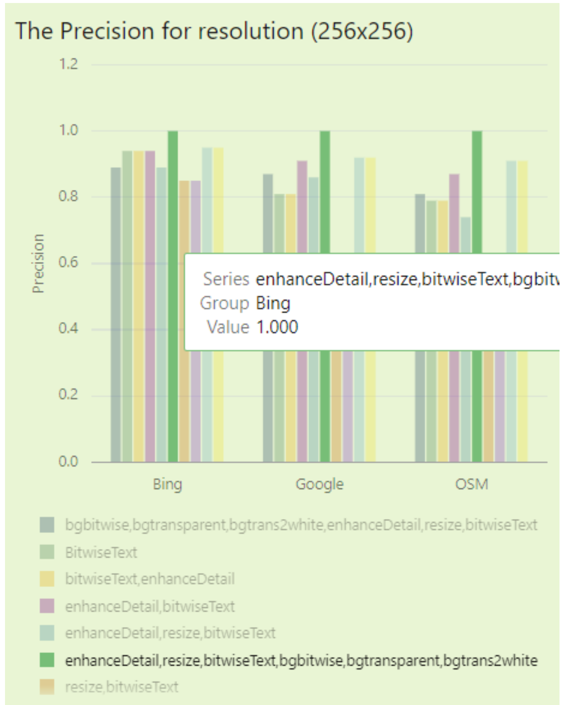


Figure 6.6. Precision for resolution (256x256) for each “effects” per map provider

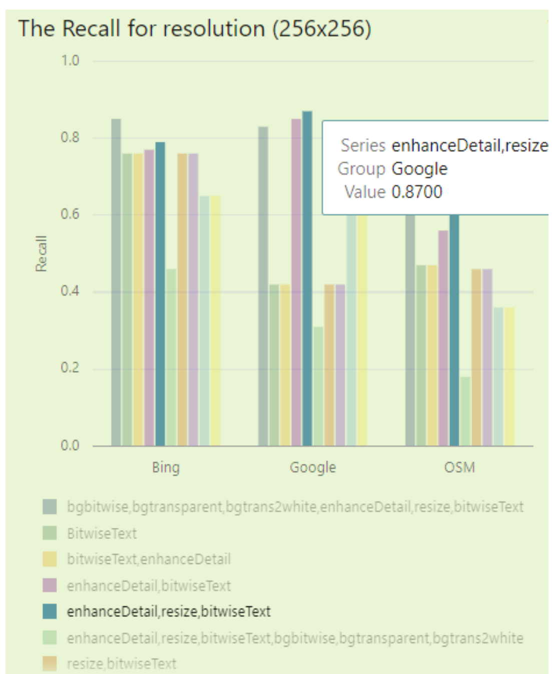


Figure 6.7. Recall for resolution (256x256) for each “effects” per map provider

Figure 6.8 shows the F1(harmonic mean between the precision and recall) for each set of image modifications, for each map provider. The results are the same as Table 6.6, but the chart is for all the "effects", and of course that's only for the lower resolution.

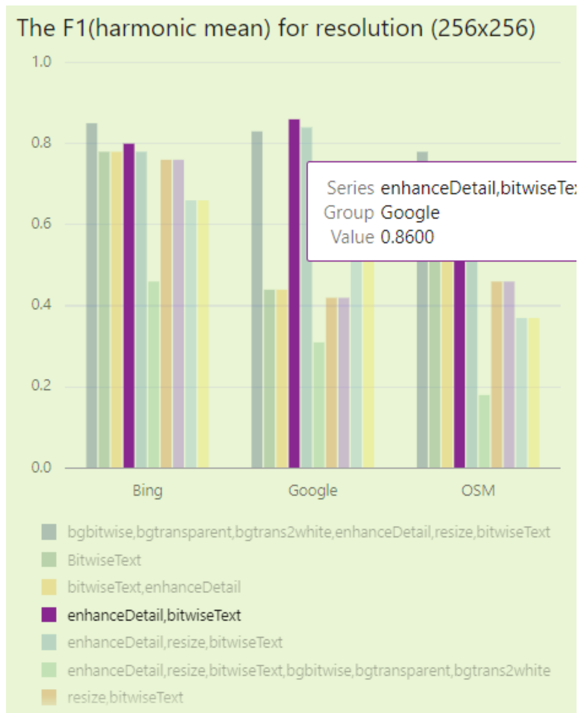


Figure 6.8. F1 for resolution (256x256) for each “effects” per map provider

6.1.6 *Different measurements show the best "effects" based on images resolutions*

