**THE IMAGE OF THE CITY THROUGH THE EYES OF MACHINE REASONING**

**Abstract**

The majority of humans will live in cities by the end of 2050, and according to Forbes magazine, hundreds of new cities will be built in China alone in the coming decade. City layouts have an important impact on many aspects of city life, e.g., transportation, urban dynamics, urban morphology, and quality of life. There is a deep correlation between urban form and urban performance and it is important to consider urban layouts in terms of their effects on the quality of life in cities. This chapter investigates the development of a Machine Reasoning Engine that learns common design patterns from the layout of a small number of reference cities. In this study, we selected six cities as positive and negative examples based on their quality of life indexes. The Machine Reasoning Engine identifies latent patterns (such as Kevin Lynch’s design elements) within these city layouts based on a set of policies and builds a vocabulary starting with lower-level urban patterns that are combined hierarchically to develop higher-level concepts corresponding to more complex city patterns. Our approach differs significantly from current neural network based approaches that require big data instead, we identify seminal patterns within cities with limited data. Our approach has the potential to open up research in fields where data is scarce, i.e., there is a limited number of city layouts in the world. This research has the potential to automate and guide city planners to develop high-performing city layouts in the future.

**KEYWORDS:** Design Patterns, Quality of Life, Artificial Intelligence, Machine Reasoning, Urban Design

**1.Introduction**

According to the 2018 report published by the United Nations, more than 55% of the world population currently resides in urban settlements (United Nations, 2018). They predict that 6.5B people, i.e., 66% of the world population will live in cities by the end of 2050. With the increase in population, city layouts are expanding and becoming more complex. However, traditional urban planning processes do often not meet the needs of an ever expanding city world. Considering that urban design and planning need to deal with increasingly more complex issues that affect the quality of life (QoL) and the performance of cities, novel and innovative tools can help planners and designers create new cities and reshape existing ones. Over decades, various methods of urban design have been explored, i.e., designing with patterns (Alexander et al, 1977; Salingaros, 2000) or codes (Blum et al, 2006), and many others. These methods dealt mostly with quantifiable aspects of the city. However, Artificial Intelligence (AI) advances aim to build cognitive systems and AI can assist us to explore qualitative problems, too. The integration of AI into the design processes of cities opens up enormous opportunities to understand latent qualities of urban spaces that were impossible to quantify and hence were mostly omitted in design deliberations. There are few AI-driven urban planning studies, such as the “Smart Design framework featuring urban design decision-making reinforced by artificial intelligence-aided design (AIAD)” (Quan et al, 2019) or “AI for Earth Land Cover Mapping user-centric tool which helps urban planners make decisions,” developed by Ho Chi Minh City Planning Department in Vietnam (Traunmueller et al, 2021). Nevertheless, there has not been a thorough attempt to learn what makes high-performing cities viable nor to apply these to upcoming challenges in order to solve the world’s urban expansion. According to Forbes magazine, hundreds of new cities will be developed around the world. The best example is of this approximately 400 new cities were built since 2013 in China according to obtained official numbers from Beijing’s National Development and Reform Commission (Shepard, 2017). Thus to find a solution for irrepressible expansion, our main questions are ‘Can we leverage AI to discover recurring urban patterns on a small number of cities that have been evaluated according to the QoL index?’ and ‘Can we use the output to compose new high-performing cities?’.

We developed a Machine Reasoning (MR) Engine that allows conventional machine learning to work with more structured and combinatorial data. The proposed engine is able to learn the structure of underlying functions, grammars, or relationships, within the city data. Our goal is to develop a MR-based software tool that offers multiple design solutions for a given area in the city. As illustrated in Figure 1, our research has five phases:

* Phase 1: Identify common architectural patterns (base concepts) using 2D grid maps of select cities.
* Phase 2: Generate higher-level concepts (corresponding to higher-level architectural patterns) by building a hierarchy of the base concepts identified in Phase 1.
* Phase 3: Generate architectural options for a given rectangular region by composing urban patterns identified in Phase 1 and Phase 2.
* Phase 4: Add contextual data to Phase 3 when generating architectural options so that the proposal takes the rest of the city into account.
* Phase 5: Enhancing the work from 2D to 3D.

Phase 1 and Phase 2 are discovery problems; whereas Phase 3 and Phase 4 are design problems. In Phase 1 and Phase 2, we determine common design patterns of selected cities. In Phase 3 and Phase 4, the outputs of the tool will be tested within a defined boundary condition. Finally, in Phase 5, the tool will be able to deal with 3D spatial data. In this chapter, we will limit our work to, and illustrate the results of Phase 1 and Phase 2.

