Machine Learning in Architecture

The new relationship between architect, image and tool

Qingling Wang

Superviser: Elin Daun
Examiner: David Andréen
AAHM10 Degree Project in Architecture, 2022
Faculty of Engineering, LTH, Lund University

Abstract:

The development of image-based machine learning technology brings new ways of how we use and produce images. Due to the significant reliance on images and drawings, machine learning technology might achieve new potentials for the architecture discipline. My thesis explored these potentials through conducting experiments. In the first part of my thesis, I used the image-based machine learning algorithm to conduct facade and church generation experiments to discover the relationship between architect, image, and algorithm. In the second part, I evaluated the quality of generated results and considered the future potential of using machine learning technology in architecture.

Content

1.Introduction:

1.1 Machine learning and artificial intelligence

Machine learning is a computer algorithm for learning specific tasks from data. The development of the Internet and electronic devices produced countless data which can be used to train machine learning algorithms into many different functions.

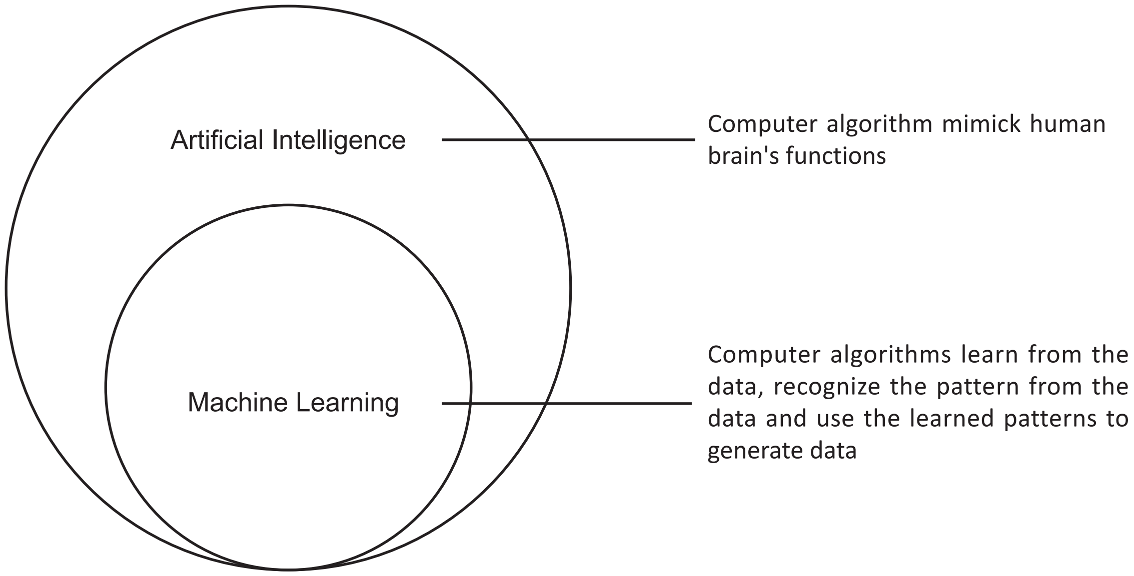


Fig1: The conceptal relations between artifical intelligence and machine learning

Looking back on history, machine learning is much related to artificial intelligence that uses the computer to simulate the function of the human brain. In the early stages of artificial intelligence, computer scientists program knowledge from a specific domain to replace human tasks. This early approach only allows computers to execute the pre-designed process, and there is no ability for self-learning compared to the actual human brain. In 1959, computer scientist Arthur Samuel raised the concept of machine learning which gives computers the ability to learn without being explicitly programmed. Arthur Samuel first used machine learning to train the IBM computer to play chess. This movement brought lots of attractions to the machine learning research domain. However, due to the limitations of hardware and computer technology, machine learning did not achieve the expected development, and a large amount of funding was withdrawn during this period. In 1997, the chess-playing program Deep Blue trained by machine learning, defected the grand-master of Garry Kasparov. This milestone regained attraction for machine learning technology and proved that computers can do a better job than humans in some domains. In the past two decades, the Internet produced countless data and it brought difficulty for the human brain to analyze each of them. In contrast, machine learning technology can use the tremendous computational ability to automatically recognize the elusive patterns from a large amount of data and then use the learned pattern to generate new data to make predictions and decisions. The benefits of machine learning are widely used in many fields, such as language translation and medical investigation.

1.2 Machine learning in architecture

In the building industry, machine learning is mainly used by software companies such as Autodesk to train their product with more powerful functionality. However, It is rare to see design offices use machine learning on design tasks. Although machine learning technology is still not widely used in architectural design, many scholars and companies have grown interested in adding machine learning to the future building process. Phil(2022) argues that the application of machine learning increases the efficiency of the building process and strengthens the architectural profession overall.