Table 6.7 presents the best "effects" for google maps that achieve the most harmonic mean between precision and recall "f1". It also shows the difference between the highest percentage obtained on the system and the image of the tile in its normal without using any "effects" on it. Effects improved the "f1" by 38% for the low resolution 256x256. It is improved by 39% for resolution 512x512. and improved by 43% for resolution 768x768. Effects improved the "f1" by 40% for the high resolution 1024x1024.

Table 6.7. Best image modifications with highest matching degree per resolutions on Google Maps

Resolution	Effects	matchingDegree	Precision	Recall	F1
512	BitwiseText	96.25	0.94	0.89	0.89
768	BitwiseText	96.52	0.94	0.89	0.89
1024	BitwiseText	96.01	0.94	0.89	0.89
256	enhanceDetail,bitwiseText	89.45	0.91	0.85	0.86
512	WithoutEffects	54.94	0.96	0.5	0.5
1024	WithoutEffects	55.06	0.96	0.48	0.49
256	WithoutEffects	50.99	0.95	0.46	0.48
768	WithoutEffects	55.76	0.93	0.47	0.46

Figure 6.9 shows the matching degree for each set of image modifications per different resolution for google maps. The results are the same as Table 6.7, but the chart is for all the "effects".

Figure 6.9. Matching degree for each “effects” per different resolutions for google maps

Figure 6.10 shows the precision for each set of image modifications per different resolution for google maps. The results are the same as Table 6.7, but the chart is for all the "effects".

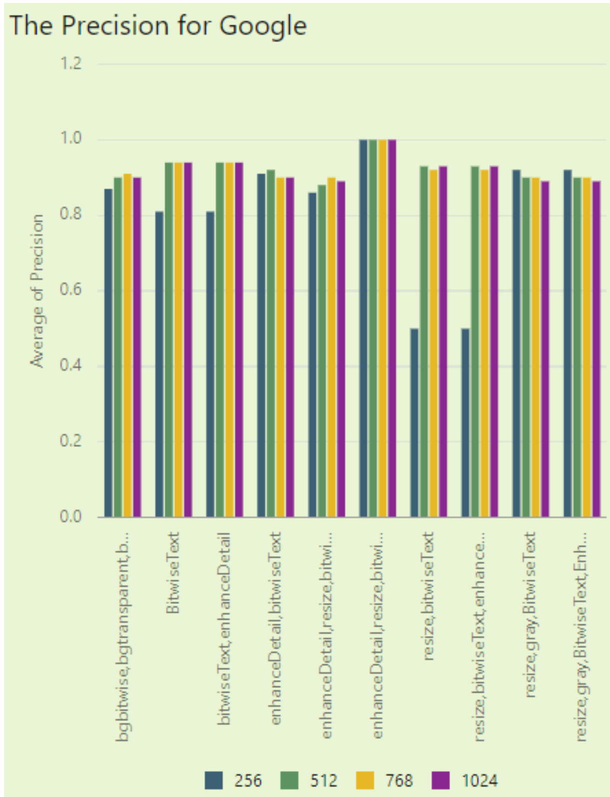


Figure 6.10. The precision for each "effects" per different resolutions for google maps

Figure 6.11 shows the recall for each set of image modifications per different resolution for google maps. The results are the same as Table 6.7, but the chart is for all the "effects".

Figure 6.11. Recall for each “effects” per different resolutions for google maps

Figure 6.12 shows the matching degree for each set of image modifications per different resolution for google maps. The results are the same as Table 6.7, but the chart is for all the "effects".

