JVLC

Journal of Visual Language and Computing

Volume 2022, Number 2

DOI: 10.18293/JVLC2022-N2

Journal preparation, editing and printing are sponsored by KSI Research Inc.

Journal of
Visual Language and Computing

An Open Access Journal published by

KSI Research Inc.

156 Park Square Lane, Pittsburgh, PA 15238 USA

Journal of
Visual Language and Computing

Volume 2022, Number 2

September 2022

Table of Contents

Regular Papers

Computer-Assisted Visual Reasoning for Territorial Intelligence. 1
Robert Laurini and Rosa Marina Donolo

Indicators in *Super Mario Maker 2*: Evolution and Rhetorical Signification in Visual Languages 12
Nathan W. Eloe and Trevor C. Meyer

Implementing BERT With Ktrain Library For Sentiment Analysis. 26
Hasnan Othman and Mohd Ridzwan Yaakub

Research Notes

Construction of Art Education Platform in Primary and Secondary Schools Based on Virtual Reality Technology 35
Lu Qi

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Computer-Assisted Visual Reasoning for Territorial Intelligence

Robert Laurini[a], Rosa Marina Donolo[b]

[a]Knowledge Systems Institute, USA, and University of Lyon, France

[b]Dept of Civil, Construction-Architectural and Environmental Engineering, University of L'Aquila, Italy

ARTICLE INFO

Article History:
Submitted 7.31.2022
Revised 8.27.2022
Accepted 9.5.2022

Keywords:
Geographic knowledge
Visual Reasoning,
Geovisualization,
Territorial Intelligence
Visual Analytics,
Datascapes.

ABSTRACT

In territorial intelligence, it is very interesting to provide computer-based tools to help reasoning especially in urban, regional and environmental planning. Traditionally, decision-makers use maps in their daily work, but they are limited in the expressive power to help reasoning, i.e., to assist them in deducing knowledge about salient problems and opportunities, and generating ideas and future projects. By means of visual analytics, and more specifically geovisualization, it seems possible. The scope of this paper is to rapidly analyze how painting has passed from representing objects as they are recognized to showing their relationships as a first step for reasoning. A similar study is made from conventional mapping to geovisualization, beyond traditional cartography, namely cartograms, chorems, datascapes, etc. as a way to base visual reasoning.

1. Introduction

For decision-makers in local authorities, it is important to capture and manage data, but overall information and knowledge in order to govern the territory under their jurisdiction. For that purpose, various software products have been created ranging for GIS systems, spatial analysis tools to recent systems based on deep learning and knowledge management, i.e., systems with some reasoning capabilities, for instance in urban and regional planning. In geographic applications knowledge has no meaning in itself but derives its value from its use in practice.

For a territory, knowledge corresponds to information potentially useful in order to make reasoning such as (Laurini et al. 2022):

[a]Corresponding author
Email address: Robert.Laurini@liris.cnrs.fr
Website: http://www.laurini.net/robert/
ORCID: 0000-0003-0426-4030

• explaining and making understandable the dynamics of a territory as well as its interactions with other adjoining places in the same or neighboring countries.
• managing a territory by some local authorities, i.e., by means of some decision-support system, in the spirit of territorial intelligence;
• monitoring its daily development through feedbacks and adaptation;
• simulating the future, and design novel projects;
• orienting actions for the future.

In parallel, visual representation of territories has evolved from conventional cartography to geovisualization systems. According to MacEachren (2004) geovisualization (short for geographic visualization), also known as cartographic visualization, refers to a set of tools and techniques supporting the analysis of geospatial data through the use of interactive visualization. Like the related fields of scientific visualization and information visualization (Stuart et al., 1999), geovisualization emphasizes knowledge

DOI reference number: 10.18293/JVLC2022-N2-021

construction over knowledge storage or information transmission. To do this, geovisualization communicates geospatial information in ways that, when combined with human understanding, allows for data exploration and decision-making processes (Jiang and Li, 2005; MacEachren, 2004).

What is visual reasoning? First, according to the Merriam-Webster dictionary, reasoning is defined as the use of reason, and especially the drawing of inferences or conclusions through the use of reason.

So, the research question addressed in this paper is to analyze how computer-based visualization can provide novel methods of reasoning about a territory. To help answer this question, we will study a few historical issues before the advent of computers, then examine what the possible solutions are existing now for visual reasoning.

2. Some Historical Landmarks

In this section, it seems interesting to study in what degree the visual representations were made in order to help reason. From a historical point of view, two directions will be detailed, namely painting and cartography.

2.1 Painting

The goal of this section is not to make a global history of painting but rather to examine a few ideas regarding the relationships between painting and the reality.

Fig. 1. Egyptians used to represent objects as they are recognized. Source https://www.britishmuseum.org/collection/object/Y_EA37983

Regarding prehistoric people, it seems that some paintings in cave have a sort of magic power to act on reality (Bégouen 1929). Later during the Egyptian period, the idea was to represent objects as they are recognized. For instance, in the famous Nebamon tumb, there is a painting (Figure 1) representing plants and animals in a garden: their flat representation allows anybody to recognize them without any problem. In the same idea, ask a child to represent a fish, s/he will not draw it from the top, from the bottom nor the face but rather from its side to easily recognize it.

Then, in painting, the dominant idea was rather to show some mythologic or religious paintings, sometimes far from reality. Later with the discovery of perspective, in 1435 Alberti wrote a treatise entitled *De Pictura* (On Painting) in which he outlined a process for creating an effective painting using one-point perspective (Sinisgalli, 2011). Now it was possible to represent objects as they are seen (see a painting in Figure 2 from Piero Perugino in 1481. The summum was reached by Leonardo da Vinci (1452-1519) in his *Trattato della pittura* (Treatise on Painting).

Fig. 2. Pietro Perugino's painting representing objects as they are seen. Source: https://www.analisidellopera.it/consegna-delle-chiavi-a-san-pietro-perugino/

Fig. 3. Pablo Picasso's Guernica. Painting emphasizing relationship between objects. Source: https://www.museoreinasofia.es/en/collection/artwork/guernica

Later during the early 20th-century art rebelled against the traditional understandings of painting. They focused on the relationship between the objects rather than on the traditional single-point perspective.

To conclude this section about painting, a sort of evolution schema can be outlined ranging from the visual representation of objects as they are recognized, as they are seen and finally the relationship between them which will be considered as a first step for reasoning.

2.2 Cartography of Small Territories

This is not the goal of this section to write a history of cartography and not to detail the ways of representing the whole earth, rather to give a few salient characteristics for maps made for reasoning about a territory.

The majority of maps are made to describe a territory, for instance in physical, economic and political geographies. Of course, even if they are limited to the description, they can assist any human to make reasoning, but their initial purpose was to explain.

Apparently, the first map was a slab discovered in Saint-Bélec, Brittany from the Bronze Age as presented in Figure 4. It shows a region, presumably in 3D.

Several centuries after, there was the well-known Peutinger map. This is the only Roman world map known to have survived antiquity showing the Roman road network. Preserved in a single, medieval copy now housed in the Austrian National Library in Vienna, the map stretches from Britain in the west to India in the east, covering a series of 11 parchment rectangles. The idea behind this map is not to show the shape and locations of cities and rivers, but overall to help find an itinerary from one city to another city. For instance, the Adriatic Sea is very narrow (at the top of Figure 5), whereas Carthage is depicted just below Rome! Knowing where you are and where you want to go, by reading this map, you can get the list of cities to cross. Of course, at this period, there were not signposts to give directions, but the solution was to ask to a person living in this city.

Fig. 4. Saint-Bélec Slab showing a map of the Bronze Age. Presenting things as they are located Source: https://allthatsinteresting.com/saint-belec-slab.

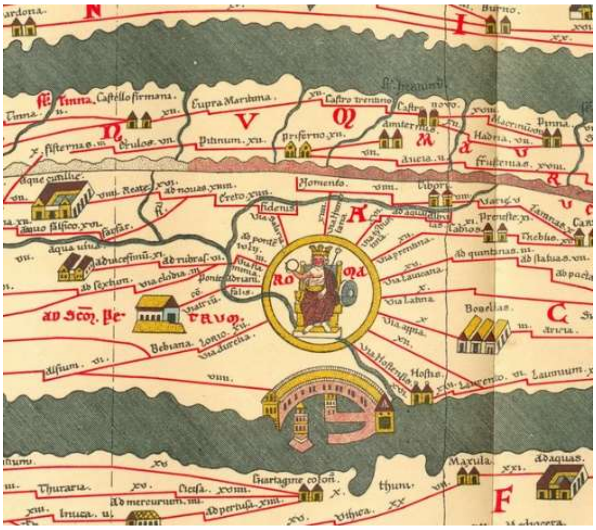


Fig. 5. An excerpt of the so-called Peutinger map centered in Rome, showing the Roman road network. Can be seen as a way of reasoning for finding itineraries. Source: https://www.onb.ac.at/

3. From Visual Data Mining to Geovisualization

In this section, we intend to clarify the differences between several notions such as visual data mining, visual analytics, visual reasoning and geovisualization

3.1 Visual Data Mining

According to Simoff (2020), visual data mining is the process of interaction and analytical reasoning with one or more visual representations of abstract data. At the difference with conventional data mining in which knowledge chunks are automatically collected often as patterns or associative rules, in visual data mining, the human interaction is the key: by observing particularities or regularities, someone can identify interesting issues, maybe leading to novel knowledge chunks.

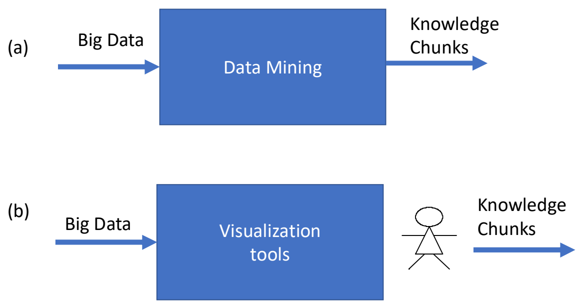


Fig. 6. Two ways of capturing knowledge. (a) via data mining. (b) via visualization tools (visual data mining).

In Figure 6, one can see the main difference between conventional data mining (a) in which knowledge chunks are identified by a computer whereas in visual data mining (b), knowledge chunks are identified by a human.

3.2 Visual Analytics

Wong and Thomas (2004) gave the following definition: visual analytics is an outgrowth of the fields of information visualization and scientific visualization that focuses on analytical reasoning facilitated by interactive visual interfaces. In Figure 7, Keim et al. (2010) depict a visual analytics workflow as a methodology to produce knowledge through visualization.

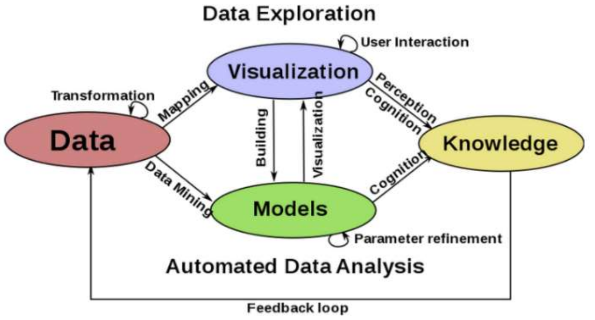


Fig. 7. Visual analytics workflow. From Keim et al. (2010).

According to Thomas-Cook (2005) here are a few recommendations in research about visual analytics:

- build upon theoretical foundations of reasoning, sense-making, cognition, and perception to create visually enabled tools to support collaborative analytic reasoning about complex and dynamic problems;
- create a science of visual representations based on cognitive and perceptual principles that can be deployed through engineered, reusable components. Visual representation principles must address all types of data, address scale and information complexity, enable knowledge discovery through information synthesis, and facilitate analytical reasoning;
- develop a new science of interactions that supports the analytical reasoning process. This interaction science must provide a taxonomy of interaction techniques ranging from the low-level interactions to more complex interaction techniques and must address the challenge to scale across different types of display environments and tasks;
- create methods to synthesize information of different types and from different sources into a unified data representation so that analysts, first responders, and border personnel may focus on the meaning of the data.

3.3 Visual Reasoning

Emily Daw (2022) defines visual reasoning as the process of analyzing visual information and being able to solve problems based upon it. In other words, visual reasoning produces knowledge. But what are the characteristics of this knowledge? For humans, overall knowledge is verbal but multimedia knowledge is also important: think about music, images, videos, gastronomy, domains for which sometimes it is difficult to transmit knowledge with words.

A good example of visual reasoning is a crime board (Figure 8) which can be seen in practical all police series and detective shows: in this board are pinned suspects' photos, crime maps, relationships between them, pieces of evidence, etc. As soon as new information is discovered, this is put onto the board. By looking at it, detectives formulate assumptions. Often the solution comes from a missing relation.

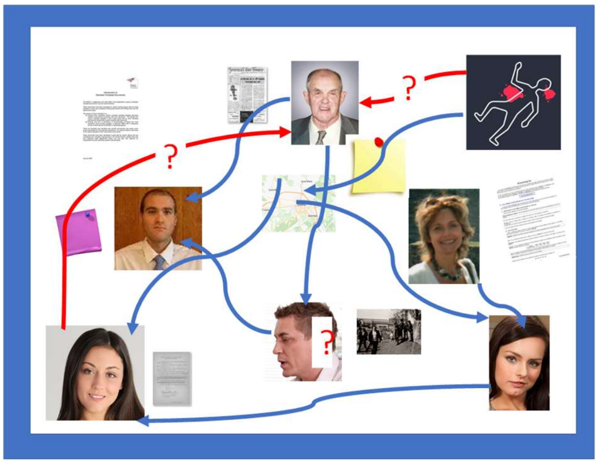


Fig. 8. Example of a detective crime board. By visual reasoning, the solution of the crime is found.

In this paper, we only try to examine how knowledge can derive from figures, drawings, schemata and especially from information mapping, i.e., coming from visual analytics, and geovisualization.

3.4 Definition of Geovisualization

According to MacEachren and Kraak (1997), Geovisualization, short term for "Geographic Visualization" can be defined as a set of tools and techniques to support geospatial data analysis through the use of interactive visualization. Like the related fields of scientific visualization and information visualization, geovisualization emphasizes information transmission. Geovisualization communicates geospatial information in ways that, combined with human understanding, allows data exploration and decision-making processes. Beyond cartography whose goal is representing territory with fidelity (usually physical or topographical), geovisualization tries to highlight the more important issues.

To summarize, geovisualization is an interesting and useful field of research for different reasons:

• it can reduce the time to search information, and support decision-making;
• it can enhance the recognition of patterns, relations, trends and critical points etc.;
• it can give a global vision of a situation, a phenomenon, etc.;
• it enables the use of human visual memory and the capability of perceptual processing of data;
• it permits a better interaction between user and the information system;
• and it can possibly lead to the discovery of new bunches of knowledge.

4. Geovisualization For Reasoning

According to Lacoste (1976), in his provocative book, explained that geography was a form of strategic and political knowledge, central to the military strategy and the exercise of political power. In other words, geography help reason for war. But the role of geography is more than that since it permits reasoning in other domains such as urban, regional and environmental planning. For instance, reasoning is useful for the following issues (Laurini, 2020):

• Where to put a new airport, a new hospital, a new stadium, a new recreational park, etc.?
• Is this new construction project compliant with planning rules?
• What is the best mode or the best way to get from *A* to *B*?
• How to organize a plan for green spaces in a city?
• How to reorganize common transportation?
• How to fight pollution?
• What could be the cost of a projected development?

4.1 Generalities about Geovisualization

Anyhow, perhaps one of the first geovisualization display was made by Minard regarding the march of Napoleon against Russia as depicted in Figure 9 in which the size of the line is proportional to the number of soldiers, yellow when going and black when returning. Due to bad temperatures, he lost more or less 2/3 of his army when marching; during the battle few soldiers died, and in the way back very few soldiers returned home.

Among geovisualization methods, let us rapidly present cartograms, chorems, datascapes and 3D representations. The first two can be seen as 2D representations as extensions of thematic cartography whereas datascapes be considered as 2½ D visualization tool.

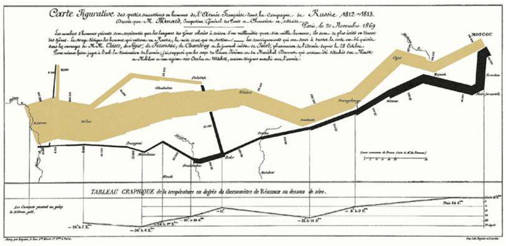


Fig. 9. March of Napoleon against Russia by Minard. Source: https://www.edwardtufte.com/tufte/minard. It is considered as a first geovisual representation.

4.2 Cartograms

According to Grover (2014), cartograms are a kind of maps which take some measurable variable: total population, age of inhabitants, electoral votes, GDP, etc., and then manipulate a place's area to be sized accordingly. The produced cartogram can really look quite different from the maps of cities, states, countries, and the whole world. It all depends on how a cartographer needs or wants to display the information. An example is given Figure 10 showing GDP wealth in 2018; look at Russia, China and Africa for distortions.

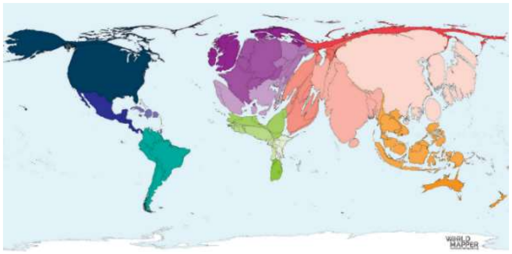


Fig. 10. Example of a cartogram emphasizing Gross Domestic Product wealth in 2018. Source https://worldmapper.org/maps/gdp-2018/

There are various forms of cartograms (Field 2017):

• non-contiguous cartogram: adjacencies are compromised as areas shrink or grow; individual area shapes are preserved but they become detached from the overall map;
• contiguous cartogram: adjacencies are maintained but shape is distorted to accommodate the mapped variable
• graphical cartogram: maintains neither shape, topology or location; instead using non-overlapping geometric shapes (e.g., circles or squares) to represent the mapped variable (see example Figure 11);
• gridded cartogram: uses repeating shapes of the same or different size to create a tessellated representation of the mapped variable;
• topology: non-metric spatial relationships that are preserved under continuous transformation e.g. adjacency.

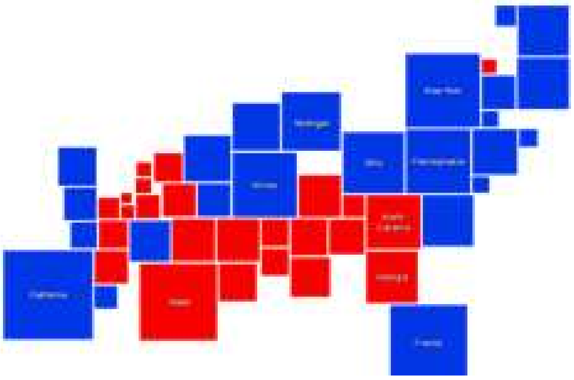


Fig. 11. Example of a cartogram presenting the U.S. Presidential election results as a cartogram based on squares. Source: https://gistbok.ucgis.org/bok-topics/cartograms.