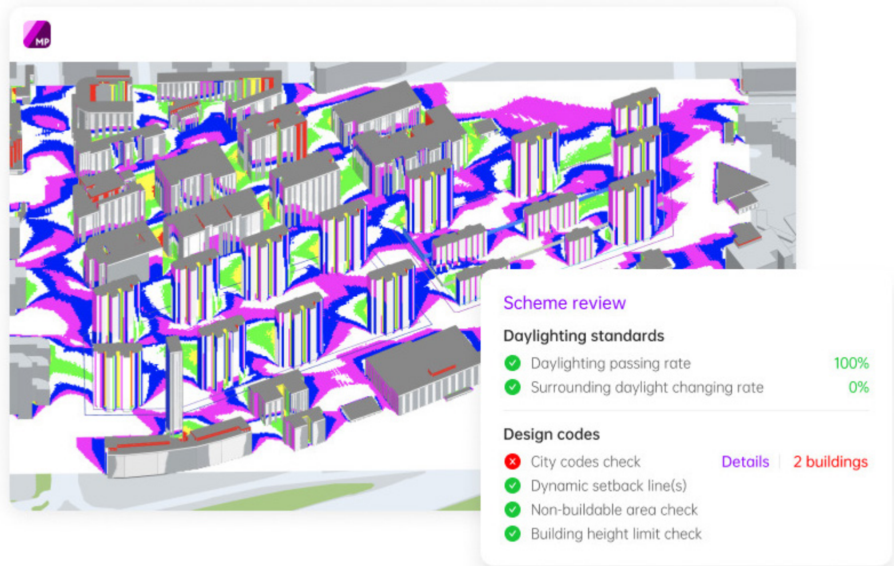


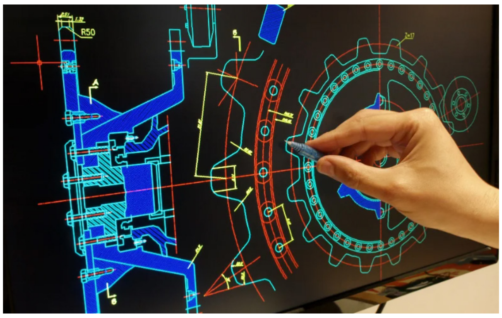
Fig2: Layout generator by Xkool
https://www.archdaily.com/974602/

In recent years, several newly established companies, such as Spacemaker AS and Xkool are developing machine learning functions for the architect. Wanyu, the CEO of Xkool, argued their goal of using computer technologies to improve efficiency and logical thinking in architectural design.

The published products by Xkool mainly focus on design analysis and design task automation. As for the design analysis, the machine learning function skips the massive calculation in daylight and CDF simulation, directly using the learned data pattern from previous cases to generate the result. As for the design task automation, the algorithm translates input parameters such as climate and building regulations into building layout strategy. The new workflows by machine learning replaced the time-consuming parts in the design process and significantly improved efficiency.

1.3 From computer-aided to computer decide

The main difference between machine learning and computer-aided design is whether let the computer make the decisions. In the current architectural process, the computer-aided design (CAD) is widely applied to shift the hand-drawing on paper into digital space. This improves the efficiency and accuracy of architectural drawing. In the CAD process, all the design decision is made by human. Drawing software only follows the command from the architect or executes a process when matching specific conditions. In contrast, machine learning allows architects obligate decisions to the algorithm. The machine learning process is a cooperation between human and computer decisions.



Human decide (CAD)

AI / ML

Trained Function

Human decide

Computer decide

human + computer decide

Fig3. Computer-aided and computer decide

1.4 Machine learning as creative tools

Since the innovation of Generative Adversarial Network(GAN) and Convolutional Neural Network(CNN), the image-based machine learning algorithms not only bring practical functionality to media industries but also are used as artistic creation tools.

GAN

Generative Adversarial Network (GAN) is a machine learning algorithm developed by Ian Goodfellow. The framework of GAN consists of a generator and a discriminator. The generator produces ‘faked images’ to mimic the dataset images during the training process. At the same time, the discriminator learns from the dataset and evaluates whether the generator is making the appropriate output. In each evaluation, the discriminator passes data to the generator to help it improve the output. When the discriminator cannot judge whether the output image is real or fake, we usually consider the algorithm recognized the data pattern.

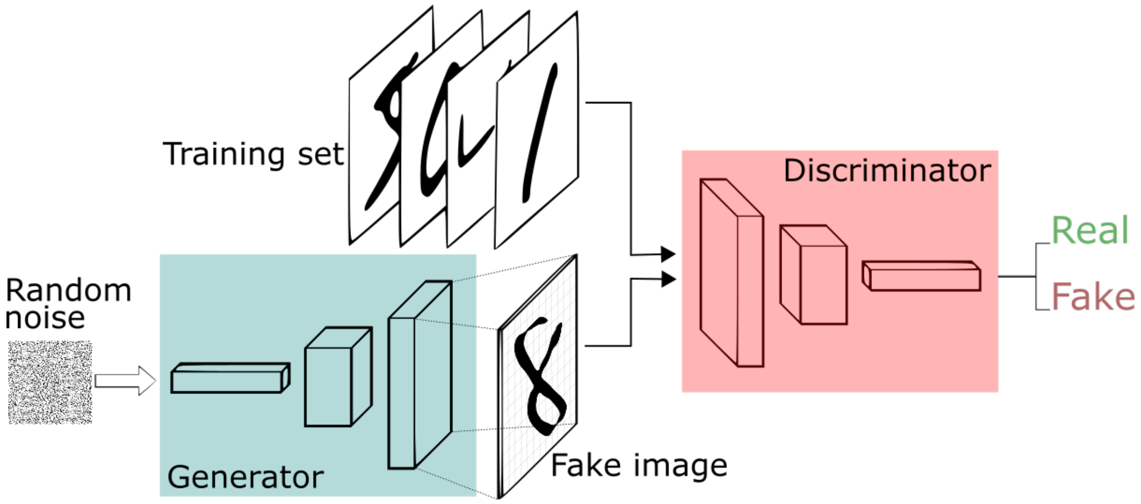


Fig4. Generative Adversarial Network
Generative Adversarial Networks, Ian J. Goodfellow

StyleGAN

In recent years, GAN became drive force for using machine learning in artist creations. The most famous derivative of GAN is StyleGAN by NVIDIA researchers. In 2018, the artistic group Obvious collected 15000 portraits made between 17 to 20 centuries as the dataset for StyleGAN. After the training, StyleGAN used learned principles from collected portraits to generate the famous artwork named by Edmond de Belamy and sold for 337,000 euros.