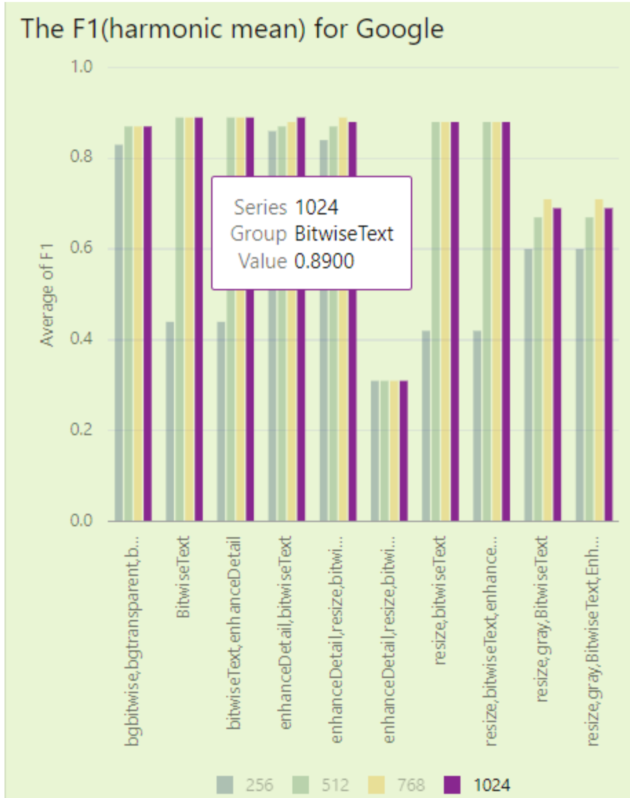


Figure 6.12. F1 for each “effects” per different resolutions for google maps

6.1 TEXT ANALYZER

This is the desired step, which uses all of the above to show some straightforward results. The dataset that the text analyzer was run on was a random sample distributed among the different countries also randomly. And of course, the same areas for all the map providers. The dataset has a size of 9,918 images, which are 3,306 images from Bing, such as that from Google, and the same from OSM.

6.1.1 *Statistics on the percentage of words extracted at the country level:*

Figure 6.13 presents a statistic that is the percentage of accuracy of words in all the pictures per countries. Table 6.8 presents a sample in numbers from this statistic for three countries.

Table 6.8. Sample of three countries and the percentage of words appearing in each map provider

Map provider	Percentage	country
Bing	66.67	Russia
OSM	66.67	Russia
Google	62.67	Russia
Google	56.67	Turkey
OSM	43.33	Turkey
Bing	3.33	Turkey
Google	53.67	Turkmenistan
OSM	44.89	Turkmenistan