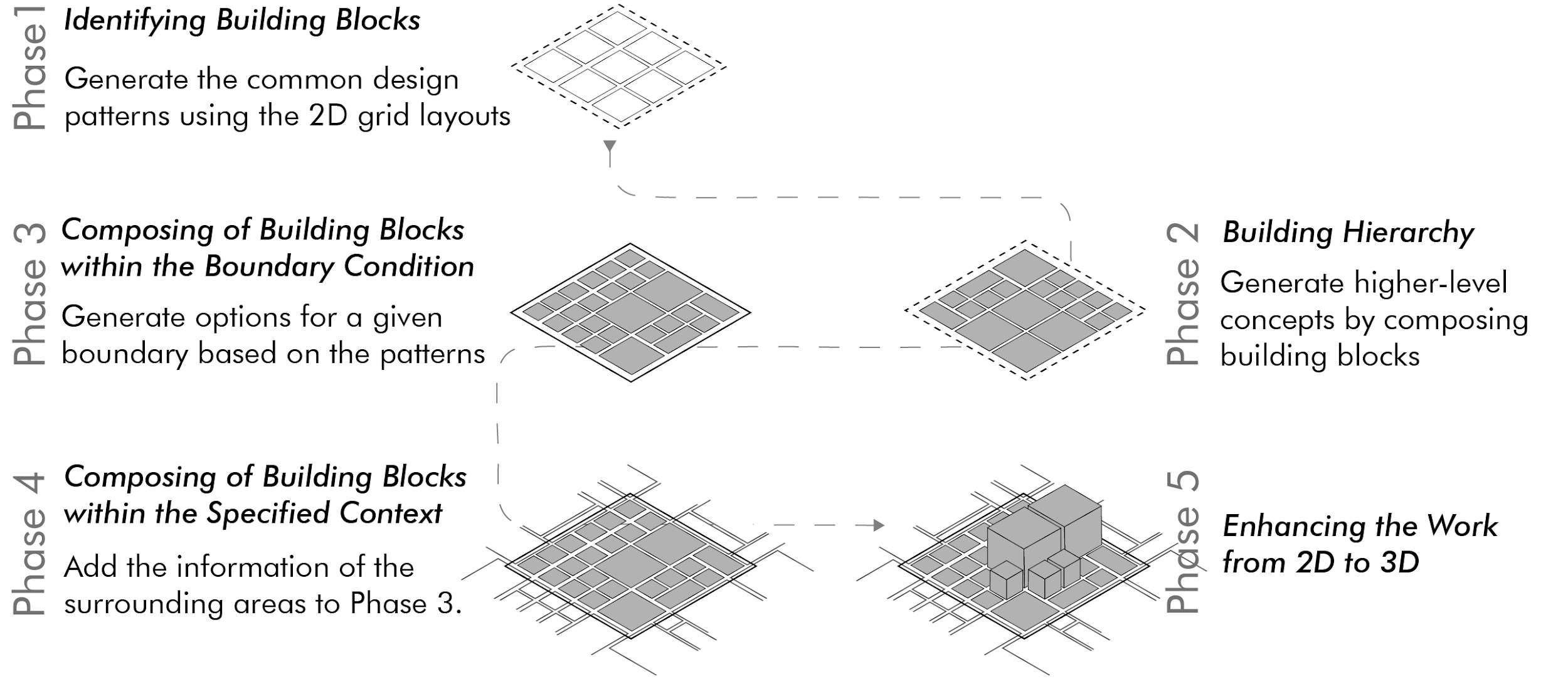


Figure 1. Phases of the study

**2. Background**

Christopher Alexander set out the basic principles of computational design in his work "Notes on the Synthesis of Form" (1964). Nicholas Negroponte demonstrated that human-machine dialogue can be actively provided in the creation of space compositions with his work "The Architecture Machine" (1970). The emergence of computational design as a new area of research was with Alexander's work "A Pattern Language" (1977) on how and why computers should be used by users to formalize the concept of design patterns. Alexander proposed patterns as helpful conceptual tools to deal with complex new design structures. More recent projects include the “City Matrix” (2017) at MIT Media Lab, which leverages machine learning to generate multiple urban simulations within a given urban field. The simulation results are then used to train a Convolutional Neural Network (CNN) to predict traffic and solar performances of unprecedented city configurations. Another recent project is “City Induction” which aims to develop an urban design machine for producing site plans (Duarte et al, 2012). The proposed model consists of three sub-models. To achieve syntactic and semantic interoperability among these three sub-models, they developed a common urban space ontology. Moreover, the “City Scope” project developed by MIT Media Lab, facilitates interaction between people and machines on 2D and 3D interfaces. It has set an example for similar studies in terms of combining computational and physical models (Alonso et al, 2018). In another recent study, “Urban Design Optimization”, computational optimization techniques at the urban design scale were examined and a tool that can code different algorithms has been produced. Urban fabrics optimized in terms of accessibility have been suggested as the outputs of this tool (Lima et al, 2021).

On the other hand, Kevin Lynch assesses environmental quality based on perception in “The Image of the City” (1960). Lynch analyzes two qualities of the built environment as *legibility* and *imageability*, to understand how people perceive a city and how they represent it. Lynch defines *legibility* as ‘the ease with which the city’s parts can be recognized and organized into a coherent pattern’ and *imageability* as, ‘that quality in an object which gives it a high probability of evoking a strong image in the observer’ (Lynch, 1960, p. 60). Lynch conducted a study in which participants experienced the cities of Boston, Jersey City, and Los Angeles. The participants were asked to describe the main elements in the mental maps formed in their minds about the cities. As a result of the study, Lynch discovered that in the mental maps formed in the minds of participants, there are important elements that organize and promote that city. Hence, Lynch classifies the contents of these city images into five types of elements: *Paths, Edges, Districts, Nodes,* and *Landmarks* as illustrated in Figure 2. For Lynch, these elements are viewed as the main materials of the environmental image and they must be provided together at the city scale for a satisfying city form to emerge.

Lynch's *nodes* element could be identified as the highest central focus point in the street network. Nodes may be junctions, a crossing/convergence of *paths*, and may also be the central square, the park where more functions happen simultaneously. *Paths* are the predominant elements that guide people’s movement by supporting orientation in the city. People observe the environment and relate environmental elements while moving through paths. Paths are characterized by continuity, directional quality, and gradients e.g. main streets and boulevards. *Districts* are the relatively large parts of the city that have common characteristics like shape, texture, class, ethnic area. These characteristics determine an endless variety of district types. Neighborhoods or blocks with clear edges may be given as examples of *districts*. *Edges* are linear elements that act as lateral references not used or considered as paths. Urban barriers e.g. shores, railroads, cuts, walls may be edges of areas. *Landmarks* are physical objects which are identified by uniqueness, singularity, and specialization. Landmarks must be visible over long vistas and represent a reference point. Buildings, monuments, etc. might be landmarks (Lynch, 1960, p. 47-48).

Can AI automatically identify Lynch's five elements from city data? We hypothesize that an MR tool can discover these elements on 2D city layouts. This approach contrasts with earlier approaches of generative design, such as shape grammars, where given design rules dictate which layouts might be formed by repeated application of basic building blocks.

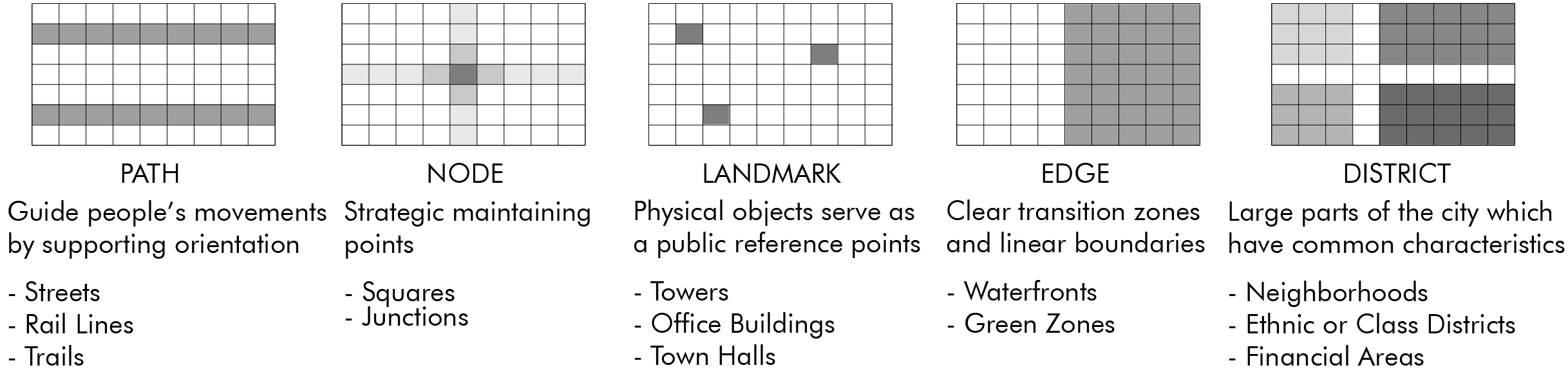
****

Figure 2. Lynch’s five elements

**3. Methods/Tools/Techniques**

Machine Learning (ML), in particular Deep Neural Networks (DNNs), forms the backbone of the current phase of AI revolution. ML techniques have proven successful in leveraging big data to answer questions narrowly posed around the training data sets, in particular for applications using voice, image, and text data. However, architectural data for cities is more structured and combinatorial than these modalities (As et al, 2018). We can consider leveraging DNNs to identify common design patterns in cities, but the lack of availability of big data is also a bottleneck. The number of cities is in thousands not in millions, and data may exist only for a small fraction of these. Therefore, a plausible approach should be able to learn from a small number of examples, and not require big data.

The very nature of the research objectives of discovering design patterns from data requires the ability to make abstractions and generalizations. Even though there are some successes in transfer learning, deep learning has proven difficult to generalize beyond their initial set of training questions. Pushing the boundaries of AI towards a general-purpose design tool requires a new approach that addresses such weaknesses of the current machine learning techniques.

In this study, we used an MR Engine to identify common design patterns in cities, such as the ones identified by Lynch. Similar to ML, the reasoning engine can be used to train MR models in a supervised or unsupervised manner. Supervised training requires annotated data with class membership, and learns to identify which patterns differentiate one class from another. It can for example be used to reason about what characteristics make a city liveable. Unsupervised training does not require annotations and is used to identify common patterns as intermediate vocabulary so that the data can be compressed without losing critical information. Discovery of Lynch’s 5 design elements falls into this category.