Fig5. Edmond de Belamy, art creation by machine learning
https://www.christies.com/img/LotImages/2018/NYR

In the StyleGAN process, the role of the art creator only takes responsibility for collecting and classifying the image. There are no interactions between creator and algorithm when generating the result. Compared to the traditional art creation, the result by StyleGAN is not directly influenced by drawing skills from the creator.

Image translation

The Vincent AI is another derivative of GAN that translates one outline drawing into a painting. Unlike StyleGAN, Vincent AI allows the interaction between creator and algorithm. The creator decides the content and uses outline drawing to represent human decisions. Based on human input, the algorithm decides the rest information, such as the color and atmosphere. To train the Vincent AI, the dataset contains two parts. The first part uses VanGogh's painting as the desired output, while the other is outline drawings that label the content on the desired outcome. During the training process, machine learning algorithms find the data pattern between input and desired output, then use the learned pattern to translate the input into painting. The achievement of Vicent AI demonstrates that the learned data pattern by machine learning can reflect artistic factors such as style and principle, then dress those factors to produce new paintings.

Monty Barlow, machine learning director of Cambridge Consultants, argues that the interactivity of Vicent AI takes the germ of a sketched idea and allows the history of human art to run with it.

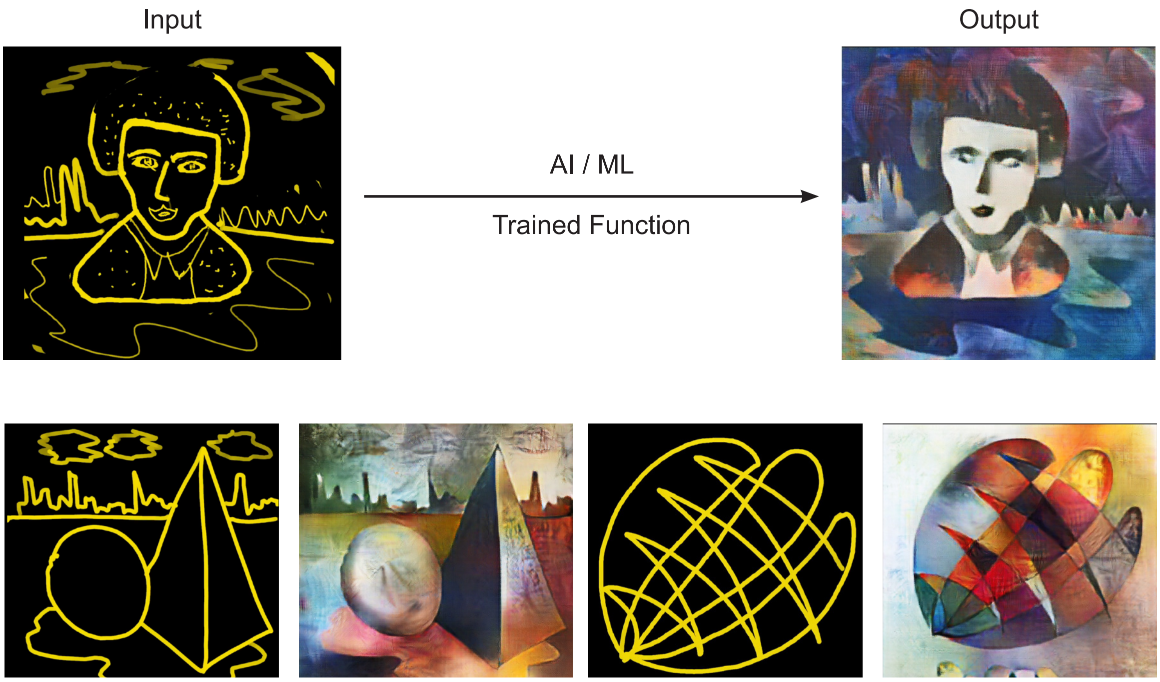


Fig6: Vincent AI
https://blogs.nvidia.com/blog/2017/10/11/vincent-ai-sketch-demo-draws-in-throngs-at-gtc-europe/

In the architectural research, Zheng (2020) tested the image translation algorithm on the residential plan. Zheng's experiment demonstrated the possibility of using machine learning as a decision-making tool at the early stage of architectural process.

Fig7. Using Chinese residential plan to predict
Apartment floor plans generation via generative adversarial networks, Hao Zheng

CNN

Convolutional Neural Network (CNN) is an artificial neural network used to train the image analysis functions. Computer technologies such as facial recognition and medical image diagnosis are related to CNN. In 2015, computer scientist Leon Gatys took an artist approach to develop Neural Style Transfer(NST) based on CNN.