Cartograms, by adapting shapes in accordance a very well-defined variable, permit reasoning which were not possible with traditional.

4.3 Chorems

Chorems were created in 1980 by Pr. Roger Brunet, a French geographer as a schematic representation of a territory. This word comes from the Greek χώρα which means space, territory. It is not a raw simplification of the reality, but rather aims at representing the whole complexity with simple geometric shapes. Even if it looks a simplification, the chorem tries to represent the structure and the evolution of a territory with a rigorous manner. The basis of a chorem is in general a geometric shape in which some other shapes symbolize the past and current mechanisms. Brunet has proposed a table of 28 elementary chorems, each of them representing an elementary spatial configuration, and so allowing them to represent various spatial phenomena at different scales.

According to Brunet, chorems are a tool among other to model the reality, but it is a very precious tool not only as a visual system, but also as a spatial analysis too. Considering Mexico, Figure 12 presents both a traditional (physical) map and a chorematic map in which the salient issues are considered (Lopez et al. 2009).

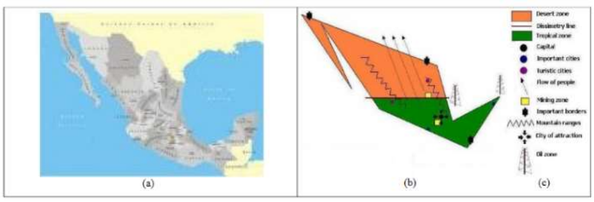


Fig. 12. Fig. 12 Example of chorem; (a) A traditional map, (b) a chorematic map of Mexico and (c) its legend. (Lopez et al. 2009).

In Bouattou et al. (2017) an experimentation of chorem generation in real time is presented with an application in meteorology in Algeria. See Figure 13.

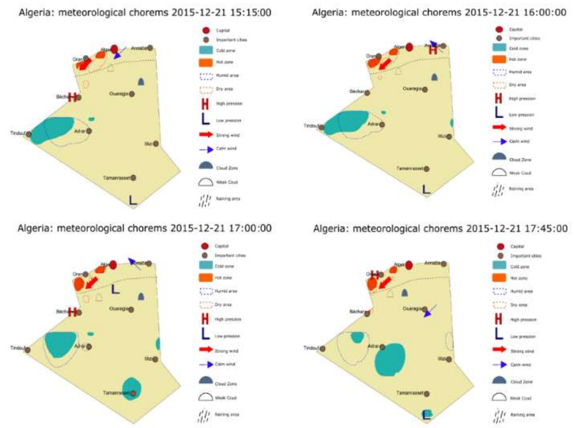


Fig. 13. Example of animated chorem showing weather evolution at different dates in Algeria (Bouattou et al. (2017).

To conclude this rapid presentation about chorems, let us say that they can be seen as a sort of generalization, both geometrically and semantically. Those salient issues can be extracted by data mining and then visualized; thanks to the chorematic presentation, new knowledge bundles can be discovered.

4.4 3D Representations and Datascapes

An innovative data driven graphic approach to model environmental, territorial and urban systems is the representation of natural and anthropic systems and phenomena as "datascapes", literally "data landscapes".

Fig. 14. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; presence of CO_2 in Grenoble, France. Source: https://archinect.com

This approach integrates the methods that describes the elements of the "physical and real" landscape systems, that can traditionally be represented by means of the 2D GIS cartography. And with the datascape representation, it is possible to describe the "visible and invisible" phenomena that spatially represent them using the third dimension, in a "virtual landscape of data" (datascape) in 3D.

Nadia Amoroso, an expert in Data Visualization, defines datascapes as “densityscape maps” and "a visual representation of all the quantifiable forces that influence a system" (Amoroso, 2010) and also as "digital landscape", literally "digital landscapes". See Figures 14, 15 and 16.

The datascape representation is particularly suitable to model different kinds of territorial systems: from simple structural systems that only describe the physical structure of a system, to more complex systems such as urban systems and ecological networks. So, many non-structural, and not visible aspects are added, such as invisible phenomena that are related to the visible structure of the system, and also relations among different kinds of phenomena. In fact, in the datascape approach, it is possible to describe all the different elements and phenomena that are present simultaneously and in the same place that can be quantitatively described, and that require a large amount of data

Fig. 15. Fig. 15. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; Traffic accidents and pedestrian traffic in Toronto. Source: https://archinect.com

This consideration is even clearer when considering the evolution of the concept of environmental and urban systems towards more complex approaches, which, in particular, take into consideration not only the monitoring of natural and territorial phenomena, but also of anthropic phenomena and with them also of new functional and management managerial and new aspects.

This is because we move from a "structural" approach, in which the identification of the areas and the connections between them has a single specific purpose, i.e., for the ecological networks, the protection of biodiversity, to a "multi-purpose" approach that combines different objectives.

Fig. 16. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; Population density in NY. Source: https://archinect.com

In the example of the representation of ecological networks, there is a combination of objectives to reach and to describe: the aim of the biodiversity protection is combined with the aim of implementing eco-systemic services for the population living in the territory; therefore, new elements are taken into consideration and are integrated into the system description: this fact increases the degree of complexity of the system-network and of the phenomena associated with it. Furthermore the "datascape modeling", as we said, is suitable to represent variables of non-visible urban and extra-urban phenomena (temperature trend, pollution and pollen dispersion etc.), and this potential can also greatly increase the complexity of the representation.

4.4.1 Differences between cartograms, chorems and datascapes

The main difference between cartograms, chorems and datascapes is that cartograms and chorems deform the real cartographic representation, instead the datascape representation displays simultaneously the real cartographic representation and the virtual representation of the variables that we need to represent (Donolo-Laurini 2015).

4.4.2 Datascape Modeling

As previously said, the cartographic representations of territorial data can be very complex, since they present different levels of information at the same time and in a multi-dimensional space; one of the main advantages of datascape modeling is given by the potential of representation, for both temporal and spatial dimensions, of the numerous geo-spatial elements and associated phenomena that are present simultaneously in a territory; in particular, it should be noted that:

a) Datascape potential of representation for the "spatial dimension": in a territorial system, in addition to the physical elements visible in the three dimensions of the "physical" space (x, y, z), (e.g. lakes, woods, deserts, etc.), there are simultaneously both visible phenomena (and graphically easily representable (such as the presence of a population of wild animals) and non-visible phenomena more difficult to be represented (such as the presence of atmospheric or aquatic pollutants, or the variation of the average temperature, etc.). Those elements can be visualized and represented using the three dimensions of a "metaphorical or virtual graphic space" (x_1, y_1, z_1) which can be superimposed on the physical space (x, y, z), since it can be translated along the z axis.

b) Datascape potential of representation for the "temporal dimension": in a territorial system, different phenomena occur simultaneously, and the "metaphorical space" (x_1, y_1, z_1) helps visualize different phenomena in the same moment, or one phenomena in different moments, because the "metaphorical space" is replicable for each time considered (x_1, y_1, z_1……… x_n, y_n, z_n) and is superimposable n-times on the physical space (x, y, z), and also because it is a space in which the quantities that vary over time can be represented dynamically in real time.

In other words, we want to highlight that the use of the three-dimensional digital data models allows to view urban and extra-urban systems with a greater visualization and analysis potential than the two-dimensional "flat" models; this is not only due to the insertion of the third dimension (z) to integrate the traditional graphic representation of the GIS in 2D (x, y), which would lead to a representation of the territory in 3D, (x, y, z), that is already used in the digital terrain models (Digital Terrain Model, DTM, Digital Surface Model DSM, Digital Elevation Model DEM), but because a real new "metaphorical space" is added in 3D (x_1, y_1, z_1). This additional graphic representation constitutes a kind of "augmented reality" and in fact produces an alternative and integrative morphology (or more morphologies) with respect to the real morphology of the selected territory, and therefore is also called "meta-morphology".

4.4.3 Advantages of the Datascape modeling

The advantage of having an additional virtual space for displaying data and indicators brings with it other positive aspects:

a) Since the variable z of the "metaphorical space in 3D (x, y, z)" can assume both positive and negative values, it is also possible to use the representation space underlying the surface identified by the plan ($z = 0$), consequently it increases the space that can be used to graphically represent more phenomena in a single "view" of digital cartography; As an example, Figure 17 shows the representation in the urban context of the continuous variable "crime density" / "population density".

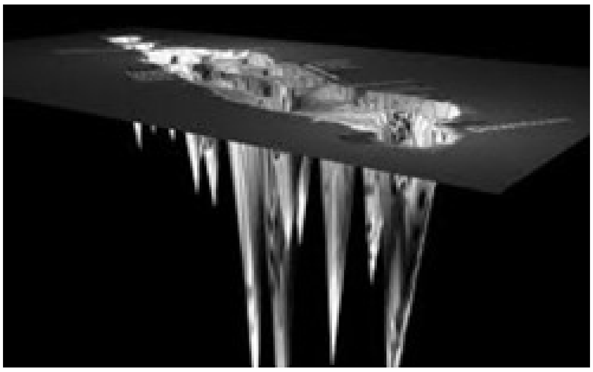


Fig. 17. Criminality datascape (Crimescape) of London, elaboration by N. Amoroso. University of Toronto Magazine, 2013.

b) As already known for the representations of ecological networks in 2D with GIS tools, an advantage of the graphic representation of datascape in 3D is the possibility to customize and optimize the use of graphic variables (color, shape, size, etc.), also called "visual variables", and their combinations. In fact, as the global number of the variables available increases (considering that the variables of the new "metaphorical space" (x_1, y_1, z_1), and the time variable t, are added to the spatial variables (x, y, z) of the territory under examination), also increases the number of visual variables that can be associated to the total spatial variables (x, y, z, x_1, y_1, z_1, and t) and therefore also increases the number of possible combinations between the visual variables and the possible ways of representing the same phenomenon. Visual variables have been defined as "a way to modify graphic signs" (Pumain et al., 1989). In particular, datascape modeling presents an additional visual variable compared to the traditional 2D GIS representation which is possible exactly the third additional dimension: "the perspective". The main visual variables whose most effective use was analyzed to highlight qualitative or quantitative differences between the graphic objects represented by Bertin (1967). It should be noted that other researchers have subsequently discussed and expanded the systematization of Bertin's visual variables, and their optimal use based on the properties of the data to be represented.

c) A third positive aspect, is that the datascapes can be processed online with specific software products that use the interactive 3D map base of Google-Earth on the Web, which can be rotated and zoomed; with it, you can therefore rotate and zoom all the datascape representations superimposed on it: so the datascapes not only are dynamic, but also interactive and they allow you to explore the map from different perspectives, in order to extract out relationships between phenomena, trends, hidden criticalities, etc.

d) Another advantage is given by the fact that the datascape modeling not only allows to manage a large amount of data, but also different data configuration; in fact, through appropriate dashboards, they make possible the management and the modification in real time of both single datascape and multiple datascapes combined and correlated with each other: although a configuration of natural and anthropic phenomena can be temporally stabilized, at times, it is sufficient for a single phenomenon to undergo a modification, to modify whole series of phenomena; so it will not be sufficient to analyze a single reality, but a set of different possible realities. Even the digital environments of datascapes are multiple, fluid and are able to represent and monitor different configurations of phenomena; as an example, different datascapes can be used to represent ecological networks belonging to different animal populations.

e) It is possible to elaborate datascapes not only of the phenomena detected at a given moment, but also of possible future scenarios, which can also be compared with each other. One could also compare datascapes of phenomena belonging to different categories, for example by representing all datascapes deriving from anthropic phenomena under the $z = 0$ plan and all datascapes of natural phenomena above the $z = 0$ plan.

f) By working on datascapes on a large area scale, as the complexity of the ecological network increases with the increasing size of the concerned territorial area, with the datascape modeling, it is possible to work at different scales, which can also be compared.

g) With the datascape modelling it is possible to customize and optimize of the graphic symbology of the single datascape and therefore the management of the visual variables.

h) With the datascape modelling, the identification of hierarchies between datascapes is possible, which can be highlighted.

4.4.4 Critical aspects of the Datascape modeling

In this paragraph, are described some critical aspects to be considered in the representation of territorial systems/networks using the datascape approach.

a) The first aspect to be considered is the tendency to abuse the graphic technological potential and to represent too many territorial and visual variables at the same time in a single "view/display", decreasing the readability of the maps for both expert and non-expert users. There are studies in the psycho-cognitive literature (Sedig and Parsons, 2013) that aim to evaluate how many territorial and visual variables are able to simultaneously process human visual and cognitive capacity. In particular, it emerged that some combinations of territorial variables and visual variables are more effective than others for representing certain quantities or phenomena.

b) A second aspect to note is that, as previously mentioned, several information layers necessary to describe the environmental and anthropic phenomena of a territory. This will be produced not only one, but more datascapes, and therefore the development of the "datascape approach" would be necessary, and will concern the modification, analysis, comparison and display of multiple datascapes at the same time; although this is generally possible through a dashboard, which is used to manage the different datascapes, it remains a critical aspect, because it requires the use by experts at least in the design phase. A dashboard, however, can be made available online, and can allow even non-expert users to interact on the Web, and make changes and views in real time shared between multiple users. Figure 18 shows an example of the interface of the DataAppeal software Dashboard implemented by Nadia Amoroso.

Fig. 18. Example of datascape representation on the Web with the Dashboard for customization (to make graphic and analytical changes, including interactive online and in real time).

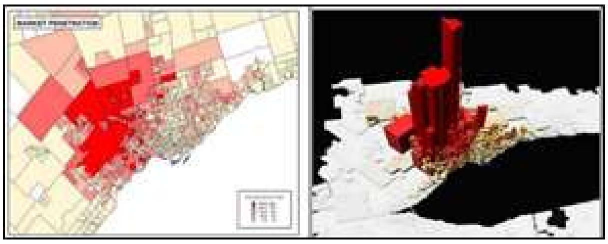


Fig. 19. Representation of the variable “diffusion of a product on the market” with two different approaches.

c) A third critical aspect concerns the poor readability of any graphic symbols / labels: also, in the case of datascapes, as for GIS maps in 2D, the use of graphic symbols superimposed on the cartography, adds information to the representation, but in the case of datascapes the perspective view in 3D could in some cases prevent them from being easily read.

4.4.5 Other Examples of 3D Representations

Figure 19 shows a comparison between the representation of the variable "diffusion of a product on the market": on the left the "traditional" GIS approach in 2D, on the right the "datascape" approach in 3D.

Below are shown other examples of 3D representations, created with opensource software; in Figure 20, is depicted an example of the temperature (or air pollution, etc.) in a canyon.

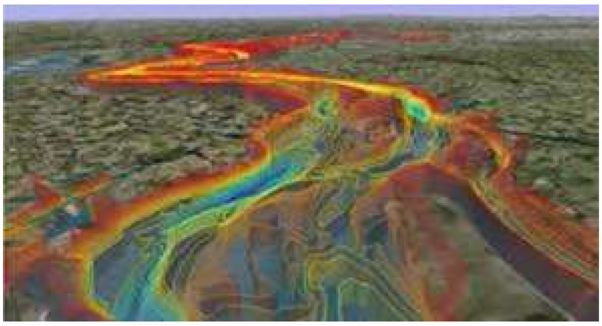


Fig. 20. Example of 3D representations of temperature / air pollutants in a canyon.

In Figure 21, are shown on the left an example of representation of green areas in an urban area, on the right: hen ta representation of flows (of information, passengers, etc.); from a datascape perspective, in this example, all the possible spatial and visual variables can be explored and exploited.

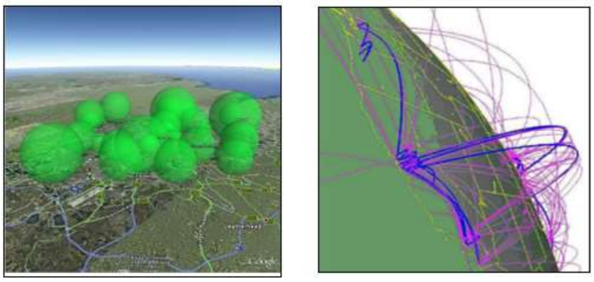


Fig. 21. Examples of datascape representations: on the left, green areas in an urban area and on the right, flow of information / passengers; (Internet source).

5. Towards Visual Reasoning

For territorial intelligence, those geovisualizations can assist visual reasoning in several directions:
• Detection of outliers: by observing a place where data are strange, a quality control action must be launched; otherwise the reasons why those data values are outside the range of other data must be identified;
• Detection of new critical points in geovisualizations, either social or environmental;
• Detection of new patterns, for instance polarization around a CBD (central business district) or along a traffic route;
• Detection of spatial correlations by comparing two datascape of the same place;
• Detection of hidden salient features;
• Confirmation of already-known information or knowledge.

The first step is to identify the main stages for visual reasoning; although it is too early to formalize them, let us list the main key aspects:
• When one is facing an unknown phenomenon exhibited by geovisualization, some questions arise such as its characterization, its magnitude and its origin.
• Then, another set of questions emerges such as how either to fight it (if the trend is negative) or to enhance it (if the trend is judged positive), leading to some action plans.

For instance, detecting a pick of air pollution, its location combined with location of plants can suggest the origin. However, for water pollution, this is more complicated, and the reasoning will differ whether this pollution is along a river or within sewerages. Regarding crimescapes, some comparisons with sociological data can suggest further actions: of course, those actions will differ when the problem is analyzed by the police or by a potential house purchaser.

More, another aspect concerns located data coming from sensors: generally, they can be examined through conventional techniques of spatial data mining, but through datascapes could be challenging; indeed, one can rapidly detect outliers, discover patterns or regularities.

Now, we are facing a kind of paradox. In one part, remember the Mark Monmonnier's book (1991) explaining that one can lie with maps deliberately or unconsciously: indeed, some persons want to mislead people with correct maps. And in the other part, some controlled map distortions can lead to, and bring the truth. In fact, Nadia Amoroso (2010) wrote: "the data should be truthful, but they have to be vivid and exiting. The data can be dramatized in an unrealistic way, which nevertheless is true" …. "This is meant to shock the audience to the risks of continuing to consume for example high levels of energy or water". In the crimescape maps, for instance, "sharp and deep pit-like points are represented that resemble an eerie landscape, in order to draw attention to the districts with high crime".

The original concept of the term datascape redefines information as "possible space" and "possible dramatized scenarios" that can lead to reasoning and in particular influence planning processes.

6. Conclusion

The scope of this paper was to examine how computer-assisted visual reasoning can help territorial intelligence. Starting from the evolution of painting and cartography, we have tried to show how geovisualization can highlight reasoning. Of course, other geovisualization methods may exist, but the general mechanisms are given Figure 22: sensors about a city send data to a geovisualization tool; the results are examined by an expert who can capture some knowledge chunks; those chunks added to already-known chunks are sent to a reasoning tool which infers suggestions; those solutions are studied by local decision-makers to launch actions plans.