MR models capture the underlying patterns that explain the training data. MR Engine generates hypotheses from simple to more complex to explain data, where each hypothesis corresponds to an MR model. MR Engine is configurable in that it can operate with different sets of hyperparameters. It can learn arbitrary concepts from a few examples, and can also discover higher-level cognitive relationships among them. MR Engine complements the existing ML Systems in three dimensions:

1. It does not require problem-specific model development;
2. It can be trained with small data;
3. It can identify cognitive relationships that are out of reach for ML tools and techniques.

We can view MR Engine from the perspective of an AI framework. Figure 3 illustrates our vision of how AI will be built based on the combination of ML and MR models, and programs.

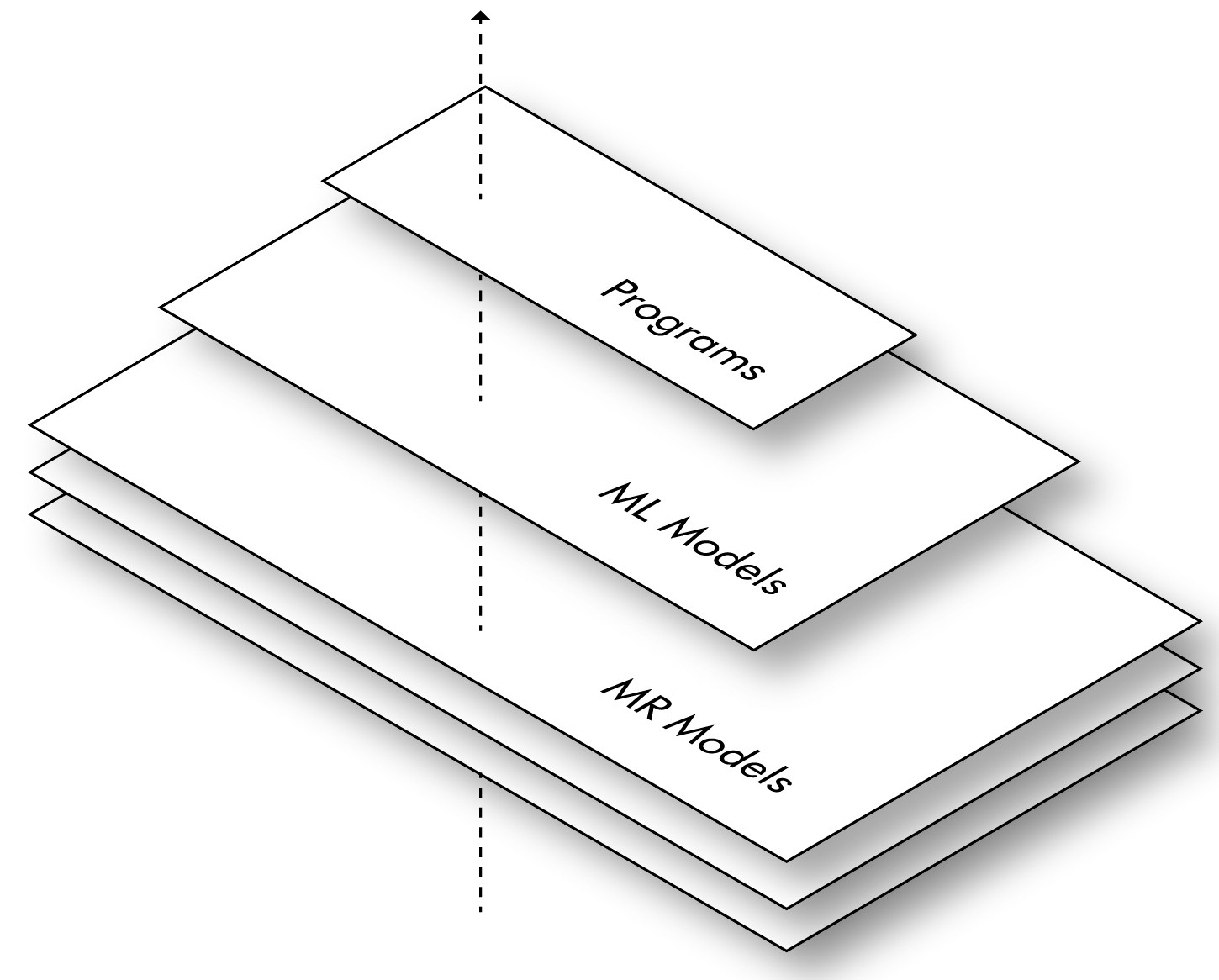


Figure 3. Artificial Intelligence (AI) framework

If we look at the figure from top to down, it encapsulates three different ways of building AI capabilities, as follows:

* In its simplest form, intelligent functions can be developed by coding them in the form of Programs. Early work in AI (including Expert Systems) falls into this category.
* Alternatively, we can build ML models that demonstrate intelligence capabilities. ML models may use “features” that are extracted using Programs as input to do classification. However, the output of the model is not within the space of the features. We cannot feed the output of a model to itself recursively to build a hierarchy of learned concepts.
* MR models do not have this limitation. They operate at the “concept” level. Both inputs and outputs of MR models are within the space of concepts. The output of an MR model can be fed into the same model recursively to build a hierarchy of learned concepts. They are also able to use ML models and other Programs as input, leveraging the existing AI systems.

MR Engine is also a step towards building more explainable AI systems. ML models are tensors, a huge array of floating-point numbers, which does not lend itself to easy interpretation. MR models on the other hand can be presented in a form that can be inspected by humans. We can debug the inference process of such models and see why certain test data is accepted or rejected. MR Engine provides the following main functions: observation, induction, deduction, hypothesis generation, concept discovery, tokenization, and abstraction. The details of how these main functions are used in our case study are given in the next section.

**4. Case Study**

For our case study, we converted the downtown areas of the selected six cities into 2D matrices. We populated the matrices with 10 peculiar symbols representing various city programs, e.g. commercial, residential, etc. We considered Lynch’s common physical characteristics in cities (Lynch, 1960, p. 105-106), and determined 3 policies: a. Frequency, b. Range, and c. Coverage policies. Our MR Engine discovered city grammar and determined common design patterns based on determined policies. Figure 4 illustrates our workflow.