The achievement of Leon Gaty is he found a mathematical way to separate the content and pattern from feature map layers. The NST algorithm recombines the content and pattern from two different images into a new synthesized image. Leon Gatys(2015) argues that the visual experience of painting is achieved through the interplay between content and style. The NST provides a new approach to interplaying the content and style by neural representation.

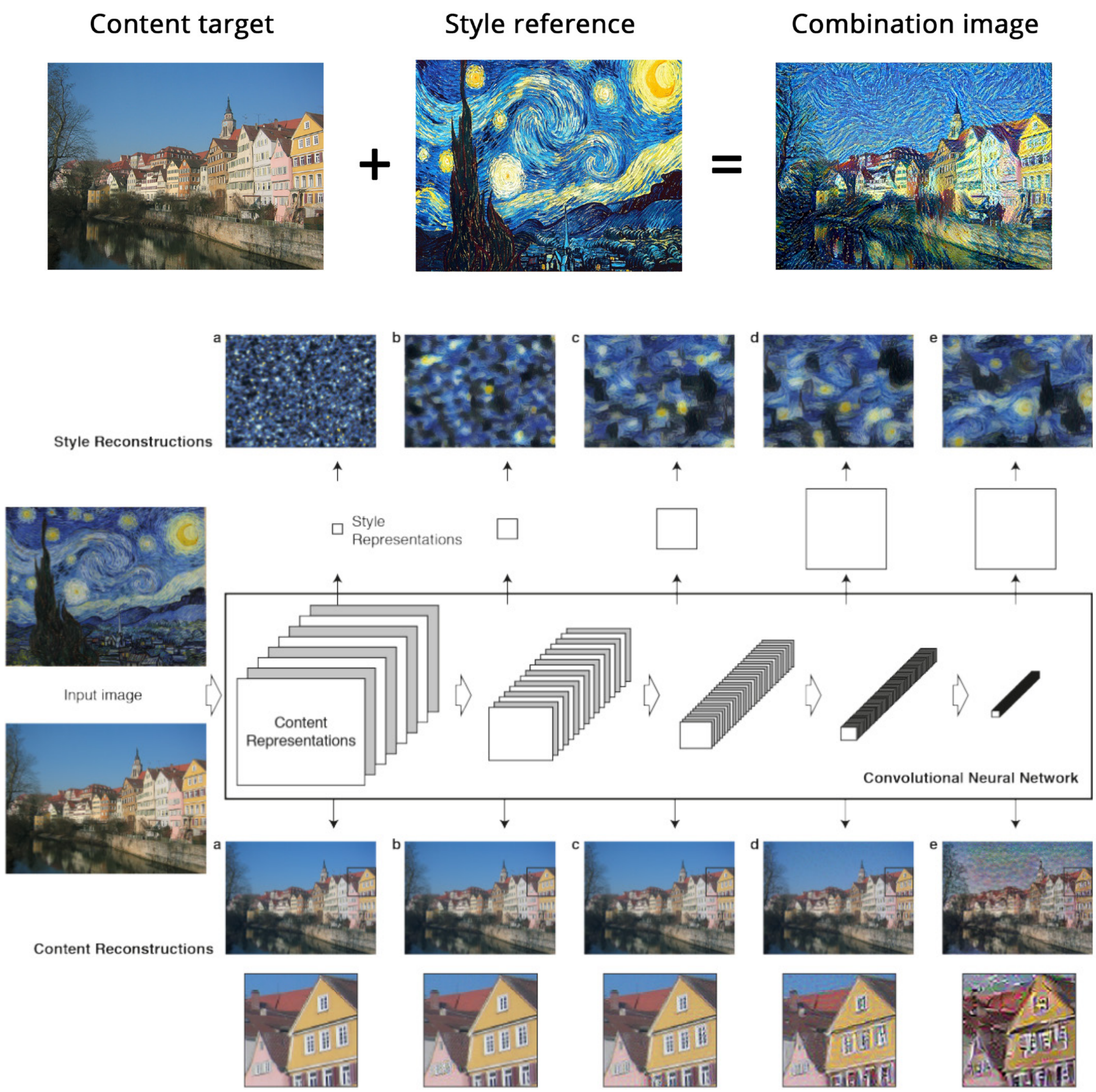


Fig8. Netural Style Transfer
A Neural Algorithm of Artistic Style, Leon A. Gatys

In architectural research, Matias(2020) suggested CNN as a tool to explore the style issues in architecture. He used historical perspective to demonstrate the cultural and technological presence in the style that drove the architectural innovation. Meanwhile, Matias dressed the artificial intelligence perspective on the style that using CNN and GAN to capture style or features from historical matter and assist architects in finding new architectural possibilities.