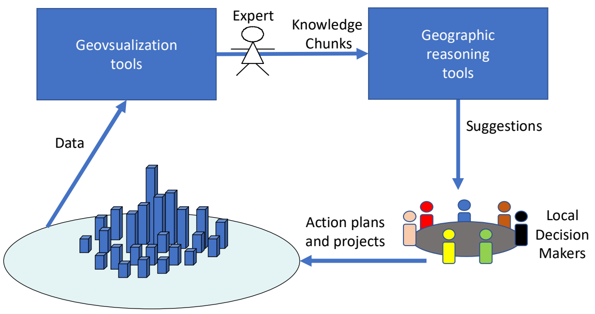


Fig. 22. Structure of visual reasoning for territorial intelligence.

Anyhow, this statement raises three kinds of questions:
1 – Are they existing other methods of visualization more adapted to derive visual knowledge which can be useful to reason?
2 – What are the main characteristics of visual knowledge chunks and bundles issued from geovisualization?
3 – How to formalize them in order to be used efficiently in inference engines?

References

[1] Amoroso N., 2010, The Exposed City, Mapping the Urban Invisibles, Routledge.

[2] Bégouen, C. (1929). The Magic Origin of Prehistoric Art. Antiquity, 3(9), 5-19. doi:10.1017/S0003598X00002933

[3] Bertin J., (1967) Sémiologie graphique, Les diagrammes, les réseaux et les cartes, Mouton, Gauthier-Villars, Paris.

[4] Bouattou Z., Laurini R., Belbachir H. (2017) "Animated Chorem-based Summaries of Geographic Data Streams from Sensors in Real Time,", Journal of Visual Languages & Computing. April 2017.

[5] Brunet R. (1980) "La composition des modèles dans l'analyse spatiale", in L'Espace géographique, no. 4, 1980.

[6] Pumain D., Sanders l., Saint-julien T. (1989) Villes et auto-organisation, Economica. Paris.

[7] Daw E. (2022) What Is Visual Reasoning? https://www.infobloom.com/what-is-visual-reasoning.htm

[8] Donolo R.M., Laurini R., (2015) Evaluation and improvement of smart representations of urban data for developing smart cities, Proceedings 18th AGILE Conference on Geographic Information Science, Geographic Information Science as an enabler of smarter cities and communities", Lisbon.

[9] Field, K. (2017). Cartograms. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2017 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2017.3.8

[10] Grover D. (2014) What is a Cartogram? https://populationeducation.org/what-cartogram/

[11] Keim D.A, Kohlhammer J., Ellis G.P., Mansmann F. (2010) Mastering The Information Age - Solving Problems with Visual Analytics. Eurographics, 2010.

[12] Jiang, B., and Li, Z. 2005. Editorial: Geovisualization: Design, Enhanced Visual Tools and Applications. The Cartographic Journal, 42(1), pp. 3–4.

[13] Lacoste Y. (1976) La Géographie ça sert d'abord à faire la guerre. Paris : La Découverte. ISBN 2-7071-0815-4

[14] Laurini R. (2020) "A primer of knowledge management for smart city governance". Land Use Policy, 2020, 104832, ISSN 0264-8377, https://doi.org/10.1016/j.landusepol.2020.104832.

[15] Laurini R., Nijkamp P., Bordogna G., Kourtit K., Duchateau F., Rinaldi A., Bouzouina L., Mehaffy M.E., Anthony B. (2022) "Regional Knowledge Management and Sustainable Regional Development: In Quest of a Research and Knowledge Agenda". In Laurini R., Bouzouina L., Kourtit K., Nijkamp P. (eds (2022)) "Knowledge Management for Regional Policymaking" Springer Verlag. On press.

[16] Lopez K., Laurini R., Del Fatto V., Sol D., Loreto R., Sebillo M., Vitiello G. (2009) "Visualizing Geographical Analysis Results from Spatial Databases Based on the Chorems". Proceedings of the 2009 International Conference on Modeling, Simulation and Visualization Methods (MSV'09: July 13-16, 2009, USA).

[17] MacEachren, A.M. (2004) Geovisualization for knowledge construction and decision support. IEEE Computer Graphics and Applications, 24(1), pp.13–17.

[18] MacEachren A.M., Kraak M.J., (1997) "Exploratory cartographic visualization: advancing the agenda", Computers & Geosciences, vol. 23, no. 4, pp. 335–343.

[19] Monmonnier M. (1991) How to lie with map? The University Press of Chicago.

[20] Simoff S.J., Böhlen M.H., Mazeika A. (2020) "Visual Data Mining: an Introduction and Overview". In Visual Data Mining pp 1-12. Springer Lecture Notes in Computer Science book series (LNCS, volume 4404)

[21] Sedig, K. and P. Parsons (2013) Interaction Design for Complex Cognitive Activities with Visual Representations: A Pattern-Based Approach, AIS Transactions on Human-Computer Interaction (5) 2, pp. 84-133.

[22] Sinisgalli R. 2011. On Painting: A New Translation and Critical Edition. Original in Latin and Italian "De Pictura" written by Leon Battista Alberti in 1435. Cambridge University Press, 2011. xvi +214 p. ISBN: 978–1–107–00062–9.

[23] Thomas J.J., Cook K.A. (2005). Illuminating the Path R&D Agenda for Visual Analytics, National Visualization and Analytics Center (NVAC), US Dept of Homeland Security, available at: http://nvac.pnl.gov/agenda.stm, 2005.

[24] Wong P.C., J. Thomas J. (2004). "Visual Analytics". in: IEEE Computer Graphics and Applications, Volume 24, Issue 5, Sept.-Oct. 2004 pp. 20–21.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Indicators in *Super Mario Maker 2*: Evolution and Rhetorical Signification in Visual Languages

Nathan W. Eloe[a], Trevor C. Meyer[b]

[a]School of Computer Science and Information Systems, Northwest Missouri State University, USA
[b]Department of Language, Literature, and Writing, Northwest Missouri State University, USA

ARTICLE INFO

Article History:
Submitted 7.31.2022
Revised 9.1.2022
Accepted 9.6.2022

Keywords:
Visual Languages
Video Games
Usability and Accessibility
Linguistics
Metadiscourse
Abusive Game Design

ABSTRACT

As a key element of user experience, communicating expectations to a user through visual elements instead of natural language can produce a more intuitive interface as users interact with a system. While such visual languages are developed by domain experts for specific purposes, these languages can also grow, change, and evolve within a community of users in the same manner as natural language. Video games are one area where communication with the user directly affects the enjoyability, usability, and accessibility of the system. While work has been done to create and leverage visual languages in order to gamify learning or improve accessibility, this research focuses on the creation and evolution of these visual languages by external experts. This exploration of a "crowd–sourced" context–aware visual language through the lens of a structural linguistic framework examines a system of indicators that has evolved over time, created by the very users of the language, to communicate the expectations and necessary actions to complete a task with other members of the community. Extending the analysis and design of visual languages to account for linguistic theory affords designers new tools and approaches in their own work, as too often disciplinary experts can be restricted by conventional understanding, "best practices", and what can be considered a "legitimate" object of study. In this paper, we examine a recent and widely popular visual language of indicators used in the *Super Mario Maker* games, and show how this use of indicators is central to making a usable text, a playable level, creating a relationship between player and designer that foregrounds the human elements in creating a visual language to assist users in a task. Or, in the case of "Troll" levels, prevent the user from doing so.

1. Introduction

Video games have long motivated progress in many aspects of Computer Science; algorithms (e.g. The Fast Inverse Square Root [1], often attributed to John Carmack in the implementation of *Quake*), hardware, and even education and pedagogy [2, 3] are but some areas of Computer Science that have been improved through video games. Games fulfill a particular need in entertainment; they can cater to a diverse audience while providing an interactive, not passive, experience that can be consumed by one or more players, whether separately or together.

Unlike storytelling, art, or video content alone, video games allow the player to interact with a tactile environment while processing auditory and visual information, creating a multisensory experience. This complexity has motivated a wide range of scholarship, as Johansen Quijano and Matthew Wilhelm Kapell explain in the *Introduction to The Composition of Video Games: Narrative, Aesthetics, Rhetoric, & Play*. Moreover, this complexity has also produced "a question that has been left largely unanswered: is it possible to consider video games as ludic, narrative, and rhetorical texts simultaneously?" [4]. That is, in what ways and by what methods should scholars, of any discipline, approach the composed "text," interwoven code and color, pixels and programming languages? Answers to this inquiry help provide the affordances and constraints for this paper's inquiry.

The ludic position, a sort of formalism, holds that video games should primarily be understood as compositions of actions and events, the game mechanics of "buying Baltic Avenue" or "jumping on a Koopa (a certain enemy that be defeated by jumping on its back, making a interactive Shell,

nathane@nwmissouri.edu (N.W. Eloe); tmeyer@nwmissouri.edu (T.C. Meyer)
ORCID(s):

DOI reference number: 10-18293/JVLC2022-N2-020

but it will 'revive' if enough time passes)". A contrasting position holds that game are "indeed, a narrative form and are best understood as such", and narratology, therefore, places games alongside television, film, literature, and comics as content, texts, with a beginning, middle, end, and tension motivating the player's actions [4].

Others, of course, have found nuance between the space of play and story, as both elements generally appear in most games, ranging from the (allegedly) narrative–free *Tetris*, to the more narratively driven, but still loosely, *Mario* franchise, to be explored in this paper. In the middle are games like *The Elder Scrolls*, *Mass Effect*, and *Fallout* series, among many others, which balance choice, consequence, and story in various ways, and at the narrative end is something like *Dragon's Lair*, which is a series of animated cut–scenes with the player's timing and choice determining whether the animated film will continue or the brave hero will meet his brutal demise!

Regardless of the emphasis on narrative and play from others, this paper concerns the third manner and method mentioned by Quijano and Kapell, and that pursued in their own work, studying video games as "rhetorical texts"; that is "not just as a medium, but also as visual and linguistic texts, as rhetorical devices, and as cultural artifacts", that directly communicate with, inform, and persuade readers, users, or players, in any number of (again) complex and situationally–specific ways, what we can understand as "protocol". In his book by the same name, Alexander Galloway defines *Protocol*, as "a set of recommendations and rules that outline specific technical standards" [5]. We can view these material networks not merely as instrument, nor as metaphorical communication, but as materially mediated communication, which requires a common ground for usability, specific goals and means to achieve them, which therefore produces normalizing assumptions.

Video games that present the player with a goal and expect them to reach it provide a challenge for developers to communicate to the player how to complete that posed goal. Often these games are presented as a series of levels (like the *Super Mario* series), though some use a more open world (ranging from gated progression "Metroidvania" games to fully open world games like *Skyrim*). In level–based games, each level must essentially act as an independent implementation of a visual language, providing cues for the player to determine the necessary series of actions to reach goal, as well as maintain the motivation to complete such a goal. In producing "normative" structure, these visual instructions allow the player to navigate all the various situational factors by producing limits and limiting expectations.

Through the example of the *Mario* series, we can see a long–standing and influential approach to visual language, both linguistically and aesthetically, as well as game design and gaming culture, because of *Mario*'s "prominent video game legacy that has been widely circulating since 1985" [6].

Indeed, Nintendo's own "professional game design best practices," promote a " 'user–centered' approach to game and level design" that works to prepare "players by gradually introducing and ramping up challenges", beginning with simple, direct information, with later texts, levels, or specific challenges within levels relying on previous understanding and practice [7]. As detailed below, the visual language of *Mario* guides the player from the first encounter, *Super Mario Brothers* (*SMB1*) level 1–1, and while subsequent releases in the franchise have modified and added elements, the core actions, choices, and consequences remain relatively unchanged. This empathetic, tiered–difficulty mirrors any number of methods from education, pedagogy, and technical communication: begin with basic, constituent parts, and add complexity and additional elements as facility comes with practice.

However, this basic structure of simple–to–complex, coupled with the increased, and still increasing, proliferation of not only hardware and software, but programming understanding and skill, has disrupted the one–way understanding of designers behind an interface projecting a world for players (like the narrative approach featuring an author and a reader, or consumerist approach of creator and customer) to produce a more dialogic relationship, both within and without of the game–world itself. That is, initially, professional designers carefully crafted these levels. The progression of technology and the availability of tools and knowledge has allowed amateur designers to create levels within the confines of a given game's mechanics to challenge their friends, develop unique puzzles, or randomize the elements of a game to allow players a radically different experience in the game every time they play.

Indeed, anyone with the desire to learn the appropriate assembly language can modify games like *Super Mario World* and create fully–fledged "ROM–hacks"; which James Newman explains as "the direct manipulation of commercially released videogame data so as to alter the original gameplay, graphics and sound or level designs", a practice that is "a clear breach of copyright" [7]. These ROM–hacks afford a sufficiently skilled creator an incredible amount of flexibility with mechanics and game assets; almost anything can be accomplished or communicated given the right assembly code. However, Nintendo's Terms of Use explicitly states "any use of the Services or the Materials other than as specifically authorized herein, without the prior written permission of Nintendo, is strictly prohibited and will terminate the license granted herein. Such unauthorized use may also violate applicable laws, including without limitation copyright and trademark laws and applicable communications regulations and statutes" [8].

While only "specifically authorized" use of Mario is allowed, the popularity and deep love for the game franchise has produced a mass of unofficial and "outlaw" content, which have been as influential, as we will see, as the official content to the overall *Mario* phenomenon. For our purposes, one particularly famous ROM–hack, *Kaizo Mario World*, has produced a whole genre of approaches to level design. Kaizo levels require immense knowledge of game mechanics and near perfect execution to complete. Rather than "gated progression" and moving from simple to complex, according to its creator, Takemoto (about whom little is known) in *Kaizo Mario*, “nothing will spare you, the game will spit on you as if you were nothing but asphalt.” [7].

In contrast to “best practices” of design, Kaizo levels feature “abusive game design” that leverage the designer’s knowledge and experience to anticipate and actively thwart players’ attempts to complete the level, often through deceptive, unfair, or subversive designs [9]. While it begins as an “outlaw,” an "unauthorized" design approach, Kaizo was only made more accessible (and sanctioned) with the 30th anniversary of the *Mario* franchise, and the release of a new product in that Intellectual Property (IP), *Super Mario Maker*.

In 2015, Nintendo released *Super Mario Maker* (or *SMM*) for the Wii U, which simplified the creation of new *Mario* levels, followed by the Nintendo Switch’s *Super Mario Maker 2* in 2019. In these games players can either “Play” levels created by Nintendo or others, or “Create” their own, using a predefined palette of sprites, enemies, powerups, and game themes, and share them with other *SMM* players.

To prevent “bad faith” design, all creators must complete their level before they can share with others [7, 6]. This gate–keeping mechanics of “play first” and the limited tools of *SMM* have produced a wide range of entirely divergent game designs, and it did not take long for Kaizo levels to be created and shared as well, incredibly challenging levels with creative setups using the limited palette of tools available in *SMM*.

Even though the intent is to prevent players from completing these levels, or to make it as difficult as possible, there is still a need for these creators (also players of these games) to meet the same challenge professional level designers do for games developed by game studios: how to teach the player what they need to do. Put differently, it’s a challenge of how the designer–players can help players navigate the Flow Channel, or the balance of challenge in the level with the skill of the player; if it’s “too hard” players can get anxiety and quit; if it’s “too easy” players become bored, and also quit [6].

In ROM–hacks, players can insert arbitrary text and assets, allowing near limitless ways to communicate intent and requirements; however, this flexibility is not afforded to level creators in the *SMM* games. Unlike the broader freedom of ROM–hacks, where instructions can be snuck into the background, the sanctioned tools of *SMM* doesn’t provide enough space to convey the necessary information. As others, including Lefebvre, Johnson, and Newman, have noted, this limitation on the available means to create realities within the game levels and communicate to players means that the “far from allowing for endless possibilities, [the creation tool] actually restricts creative possibilities” [6]. But the restrictions are not simply limited by the available tools within *SMM*, but through the additional protocol of Nintendo’s ownership and curation of *SMM* levels in their hosted “Course World,” which prohibits much direct, didactic communication with players.

Instead of the direct communication afforded by ROM–hacks, in *SMM* the player–makers developed a simple, semi context–aware visual language using the simple primitives provided by the game itself “to hint to players what they should do,” such as “when to spin–jump or where to go.” [10]. Therefore, the language of indicators fills in an important role as “metadiscourse,” combining a series of game elements to act as “commentary,” to direct players to accomplish the goals of their levels [11]. This language has no glossary or dictionary, or even perhaps a written description; it instead relies on the skill of the player to understand the actions available to them to determine what should be done next.

Over time the language has even evolved to account for updates to the game that have added new abilities and graphical elements. This evolution does not seem to be communicated explicitly between creators; instead a creator sees and understands the use of an indicator in a level, and then emulates it in their own; either copying the strategy with fidelity or modifying it for their own unique approach. In this way, the *SMM* Indicators language develops similarly to oral tradition; players “hear” a story (see the use of an indicator) and incorporate it into their own levels, as an allusion, homage, or even outright theft! The understanding of these visual lexemes is left to the level “reader”, whose requisite experience and knowledge is vital to reading and udnerstanding the message and successfully completing the challenge of the level if they have the skills and timing to do so.

This language has essentially been crowd–sourced, creating what Ferdinand de Saussure calls “a community of speakers” outside of which the language “itself” does not exist, and mirroring the change and diversification of dialects in natural language as well [12]. For example, the differences between Spanish and Portuguese develop from differing results of “imperfect copy” of the source–language, Latin. Furthermore, Spanish or Portuguese, and indeed any ‘living’ language, “never exists apart from the social fact”, while the source–language, Latin, is considered “dead” as its power as a social fact has been restricted to scientific terminology and Catholic liturgy [12].

The *SMM* community is comprised of individuals with common interest and immense expertise, so the comparison to scientific and theological jargon is quite apt. However, as this language of indicators is shared globally, with multiple millions of users and even more levels, this “specialized” discourse that functions within a community (like Post–Rome Latin) also changes because of the speed and intensity, wide–base of users, who are both players and creators, readers and authors, and (sans compensation), consumers and designers. Indeed, the impact on language change in the wake of global internet technologies is a potent site of much–needed research beyond the scope of this paper.

Even still, the specific language development and the specific dialogic relationship among player–creators of *SMM* provides insight into how a visual language develops, proliferates, and even changes. The intense practice being a player–maker leads to a common understanding, and indeed at times to setups in levels that are so common they no longer require indicators to be understood, but misunderstanding is still possible, and in the case of "troll" levels, indicators are used to actively deceive players and subvert their expectations.

In this paper, we explore this language of indicators in *Super Mario Maker* community, which illustrates and cultivates the "dialogic relationship" between game–makers and game–players who are the same people; as a microcosm of the effects increased access to computer technology, examining how visual language works among a community of player–makers or user–designers, which is likely to become increasingly common and therefore worth exploring ahead of time.