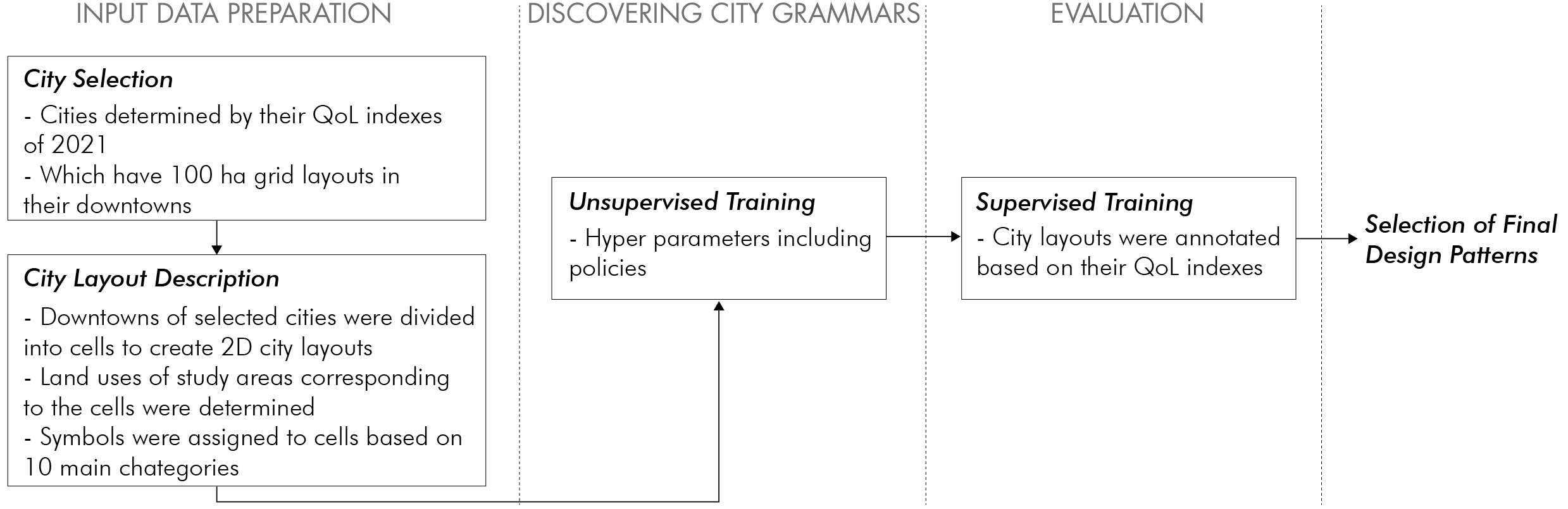


Figure 4. Conceptual model workflow

**- Selecting Cities and Creating Grid Layouts**

We based our selection of cities on Numbeo’s QoL index (Numbeo, 2021), i.e., three of them were very highly ranked: Adelaide, Australia (1st), Wellington, New Zealand (3rd), and Raleigh, United States (4th) (Figure 5); and, the other three were at the bottom of the list: Nairobi, Kenya (230th), Ho Chi Minh, Vietnam (235th), and Manila, Philippines (240th) (Figure 6). We selected cities that were built on grid layouts. Also, to keep it uniform across city examples, we selected about 100 ha within their downtown areas.

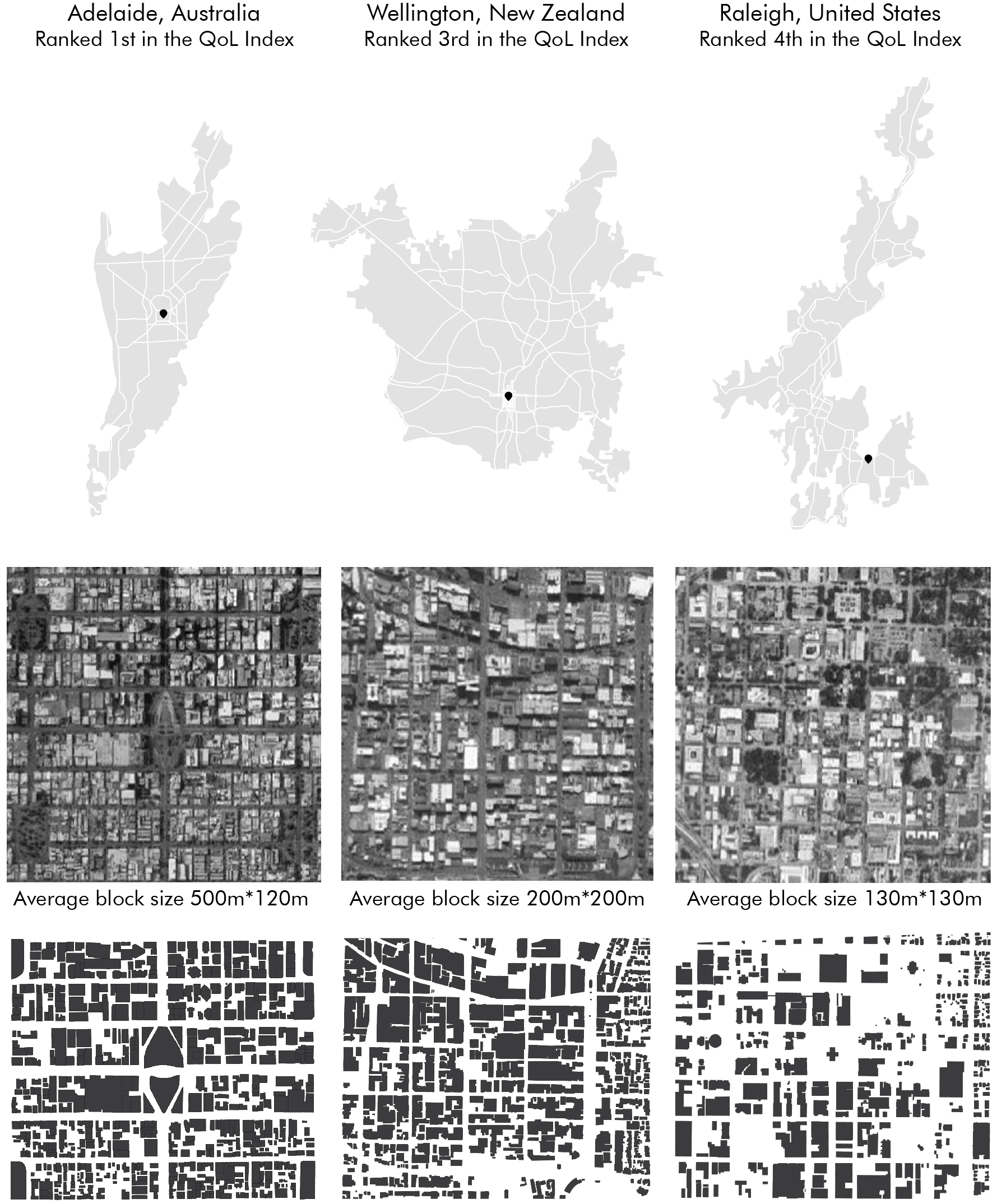


Figure 5.City layouts with the highest quality of life indexes

(prepared according to May 2021 QoL indexes on the Numbeo website)