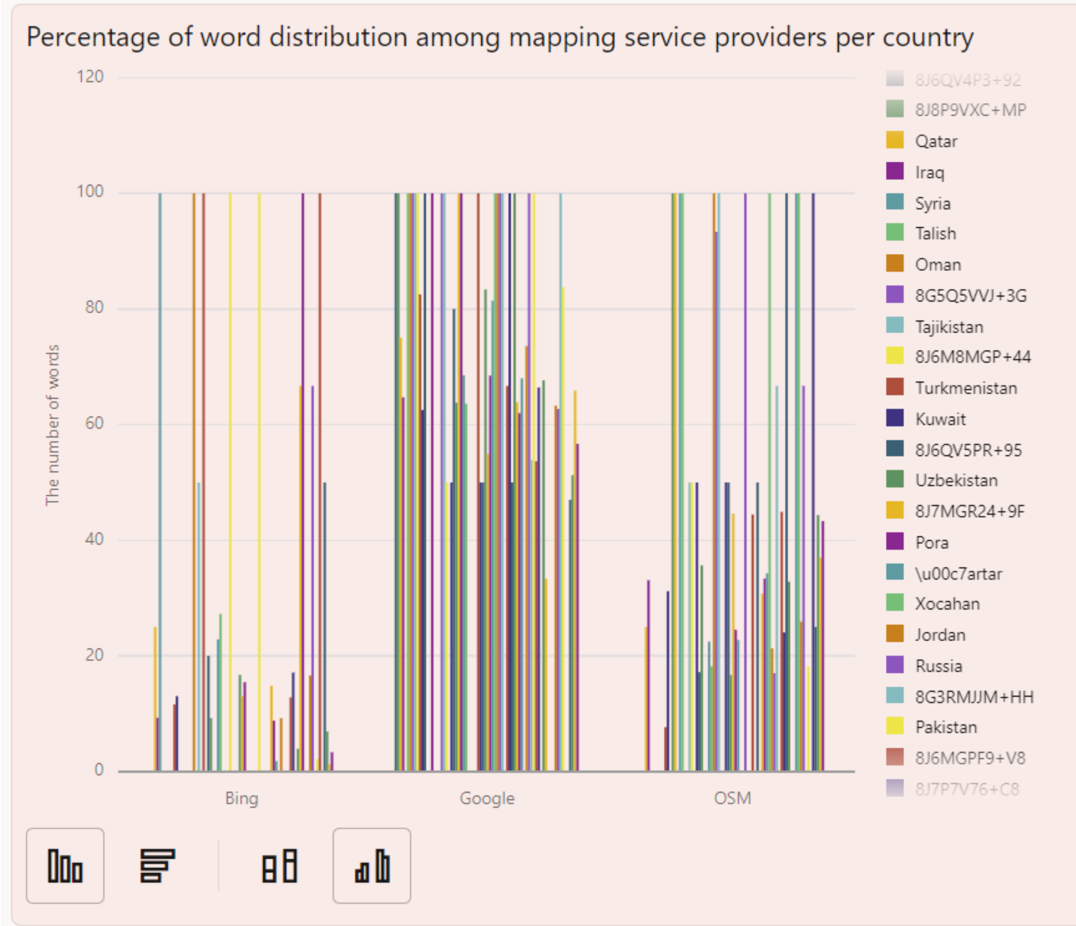


Figure 6.13.Statistics on the percentage of words extracted at the country level

6.1.2 *Statistics on the number of missing words in each map provider at the country level*

Figure 6.14 shows a statistic of the number of missing words in the map provider and found by the rest of the map providers. Table 6.9 presents a sample in numbers from this statistic for three countries.

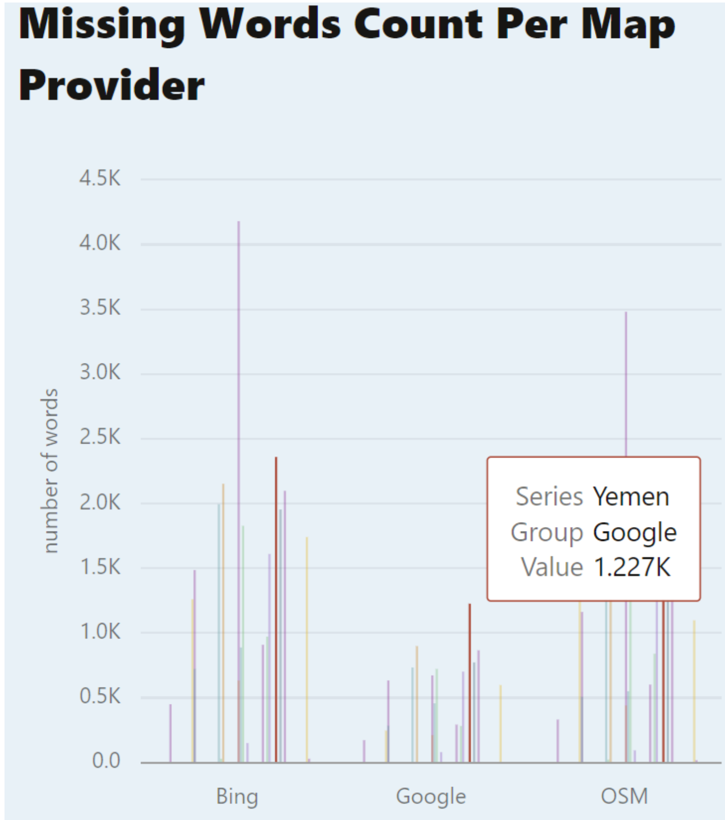


Figure 6.14. Statistics on the number of missing words in each map provider per country

Table 6.9. Sample of three countries with number of missing words in each map provider per country

Map provider	#Missing	country
Bing	1827	Lebanon
OSM	1820	Lebanon
Google	722	Lebanon
Bing	29	Turkey
OSM	17	Turkey
Google	13	Turkey
Bing	2359	Yemen
OSM	1391	Yemen
Google	1227	Yemen

We note from the table 6.9 data that the number of missing words in Turkey is small for all the map providers and at the same time convergent, which may reflect that information about this area is available to all map providers. This contrasts with the large numbers of missing words in all the map provider in Yemen, which may reflect that each map provider displays very different information from the other in this area. The missing words in Lebanon are small in Google compared to the rest of the map providers which reflects that google is full of information in this area, so the rest of the map providers lose many of the words that do not appear in their maps.