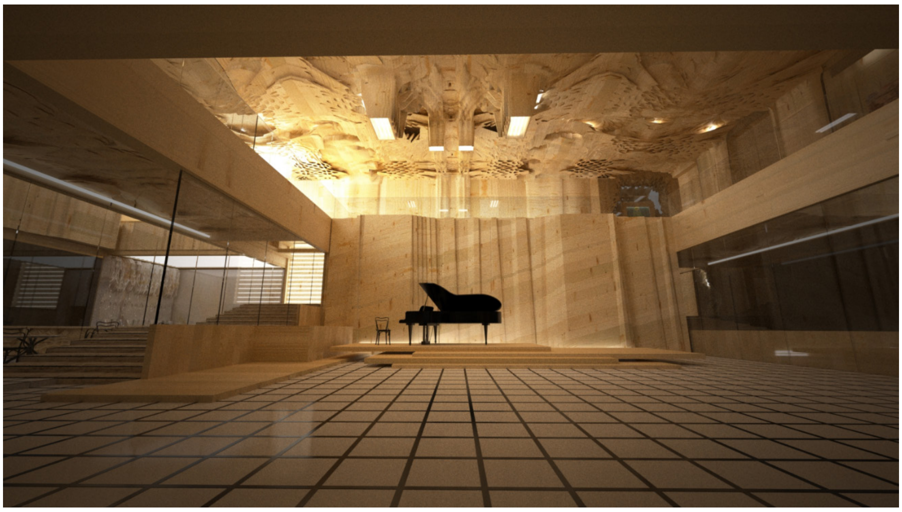


Fig9. Austrian Pavilion for the Dubai Expo 2020
A Question of Style, Matias del Campo

2.Research questions and method:

The first research question relates to the relationship between architects, images, and machine learning tools. Bringing image-based machine learning technology into architecture might influence how architects collect, use, and produce images. I aim to define the role of each agent in this process.

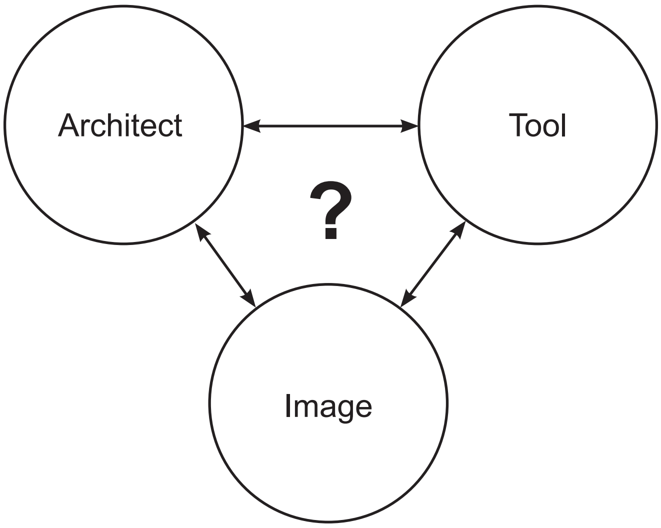


Fig10. The relationship between architect, image and tool in machine learning process.

The second research question is about the cooperation between human and computer decisions. Zheng(2020) demonstrated the possibility of training the algorithm into a design decision tool. However, Due to the input image in Zheng's experiment only labeled the boundary, the content on generated plan is uncontrollable. It might be impossible to obligate all the plan decisions to the computer. So I aim to explore how to make the algorithm more controllable to approach the desired outcome.

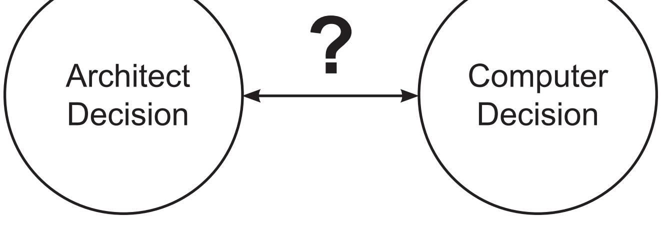


Fig11. The cooperation between human and computer decision

The third question is related to the future development of machine learning in architecture. This current machine learning application in the architecture industry mainly concentrates on problem-solving to improves efficiency and accuracy. Spatial factors such as qualities, typologies, styles, and design principles have not been heavily involved. I aim to take the problem-worrying approach to use machine learning to address those factors and seek future possibilities. In addition, developing image-based machine learning in architecture might need more cooperation between architects and computer scientists. I seek to explore the limitations and deficiencies of image-based machine learning algorithms in architecture so that computer scientists can know how to optimize algorithms and make it easier for architects to use machine learning.

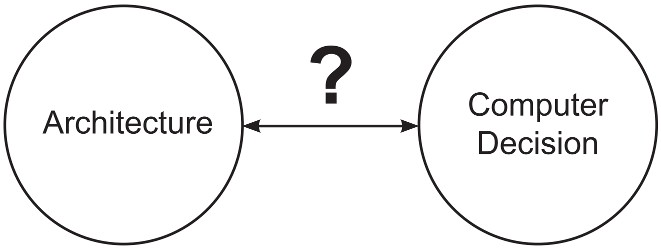


Fig12. The future potentional and development of machine leanring in arhcitecture

Method

To explore those questions, I conducted experiments to train the machine learning algorithm into facade image and church plan generation tools. The process of experiments helped me to gradually approach the answers.

3.Experiment:

3.1 Pix2pix facade generation

The first experiment explored the relationship between image and tool, specifically concentrating on the technical issues of how to train an image translation function and how the dataset influences the result. The dataset in this experiment is provided by Phillip's research(2017) which is used to train the facade generation tool. This dataset consisted of 600 pair images of the building facade. The left side of the dataset is the desired output, while the input image uses different colors to label the content of the desired output. The Pix2pix algorithm automatically calculates the data patterns between the input and target output image during the training process. It then uses the learned data pattern to translate the input color map into a facade image in the generation process.