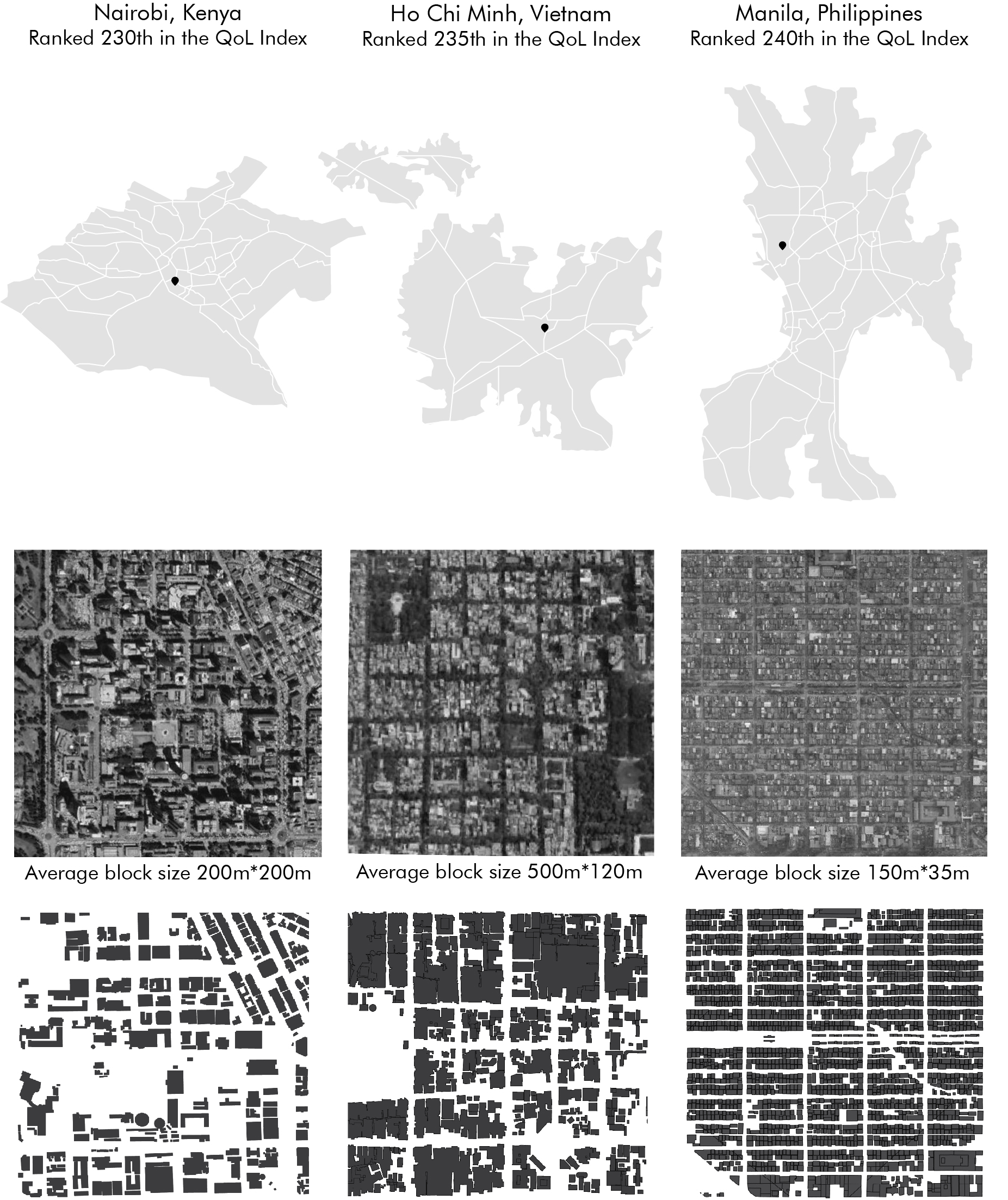


Figure 6. City layouts with the lowest quality of life indexes

(prepared according to May 2021 QoL indexes on the Numbeo website)

There are plenty of indexes to evaluate city performance i.e. Cities in Motion index, Global Cities index, Livability index, etc. We decided to determine cities according to QoL index on the Numbeo website because QoL indexes were calculated as an estimate of the overall QoL by using an empirical formula that takes into account purchasing power index, pollution index, house price to income ratio, cost of living index, safety index, health care index, traffic commute time index and climate index (Numbeo, 2021).

where, *ppi* = purchasingPowerInclRentIndex*, hpr* *=* housePriceToIncomeRatio, *cli* = costOfLivingIndex*, si* *=* safetyIndex*, hi* *=* healthIndex*, tti* *=* trafficTimeIndex*, pi* = pollutionIndex, and *ci* = climateIndex*.*

The downtowns of selected cities with a grid layout of 100 ha were divided into 10m\*10m cells in order to convert them into 2D matrices. The land uses of the city downtowns corresponding to the cells were determined and category names such as residential buildings, transportation, and health centers, etc. were given to the cells. 96 sub-categories determined for the 6 selected city downtowns were gathered under 10 main categories and symbols were assigned to the categories. These symbols and 10 main categories are M (mixed-use), R (residential buildings), G (open areas), H (health centers), + (transportation), E (educational buildings), S (socio-cultural buildings), P (public buildings), C (commercial buildings), and O (others). These correspond to the base vocabulary of the grid data. Within the scope of this study, we limited the top-level categories to 10 symbols.

**- Guiding Policies To Discover Design Patterns**

As indicated in Lynch's book, there are common themes that repeat references to certain general physical characteristics that run through the whole set. *Singularity* or *figure-background clarity* are the qualities that make an element remarkable, noticeable, and recognizable; *form simplicity* allows us to easily incorporate it in the image; c*ontinuity* facilitates the perception of complexity as interrelated; *dominance* is a quality that dominates one part over others; c*larity of joints* are the strategic points of structure that show clear relation and interconnection; d*irectional differentiation* quality differentiates one end from another by means of asymmetries, gradients; v*isual scope* increases the range and penetration of vision by using transparencies, overlaps, vistas and panoramas; m*otion awareness* quality provides an observer to reinforce and develop sensing form in motion; *time series* are sensed sequences and provide linkage items; n*ames and meanings* are the non-physical characteristic qualities that may enhance the imageability of an element (Lynch, 1960, p. 105-106).

We introduced three different policies as Frequency, Range, and Coverage (Figure 7), to guide the MR Engine’s discovery process of design patterns. While determining these three policies, the themes that Lynch mentioned as the common physical characteristics of cities were taken into consideration. Determined Frequency policy has similar features with the Continuity quality; Range and Coverage policies have similar features with the Dominance quality.

1. Frequency policy counts each occurrence of the candidate patterns and prioritizes the most frequent ones over the less frequent ones.
2. Range policy looks at the areas (size) of occurrences for each pattern candidate. In this policy, regardless of whether the pattern repeats to other areas or not, the pattern that is associated with the largest area is prioritized over the others.
3. Coverage policy adds up the areas of every occurrence of a given pattern and prioritizes the ones which cover the largest area over the others.

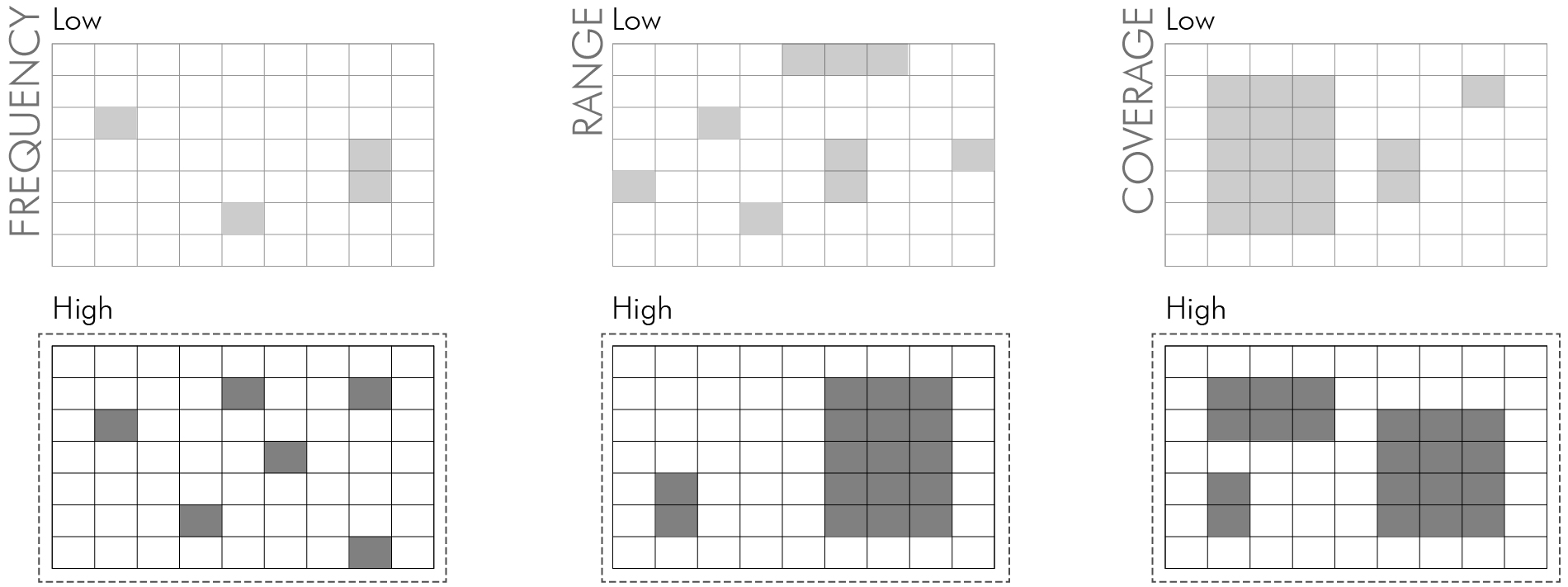
****

Figure 7. Frequency, Range, and Coverage policies for identification of urban patterns

The MR Engine goes through three steps to discover design patterns. First, it generates numerous pattern candidates, where each can be represented as a regular expression. Then each candidate is evaluated and ranked based on available policies. Then, highly ranked candidates are evaluated further based on their fitness to contribute to the specification of the city grammar in a compressed manner. If a candidate pattern leads to a more concise representation of the grid data, then it is favored over the others.