6.1.3 *Statistics on the number of mismatched words between the map provider per country*

Figure 6.15 shows a statistic about the number of words written with some letters in the map provider and with other letters in the rest of the map providers despite their similarity (i.e. transpositioned characters or not present, and so on). Table 6.10 presents a sample in numbers from this statistic for three countries.

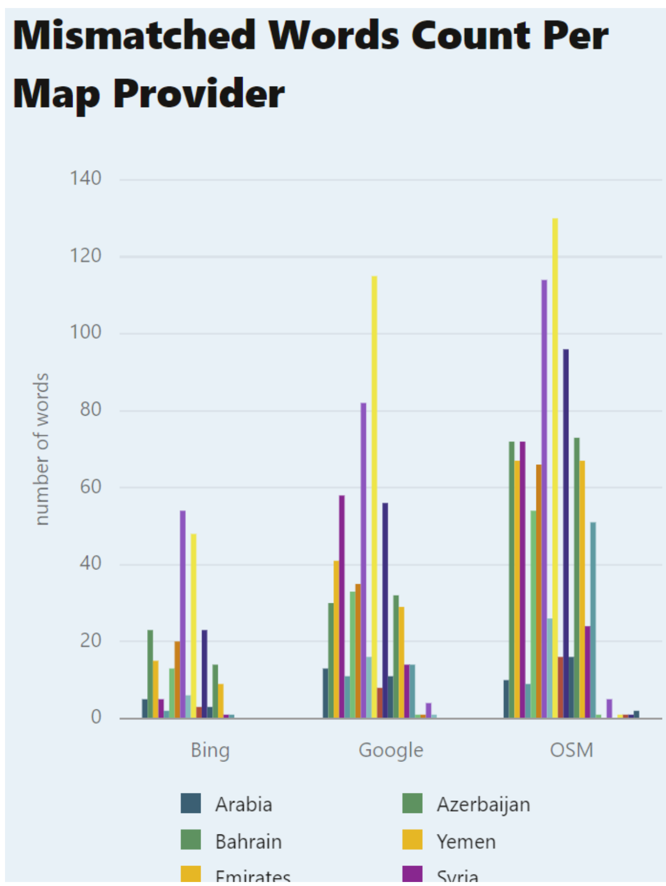


Figure 6.15. Statistics on the number of mismatched words in each map provider per country

Table 6.10. Sample of three countries with number of mismatched words in each map provider per country

map provider	WordsMismatched	country
OSM	26	Armenia
Google	16	Armenia
Bing	6	Armenia
OSM	130	Lebanon
Google	115	Lebanon
Bing	48	Lebanon
OSM	67	Yemen
Google	29	Yemen
Bing	9	Yemen

We notice from the numbers in Table 6.10 that in Lebanon the numbers between Google and OSM are large, which may reflect that the conversion of Lebanon's local language to English characters created many inconsistencies even though the words are similar. This is in contrast to Armenia which has mismatched words with small numerical differences.

6.1.4 *Statistics on words of different sizes:*

Table 6.11 shows the number of small words S for each map provider in comparison to the other map providers. It also displays the number of large words B for each map provider in comparison to the other map providers. And that data is according to the country, as the number of words in the country and the number of locations downloaded in this country in our dataset.

Table 6.11. Sample of 10 countries with statistics on words of different sizes

NoOfWords	B-S-Count	G-S-Count	O-S-Count	B-B-Count	G-B-Count	O-B-Count	country	locations
4275	13	58	137	36	131	43	Pakistan	302
2538	40	103	147	115	87	83	Qatar	55
2538	12	18	66	25	47	24	Yemen	133
2390	70	131	196	200	150	52	Lebanon	83
2379	52	82	117	117	84	53	Kuwait	50
2365	21	86	121	119	71	38	Emirates	78
2305	21	35	70	44	52	30	Iraq	120
1939	31	47	68	57	48	39	Jordan	66
1763	1	15	70	4	70	12	Iran	220
1717	27	30	92	57	59	34	Bahrain	29

6.1.5 *Statistics on different positioned words from one map provider to another:*

Table 6.12 shows the number of different positioned words between each two-map provider. We note, for example, in the Google and OSM comparison column, that the numbers are always higher than the rest of the comparisons. This may reflect the large difference in word positions between google and OSM, or the lack of words in Bing.