While we draw on the ideas of "abusive game design", "productive play", and the normative constrains in *SMM*, we focus specifically on how these indicators are functioning as language and visual language, as other analyses of *SMM* have mentioned, but not explored this dimension of the game player–designer community. Indeed, much of the work referenced thus far emphasizes the work done to make a level "challenging" or "seemingly impossible," but alongside that work, and as an important supplement to it, we should look more closely at the ways in which designer–players tell player–designers how to overcome the challenge, even if the player themselves might lack the skills to do so. The indicators are instructions on how to open the door, but it's up to players to walk through it. If they can. If they dare.

To accomplish this task, we will provide an introduction to the relevant limitations in the level creator in *Super Mario Maker 2*. Next, we provide a rough survey of the main styles of *SMM* levels, with a more in–depth discussion of Kaizo and Troll levels, in which the use of indicators is quite prominent. After analyzing some individual elements of the language of indicators, this paper will proceed into a few full screen examples demonstrating how the language can be read in the context of a level. Definitions of relevant player abilities will be provided as necessary. Finally, we will present an analysis of some of the lessons visual language developers can take from such a language that has naturally evolved in an environment of mushroom people and sentient dinosaurs.

2. Related Work and Background

2.1. Visual Language and Level Design in Mario Games

In an interview [13], Shigeru Miyamoto discusses the design of the classic Level 1–1 of the original *Super Mario Brothers*. In this interview, he discusses not only how the level was designed last, but also how it trains the player to play the game through a process of rewards and punishments. On the first screen alone, if the player just walks (or doesn't move at all) the Goomba plodding across the screen will cause the player's death. If the player moves, avoids the Goomba, and jumps into the classic question mark block, they are rewarded with a coin. In this way, the player learns that question mark blocks are positive, and that they must jump on certain enemies to defeat them. However deep pits should be avoided, as should being touched by enemies as these will lead to Mario's demise (or at least a loss of a powerup).

The careful training of players over time into the "official" Mario game–level language is a foundational part of the language of indicators that has evolved over time. By putting the player in situations where they can escape danger or be rewarded by experimenting, the design team trains the player to perform certain actions simply by introducing basic elements that will be multiplied and complicated as the player progresses. This design exemplifies the "user–centered" approach mentioned previously as "good design" of game levels.

The following assumptions will be made about the language that *Mario* level designers have trained players to understand since *SMB1*:

- Coins are rewards, and they should be collected (motivating the player to move Mario to collect them)
- Question mark blocks, disguised question mark blocks, and turn blocks can be struck from below by the player (or from the side using a thrown item which is introduced), which often rewards the player with a power–up or some other form of level progression
- Enemies can be safely jumped on unless they have some form of protection (like spikes or fire); later games gave Mario the ability to safely land on these enemies using a spin jump move.

The core mechanics of the game are exposed to the player through low–stakes experimentation in the first few moments of the first level, not explicitly through text. This is truly a marvel of game design principles, and it exemplifies Nintendo's " 'user–centered' approach to game and level design that places considerable emphasis on engineering particular kinds of experiences for the player...gradually introducing and ramping up challenges" [7]. The mechanics and visual elements from *Mario* games are the foundation for level creation in the *Super Mario Maker* games, released in concordance with the 30th Anniversary of the original *Mario*.

In like fashion, when players first open *SMM*, they are given a choice of "Create" or "Play": either delve into the Maker side of the game, or play through the results of others' efforts, whether official content from Nintendo or any of the over 26 million player–created levels (as of 2021) [14]. While many of the levels created explicit reject "good design", Nintendo still uses the informative–design that has been characteristic of *Mario* games since Level 1–1. As LeFebvre explains "when a player plays *Super Mario Maker* for the first time, she starts in a level reminiscent of the first *Super Mario Brothers* (Nintendo 1985)", which plays directly into the established expectations of *Mario* games [6].

Although, at the end of the level, there is "a bottomless pit that is too wide for Mario to traverse", and when the player inevitably fails to make the jump, rather than return to the start a message appears "Whoa! Looks like someone left this course unfinished...It's up to you to complete it", after which the player is provided a basic Maker toolset to complete the level themselves [6]. That is, in a game franchise built from its earliest manifestation of providing in–game tutorials using visual indicators to teach the player, *SMM* provides the same experience in the emphasized element of this two–sided program, "making of *Mario* levels by the player" [6]. They also do so through direct commentary to the player.

However, even though making is emphasized within the first level that can be played, Nintendo also requires all levels to be completed by their creators before they can be shared [6, 7, 10]. In the same way that Create mode is emphasized within Play mode, so too is Play mode emphasized within Create mode. Furthermore, player–makers can switch between modes at will, showing how this game–development game disrupts the common monologic, monodirectional practice of developing a game using special skill and insight for publication and consumption by players who may or may not share some of that skill, but certainly don't rely on it to play the game.

2.2. Limitations in Mario Maker 2

The way *Mario Maker 2* conveys information has two major limitations. First, each screen of the level is a grid that is 24 "blocks" wide by 13 or 14 blocks tall. This space limits what knowledge can be conveyed through simple "pixel art". Additionally, the only free–form text that can be distributed with levels comes from the level description, which has a limited character count. Players can leave small comments in the level, but they may not always be trustworthy or helpful (as we will see with "troll" levels below).

Additionally, because the interface for both play and design are the same, and all items and design tools embody "the aesthetic of Mario's worlds," Newman argues that "*SMM* gamifies game–design", rendering a space to not exactly Make Mario, but rather a space to Play at Making Mario, while the real tools used by the original designers are left to automation or out of the player–designer's control [7]. That is, *SMM* does not provide the real tools and approaches of the Nintendo designers, any more than Mario provides real training in plumbing, but rather a sand–box and toy–box producing something slightly different.

Furthermore, Nintendo is a corporation interested in creating profits, for which games are products, the results of budgets and timelines. Indeed, as LeFebvre argues "the *Mario* franchise does not depend as much on the development of rich narratives within a complex fictional universe as it does on the diversification of cultural products and experiences for a commercial purpose." [6]. Certainly, a princess being in another castle and the increased amount and intensity of antagonists do not attempt "world–building," and Mario's own complete lack of character development over almost 40 years push it away from being a strongly narratological game; the proliferation of Mario spinoff games, like *Mario Kart*, *Mario Paint*, and *Super Mario Maker (1&2)*, further support LeFebvre's position, in addition to the merchandising that have placed Mario alongside any number of other immediately–recognizable cultural figures of late 20th and early 21st century culture writ large. In addition to being something a user plays, Mario is also something a customer buys.

As a product, the "good design" of progressing simple–to–complex, easy–to–difficult, managing the challenge to offset both anxiety and boredom, is easy to understand. If players (users/customers) find something accessible and usable, they will buy it, and keep buying. But as a product, an Intellectual Property (IP) still owned by Nintendo, the "freedom" of *SMM* is constrained by the limits imposed by the company itself. While anyone can make and upload levels to "Course World" to be played by others, these creations exist only within the legal bounds outlined by Nintendo.

Levels can be deleted for "Inappropriate Content," as defined in the Nintendo Network Code of Conduct, which is vital for a "family–friendly" franchise like *Mario* [7]. Additionally, and interestingly for us, but not something to fully explore here, levels can be deleted for including "bugs... because letting these levels remain in Course World [the shared collection of *SMM* levels] can lead to negative outcomes for many players such as players experiencing levels in unfair ways that the original course creator did not intend, or re–writing 'World Record' times" [7].

Considering the essential "unfair" nature of "abusive game design" in general and the "the creative utilization or even abuse of object/item behaviours and interactions," in Kaizo design in particular, makes this process "especially problematic" because it isn't always clear "what constitutes a 'bug' at the point of level creation and uploading. That the decision to designate as a bug what might have previously been apprehended and utilized as a creatively exploitable behaviour is, demonstrably, one taken by Nintendo and leaves levels created and uploaded in the liminal zone before the recoding (and potential removal/recalibration of the 'offending' behaviour) vulnerable to deletion" [7].

Even still, Kaizo level designs remain the most popular and pervasive genre of *SMM* levels, which exemplifies the creative work of player–designers, and these player–designers develop their own aesthetic style and followings themselves, which can lead to other connections and opportunities [7].

Furthermore, "after a fixed period of time, courses with low stars/plays" are deleted automatically, making successful completion, shown as a percentage, and subjective satisfaction of the players, indicated by stars, as well as motivation to share or play the level itself, a vital part of levels' survival in the massive Course World [7]. However, levels are also deleted for "requesting stars from other users," which makes the direct communication with users not impossible, but risky. Therefore, to maintain the existence of levels, even (and especially) complex and difficult ones, designers have a strong motivation to not only challenge players, but to provide insight into how to complete these challenges.

Even with these limited affordances, and sanctioned tools that differ from the original designers', players have made ample use of available resources, and the flexibility surrounding them in the *SMM* toolbox, to communicate directly with players to help them complete challenging levels.

2.3. Kinds of *Mario Maker* Levels

Creators can create any level they can imagine if it is completable and complies with the terms and conditions Nintendo places on the levels it hosts. This wide range of levels also occupy different places within the ludic–narrative spectrum of video games. This range has produced several distinct types, or genres, of level design, each with a different design approach, effect, aesthetic, and particular audience. Some approaches are standard levels that would be appropriate for inclusion in a commercial Mario game. These levels can be understood as the most literal and direct result of a game called *Super Mario Maker*.

Other levels are "AutoMario", in which the player does nothing or simply moves forward, the level having been meticulously constructed to reward the player's inaction with a cinematic, dramatic sequence of Mario braving hazards and only–just–escaping certain doom before arriving safely at the end. Still others emphasize a narratological approach within a level, complete with dialogue, while others feature challenging puzzles, a ludic approach, that work within the constraints of *SMM* and the game's longstanding influence.

However, the most popular style of level, as discussed previously, is Kaizo, which uses the limited toolset within *SMM* to emulate the difficult and complex designs of the ROM–hack original. First appearing in a video in 2007, *Kaizo Mario World*, or "*Jisaku no Kaizō Mario (Sūpā Mario Wārudo) o Yūjin ni Purei Saseru*" which translates as "Making my friend play through my Mario hack (Super Mario World)", features the extreme opposite of Nintendo's game design best practices. Rather than Nintendo's advice, "you want the player to feel like they're about to get crushed," producing enough challenge to motivate play without quitting, in *Kaizo* "the player not only feels like they are about to get crushed, they get crushed. Many times over." [7]. Drawing from the ROM–hack approach, Kaizo levels push the limits of the mechanics afforded to the creators and the abilities of a player to jump, aim, and time everything just right. These levels often require precise tricks and near perfect execution to complete, and the payoff in these levels is the satisfaction that the player has obtained a level of skill well beyond that of a causal player. Indeed, the prevalence of Kaizo levels, as well as play–through videos on Youtube and Twitch, has become the most visible examples of *SMM*, and bringing back those hardcore gamer fans of the franchise who might otherwise pursue ROM–hacks in search of a challenge.

Additionally, this purposefully–nigh–impossible design in Kaizo is accompanied by a "markedly more dialogic relationship between player and designer," which Newman cites as characteristic of "abusive game design" coined by Wilson and Sicart [9]. While so–called "good design" causes the designer's presence to "recede into the background" becoming "anonymous" (in the same way "good" language use is "clear", emphasizing the erasure of the words themselves), abusive game design in Kaizo creates a "clear exchange existing between designer and player," with the level being "the site of their contest" as "one between two (or more) human participants [rather] that one between a player and a system" [7]. That is, a Kaizo level is more about the player and the designer than the level itself, foregrounding the communicative medium of the video game as one between people; the game is an instrument to build relationships.

Furthermore, as levels are shared across the Course World, they are "assiduously attributed to their nicknames" and named by creators, and "once one alights upon a designer whose levels one enjoys, they can be followed, ensuring no new upload is missed and cementing the relationship yet further in much the manner of social media" [7]. After all, the original *Kaizo* video is making a friend play a challenging game.

However, not all purposefully–nigh–impossible designers aim to reward skill and timing alone, but some levels require in–depth thinking and suspicious play. In Kaizo levels, the challenges are extreme and arduous, but essentially honest; in contrast, Troll levels are extreme, arduous, and frustrating, as they are they are built on deception. A relationship is built here too, but through a practical joke. Troll Levels are similar to Kaizo levels, but they are designed to subvert the expectations of the player, often leading to an excessive number of character deaths. Often, troll levels will evoke schadenfreude in those watching the player, and they are entertaining in the creative ways the player can be killed more so than in the successful completion of the level.

These levels rely on the learned expectations of expert players, second– and third–level thinking, to encourage them to "troll themselves," as described by Johnson. Drawing from studies of poker, Johnson explains first level thinking as "simply looking at the level as presented, while second level thinking means considering that the designer was thinking, and third level means considering what the design thought the player would be thinking at a given movement and what decision they (the player) might be poised to make" [10]. These levels will often provide the player with one or more "obvious" paths to take, then punish the player when clever mechanisms cause a new path to appear after the player has committed to the wrong choice.

These levels can be often removed because of the presence of glitches or unintended mechanics, but they have a cult following where people will upload a level long enough for it to be downloaded by others before it is removed from Nintendo's servers. Streamers on services like Twitch will hold contests to see who can build creative trolls [15]. Their popularity likely stems from the comedy that comes from creatively subverting expectations and the streamer's reactions to the creative and unexpected deaths. The use of indicators adds to the psychological duress on the player as there is never the certainty that an indicator is lying, telling the truth, or even partially lying.

3. A Language of Indicators

Building on the previous assumptions (and referencing them), this paper will focus on three extensions to the visual language: the P–Block, a C–shaped track, and a curved track. These indicators communicate modifications to position, action, and movement, respectively. All images below were captured from the author's console using the screenshot feature and edited as necessary. Interestingly, it may be almost impossible to track down the first use of these indicators; as discussed above, levels sometimes get removed for a variety of reasons and as such the genesis of some of these indicators and any textual description they might have had may be lost.

While these visual lexemes are only a selection of the various tools available and in use, they provide the most telling examples of the "visual metadiscourse" of indicators at work in *SMM*, which cultivate the "dialogic relationship" between player and designer central to Kaizo design. Drawing on Vande Kopple and others, Eric Kumpf explains that "metadiscourse helps writers arrange content by providing cues and indicators that both help readers proceed through and influence their reception of texts...To omit this metadiscourse would blur the separation of content, making the text less cohesive and less considerate of readers" [11].

While discourse is the communication, exploration, and presentation of the ideas, arguments, and impressions of a text, metadiscourse is writing that helps discourse accomplish its goals. That is, metadiscourse is writing explicitly designed to help the reader read and understand a text, and as Kumpf explains, this is not limited to textual commentary (like "in this paper will first, then, next, later...,") but also visual elements, like abiding by convention, maintaining consistent design, and dividing content into accessible chunks, "may help readers understand and interpret the whole document." [11]. In like fashion, level designers (text writers) use metadiscursive indicators to help level players (text readers) to read the level and understand how to complete it.

Furthermore, like all semiotic units, these indicators can be understood through the structural linguistic frame of Ferdinand de Saussure. While de Saussure's initial discussion is primarily focused on words in phonetic languages, it also applies to the elements of visual languages. Rather than the indicator, or token, or *Sign* being a singular entity, de Saussure identifies two interconnected, yet distinct, elements: the *signifier* and the *signified* [12]. The *signifier* is the word the reader sees or the sound the listener hears. In the context of visual languages, the graphical elements constitute the signifier, that which is seen by the user. The *signified* is the meaning that the listener (or reader in the case of written and visual languages) attributes to the signifier, the idea intended to be thought about, the concept being communicated, or the action intended in response.

The *SMM* Indicators are then Signs, combinations of a graphic element drawn from a collection of possible elements with a corresponding idea or action for the player to actualize in response, and the language a Language, "a system of distinct signs corresponding to distinct ideas." [12]. This distinctness is central to coherence and communication: "M" is a distinct shape for a distinct sound, because it's different from all other shapes/sounds, and by adding other distinct shapes and sounds, a fuller distinct concept can be communicated, "Mario" (with further distinctions needed to specify which Mario is intended, whether real chef or fictional plumber). However, the connection between the signifier and signified is not natural, but the result of conventional use and rhetorical repetition.

As a First Principle of semiotics, the study of signs (of which linguistics is one, yet paradigmatic, example), de Saussure argues that Signs, combinations of Signified and Signifier, are arbitrary. That is, a signified idea is "not linked by any inner relationship to succession of sounds...which serve as its signifier," and "it could be represented equally by just any other sequence [, which] is proved by differences among language and by the very existence of different languages" [12]. As a Second Principle, emphasizing phonetic, spoken language as the primary "object" of study, is that the signifier, the sound sequence, is linear, which is "obvious," "too simple," and therefore assumed, but yet "fundamental, and its consequences incalculable." [12]. Indeed, as a corollary to Principle II, de Saussure contrasts the "auditory signifiers [that] have at their command only the dimension of time," with "visual signifiers...which can offer simultaneous groupings in several dimensions" [12]. From these two principles, de Saussure developed a massively influential theory that shaped inquiry in the humanities and social science quite broadly, not limited to language study.

It is interesting to draw comparisons between de Saussure's structural linguistic frame and the established frameworks for visual languages, such as those by Green [16], Moody [17], and others, because the structural approach not only precedes any development of (and analysis of) programming languages, but it would also therefore inform the understanding of "language" as developed within programming, visual language included. The nature of this work also relates to the field of Computational Semiotics, but is outside the scope of this paper. For more on this field, see [18].

Chang et al. discuss the use of formal language theory in relation to visual languages, but call for an interdisciplinary approach to extend the field [19]. The formalisms specified by formal visual language theory apply when working to intentionally design a visual language, but there are aspects that fall apart when dealing with a language changing in real time. De Saussure explicity addresses these different approaches; there is a formalistic, synchronic study of language, drawing on the method of comparative philologists, whose work begat proper linguistics for de Saussure, and in contrast, a diachronic study of language examines how language use changes in time [12]. While limited in his time by technology, de Saussure's emphasis on the importance of structure, rather than use itself as in speech, is understandable; in the *SMM* Indicator language, we can see, though only in part, the development, proliferation, and change of a distinct, visual language with a massive community of speakers.

This is not to say that the standing theories of visual languages are at odds with structural linguistics, though we may need to interpret elements in both fields with some flexibility. For example, Green's Cognitive Dimensions [16] address the closeness of mapping of a notation to the problem world, which would at first seem at odds with de Saussure's idea that the signifier is arbitrary. In a visual language, however, closely mapping to the problem space means the signifier is chosen specifically to communicate the signified. As a "metadiscourse" the closeness of a signifier to the signified helps assist the user by relying on already existing connections outside of the text, and because the goal is not to vocalize the signifier, we are not limited to phonetic signifiers.