**- Discovering City Grammars**

When we train the MR Engine with the dataset, it will first try to identify the best *vocabulary* to represent the city layouts. The vocabulary consists of the base concepts (R, G, H, …, O) that appear in the city layout, plus the new concepts that we discover to serve well to achieve a concise overall representation. Each new concept corresponds to a pattern that is expressed as a regular expression. The first level of vocabulary contains commonly used simple design patterns within the training data. It will then discover a hierarchy of more complex design patterns, using not only the base concepts but also the newly discovered ones, which will gradually build the grammar of the city layouts, as shown in Figure 8. For example, the tool may discover the concepts of residential areas (RA) as a function of residential buildings (R) and parks (P); commercial areas (CA) as a function of commercial buildings (C) and schools. These two new concepts (RA and CA) then may serve as the building blocks of higher-level concepts, such as towns (TT), as follows:

RA = ***f***(R,P)

CA = ***g***(C,S)

TT = ***h***(RA,CA,\*)

MR Engine can be trained in a supervised and unsupervised manner. Unsupervised training is used to detect common patterns from unlabelled data. Supervised training, on the other hand, will require annotated data. The data samples from the above six cities are annotated based on membership to two layout categories. The first categorycontains cities with high QoL index, and the second category contains cities with low QoL index. The remaining data is unannotated. The aim here is to determine the basic features that distinguish the cities into two categories. The tool uses all data to capture commonly seen design patterns but uses only the annotated data to decide which patterns should be used and which ones to be avoided when generating new designs. The training produces a model, which describes high-performing layouts as a function of all the input symbols plus the newly discovered concepts. The tool also provides a utility, which generates an unbounded number of new designs using the high-performing layout models (Figure 8). The tool allows experimentation to see the impact of adding new layouts to the dataset.

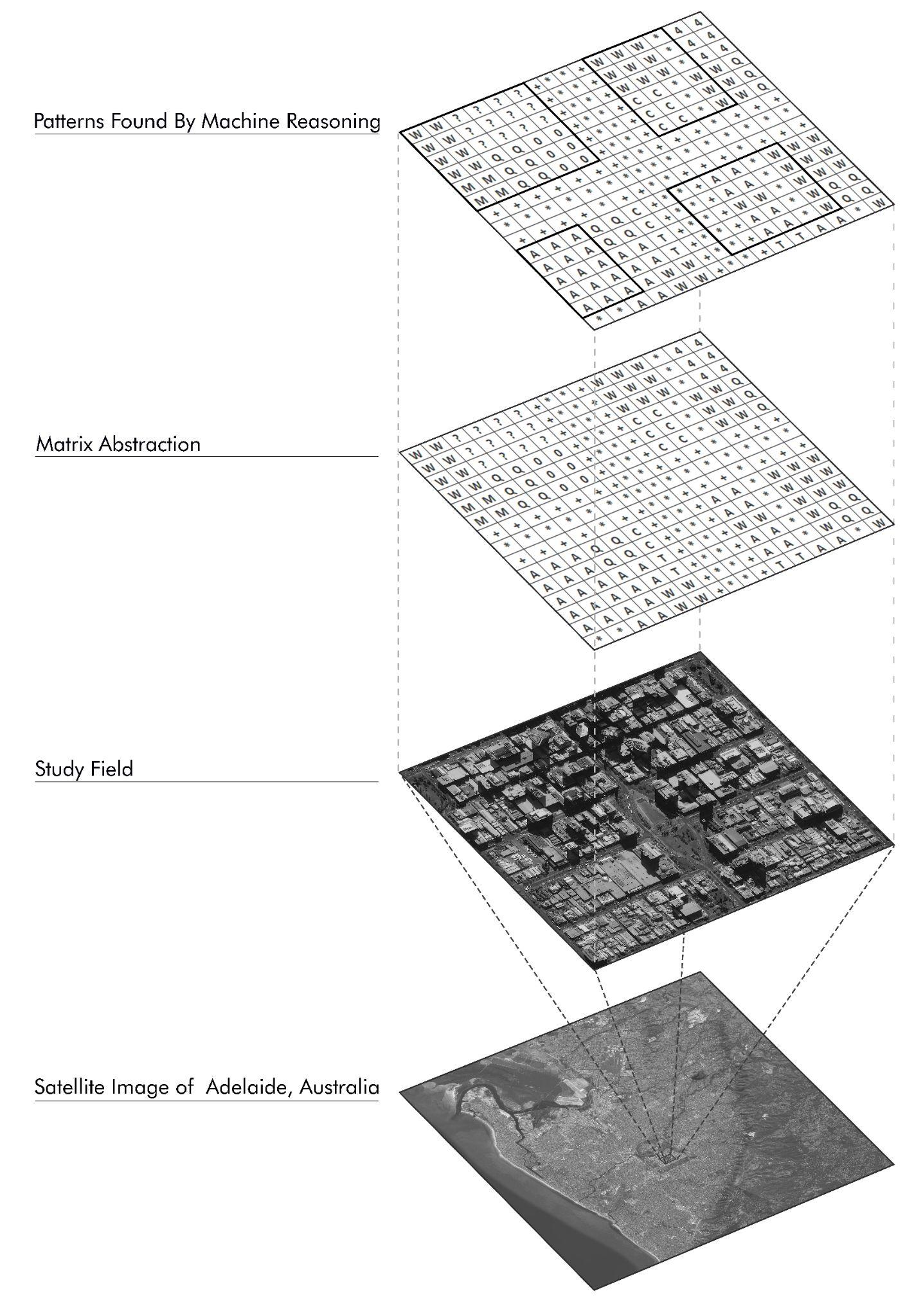


Figure 8. From raw data to city grammar

**- Results**

In this section, we provide a peek into the preliminary results in relation to our research objective of automatically discovering Lynch’s design elements as paths, edges, districts, nodes, and landmarksusing the MR Engine. The following is a summary of a typical training session using MR Engine:

Create a new concept A and introduce positive training samples

Introduce negative training samples

Repeat (i = 1 ... n)

Set the hyperparameters

Discover a pattern X(i) within A

Visualize X(i)

Express A as a function of X(i)

Generate a theory to express A as a function of X(0), X(1), …, X(n)

Visualize A = f(X(0), X(1), …, X(n))

The following illustrates the command line interface of the training session performed to discover the city grammar for this research:

u L+ Adelaide.txt */\* Create a new concept L for “liveable cities” \*/*

u L+ Wellington.txt */\* Introduce Adelaide, Wellington and Raleigh \*/*

u L+ Raleigh.txt */\* as positive samples \*/*

u L- Nairobi.txt */\* Introduce Nairobi, Ho Chi Minh and Manila \*/*

u L- HoChiMinh.txt /\* as negative examples \*/

u L- Manila.txt

s l C */\* Set the hyperparameters \*/*

n f X L */\* Discover a pattern X within L \*/*

c X */\* Visualize X \*/*

u L< X */\* Express L as a function of X \*/*

n f Y L */\* Discover a pattern Y within L \*/*

c Y */\* Visualize Y \*/*

u L< Y */\* Express L as a function of Y \*/*

n f Z L */\* Discover a pattern Z within L \*/*

c Z */\* Visualize Z \*/*

u L< Z */\* Express L as a function of Z \*/*

Notice that training the MR Engine is as simple as introducing the data (L), setting hyperparameters, and successive rounds of discovering new concepts (X, Y, Z, and so on) and expressing L as a function of these new concepts. With these training sessions, the MR Engine is able to identify Lynch’s five design elements as illustrated in Figure 9 Manila city and Figure 10 Raleigh city examples.