Table 6.12. Sample of 10 countries with statistics on different positioned words from one map provider to another

NoOfWords	PosDiff-B-G	PosDiff-B-O	PosDiff-G-O	country	locations
4275	14	10	77	Pakistan	302
2538	41	47	72	Qatar	55
2538	5	21	34	Yemen	133
2390	67	52	68	Lebanon	83
2379	43	59	61	Kuwait	50
2365	45	35	64	Emirates	78
2305	15	17	37	Iraq	120
1939	19	31	33	Jordan	66
1763	3	0	23	Iran	220
1717	21	31	41	Bahrain	29

We have presented some results that show what the system offers, and we were not able to put all the results because of their large size, despite the smallness of the dataset to some extent. But now we're doing more than one different dataset that's focused and expressive of reality, and then deduces numbers that express the actual differences between the real-life map providers and not just a simple sample.

Results and discussion in this section about the dataset used is currently an experimental use to verify the results of the system, but we are adding integrated data to the system according to different countries through a group of professional editors and cartographers to help them correct and improve the quality of digital maps, and thus This system will benefit from the additions and modifications that they will propose.

BIBLIOGRAPHY

[1] Zhu, A-Xing, Fang-He Zhao, Peng Liang, and Cheng-Zhi Qin. "Next generation of GIS: must be easy." Annals of GIS 27, no. 1 (2021): 71-86.

[2] Bandil, Ayush, Vaishali Girdhar, Hieu Chau, Mohamed Ali, Abdeltawab Hendawi, Harsh Govind, Peiwei Cao, and Ashley Song. "GeoDart: A System for Discovering Maps Discrepancies." In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 2535-2546. IEEE, 2021.

[3] Tabet, Fares, Sikha Pentyala, Birva H. Patel, Abdeltawab Hendawi, Peiwei Cao, Ashley Song, Harsh Govind, and Mohamed Ali. "OSMRunner: A System for Exploring and Fixing OSM Connectivity." In 2021 22nd IEEE International Conference on Mobile Data Management (MDM), pp. 193-200. IEEE, 2021.

[4] L. Cohen, Time-Frequency Analysis, Prentice-Hall, 1995.

[5] Google Maps Market Share and Web Usage Statistics (similartech.com), https://www.similartech.com/technologies/google-maps, last access 2021

[6] • Top U.S. mapping apps by downloads 2020 | Statista, 2020, https://www.statista.com/statistics/865413/most-popular-us-mapping-apps-ranked-by-audience/, last access 2021.

[7] Bing Maps Tile System - Bing Maps | Microsoft Docs, 2018 https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system, last access 2021.

[8] Scheuerman, Morgan Klaus, Alex Hanna, and Emily Denton. "Do datasets have politics? Disciplinary values in computer vision dataset development." Proceedings of the ACM on Human-Computer Interaction 5, no. CSCW2 (2021): 1-37.

[9] Park, Seulki, Dae Ung Jo, and Jin Young Choi. "Over-Fit: Noisy-Label Detection based on the Overfitted Model Property." arXiv preprint arXiv:2106.07217 (2021).

[10] Ranjan, Ashish, Varun Nagesh Jolly Behera, and Motahar Reza. "OCR Using Computer Vision and Machine Learning." In Machine Learning Algorithms for Industrial Applications, pp. 83-105. Springer, Cham, 2021.

[11] Khan, Arjumand Masood, and Khan Rahat Afreen. "An approach to text analytics and text mining in multilingual natural language processing." Materials Today: Proceedings (2021).

[12] Deng, J. X., and Yong Deng. "Information Volume of Fuzzy Membership Function." International Journal of Computers, Communications & Control 16, no. 1 (2021).

[13] Bandil, Ayush, Vaishali Girdhar, Kivanc Dincer, Harsh Govind, Peiwei Cao, Ashley Song, and Mohamed Ali. "An Interactive System to Compare, Explore and Identify Discrepancies across Map Providers." In Proceedings of the 28th International Conference on Advances in Geographic Information Systems, pp. 425-428. 2020.