Furthermore, de Saussure includes "modes of expression based on completely natural signs [closely mapped], such as pantomime" among the science of signs, because "arbitrary" does not mean totally random, but only that a specific signifier "actually has no natural connection with the signified" [12]. Indeed, "every means of expression used in society is based, in principle, on collective behavior or–what amounts to the same thing–on convention" [12]. Calling a herbivorous quadruped with mane and hooves a "horse" isn't more true or accurate than calling it a "*caballo*" in Spanish, or the Portuguese "*cavalo*" or "*égua*" (depending on gender), or the Latin *equus*; put differently, a horse isn't any more "horsey" than it is equine(although Latinate roots in English are used widely within specialized, scholarly discourses to afford so–called precision), but the effect and effectiveness of those Signs will depend on the context, convention, and situation.

Moody's Principle of Semiotic Clarity [17] particularly falls directly in line with de Saussure's concept of a sign being the union of the signifier and signified and the immutability of the signifier within a given community of speakers. Moody does specify that there should be a one–to–one relationship between signifier and signified in visual notations, and while de Saussure is more relaxed in the definition, due to the messiness of natural languages changing over time, he is explicit that distinction, one shape to one sound to one idea, within a community of speakers is constitutive of the language itself. Indeed, a sign "is never wholly arbitrary [totally random]; it is not empty, for there is the rudiment of a natural bond between the signifier and signified. The symbol of justice, a pair of scares, could not be replaced just by any other symbol, such as a chariot" [12].

In this sense this principle of the Physics of Notations is not a departure from the principles of linguistics, but a subset, and while Sassure grapples with the complexity of language being conventional, but not random, but not natural either, the Physics of Notation must work toward more clarity. Put differently, programming languages work to create an ideal language with much less ambiguity and misunderstanding than natural language; code works or it doesn't.

However, as we've seen with ROM–hacks, Kaizo level–design, and trolling, and as de Saussure states that convention is not total control, which "even holds true for artificial languages. Whoever creates a language controls it only so long as it is not in circulation; from the moment when it fulfills its mission and becomes the property of everyone, control is lost" [12]. That is, language, a series of connections of signifier and signified conventionally constructed and used by a community of speakers, changes and persists, and once out into that community, and beyond, the creativity of those speakers moves it well beyond the control of the originators.

Unless, of course, they violate Terms of Service, IP, or Copyright, laws and rules that help originators, like Nintendo, maintain control. However, the challenge of communication is embodied in fictional voices, like Yamamura, a pigeon that inexplicable understands the nuances of level design in *SMM*, even if the effect is an erasure of the designer, fading back into the system itself, which is the main way *SMM* communicates with player–users.

But whether they're ROM–hackers illegally ripping and remixing source code, *SMM* designer–players creating complex level–texts out of a limited set of items, Kaizo–designers eschewing 'good design' in favor of 'abuse' with indicators to help understanding, or Trolls using the same indicators to lie and deceive, and 'abuse' their players, even with legal weight and force, there is movement, change, and evolution.

However, it's important to note that difficulty and deception are not necessarily correlated to the popularity and fun of "abusive" levels; Kaizo and Troll levels use indicators differently, and the quality of their use shows the quality of the level. That is, both Kaizo and Troll levels are manifestations of "abusive game design" and examples of "masocore" gaming, "the portmanteau...combining 'masochism' and 'hardcore'", which emphasizes repeated attempts and failures on "punishingly difficult" challenges [10, 7]. The "fun" of these designs might be called "suffering" by more casual gamers, and coupled with the powerful schadenfreude provided by streaming play–throughs, "abusive" design is a powerful force. However, in a set of game designs where difficulty and suffering is the point, there are still distinctions between "good" and "bad" designs, and the use of indicators is a key factor in those distinctions.

Kaizo levels follow the tradition of the first *Kaizo Mario* game, which was intended to both challenge and frustrate the creator's friend. These levels have varying use of indicators to show the intended actions, but they still rely on the player's skill to complete the level. The challenge in these levels is not only in determining how to reach the goal, but also the execution of the level itself.

A "Good" Kaizo levels "presents a player with a unique set of precise gameplay challenges which ideally look compelling or beautiful to perform," while a "Bad" Kaizo level is marked by "its lack of creativity, its pure repetition, its almost complete lack of aesthetic detail, and its absence of pleasing sequences," which Johnson sees as evidence for "new *surprises* being indicative of good abusive design" [10]. Creativity, beauty, nuance, and craft, as well as being playable with hours of practice, or not, are all central to Kaizo design, and indicators form a central part of that; Kaizo indicators show the player the door, even if they might not be able to see how to walk through it. It's the joy of a challenge overcome.

Similarly, Troll levels will often (but not always) lie with indicators, following the tradition of subverting a player's expectations. Johnson explains that "a good troll level is one which toys with the player's expectations, gets into the player's head, encourages the player to get into the designer's head, and perhaps most importantly...provokes entertainment (falling into a trap) just as much as it provokes frustration (at falling into a trap)" [10]. Some levels even work to subvert the expectations of expert players, providing an obvious path to success within an otherwise 'normal' Troll level, leading an expert *SMM* to "troll themselves."

In contrast to the nuanced, psychological approach of "good" Troll designers, a bad Troll level "is one where your required choices are arbitrary (such as just choosing a door at random), or the level is packed with enemies to the point where nothing except purely chance–based trial and error will get you through."[10]. Creativity, beauty, nuance, and craft, as well as being playable with hours of practice, or not, are all central to Troll design as well, and indicators form a central part of that, even if they are not to be trusted as in Kaizo.

That is, the "thin line" between good and bad levels, for both Kaizo and Troll levels, seems to be their use of Indicators, reflective and purposeful construction of metadiscourse (even if it isn't always 'honest', but communication between humans is rarely as full, honest, and complete as we might imagine or hope). This metadiscourse and the "dialogic" relationship that it creates is foundational in "abusive" game design, in which the player experiences the designer's presence and persona, through both aesthetic and direct communication with indicators.

These indicators are motivated by Nintendo's rewards for plays and stars, and the relationship between designers and player, as well as the prohibition of explicitly soliciting stars. In balancing anxiety and boredom, player–designers in Super Mario Maker use indicators as metadiscourse in their levels to make them "considerate texts", to be read and understood by their readers, even, and especially, when consideration for the player is couched within abusive game design. It's not a matter of failure, in Kaizo or Troll, but that the player understands how and why they fail, and they have fun doing it.

What follows is a subset of the language of indicators. These are specific examples that help inform the "reading" of the level presented in Section 4 or are particularly interesting in how they signify meaning.

3.1. The P–Block: "Aim Here"

In *Super Mario Maker 1* and earlier updates for *Super Mario Maker 2*, creators were limited to using the coin to indicate to players where they should attempt to send their character. Due to the limited palette, the coin was a multi–purpose reward indicator, promising progression if the player performs the correct action at that location. Level creators were afforded a new option with a later update to the game: the P–Block (Figure 1), which occupies a single grid space on the screen.

Figure 1: The P–Block

Mechanically, when this block is in the state shown in Figure 1 it behaves like a background object the character can pass through. If the character hits a P–Switch, the block becomes a solid object, which can serve as a platform or wall. The background state of these blocks makes it useful as an indicator, but particularly clever creators can use it as an indicator in the background state and as a platform or wall in its activated state. The P–Block is especially useful when the item or enemy the player will be landing on is not yet apparent but will be by the time Mario reaches that location, assuming all preconditions have been met.

This indicator then acts as a general "aim here" message to the player, and is used in many of the situations the coin was in earlier releases of the game. This change allows the coin to be used as a different kind of indicator, which illustrates de Saussure's point that "everything that changes in the [language] system is internal" [12]. While the new signifier P–Block takes up some of the signified action of the Coin, this change in aligning a new signifier with part of the signified of another signifier is an internal change: a change in the connection of a signifier to a signified. The Coin no longer means what it did, signifies the same signified, as it did before the introduction of the P–Block. Furthermore, and interestingly, this indicator's meaning can change based not only on how it is used in a level, but also within the style used to create the level.

Visually, the negative space in this indicator in its inactive state somewhat resembles a cross–hair or target, which evokes the real world idea of aiming for a particular location (an example of Green's Closeness of Mapping and de Saussure's emphasis on convention). While the color cannot be changed by the level creator, extra meaning can be conveyed through placement or number of indicators (which is related to Green's Secondary Notation and Escape from Formalism). For example placing one P–Block directly above another often indicates that the player should land in that location twice. An additional example is provided in the partial level read in Section 4.

This indicator can be confusing if there are multiple reachable from the player's current location, making Premature Commitment particularly a concern at times, though often after one or two mistakes it becomes more clear as to which location should be the player's next target. As discussed previously, Indicators "communicating effectively" depend on the audience, the purpose, and the context of its use, and players need the skills to read and interpret what the designer had in mind.

3.2. Shaped Tracks: "Throw it this way!"

In all level styles except *SMB1*, the player has the ability to pick up, carry, and throw objects. These sprites can interact with the world, triggering switches, breaking blocks, or just providing an object the player can land on later. Kaizo levels often require objects to be thrown precisely to allow the player to proceed. As quite flexible game–objects, tracks can be placed in relatively arbitrary shapes, but must either be a closed loop or a simple path. Often, a closed square loop in a 3x3 grid is used to indicate to the player that they should wait at a location (or something will appear there). Alternately, Tracks can be placed in the shape of a "Z" to indicate that the player should press the Z button on the controller (this is a fairly explicit use of indicators to communicate intent to the player, as part of the "dialogic" relationship). While arrows pointing in various directions can be placed in levels, creators often rely on a ⊏, ⊐, ⊓, or ⊔ shaped set of tracks (Figure 2).

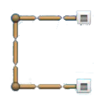


Figure 2: Tracks shaped to indicate the direction to throw a held item

When one side of the closed square loop is omitted, it tells the player to throw or drop the item they are currently holding in the direction of the missing edge when they are inside the square. Usually if the player releases or throws the correct held object in the direction indicated by the opening while Mario is in the outlined box the item will end up exactly where it needs to be for the next required steps in the level to barely work, only just so.

This indicator is interesting with respect to the Cognitive Dimensions of Notation. The size of the symbol is larger than the corresponding arrow (3x3 instead of roughly 1x3), but provides additional context of the player's location when the object should be thrown, as a metadiscourse commentary [11]. Choosing to trade terseness for additional information may depend on the available space. Arrows are also frequently used to indicate a direction the player should go if it's not immediately clear; this may be a reason why the shaped track was chosen.

Additionally, this particular indicator has a hidden dependency: the player must be holding an object for it to have meaning, and if there are multiple objects available they need to be holding the correct item. If a player reaches this indicator without an object, it may be a hint that a previous part of the level has been misread or the player did not execute the previous parts of the level as intended (and instructed).

Also, there is nothing obvious about this particular indicator that indicates throwing. The closest it comes is indicating a position and a direction. Closeness of Mapping may be this indicator's weakest Cognitive Dimension, though there is not a particular symbol available that would necessarily directly communicate the idea of "throw" to the player. Rather, this signifier has connected to the signified through conventional use. However, in concert the mechanic of holding and moving items, the conventional assignment of "tracks" with "movement" and the repeated, conventional use of this indicator to communicate the intent takes up the communicative work than a more Closely Mapped indicator would perform.

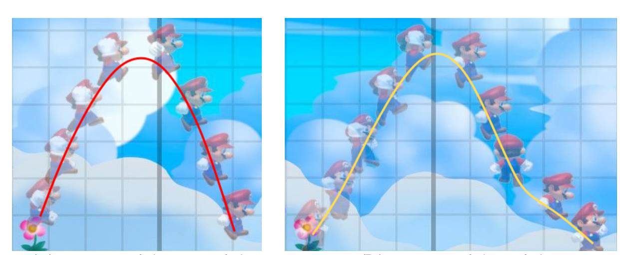
One Cognitive Dimension this notation rates particularly well at is Consistency, a name shared by one of Kumpf's visual metadiscourse devices [11]. The orientation of the notation (which side is open) indicates the direction the player should send the held object. Once a player has learned that the ⊏ shaped track means "throw the held object right", the convention at play, the player can then infer that ⊔ shape means throw the object up (in game styles that support that), a ⊓ means drop the object, and a ⊐ shaped track indicates that the object should be thrown backwards. Once we learn about a distinct sign, modifications to it, which combine conventional understanding of the significance of the sign itself with other visual language elements, like direction, we can extrapolate one sign into many, each with a distinct signifying form and a signified action/concept.

3.3. Curved Track: "Twirl Jump Here"

Super Mario Maker 2 allows an additional trick with Tracks: a Track placed diagonally can curve instead of going in a straight line (as in Figure 3). Unlike the track configuration in Figure 2 that encodes its meaning in the empty space contained within the track and its missing side, the path this indicator expresses its meaning in its shape.

Figure 3: Tracks curved to indicate when to twirl

In the *New Super Mario Bros. U.* and *Super Mario 3D World* level styles the player can perform a maneuver called an air twirl. This technique is performed during a normal jump. Players make Mario spin once in midair which briefly stalls the character's downward momentum. This effectively extends the reach of the jump slightly as in Figure 4.



(a) Jump without twirl **(b)** Jump with twirl

Figure 4: A comparison of Mario's jumps with and without an air twirl

Note the curve in the path at the end of the jump in Figure 4b.Mario's path is almost exactly the shape of the curved Track, an excellent example of Closeness of Mapping. Treating the grid as a standard Euclidean graph, Mario begins the air twirl at $(6, 2)$ which stalls his downward momentum as he travels to $(7, 1)$; those are exactly the grid spaces the track would occupy; beginning an air twirl at the top square of the track will cause Mario to follow the path indicated by

Figure 5: A partial level with indicators. Footage of a player reading and completing this level can be found at [20]

the track very closely. Here the "natural sign" of movement in space is illustrated by the shape of the indicator, but the conventional understanding of what moves are available in what styles of *Mario* is also necessary to understand the message. A player who understands the mechanics afforded by the level style can read this indicator and map that to the necessary controls to make the character follow the path shown by the notation. Like other Indicators, within and without Mario, it is arbitrary, but not empty.

4. Reading the Level: An Example

Figure 5 contains an annotated representation of the first few screens of the level Mechanical Manacles created by Donkeymint (Level Code 8QH–JRX–GLG). This image was created by splicing live game play images captured directly on the Nintendo Switch, and it is annotated to explain how a player who knows the mechanics of both the game and level style can complete the level with enough skill or practice. The difficulty of playing these levels, as well as access to Nintendo products, can make examining these levels directly rather challenging itself. Luckily, the massive popularity of *SMM* and the proliferation of level–play videos provide ample secondary sources to examine.

The player begins on the left side of the screen. The actions that must be taken do have a strong ordering, and it is not immediately obvious how to get from the starting location to the first safe platform (the coins at indicator I3). Because the timing of the actions is also critical, skilled level creators will sometimes add "reset doors" as seen at the beginning of the level to give players time to analyze the situation or retry a particularly difficult trick, which helps avoid premature commitment. The indicators themselves do not necessarily have a way to indicate order of actions, often the only way to determine what action to take next is to look at what the only reachable indicator is. While the indicators themselves do not necessarily follow de Saussure's Principle II of linearity, their sequence does, as the level itself exists in time for the player.

While there is no immediately clear path forward, the indicator I1 tells the player that they should land there. Because this level is in the style of *Super Mario World* there are two choices of how to get there: a regular jump or a spin jump; one of these jumps will lead to success while the other will cause the player to fall into the Piranha Plant (the bottom enemy in the stack) or the saw blades below where they will die. Since there are only two options and this is the first jump in the level, failure is not too punishing; however a player who understands the mechanics of the game can glean more information from the indicator.

I1 is placed to the left of the top enemy, called a Mechakoopa, and if the player was simply using it as a way to bounce higher, then the player would land directly on top of the enemy. However, here we see another convention at work: not every enemy reacts to jumps in the same way; the player needs either learn through experience or research how enemies interact with different player actions. The placement of I1 indicates some directionality in the desired end result. With a regular jump, the Mechakoopa would collapse and fall straight down (or be knocked slightly to the side), but a spin jump will send this particular enemy in a direction opposite to the side they were hit from (if they were hit from the left, they would move right, and vice versa) [21].

By using a spin jump on the Mechakoopa at I1, the enemy will be stunned and sent to the right, landing on or near the indicator at I2 (depending on the exact timing of the jump). Because the character continues spinning, they can land on the stunned Mechakoopa and safely make it to the coins at I3. At this point, the next indicator (I4) tells the player that they should throw something to the right, but currently has nothing in hand to throw. Depending on the exact timing of these first two jumps, one of two things will happen. If the player landed again on the left side of the stunned Mechakoopa it will be knocked to the right where it will land on the note block and bounced up to the top of the icicles (directly next to the player). But if the player doesn't knock the enemy to the right, it will eventually recover and begin walking around, turn around at the one–way gate to the left of I2, and then walk to the note block and bounce up to the player waiting at I3. At this point the Mechakoopa can be picked up (or re–stunned and then picked up), giving the player something to throw.

After jumping and throwing the enemy to the right at I4, there is only one safe place to land: the checkered platform below the indicator. However, this platform will start to fall quickly, so the player will need to quickly decide what to do. The next indicator (I5) tells the player to jump there, but at this point there is nothing to land on. However, when the Mechakoopa is thrown from I4, it lands on top of the note block to its right, which spawns a flying enemy directly at I5. The player can then bounce off of this enemy and land on the checkered platform that is in the middle of indicator I6.

Once again, the player is told to throw an object from a location but isn't holding anything. The mechanism above the note block above I5 causes the Mechakoopa (which was thrown at I4) to bounce up and then over, landing on the checkered platform. This enemy should then be thrown at I6; there are P–Block indicators behind the tall yellow enemy (a Pokey) that can be seen as the enemy moves. The Mechakoopa destroys the Pokey and falls to land on the spike at the bottom of the screen. A spin jump will allow the player to jump off of the stunned Mechakoopa and land on the right most checkered platform, which then begins to move up and down.

At this point the creator forgoes the use of custom indicators and relies on the visual language established over time by the developers of the *Super Mario Brothers* series. After landing on the right most platform (which begins to move), the player has exactly one choice left: hit the turn block B1. This triggers the mechanism above the turn block, which sends another stunned Mechakoopa to land on the right side of the checkered platform. At this point the player again has exactly one choice that does not result in death: pick up the stunned Mechakoopa and throw it up into the On/Off blocks which are hidden by the timer in the upper right corner of Figure 5. Hitting these blocks will cause blue blocks to become solid and red blocks to become background objects, allowing the player to progress further in the level.

An important consideration for this level and the spliced image is illustrated in Figure 5. The images were captured by a player who has not devoted hours of practice to develop the skill that professionals will hone over time. In fact, the amount of skill required to beat the level is independent of the knowledge and ability to translate the language; despite being a mediocre (at best) player, one author was able to progress this far in the level through repetition and practice; the intended action was always clear, even if the muscle memory and skill was not developed enough to allow successful execution of every required move. The other author, to the best of his recollection, has never successfully completed a single *Mario* level alone.