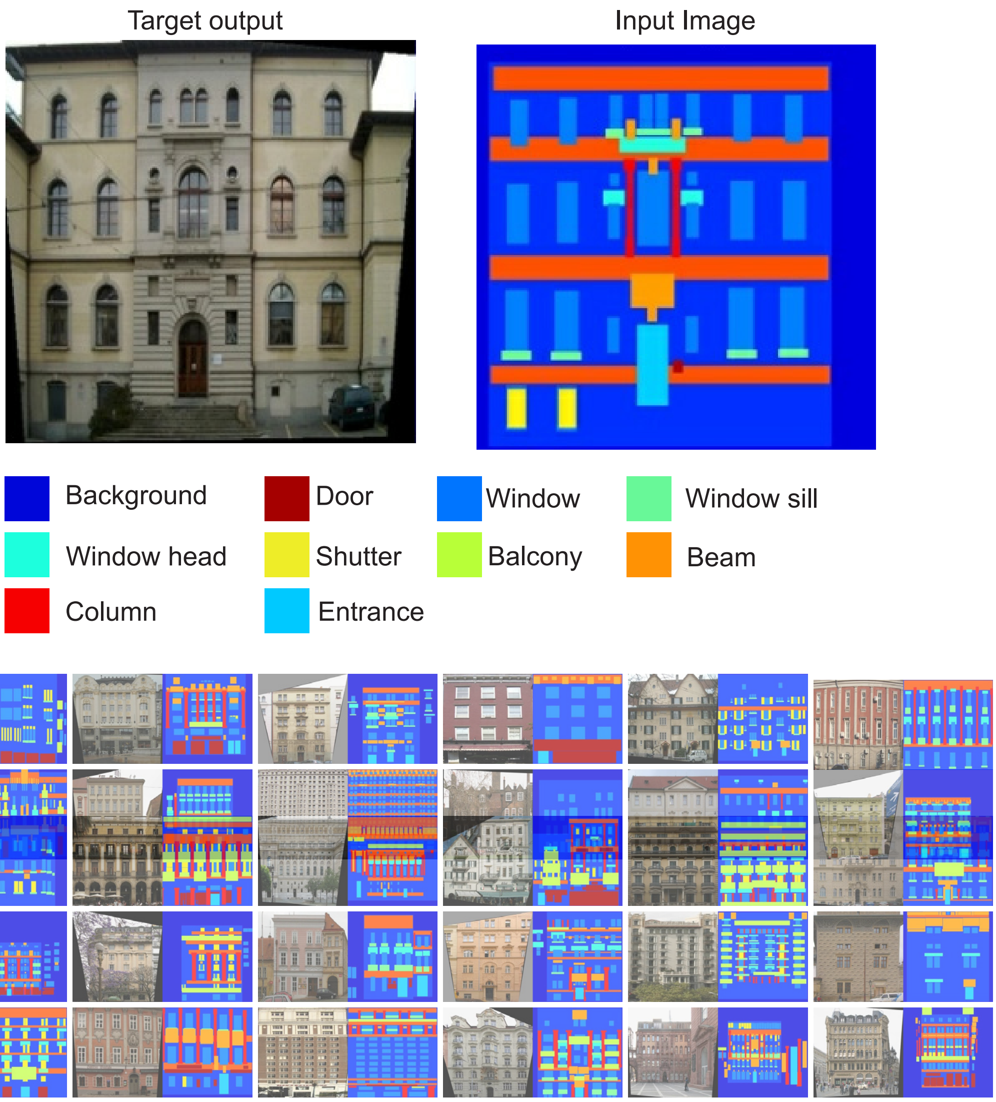


Fig13. Dataset
Image-to-Image Translation with Conditional Adversarial Networks, Phillip Isola

Training

The training process showed a gradual improvement that the generator constantly increases the quality and resolution of the output image. As for the result, this test accomplished 30000 steps of training within 20 minutes. The generated results could be identified as a facade image.

Fig14. Training process

Testing

Several color maps out from the datasets were input to the algorithm to evaluate the learning outcome. Although some of the generated results had a low resolution, they still can be identified as facade photos.