Path is the first design pattern that the MR Engine discovers. MR Engine specifies path as a function of “+” symbols distributed in 2D rectangular regions. The path elements work well for compositionality, for building the city grammar as it breaks the grid data into smaller modular regions. Figure 9 illustrates the result of one such run for the grid data of Raleigh. Here we can see the roads that appear horizontally and vertically in the grid. We can also see the nodes in Lynch’s terminology that correspond to the intersection of path elements. As can be seen in Figure 9, the identified path elements divide the city map into rectangular areas and serve as the intermediary concepts in building the city grammar.

******

Figure 9. Patterns discovered by MR Engine in Raleigh’s downtown

The rectangular areas that stay between these roads are the unprocessed portions of the data at this stage of processing. Notice that Kevin Lynch’s districts correspond to the rectangular regions separated by path elements. MR Engine does not immediately assign any symbol (concept) to these rectangular regions but treats them as raw data that can be taken into account for further concept discovery. In the next round, MR Engine identifies regions that have both commercial (C) and residential buildings (R) as a new concept, which is an example of a district. Notice that there may be many different patterns for districts. MR Engine’s discovery process works based on the discovery policy and prioritizes the candidates that contribute to the most concise expressions.

MR Engine also generates patterns that correspond to Lynch’s edge elements. As can be seen in Figure 10, there are examples of edge elements, such as the border between the green areas and the rest of the rectangular region. Landmark elements have a special place from the perspective of this analysis in that they do not correspond to a pattern that appears in many places, but as a unique instance that appears once. We use the input data to mark the landmark instances.

******

Figure 10. Patterns discovered by MR Engine in Manila’s downtown

This outcome is not unique to a single city but everywhere. Figure 11 illustrates samples of the design elements that we identified automatically for each of the selected six cities.

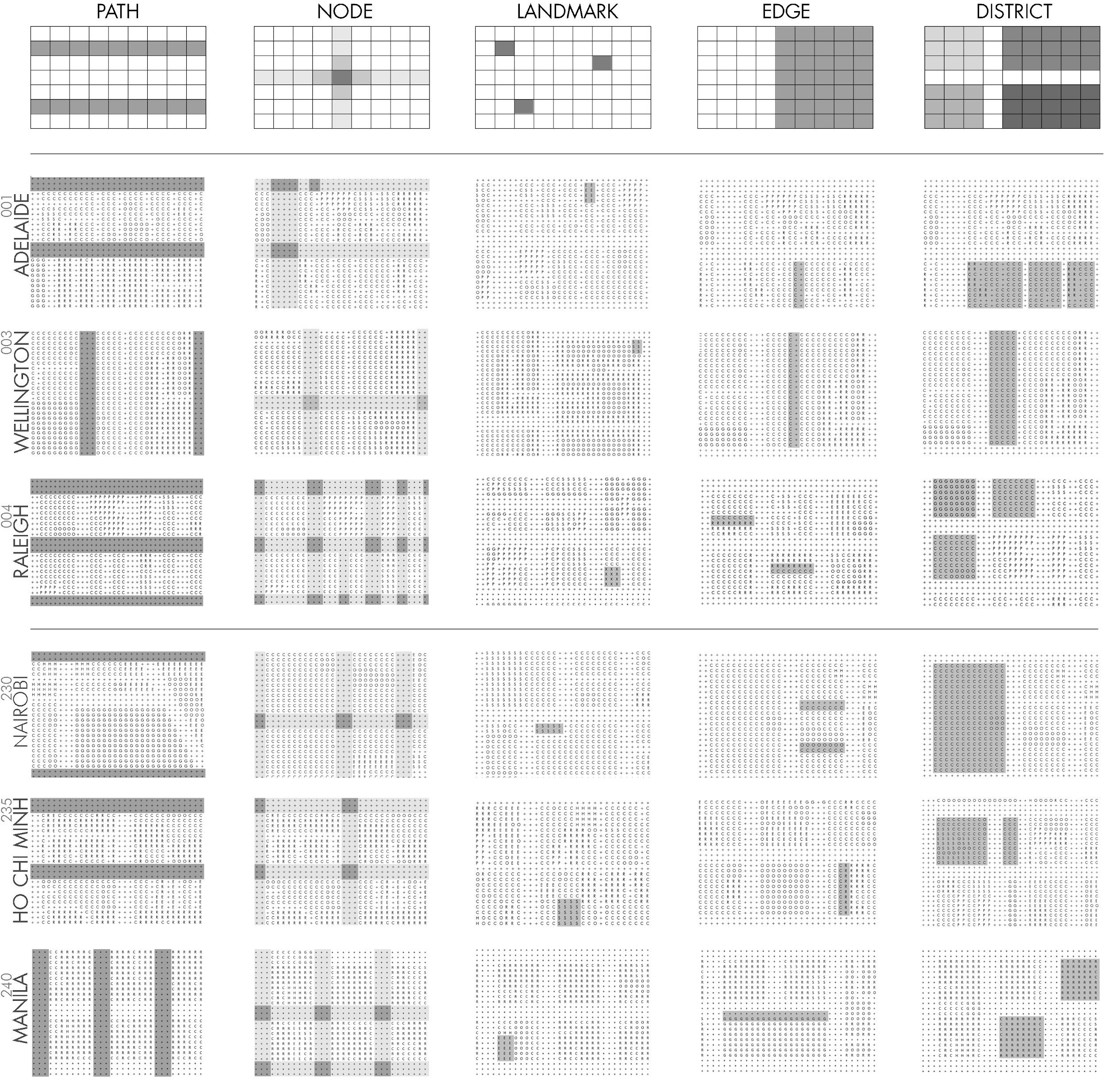


Figure 11. Samples of Lynch’s design elements were automatically discovered using the MR Engine

**5. Conclusion**

The urban design process has a crucial role in shaping our environment and cities. However, this process remains plenty of challenges in practice, and also traditional urban planning methods are failing to meet the needs of an ever urban expansion in the world. Recently, the design process has also been an important topic in AI research. Studies on this subject are about how AI will determine design decisions by computational methods, taking into account different parameters. In order to solve the problem of unpredictable urban expansion occurring around the world, AI offers various opportunities to learn what makes cities with a high quality of life and to apply it in urban expansion areas.

We investigated whether AI can contribute to developing new languages by automating the discovery of common design patterns in cities from a small number of examples. We developed a MR Engine that discovers common design patterns and applies this knowledge to new situations to generate novel conceptual designs. The MR Engine identified latent patterns within these city layouts using a set of policies and built a city vocabulary starting with the lower-level patterns that were directly observed, and then by combining these hierarchically to develop higher-level concepts that correspond to more complex design patterns. To create and analyze grid city layouts in 2D we selected six cities with high and low QoL indexes according to data on the Numbeo website in May 2021. Unlike neural networks, which require big data, our approach was able to train MR models using data from only six cities and be able to automatically detect Lynch’s five design elements in every one of these cities.

Our research roadmap consists of five phases, and this article presented the preliminary results of the first two phases, where the focus is to discover design patterns from base concepts and build hierarchies expressing more complex patterns with the ultimate goal of discovering concise city grammar that explains the data. We envision that such grammar will enable us to distinguish designs with high livability scores from the ones with low livability scores. In future works, we intend to add more complexity to the study by considering other parameters, like the location of public buildings, services, amenities, and housing, and by considering the relation and adaptation of predicting design patterns by MR Engine to its environment. In Phase 3, we will use the city grammar discovered to generate unique designs for a given empty rectangular area. In Phase 4, we will extend our approach to take into account the rest of the city (in particular the neighboring areas) to produce designs that are also compatible with the rest of the city. In Phase 5, we will enhance our approach from 2D rectangular grid layouts to 3D spatial forms.