5. Discussion and Conclusion

Language designers can take multiple lessons, or reminders perhaps, from investigating what we've shown here. Visual languages are often used to lower the barrier of entry for individuals completing a new task, as metadiscourse to make a text more considerate of its users, as Kumpf puts it. But experts create mechanisms for communicating succinctly that require a learning curve. For example, experts can use shorthand, a glyph–based method of shortening written communication, to increase the speed with which information can be recorded. Quickly decoding shorthand requires practice and knowledge of which system encoded the information. While useful, such methods are not explicitly designed for simplicity or guiding novice users through completing a task.

This language of indicators serves much the same purpose as shorthand: meaning must be conveyed using limited resources in a constrained space. A visual language is used because unrestricted words and lettering are mostly unavailable to level creators. By removing the expectation that the language be immediately accessible to novice players and their knowledge of the game mechanics, the language can be concise and still convey varied meanings with a single indicator based on the context in which it is used.

This is not to say that accessibility is not important; in fact visual languages should continue to be an important part of aiding in accessible design goals, regardless of whether we're working on digital interfaces, analog written documents, or any of the wide range of texts and mediums available to contemporary designers and users. The role of visual languages in wider accessibility is beyond the scope of this paper.

Even still this language of indicators enables communication within an audience that shares common knowledge and skills, and in the case of *SMM* to reward that knowledge and skill. Understanding the target audience is important when devising any form of communication, as the conventional understandings of one community of speakers are not going to be the same as another. Domain specific visual languages should be designed with a specific audience in mind, and the intended audience needing certain domain specific knowledge to comprehend the visual language should not be a barrier to the design of the language.

The language of indicators is a prime example of "designing" a language with a specific target audience in mind. How often is the user experience and functionality of a deployed system designed and developed seemingly without consideration for the intended users? Updates resulting in removal of used features or strange UX choices often result in members of the user base reaching out to each other and the developers of the system as in [22]. To quote one reply from that particular thread: "Any time we need to tell instructors 'you have to redesign your course / assessment to fit into the software constraints' you know there is a poor software design." Any visual language must be designed and implemented with the target audience in mind, and while there must be common understanding for any communication to happen, we cannot assume that the audience will be of the "same mind" as the author. Indeed, the kinds of "level thinking" employed in Troll design are essential here; it's not enough to see what a text looks like to us, as makers, but also what we think it will look like to users, readers, and players.

However the most intriguing aspect of this language is that it was designed by players for players. Unlike the Hmong Script or the Sequoyah Syllabary, with a single creator endeavoring to preserve a language from annihilation, many people have built the language over the life cycle of two different *Mario Maker* games. This language has evolved as updates have changed elements used to indicate meaning, external changes in the language that lead to internal changes.

But tracing the origins of specific indicators, like "throw" are impossible because the online services for the original game have been shut down, and levels can (and could) be deleted by both Nintendo and the creator. There was no meeting where a group sat down and decided how that configuration of lines should mean "throw the thing".

Perhaps the originator of the indicator put a note in the level description as to what it meant; over time though that indicator has become widespread. A player would play a level, determine the meaning of an indicator, and then use it in their own creations. This interaction is also intensified by the presence and popularity to Kaizo and Troll design, which explicitly foreground the persona of the designer and the player; as Wilson and Sicart's title indicates: "it's personal." If that creator's understanding of the indicator was slightly different, they might use it in a different way. As such, the meaning and usage of this language has evolved in much the same way as natural language, although its change intensified by the strong presence of the original "source language" of Mario, as well as the global "Course World", and popularity of game streaming services like Twitch.

Language and communication evolve, signs grow and change, signifiers and signifieds shift and move with use, misuse, and abuse, imitation and bad copies. The role of visual languages in enabling accessible and intuitive experiences for all users has been increasing, and it will continue to evolve with language as a whole. As visual language designers it is important to keep the intended audience firmly in mind when building an experience and a mechanism for guiding users through completing a task.

However, the intended audience might be a collection of skilled experts who've developed ways to communicate succinctly with each other. This language of indicators serves as a fun reminder couched in a video game that intention can be communicated using limited tools to leverage the audience's knowledge and skill. It can serve as a reminder to UI and UX designers that perhaps they should look at how members of the target audience are already conveying information to each other and common knowledge within the intended users. That may give the language designer a starting point to provide an experience that is familiar, more intuitive, and more comfortable to the user while leveraging their experience and knowledge to enhance how information is communicated. This can also ease the burden on the designer, as much of the design work may already have been completed as the users allowed their communication to change, allowing more time for implementation, testing, and refinement.

Visual language in *SMM* also serves to re–emphasize the human and linguistic dimensions of digital interfaces; regardless of the bits, bytes, and bots at work, the practice of language, visual or written, digital or analog, is about connecting with other people. That is, far from the negative connotations it has "abusive" game design requires the same kind of empathy for the player as the "'user-centered' approach...[that] places considerable emphasis on engineering particular kinds of challenges for the player" [7]. The difference between abusive game design and "good" design is a difference in audience, rather than a quality inherent in the games themselves. After all, even challenging, brutal, and dishonest levels, like Trolls, can and are enjoyed widely because they satisfy the expectations and needs of their user; whether Kaizo, Troll, or Regular, a level is "bad" because it doesn't meet those expectations, different as they might be for different audiences.

References

[1] J. Blinn, Floating-point tricks, IEEE Computer Graphics and Applications 17 (1997) 80–84.

[2] P. D. Bitonto, T. Roselli, V. Rossano, E. Frezza, E. Piccinno, An educational game to learn type 1 diabetes management, in: Proceedings of the 18th International Conference on Distributed Multimedia Systems, DMS 2012, August 9-11, 2012, Eden Roc Renaissance, Miami Beach, FL, USA, Knowledge Systems Institute, 2012, pp. 139–143.

[3] E. J. Marchiori, Ángel del Blanco, J. Torrente, I. Martinez-Ortiz, B. Fernández-Manjón, A visual language for the creation of narrative educational games, Journal of Visual Languages and Computing 22 (2011) 443–452.

[4] J. Quijano, The Composition of Video Games: Narrative, Aesthetics, Rhetoric and Play, McFarland, 2019.

[5] A. R. Galloway, Protocol, Theory, Culture & Society 23 (2006) 317–320.

[6] I. Lefebvre, Creating with (un) limited possibilities: Normative interfaces and discourses in super mario maker, Loading... The Journal of the Canadian Game Studies Association 10 (2017) 196–213.

[7] J. Newman, Kaizo mario maker: rom hacking, abusive game design and nintendos super mario maker, Convergence 24 (2018) 339–356.

[8] Nintendo, Nintendo Terms of Use – Official Site, https://www.nintendo.com/terms-of-use/, 2016. Accessed: 2022-28-07.

[9] D. Wilson, M. Sicart, Now it's personal: on abusive game design, in: Proceedings of the International Academic Conference on the Future of Game Design and Technology, 2010, pp. 40–47.

[10] M. R. Johnson, Playful work and laborious play in super mario maker, Digital Culture & Society 5 (2019) 103–120.

[11] E. P. Kumpf, Visual metadiscourse: Designing the considerate text, Technical communication quarterly 9 (2000) 401–424.

[12] F. De Saussure, Course in general linguistics, Columbia University Press, 2011.

[13] Eurogamer, Miyamoto on World 1-1: How Nintendo made Mario's most iconic level, 2015. https://www.youtube.com/watch?v=zRGRJRUWafY.

[14] S. Baird, Super Mario Maker 2 Players Have Uploaded More Than 26 Million Levels, https://screenrant.com/super-mario-maker-2-upload-26-million-levels/, 2021. Accessed: 2022-28-07.

[15] FANDOM, Carl's Troll Contests | Mario Maker Trolling Wiki | Fandom, https://mario-maker-trolling.fandom.com/wiki/Carl%27s_Troll_Contests, 2020. Accessed: 2022-27-07.

[16] T. R. Green, Cognitive dimensions of notations, People and computers V (1989) 443–460.

[17] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (2009) 756–779.

[18] C. Shackell, Finite Semiotics: A New Theory of Semiotics With Applications To Information Technology, Ph.D. thesis, Queensland University of Technology, 2019.

[19] S. Chang, M. Barnett, S. Levialdi, K. Marriott, J. J. Pfeiffer, S. L. Tanimoto, The future of visual languages, in: Proceedings 1999 IEEE Symposium on Visual Languages, IEEE, 1999, pp. 58–61.

[20] GrandPOOBear, RACE this arrow from MAX MAP HEIGHT [SUPER MARIO MAKER 2], https://www.youtube.com/watch?v=5wQlFpLYVsg, 2021. Accessed: 2022-03-02.

[21] FANDOM, Mechakoopa | Super Mario Maker Wiki | Fandom, https://supermariomaker2.fandom.com/wiki/Mechakoopa, 2020. Accessed:

2022-03-02.
[22] jlubkinchavez, Solved: New quizzes survey workaround and complete/incomplete, https://community.canvaslms.com/t5/New-Quizzes-Users/New-Quizzes-survey-workaround-and-complete-incomplete/td-p/194757, 2020. Accessed: 2022-03-09.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Implementing BERT With Ktrain Library For Sentiment Analysis

Hasnan Othman[a,*], Mohd Ridzwan Yaakub[b]

[a] Malaysian Administrative Modernisation and Management Planning Unit (MAMPU),Prime Minister Department,Putrajaya, 62502 Malaysia
b.Centre for Artificial Intelligence Technology, Universiti kebangsaan Malaysia, Bangi 43600, Malaysia

ARTICLE INFO

Article History:
Submitted 10.1.2022
Revised 10.2.2022
Second Revision 10.5.2022
Accepted 10.6.2022

Keywords:
Bi-directional Representation
Bi-directional Encoder
BERT
ktrain library
Sentiment Analysis
Transformers

ABSTRACT

Building the Sentiment Analysis (SA) model for classification tasks is challenging to boost computational efficiency while writing Machine Learning (ML) applications with the Python Library package. Its implementation with the pre-trained Bi-directional Encoder From Transformers (BERT) will produce better results for the experiment. However, one of the implementation issues is choosing library resources to develop the SA model in solving the Natural Language Processing (NLP) tasks. Implementing BERT with the Ktrain framework can make such jobs more accessible by allowing domain experts to further democratize ML with minimum code. The Ktrain Library package's experiment in implementing the BERT model on IMDB movie datasets obtained an accuracy of 92.8 percent compared to based line models using Support Vector Machine (SVM) and Logistic Regression (LR). Future works will apply the Ktrain library package with other ML models on diversified domains of datasets.

1. Introduction

Sentiment analysis(SA) on the text is the form of indirect assessment based on information gathered in a text which can be classified as positive, negative, or neutral. Three approaches to conducting SA on text are machine learning, lexicon-based, and hybrid. The machine learning strategy uses machine learning algorithms to classify text, whereas the lexicon-based approach requires the construction of manual lexicon dictionaries, which is more time-consuming [1].

The difficulties in building a model for SA inherent in mastering machine learning coding workflows might make it challenging for beginner programmers to determine appropriate library resources to be used. Machine learning workflows were composed of several components, such as building the model, inspection, and application, to ensure that the models become operational [2]. Python's ability to be used as a programming language is significantly more efficient than other programming languages in performing matrix operations and SA tasks [3]. Ktrain library lets you use many pre-trained deep learning architectures in Natural Language Processing (NLP), such as BERT. Choosing the model coherently with NLP SA tasks can be improved by getting an efficient programming language and supporting resources library. Numerous studies have been conducted on sentiment analysis over the years, and most studies accurately classify data as positive, neutral, or negative polarity. Despite this, various studies have shown the diverse applications of SA research in security, tourism, and business intelligence. The approach can expand the need to use SA as part of the business strategy to increase revenues [4].

Ktrain is a Python machine learning package library that provides a simple, unified interface for performing the ML programming workflow in building the model regardless of the input type (i.e., text vs. images vs. graphs). It is a lightweight TensorFlow Keras wrapper package that can be useful in developing deep learning and ML projects. The wrapper is used to create, train and deploy deep learning and ML models. There are three phases of developing the project: building a model, inspecting, and applying it that can be completed in as few as three or four lines of code, a technique known as "low-code" ML [2]. The ktrain library package currently supports several data types and tasks: text, vision, graph, and tabular data.

Ktrain also aims to further democratize machine learning by allowing novices and domain experts with minimal programming or data science knowledge to construct advanced machine learning models with

DOI reference number: 10-18293/JVLC2022-N2-024

minimal coding. It is also handy for seasoned practitioners who need to develop deep learning solutions rapidly[2].

A deep learning transformer solution like BERT is the first fine-tuning-based representation model that outperforms numerous task-specific designs on many sentence and token-level challenges. As a state-of-art model, BERT was designed to pre-train deep bidirectional representation from an unlabeled text by conditioning each layer on the left and right context. BERT can be fine-tuned to various task questions and language inference in NLP without significant task-specific architecture modifications. BERT is advanced in eleven NLP tasks and distinctive features unified are architectured across different tasks [5].

The hyperparameters setting using the ktrain library used a simple lines code to describe what batch size to be used and what learning rate parameter to specify. The ktrain-BERT has a pre-training length of 75 words, a full feature set to consider, and a batch size of words to embed. It features a three and n-gram range for word token sequencing. Model loading also establishes column labels so the model can determine if the job is binary or multi-label categorization[6]. Fig 1. shows the codes line to set the training model with one cycle learning rate policy. The fit method will be applied with the cycle_len parameter. The hyperparameters setting using the Ktrain Library used a simple line that described what batch size to use and what learning rate parameter to specify. Figure 1 shows the line of code to set the training model with one cycle learning rate policy. The function will process the learning rate schedule and maximal learning rate reduction.

```
hist = learner.fit_onecycle(1e-5, 2)
```

Figure 1: Method to set one cycle policy rate schedules

Text classification is a common problem in SA. It is distinct from text mining and employs text classification to identify document subjects. SA classifies material depending on the writer's mood on various positive or negative themes [4]. For determining the sentiment of a text, the machine learning approach uses algorithms such as Naive Bayes, S Vector Machine, Decision Tree, Logical Regression, etc. In contrast, the lexicon-based approach relies on sentiment lexicons (i.e., a dictionary of opinion words and phrases with assigned polarities and intensities) [7].

2. Research Objectives

The primary goal of this research is to enhance the model development process using programming languages such as Python by choosing an adequate library package to increase the model's performance. Doing this can make the process more efficient with less code and improve performance on processing data to meet challenged NLP tasks in SA. The scope of work for this research is to fulfill the following objectives :

(i) Enhance the capability of a model building using the library packages to accommodate the need for NLP tasks, particularly on SA domain datasets.

(ii) Better managed the language program tools using transformer BERT correlated with the python package using ktrain library.

The works of literature related to the ktrain Library, BERT pre-trained Model, ML model workflow, and buildings of the SA model in these subsections are as follow:

2.1 BERT Model

The BERT language model has two approaches: feature-based, unsupervised, and fine-tuning. Feature-based unsupervised techniques, in general, are like word or word embedding [8]–[10]. The process to train the word representation based on previous work was to place the possibilities of the following sentence [9], the generation of left-to-right emphasis words of the last sentence [8], and the derivative of automatic denoising encoding [11]. The unsupervised approach based on fine-tuning, on the other hand, works on word embedding parameters trained from the unlabeled text. Sentence or document encoders that produce contextual token representations have been trained from unlabeled text and refined for supervised downstream tasks [12]–[14]. Figure 2 shows the pre-training and fine-tuning for the BERT model in which the output architecture is the same. In pre-training, the BERT model will use the masked sentence to train the model on how to predict the surrounding words[5]. As for fine-tuning, the model can be configured to apply a feature selection algorithm using hybrid approaches to building the sentiment analysis model.

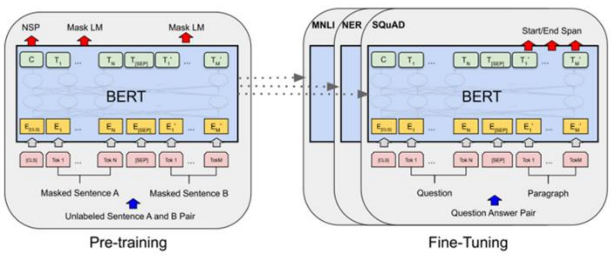


Figure 2: BERT model Pre-training and Fine-Tuning [5]

The machine learning model development consists of 3 phases: model development, inspection, and development [15]. Recent improvements in deep learning-based recommender systems have piqued researchers' interest by overcoming traditional models' limitations and reaching excellent suggestion quality. Deep learning can capture non-linear and non-trivial user-item connections and codify more sophisticated abstractions as data representations in higher layers. Furthermore, it captures the subtle relationships within the data from readily available sources like contextual, textual, and visual information. The BERT model can semantically pick up a relevant field associated with documents classified in scientific publications[16].

2.2 Ktrain Library

Ktrain is a low code library for Python that can make accessibility and application to ML tasks easier. It is straightforward for beginners and expert practitioners to build, train, inspect, and apply sophisticated, state-of-the-art machine learning models. Ktrain is also designed to make deep learning, such as BERT and Artificial Intelligent (AI), more accessible and easier for newcomers and experienced practitioners. It also includes modules that deal with textual data. Ktrain is used in the machine learning model implemented in TensorFlow Keras (TF. Keras) and supports the following data types and tasks, as illustrated in Fig. 3[2]. Four categories for the ktrain library are text, vision, graph, and tabular data. This category will relate the ktrain library with a classifying model for performing the relevant task. Tasks for text data include task classification, such as auto categorizing documents, text regression, such as predicting numerical values, and sequence tagging, which extracts sequences of words representing Named Entity Recognition. The tasks on vision data include image classification, such as auto-categorizing images across various dimensions, and image regression to predict numerical values such as a person's age from photos. The graph data types category is a node classification that auto-categorizes nodes in a graph, such as social media accounts, and predicts missing links in a social network. Lastly, tabular data includes classification and regression that store data in tables.

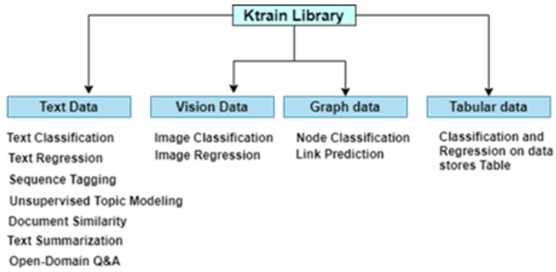


Figure 3: Ktrain data types and task [2]

Additionally, ktrain offers a straightforward and user-friendly prediction API for making predictions on fresh and unstudied samples. The model (i.e., the underlying tf. Keras model) and the preprocessing procedures (i.e., a Preprocessor instance) necessary to convert raw data into the format anticipated by the model are both included in a Predictor instance. It is simple to save and reload the Predictor instance to deploy production environments [2].