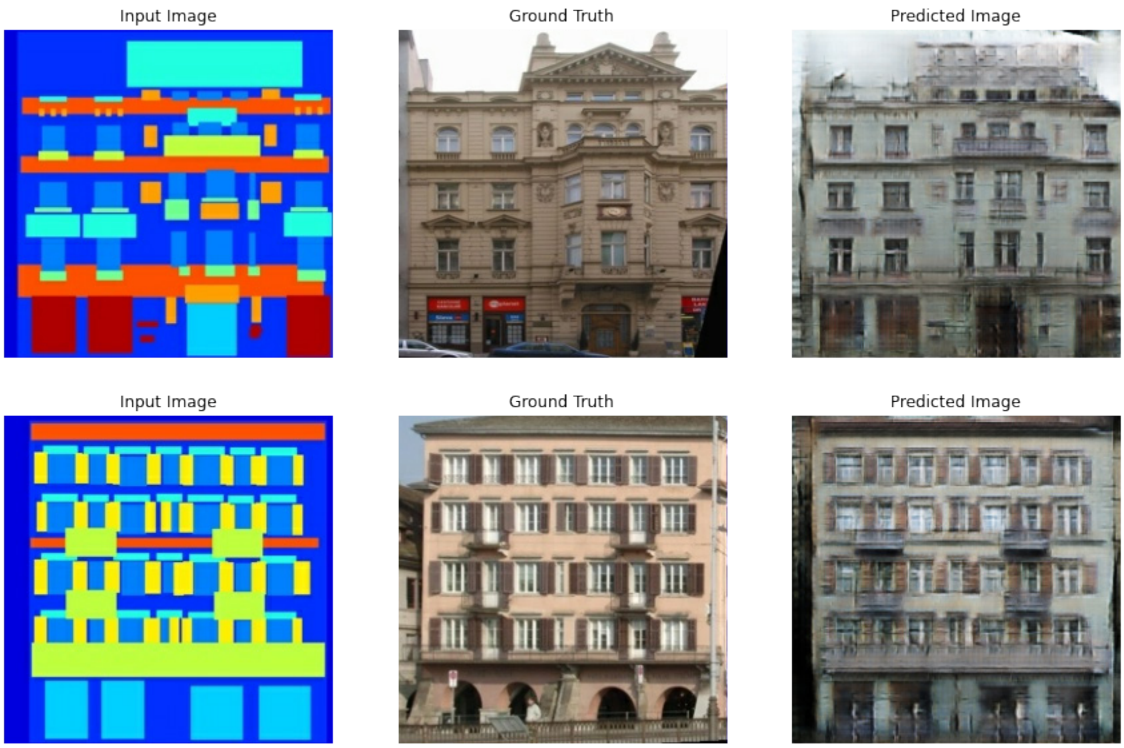


Fig15. Result by testset

Define the human and computer decision

By comparing the input and output, the labeled information such as window size, beam length, and door locations are precisely translated into output. In contrast, unlabeled information such as colors and materials are decided by the computer. This fact showed that the weight of human and computer decisions is definable by dataset. The unlabeled information in the dataset represents the computer's decision, while the labeled information in the dataset presents human decisions.

In the generation process, the human agent cannot directly control the unlabeled information but decide or modify the labeled information on input to define the content of the output.

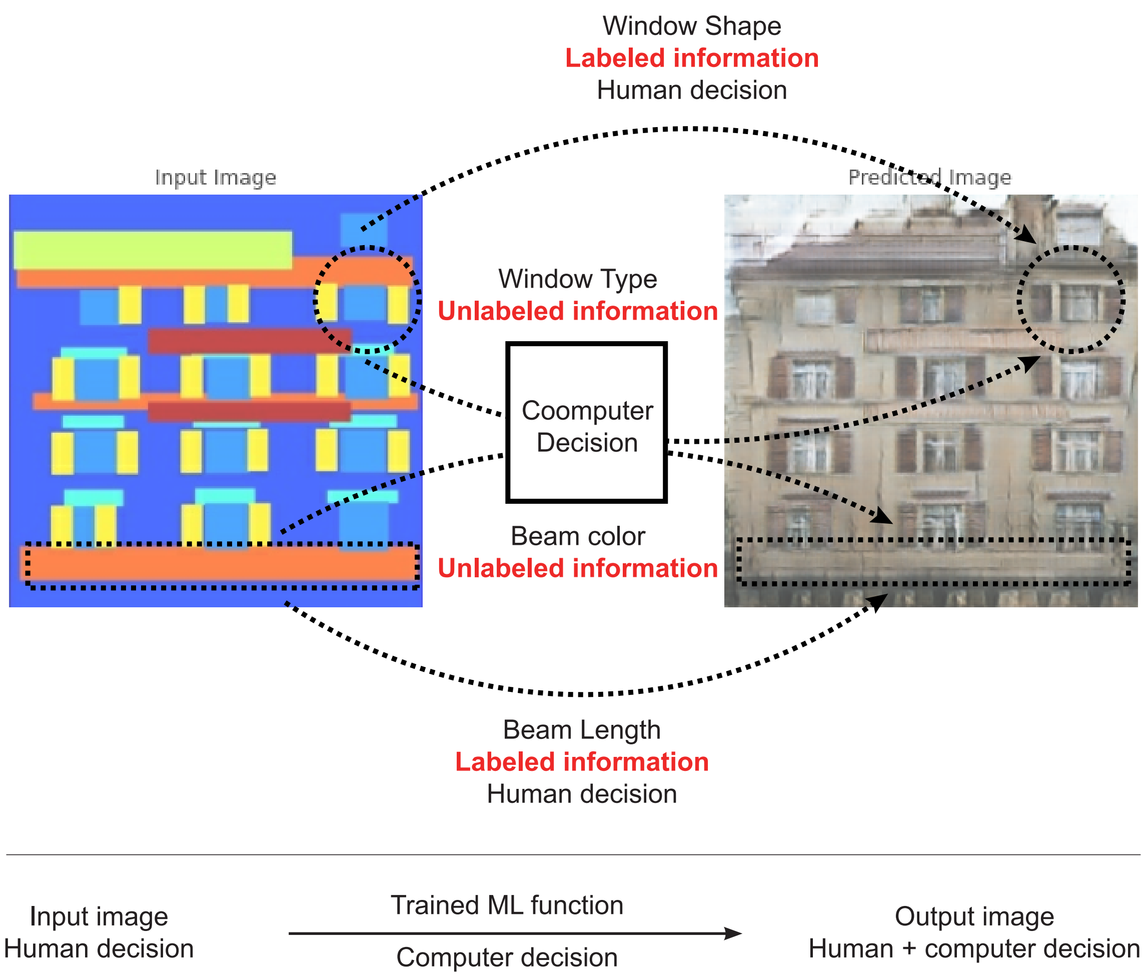


Fig16. The way of defining human and computer decision

Transformation

This step arranged the labeled information in an unconventional way to test the trained algorithm. Although the arrangement of labeled information is much different from the facade, the algorithm still translated the input into faked facade image. This test suggested a transformative approach to breaking through the original principles of input and then using trained algorithm to explore new things.