Hundreds of new cities will be developed around the world, the fact that China, the global epicenter of city building, has built more than 600 cities since 1949 is an indicator of that. The MR Engine that was developed within the scope of this study has potential as an application area that can be used by municipalities and institutions in terms of being implemented in urban expansion zones. In addition, this chapter provided a solid basis for future work with its extensive assessment of the design-decision process with AI approaches.

**References**

Alexander, C. *et al.* (1977) *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.

Alexander, C. W. (1964) *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, Massachusetts.

Alonso, L. *et al.* (2018) “CityScope: A Data-Driven Interactive Simulation Tool for Urban Design. Use Case Volpe,” in *Unifying Themes in Complex Systems IX. ICCS 2018.* Springer, Cham., pp. 253–261. doi: 10.1007/978-3-319-96661-8\_27.

As, I., Pal, S. and Basu, P. (2018) “Artificial Intelligence in Architecture: Generating Conceptual Design via Deep Learning,” *International Journal of Architectural Computing*. United States: SAGE Publications Inc., 16(4), pp. 306–327. doi: 10.1177/1478077118800982.

Blum, R. *et al.* (2006) *Design Coding In Practice, An Evaluation*. DCLG England.

Duarte, J. P. *et al.* (2012) “City Induction: A Model for Formulating, Generating, and Evaluating Urban Designs,” in Arisona, S. M. et al. (eds.) *Communications in Computer and Information Science*. Portugal: Springer, Berlin, Heidelberg, pp. 73–98. doi: 10.1007/978-3-642-29758-8\_5.

Lima, F., Brown, N. and Duarte, J. (2021) “Urban Design Optimization: Generative Approaches Towards Urban Fabrics With Improved Transit Accessibility and Walkability,” *Projections - Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia, CAADRIA 2021*. Hong Kong: 26th CAADRIA- International Conference of the Association for Computer-Aided Architectural Design Research in Asia.

Lynch, K. (1960) *The Image Of The City*. The MIT Press.

Negroponte, N. (1970) *The Architecture Machine: Toward a More Human Environment*. The MIT Press.

*Numbeo* (2021). Available at: https://www.numbeo.com/quality-of-life/ (Accessed: May 2021).

Quan, S. J. *et al.* (2019) “Artificial Intelligence-Aided Design: Smart Design for Sustainable City Development,” *Environment and Planning B: Urban Analytics and City Science*. South Korea: SAGE Publications Ltd, 46(8), pp. 1581–1599. doi: 10.1177/2399808319867946.

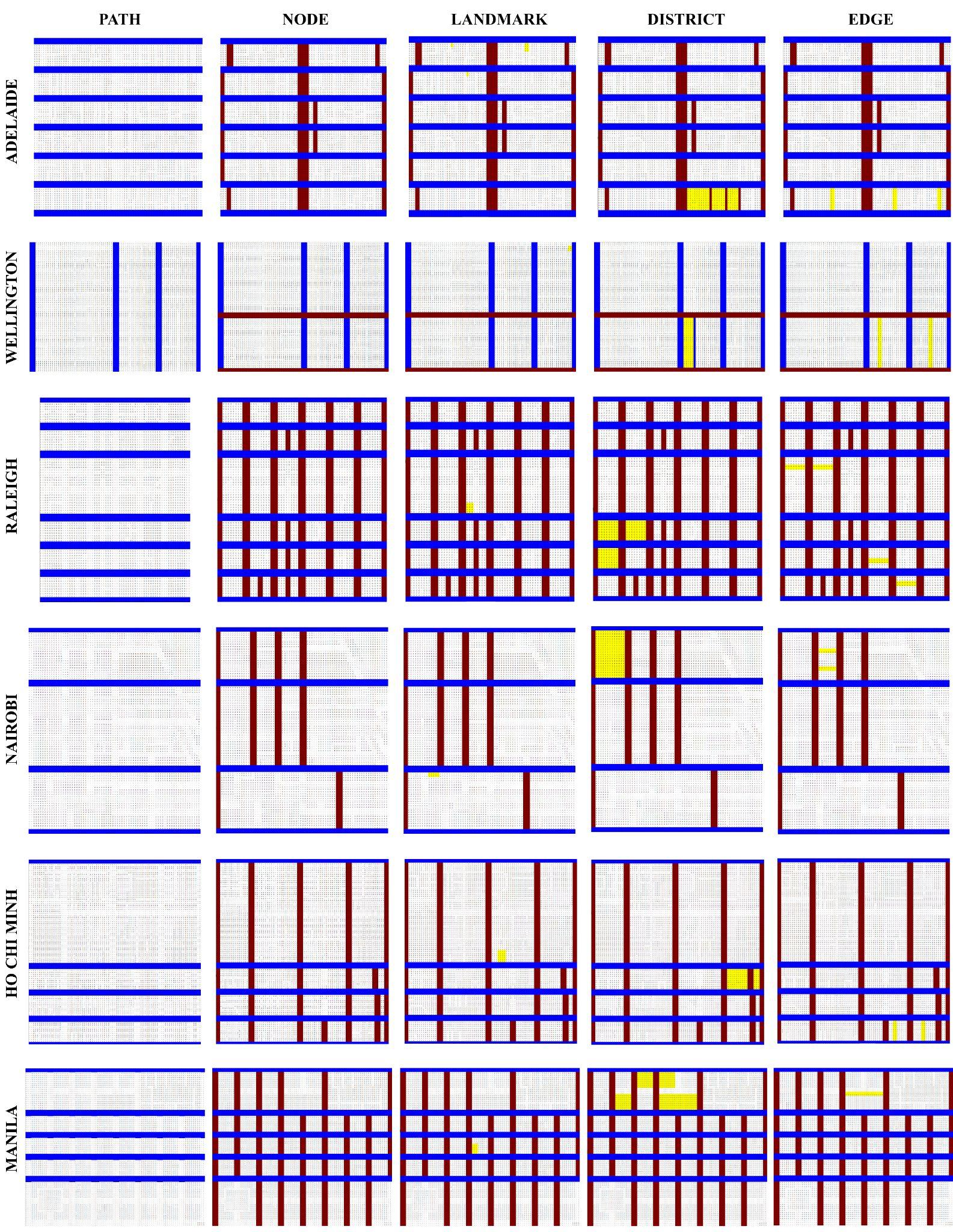
Salingaros, N. A. (2000) “The Structure of Pattern Languages,” *Architectural Research Quarterly*. United States, 4(2), pp. 149–162. doi: 10.1017/S1359135500002591.

Shepard, W. (2017) *Why Hundreds Of Completely New Cities Are Being Built Around The World*. Available at: <https://www.forbes.com/sites/wadeshepard/2017/12/12/why-hundreds-of-completely-new-cities-are-being-built-around-the-world/?sh=3839c7ac14bf>.

Traunmueller, M. *et al.* (2021) *AI-Supported Approaches for Sustainable Urban Development: Analysis of Case Studies*. Edited by T. Kaatz-Dubberke et al. Deutsche Gesellschaft für Internationale Zusammenarbeit GmbH and Austrian Institute of Technology.

United Nations (2018) *The World’s Cities in 2018*, *Statistical Papers - United Nations (Ser. A), Population and Vital Statistics Report*. doi: 10.18356/c93f4dc6-en.

Zhang (2017) “City Matrix: An Urban Decision Support System Augmented By Artificial Intelligence.” The MIT Press.

****