Ktrain simplifies the SA model workflow, from input curation and preprocessing (i.e., ground-truth-labeled training data) to model training, tuning, troubleshooting, and application. Libraries that are a core of Ktrain's explainable AI are powered by other libraries, such as Shapley Additive Explanations (SHAP) (Kiros et al. 2015) and ELI5 with LIME [9]. The prediction using both libraries significantly impacted the result by trusting a prediction or testing the model. We characterize "explaining a prediction" as displaying textual or visual artifacts that provide a qualitative comprehension of the relationship between the instance's components (e.g., words in a text, patches in a picture) and the model's prediction [9]. The SHAP library will assign each feature an essential value for a particular prediction. Hence, the local Interpretable Model-agnostic Explanations (LIME) technique interpreted individual model predictions based on a local approximation model around a given prediction [15].

The equation prediction using SHAP by additive feature attribution methods is :

$$g(z') = \emptyset_0 + \sum_{i=1}^{M} \emptyset_i \ z_i' \qquad (1)$$

Where $z' \in \{0, 1\}^M$, M is the number of simplified input features, and $\emptyset_i \in \mathbb{R}$.

This method matches the attribute with the effect of $\emptyset_i$ to each feature and sums up all the attribute features to the output *f(x)* of the original model.

As for using LIME, the equation prediction interprets the model predictions individually based on approximating the model for the given prediction. It is to simplify input by mapping the value of $x = h_x(x')$ to convert it into a binary vector of interpretable input to the original input space.

$$\xi = \arg\min L(f, g, \pi_{x'}) + \Omega(g) \quad (2)$$

The explanation of the argument is the minimum ξ =arg min, and it will enforce the loss of L over a set of samples in simplified input space weighted by the local kernel $\pi_{x'}$.Ω penalizes the complexity of g. The equation is solved using penalized linear regression [15].

Figure 4 is a sentiment analysis pipeline by applying SHAP in IMDB positives datasets for two rows of data. It shows how to score positive based on the reviewer's dataset comment.

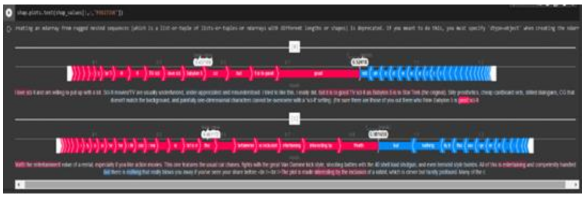


Figure 4: SHAP explained in an IMDB positives review on two sets of data

2.3 SA Model

Acquiring SA models and modifying them according to the task requirements is time-consuming and error-prone. It is because process knowledge is typically disseminated across many people, and workflow modeling is an arduous task that requires the expertise of a modeling specialist [17]. There are three fundamental problems in developing the SA method: feature selection, senti-word identification, and sentiment classification. This requirement is needed to prepare the developing model to meet SA requirements and focus on the given task. The data validated the sentiment model's integrity by developing new SA frameworks and components under SA-related issues [18]. Building models using the ktrain standard for supervised tasks described are as described in Figure 5 [2], [17].

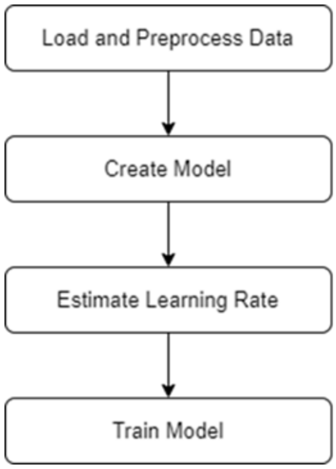


Figure 5: Phases in building model [2]

Phases step based on Figure 5 that will be applied using ktrain library packages :

- Load and preprocess Data comprises a step for loading data from various sources and preparing it according to the model's specifications. The process for language-specific preparation will include lemmatization and tokenization for text classification. Image data may involve auto-normalizing pixel values based on the model, attributes of nodes, and the link to the networks. All ktrain preprocessing techniques return a Preprocessor instance that incorporates all the preprocessing stages for a particular task. This instance can be used when utilizing the model to make predictions on new, unknown data.

- Create a Model using tf.Keras for model development and be wrapped in a ktrain. The model will be configured based on configuring the Learner instance wrapped in ktrain. Learner instance as an abstraction to facilitate training. The customization also can be made by the users whether to customize it or use defaults.

- Estimate Learning Rate can be employed to estimate the optimal learning rate for the given model and data. This step is optional for some models, such as BERT, with default learning rates that function well.

- The training model will be using the fit-one-cycle method and autofit method. Fit-one-cycle is the method that employs a one-cycle policy, whereas autofit utilizes a triangular learning rate schedule by specifying the number of epochs. After training a model, ktrain provides a simple interface to evaluate the technique and calculates extensive validation (or test) metrics.

3. Methodology

The proposed methodology using the ktrain Library and BERT model are as per Figure 6. The building frameworks for this methodology will be conducted in various steps starting from initiating the library package for ktrain library resources, processing the data, and splitting data for training and evaluation models. We will use ktrain to easily and quickly build, train, inspect, and evaluate the model. The visualization of the experience will be assessed based on a comparison will baseline models and to produce the result based on the finding.

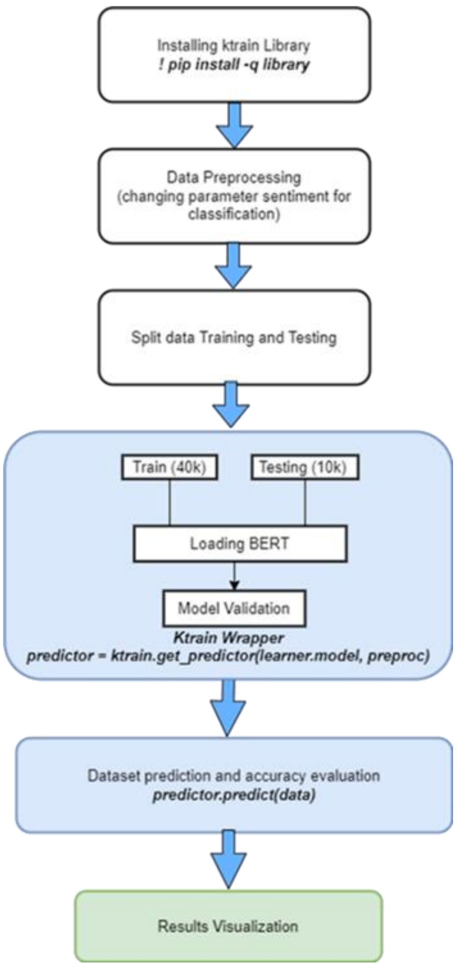


Figure 6: Proposed Methodology Using the Ktrain Library and BERT Model

Firstly, the building of the model BERT will be using the ktrain library by implementing based on hyperparameter tuning on the model structure. Firstly the pre-trained BERT will determine the column labels so that the model can know whether the task is binary or multi-label classification. The transformers class using the ktrain library will be instantiated by providing the model name, the length sequence, and the target names' populating classes. The training metrics were fit to 2 epochs using the one-cycle policy with a max LR of 1e-05. Total trainable parameters are 109,336 322.

Next is the preprocessing data in the Keras model, where the preprocessor instances are loaded with the encapsulated feature set, pre-training length, and model range. The preprocessing steps include tokenization, normalization, and stop word removal [19].

After that, we create a classifier with pre-trained weights, and fine-tuneable final layers initialized randomly. The model will be wrapped in a ktrain Learner object, which will make it simple to train, analyze, and use the model to generate predictions on new data. Dataset will be split into training and testing datasets in a ratio of 40,000 data to trained and 10,000 for testing. We will use the Learning Rate Finder in ktrain to estimate a good learning rate for our model and dataset. Then the model will perform sentiment classification by using the learner.validate() method to produce testing data results (learner.validate(val_data=(x_test, y_test)).

The following method is to inspect the model by examining the classifying datasets. This process includes loading the BERT model using the ktrain wrapper by implementing the function that predicts the classifying sentiment by declaring operating predictor=ktrain.get_predictor(learner. model, preproc).

Lastly, the prediction of the dataset and evaluation for accuracy will be using the predictor.predict(data) function. The result will be visualization by evaluating the accuracy, precision, recall, and F1 score. The visualization will illustrate the comparison with other baseline models, such as SVM and LR.

4. Results and Discussion

The experiment was conducted on the Google Colab platform with GPU using an IMDB movie review [20] with total data of 50,000 lines data. Dataset was split into 40,000 for the training dataset and 10,000 for the test dataset. It is a balanced dataset with polarity negatives and positives. The experiment reviewed the ktrain lightweight wrapper for TensorFlow Keras, which streamlines the development of the model transformer BERT. The execution of model activities through the ktrain wrapper will use the library's resources that initialize from the ktrain library. The task was done with minimal program code to initialize the

BERT model classifier. This function will process the splitting data process with eight batches based on the parts of program codes shown in Figure 7.

```
model = text.text_classifier('bert',
                             train_data=(x_train, y_train),
                             preproc=preproc)

learner = ktrain.get_learner(model,
                             train_data=(x_train, y_train),
                             batch_size=8)
```

Figure 7: Line Codes In Training The Model Using The Ktrain Library

The classifier model used the method of ktrain.get_predictor to evaluate data from the learner model to predict the sentiment outcome using predictor instances predictor.predict(data). The predictor can be implemented on individual phrases of data to check the sentiment classifier outcome. The method of predictor object is to help the model understand how those predictions were made. Figure 8 is the result based on the unigram classifier. The Unigram word is 'bad,' and the result gives 0 negative value.

```
predictor =
ktrain.get_predictor(learner.model, preproc)
data=["bad"]
predictor.predict(data)
```

Figure 8: Results based on unigram

The calculation was based model classifier for True Positive, False Negative, True Negative, and False Positive. Table 1 shows the results based on learner.validate() method for the testing of 10,000 data (0.2 from datasets of 50,000 with 40,000 for training dataset). The results are based on classification Table I, which then shows the detailed results compared with the baseline models.

Table 1. Evaluation results

Details calculations are based on the measurement equation below [21]:

Model	True Positive	False negative	True Negative	False Positive
BERT	4612	339	4676	373

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} => \frac{9288}{10000}$$

$$=> \mathbf{0.928}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} => \frac{4612}{4612 + 373}$$

$$=> \frac{4612}{4985} => \mathbf{0.925}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} => \frac{4612}{4612 + 339}$$

$$=> \frac{4612}{4951} => \mathbf{0.931}$$

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$=> 2 * \frac{0.861}{1.856} => \mathbf{0.927}$$

Compared with the baseline model summarized, model BERT gets a higher accuracy of 92.8 compared to other baseline models using Support Vector Machine(SVM) and Logistic Regression(LR). For precision, BERT obtained 92.5%, Recall 93.1% dan f1-score 92.7%. This result shows that the BERT model was better than SVM and LR. Detail as described in Table 2.

Table 2. Comparison of results with baseline models

	Accuracy (%)	*Precision* (%)	*Recall* (%)	*F1-Score* (%)
BERT	**92.8**	**92.5**	**93.1**	**92.7**
LR	88.90	88.90	90.80	88.90
SVM	89.80	89.20	90.80	90.0

The discussion on the experimental process is an initialization of the ktrain library that provides more options for the general sentiment analysis process. Ktrain package and sub-modules comprehensively cover most of the tasks in classifying the SA requirements. The experiment reviewed the ktrain lightweight wrapper for TensorFlow Keras, which streamlines the development of transformer models such BERT model.

The prediction instances show that making a predictor of the current data only needs simple code to call the predictor instances, declare it based on the data set, and thus, predict the data polarity. The predictions can be easily defined by utilizing the method predictor objects to understand how the prediction is made. An experimental sample using the unigram n bigram shows that the predictor was accurate based on the predicted results. Figure 9 shows the ktrain library's predictions using the BERT model based on bigram phrases.

```
# calling instances
predictor = ktrain.get_predictor(learner.model, preproc)
#line of data set
data=["bad movie"]
# predict the data
predictor.predict(data)
#Results 0 – negative 1 - positive
['0']
```

Figure 9: Make predictions using the ktrain library and BERT model

Once all models are trained, they can be managed using TensorFlow or the transformer Library. The predictor also can be saved than to be reloaded later in making other deployment classifications by applying the predictor. save method.

Additionally, the study has shown that the task can be done with minimal program code and demonstrated how to do the tasks with less program code. This post explained how to execute most modeling activities entirely through the wrapper. Developing ktrain Library to BERT classifier also used a few line codes for examples method such learner.validate. That will produce the result evaluation that uses a validation set by default, as shown in Figure 10.

```
learner.validate(val_data=(x_test, y_test))
              precision    recall  f1-score   support

           0       0.92      0.93      0.92      5049
           1       0.91      0.92      0.91      4951

    accuracy                           0.92     10000
   macro avg       0.92      0.92      0.91     10000
weighted avg       0.91      0.92      0.91     10000
```

Figure 10: Evaluation Results of the BERT Model

The comparison within the finding of True positives (TP), True Negatives (TN), False positives (FP), and False Negatives (FN) are illustrated in Figure 11. The result shows that the predicted TP and TN were high, and FP and FN were low, resulting in better accuracy.

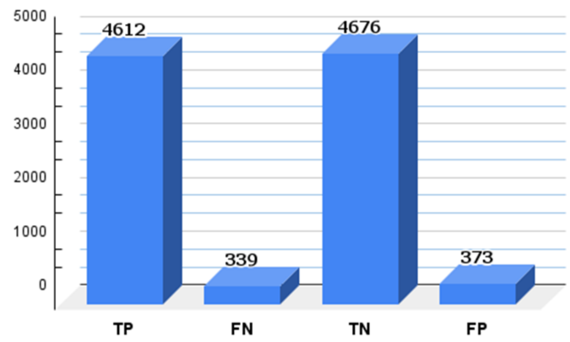


Figure 11. Evaluation results

The comparison of the BERT model to other baseline models resulted in Figure 12. The results show that the BERT classifier results better than other SA models like LR and SVM. The predictions of TP dan TN showed a higher true result. FN and FP also showed lower false results.

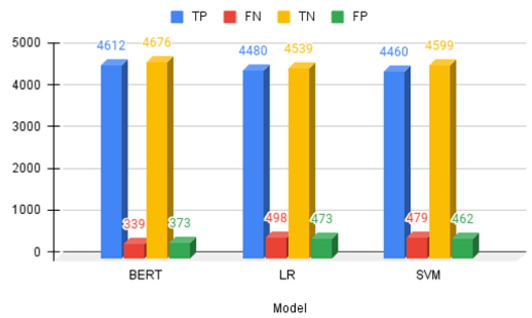


Figure 12: Comparison with baseline models

5. Conclusion

This paper shows that the BERT model works better with the ktrain library resources modules. It successfully processed the function or method to expedite programming capability in delivering results. The proposed model achieved a better result with an accuracy of 92.8%, a precision of 92.5%, Recall 93.1%, and F1-Score with 92.7% compared to Support Vector Machine and Logistics Regression. Additionally, it demonstrated that the library provided by ktrain requires less code and can produce the required task based on NLP needs. Nevertheless, further research needs to work on the limitations. For future work, we advocate wrapping the ktrain library to more pre-trained models and testing them on diverse domains of datasets.

Acknowledgement

The authors gratefully acknowledged Universiti Kebangsaan Malaysia and RHB Bank through grant code RHB-UKM-2020-001 (Geran RHB-UKM) for supporting this research. The authors also thank the Public Service Department of Malaysia(PSD) and the Malaysian Administrative Modernisation and Management Planning Unit (MAMPU) for this research opportunity.

This material was made initially and has never been published elsewhere.

References

[1] W. Kaur, V. Balakrishnan, and B. Singh, "Improving teaching and learning experience in engineering education using sentiment analysis techniques," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 834, no. 1, 2020, doi: 10.1088/1757-899X/834/1/012026.

[2] A. S. Maiya, "ktrain: A Low-Code Library for Augmented Machine Learning," vol. 10703, 2020, [Online]. Available: http://arxiv.org/abs/2004.10703.

[3] P. Ksieniewicz and P. Zyblewski, "Stream-learn — open-source Python library for difficult data stream batch analysis," *Neurocomputing*, vol. 478, pp. 11–21, 2022, doi: 10.1016/j.neucom.2021.10.120.

[4] I. S. Ahmad, A. Abu Bakar, M. R. Yaakub, and M. Darwich, "Beyond sentiment classification: A novel approach for utilizing social media data for business intelligence," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 437–441, 2020, doi: 10.14569/ijacsa.2020.0110355.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, 2019.

[6] S. Das, P. Mandal, and S. Chatterji, "Probabilistic Impact Score Generation using Ktrain-BERT to Identify Hate Words from Twitter Discussions," 2021.

[7] W. Kaur and V. Balakrishnan, "Improving sentiment scoring mechanism: a case study on airline services," *Ind. Manag. Data Syst.*, vol. 118, no. 8, pp. 1578–1596, 2018, doi: 10.1108/IMDS-07-2017-0300.

[8] R. Kiros *et al.*, "Skip-thought vectors," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, no. 786, pp. 3294–3302, 2015.

[9] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–16, 2018.

[10] C. Republic and T. Mikolov, "Statistical Language Models Based on Neural Networks," *Wall Str. J.*, no. April, pp. 1–129, 2012, doi: 10.1016/j.csl.2015.07.001.

[11] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," *2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf.*, pp. 1367–1377, 2016, doi: 10.18653/v1/n16-1162.

[12] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 3079–3087, 2015.

[13] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*, vol. 1, pp. 328–339, 2018, doi: 10.18653/v1/p18-1031.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sustskever, "Improving Language Understanding by Generative Pre-Training," *Comput. Sci.*, vol. 9, no. 1, 2018.

[15] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Section 2, pp. 4766–4775, 2017.

[16] A. Garcia-Silva and J. M. Gomez-Perez, "Classifying Scientific Publications with BERT - Is Self-attention a Feature Selection Method?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12656 LNCS, pp. 161–175, 2021, doi: 10.1007/978-3-030-72113-8_11.

[17] J. Herbst, "A machine learning approach to workflow management," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1810, pp. 183–194, 2000, doi: 10.1007/3-540-45164-1_19.

[18] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, "A review of feature selection techniques in sentiment analysis," *Intell. Data Anal.*, vol. 23, no. 1, pp. 159–189, 2019, doi: 10.3233/IDA-173763.

[19] N. Omar, M. Albared, T. Al-Moslmi, and A. Al-Shabi, "A comparative study of feature selection and machine learning algorithms for arabic sentiment classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8870, pp. 429–443, 2014, doi: 10.1007/978-3-319-12844-3_37.

[20] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," *ACL-HLT 2011 - Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol.*, vol. 1, pp. 142–150, 2011.

[21] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran,

M. Y. Alzahrani, and R. Budiarto, "Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," *IEEE Access*, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Construction of Art Education Platform in Primary and Secondary Schools Based on Virtual Reality Technology

Lu Qi[a,*]

[a]Beijing Youbang Huizhi Enterprise Management Co., Ltd

ARTICLE INFO

Article History:
Submitted 7.8.2022
Revised 11.8.2022
Second Revision 11.20.2022
Accepted 12.19.2022

Keywords:
Virtual reality
Fine arts
Education platform

ABSTRACT

Today, with the rapid development of information technology, Virtual reality (VR) has developed rapidly, and has been popularized in many fields such as film and television, sports and so on. If VR can be successfully introduced and applied in art teaching in primary and secondary schools, it will create a good situation for students to perceive art works, which will stimulate their interest in learning art knowledge and improve their learning effect. Based on 3D image VR, this paper constructs a VR visualization model of art in primary and secondary schools to promote the digital reform of art teaching and the ecological construction of digital education. The simulation results show that the stability of the art visualization model constructed in this paper is still about 90% when there are many business sets, and the subjective scores of art teachers and students are better. The results verify the effectiveness and reliability of the VR visualization model based on VR, which can provide technical support for the construction of art education platform. The VR visualization model is applied to the construction of art education platform in primary and secondary schools, which will promote the innovation and optimization of art teaching.

1. Introduction

Today, with the rapid development of information technology, VR has developed rapidly, and has been popularized in many fields such as film and television, sports and so on, and has achieved good application results [1]. The appearance of multimedia-assisted instruction brings convenience to education, and the improvement of teaching effect after multimedia-assisted instruction is obvious to all [2]. Under the background of quality education, all walks of life pay more and more attention to the cultivation and promotion of students' comprehensive quality. As an important part of quality education, art teaching in primary and secondary schools is characterized by abstract content and strong thinking. In this regard, art teachers actively explore new teaching methods, create real situations through multimedia technology, and promote the visualization of art knowledge, but its teaching effect is average [3]. VR is the product of big data, and its application in art teaching has certain advantages. That is, VR takes pictures of real scenes in real life, and then uses computer modeling to simulate the two-dimensional plan into a real three-dimensional space and present it to the viewers [4]. Network and VR can create a good learning environment for people, promote the application of high-tech information technology in human life and study, and thus accelerate the modernization of education in China. If VR can be successfully introduced and applied in art teaching in primary and secondary schools, creating a good situation for students to perceive art works will help stimulate their interest in learning art knowledge and improve students' learning effect, which has a strong application value [5].

A positive situation can stimulate students' desire to explore and learn independently and deepen their understanding of what they have learned. Because of the aesthetic particularity of the art discipline itself and the need of teaching objectives, the creation of situations is particularly important. There are various ways to create situations, for example, using language description, using pictures or using multimedia devices to create situations, etc [6]. Using virtual simulation worlds such as VR perception, simulation environment and sensor devices, people can interact with each other in the virtual world, which has attracted the attention of art education. Art teachers should actively use VR

DOI reference number: 10.18293/JVLC2022-N2-025

to implement virtual teaching, build a systematic knowledge system for students, and create a real and three-dimensional art teaching situation, so as to enhance students' understanding and application of knowledge and promote their comprehensive development [7]. As a brand-new technology in the digital era, VR uses computers to simulate real scenes, and through special head display or glasses, the real objects are blurred and virtual objects are realized, and a brand-new scene is constructed [8]. Art courses are characterized by perception, understanding and creation of visual images, and images are a very important part of art courses [9]. As a new media platform, VR can be introduced into art classroom teaching by teachers to guide students' learning, or it can be used as a platform for students to learn independently after class.

2. Application of VR technology in art teaching

2.1 Application value of VR in art teaching

Under the background of economic globalization, the competition between countries and enterprises has gradually evolved into talent competition. In order to improve the comprehensive quality of the people, the state actively carries out quality education, and the new curriculum standard of fine arts in primary and secondary schools emphasizes the cultivation and promotion of students' innovative quality, aesthetic level and autonomous learning consciousness. However, at present, some schools attach importance to the teaching of mathematics, English and other subjects, ignoring the development of art courses, which leads to some students' weak aesthetic consciousness and their comprehensive quality not being comprehensively improved. VR overcomes the limitations of teaching venues, whether it is auditory, visual, tactile, realistic sensory experience, and makes the experiencer feel as if he were there. Generally speaking, learning motivation is divided into internal motivation and external motivation [10]. Under the background of quality-oriented education, teachers should combine VR with art textbooks, use VR software to create a three-dimensional life situation, so that students can be there, understand the structure of real objects by rotating, zooming in and out, and deeply understand the meaning of art terms. VR can bring students into a completely realistic teaching situation, and gain pleasure and satisfaction through interactive experience, so as to mobilize students to take the initiative to learn, stimulate their learning motivation and increase their learning experience and participation. Situational learning solves the problem that traditional teaching is divorced from reality, challenges the limitations brought by traditional teaching venues, and promotes students' learning by setting scenes similar to their living environment.

In the past, traditional art teaching in primary and secondary schools mainly used art teachers to show related teaching contents in kind, or demonstrated some specific art operations for students on the spot in class. However, due to the different positions of students in the classroom, the observation effect of some students may be affected by the difference of observation angles. Compared with two-dimensional images, the immersive teaching brought by VR makes the classroom more lively and interesting. More importantly, this learning experience will stimulate students' creativity and imagination [11]. The purpose of art education in primary and secondary schools is not only to let students have certain artistic knowledge, but also to build aesthetic consciousness for students. The establishment of aesthetic consciousness needs to be accumulated over time, and it is the active reflection of the subjective aesthetic attributes of objective perceptual images. If people don't have a universal evaluation standard about interest, they can't have active and effective spiritual communication and aesthetic evaluation. By bringing VR into art classes in primary and secondary schools, students can learn more about painting, sculpture, architecture, design and other arts, so as to cultivate students' aesthetic ability.

2.2 Three-dimensional visual modeling of art images

Introducing VR into art education in primary and secondary schools can help teachers make effective use of resources in teaching design and achieve teaching goals more intuitively. Teachers can give full play to the advantages of VR in teaching design, assist art teaching, overcome the important and difficult points in teaching, impart knowledge more easily and improve classroom efficiency. Under the background of VR, teachers can use VR view to show students the three-dimensional design process, and use related technologies to change the color and style of objects, so as to stimulate students' design interest and potential. For example, in the teaching of packaging design, teachers use VR view to show the design process of packaging box formed by folding, cutting and designing white paper from all directions and angles. The process of image recognition and model training is shown in Figure 1.

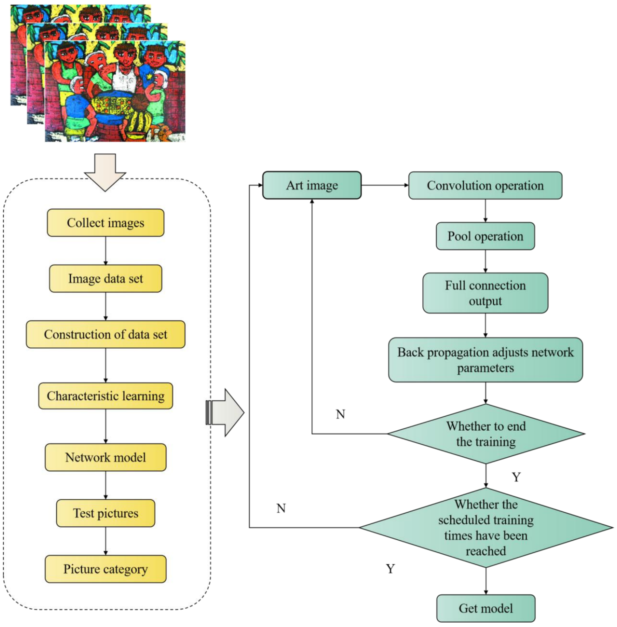


Figure 1: Art image recognition and model training process.

Introducing VR into art classes in primary and secondary schools can greatly enrich teaching means and methods, and use it as an auxiliary classroom teaching tool to make up for the time and space limitation in teaching, so that students can get immersive experience in the classroom. As the main course content of art teaching in primary and secondary schools, painting requires students to show the outline, light, structure, space and texture of objects through simple lines, which has higher requirements for students' thinking ability, spatial perspective and internal structure shaping ability. In VR teaching, teachers can use VR to show students the internal structure of three-dimensional images, and truly restore the texture, contour, structure, light and other parameters of objects. Students can observe the contour, spatial change and texture of objects from a distance, a close distance, looking up and down, etc. by using functions such as zooming and angle adjustment. The three-dimensional interactive model of art resources in primary and secondary schools is shown in Figure 2.

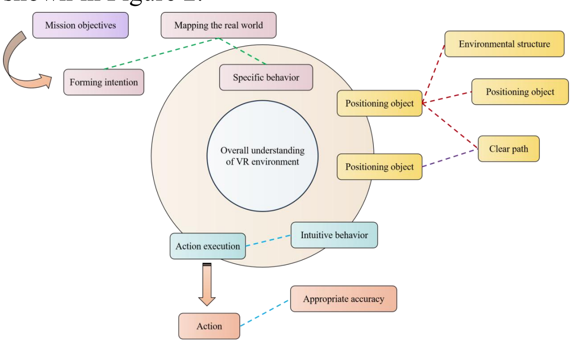


Figure 2: Three-dimensional interactive model of art resources in primary and secondary schools.

3. 3D Reconstruction of Art Images

VR, frame synchronization and integration of original point cloud data are adopted to realize the point cloud information fusion of original art images. The median filter is used to preprocess the point cloud data in the original art image, and the output value after preprocessing is:

$$\psi_{i,j} = median[\Lambda_{i+m,j=n};(m,n)\in w]\{\Lambda_{ij};(i,j)\in Z^2\} \quad (1)$$

Where: m and n represent the size of the window in the horizontal and vertical directions respectively; w , Z^2 respectively represent the plane window specification and the serial number of the two-dimensional data string; Λ_{ij} and $\psi_{i,j}$ respectively indicate that the point cloud coordinates on the image are the output values of (i,j) after median filtering.

In the three-dimensional space, on the a -th image, the minimum calculation method of the square sum of the distance between the vertical projection point of the space point g and the pixel point $\hat{\omega}$ is as follows:

$$\min = \sum_{g=1}^{g}\sum_{a=1}^{A} v_{ga} d\left(E_a S_g, \hat{w}\right)^2 \quad (2)$$

In order to eliminate the registration noise in the image, the initial value is randomly selected and linearly transformed. Local linear approximation method and nonlinear least square method are used to achieve image iteration to convergence and maximum likelihood estimation.

In order to better reflect the distribution of pixels in the image, the Gaussian Zhengtai distribution project of point cloud data optimized by VR will be used to form the reconstruction model. The change of pixels in the image frame in the time domain O is represented by $\{X_1, X_2, \ldots, X_O\}$, and the time X_i of pixels in the frame is selected randomly, so the ratio of X_i to the background of the reconstructed image is:

$$Q(X_i) = \sum_{i=1}^{L} f(X,Y) w_{i,j} \quad (3)$$

Where: $f(X,Y)$ is used to describe the Gaussian probability density function, which is a function of the mean value μ_i of Gaussian distribution and the covariance μ_0^2 of Gaussian distribution items, and L is the number of Gaussian distribution items. The pixel gray scale of the image points is described by the average value μ_i of Gaussian distribution, and the pixel variance is described by σ^2 . Then, the background image is described by the relationship function between gray scale and variance:

$$K(X,Y)=\left[\mu_i,\sigma^2\right] \tag{4}$$

By restricting the pixel gray calculation process of each frame point, according to the definition of image gray, the correct matching of the point cloud position of foreground, background image and original art image is implemented, and the 3D reconstruction of art image is realized.

4. Result analysis and discussion

After understanding the basic characteristics of VR and its effect and influence in art education, teachers should fully realize that the rapid development of VR provides a good opportunity for the development of art teaching in primary and secondary schools, and its far-reaching influence on the future development of education. Students can also use VR to enter the interior of an object to observe its internal structure, and move the light source to observe the line, color and light changes of the object. Through comparison, they can master the painting characteristics of light and structure, and then improve their painting level.

VR-assisted instruction frees art teachers from the tedious preparation of teaching materials, but it also weakens the role of art teachers. If art teachers want to continue to play a leading role in teaching, they must keep pace with the times and strive to improve their comprehensive quality. Although VR is gradually coming into our daily life, there are still some challenges in how to make use of VR in art teaching in primary and secondary schools. For example, due to the high technical threshold of educational resource design and development, it is difficult for subject teachers to independently develop VR courseware needed for teaching, and the quality of VR resources depends on developers' understanding and grasp of teaching content and teaching methods. Figure 3 shows the accuracy comparison of different algorithms.

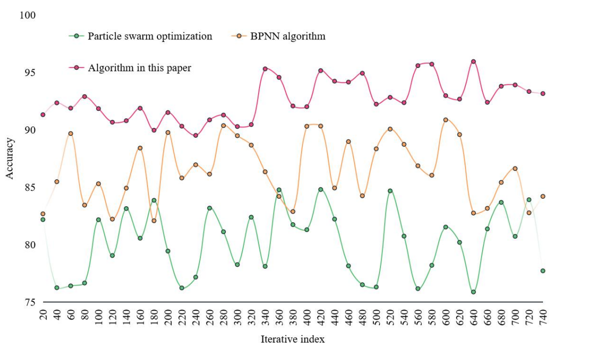


Figure 3: Accuracy comparison results of different algorithms.

The VR visual modeling algorithm proposed in this paper has higher accuracy of image recognition. During the art teaching in primary and secondary schools, VR application involves more professional equipment and instruments, and the price is relatively high. Therefore, in order to ensure the smooth application of VR, it is necessary to increase the educational investment of schools and actively purchase some related equipment for VR application. The stability results of the art education platform are shown in Figure 4.

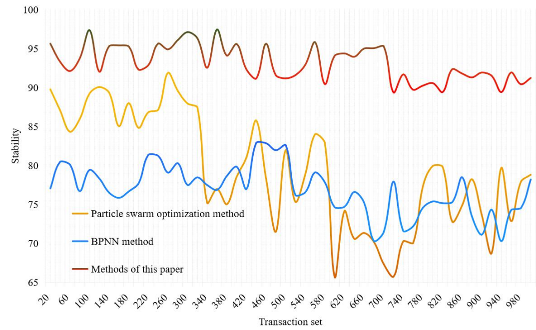


Figure 4: Stability results of the platform.

It can be seen that the stability of the art education platform constructed in this paper is relatively high, and its stability is still around 90% when there are many transaction sets. With the emergence of virtual teaching, we should inherit and learn from the old traditional teaching ideas and concepts, and reform those outdated, backward, inefficient and costly teaching methods and means. Because of the particularity of art itself, it is a course that integrates appreciation and hands-on ability, and the training of motor skills is far higher than that of general courses.

In using VR to assist art teaching, we can't ignore the cultivation of motor skills in art teaching in primary and secondary schools. In the process of teaching, students are immersed in VR, and it is difficult for teachers to monitor students' learning situation. Too many functions and rich simulation scenes in the virtual world will interfere with learners' absorption of important content. This paper selects 260 art teachers and students to evaluate the art education platform constructed in this paper. The subjective grading results of art teachers and students are shown in Figure 5.

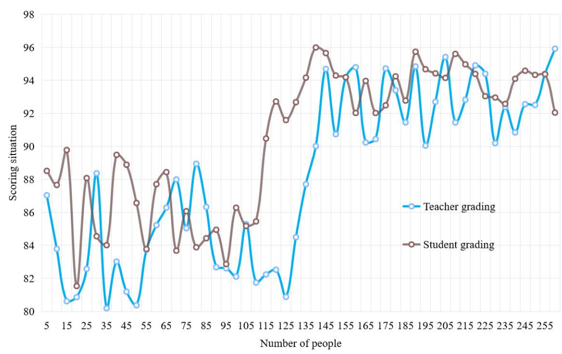


Figure 5: Subjective rating results of art teachers and students.

The results show that the subjective scores of art teachers and students are better. The results verify the effectiveness and reliability of the art education platform based on VR, which can provide some technical support for art teaching. In today's art courses in primary and secondary schools, the classroom mode dominated unilaterally by teachers still accounts for the majority, while the application of VR will improve the original teaching mode and increase the proportion of students' spontaneous exploration of knowledge in the classroom. It can not only exercise students' ability of active learning, but also cultivate students' teamwork ability and innovative spirit in the long-term teaching process, and promote the all-round development of primary and secondary school students.

5. Conclusions

Today, with the rapid development of information technology, VR has developed rapidly, and has been popularized in many fields such as film and television, sports and so on, and achieved good application results. In this paper, based on 3D image VR, a VR visualization model of art in primary and secondary schools is constructed to promote the digital reform of art teaching. The experimental results show that the VR visualization model constructed in this paper has high stability, and its stability is still about 90% when there are many transaction sets. At the same time, the subjective scores of art teachers and students are better. The experimental results verify the validity and reliability of the art visualization model based on VR, which can provide technical support for the construction of art teaching platform in primary and secondary schools. It provides a good opportunity for the development of VR art teaching in primary and secondary schools. However, in actual teaching, in order to successfully apply VR, it is necessary to increase the investment of teaching funds, actively purchase some VR supporting hardware and equipment, and strengthen the construction of teachers. At the same time, according to the teaching needs of art courses, VR should be applied to teaching links such as class opening, teaching and appreciation.

References

[1] Chang Xiaojun. (2021) “School-enterprise mutual promotion of the construction of Industry-University-Research platform for art education curriculum system under ecology”. Educational Theory and Practice, vol. 41, no. 6, pp. 4 .

[2] Liu Huaiying. (2019) “Problems and reflections on the cultivation of students majoring in art education in normal universities”. Educational Theory and Practice, vol. 39, no. 36, pp. 3 .

[3] Zhong Chaofang. (2018) “On the cooperation basis of contemporary art education between China and Africa”. Art Observation, vol. 2018, no. 7, pp. 2.

[4] Wu Jun. (2020) “Research on the integration of tea painting art in the process of higher vocational art education. Fujian Tea”, vol. 42, no. 10, pp. 2 .

[5] Zhang Lichun. (2021) Research on the application of VR in art history teaching in colleges and universities. Educational Theory and Practice, vol. 41, no. 24, pp. 4 .

[6] Wang Chenhai, Peng Chanjuan. (2018) “The application and prospect of VR as a new media in the field of art exhibition design”. World Art, vol. 2018, no. 2, pp. 5 .

[7] Xu Zhe. (2022) Research on VR and Animation Application. Art Observation, vol. 2022, no. 5, pp. 1.

[8] Ren Lifen. (2019) “Application analysis of VR in modern exhibition communication”. Publishing Wide Angle, vol. 2019, no. 2, pp. 3.

[9] Li Xiaoyu. (2018) “Research on Fashion Design System Based on VR. TV Technology”, vol. 42, no. 12, pp. 5 .

[10] Liang Zheng, Zhang Yuhong.(2021) “VR and digital media art teaching. Art Observation”, vol. 2021, no. 3, pp. 1.

[11] Shi Chunyan.(2018) “Research on VR Method of Interior Design Based on 3D Vision. Modern Electronic Technology”, vol. 41, no. 5, pp. 6.

Journal of Visual Language and Computing

Volume 2022, Number 2