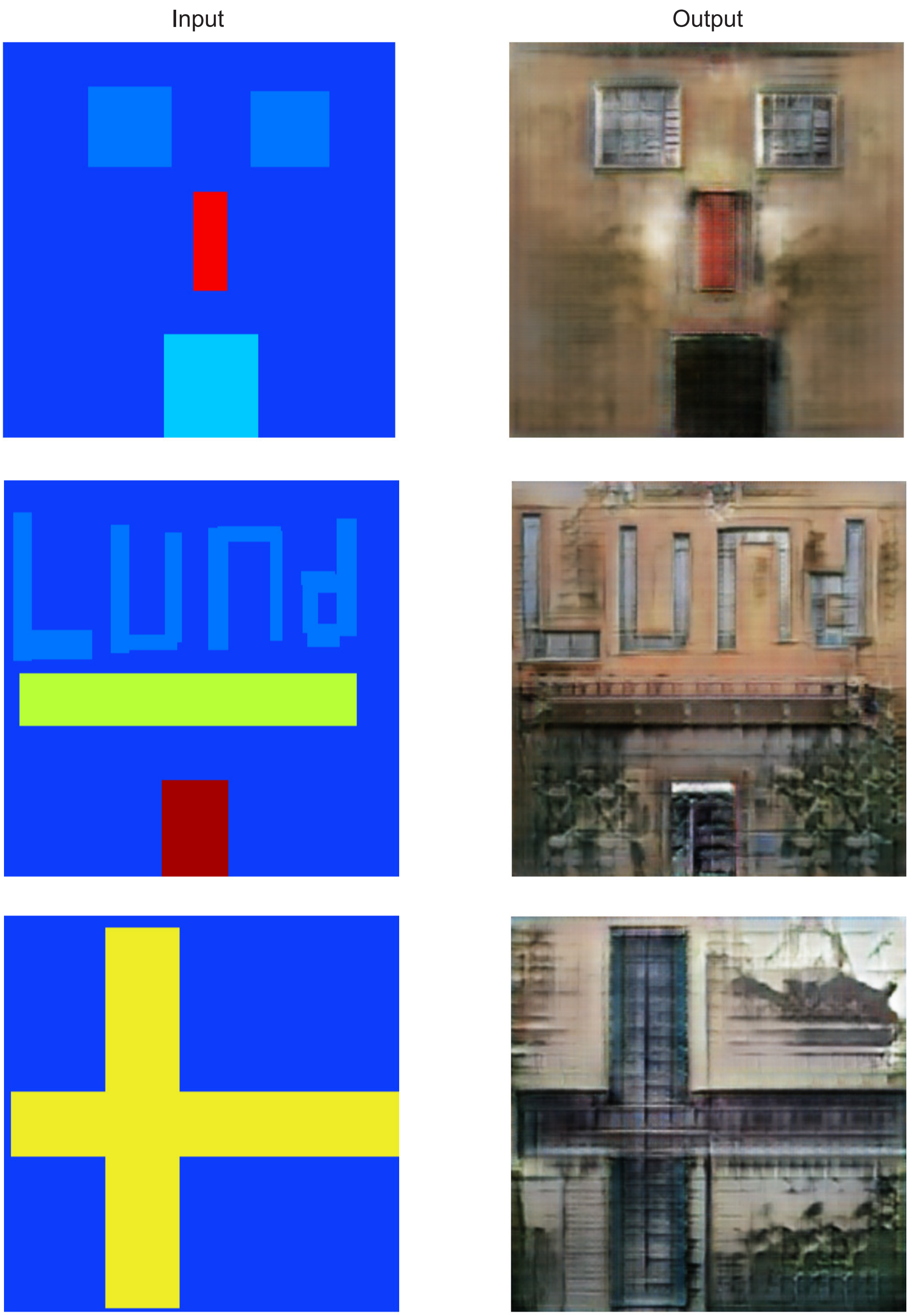


Fig17. Generation test

3.2 Pix2pix Church plan generation tool

The second experiment trained the Pix2pix algorithm into a plan generation tool. This experiment focused on three aspects: (1) The way of collecting and making dataset. (2) Whether the learned data pattern could reflect architectural design principles. (3) The transformative ways of using learned data patterns.

This experiment used 300 old church plans as image sources to produce the dataset. The reasons for choosing old church plans are the consistency and complexity of their design principles. As for consistency, the church has experienced a long history that concluded specific design methodologies for this building type. Whether it is in different geographical locations, the church plan in the western world shows morphological similarities. This might bring the possibility of using machine learning to find patterns from old church plans. As for the complexity, old church plans are typically designed in a mathematical manner that coordinates all the plan elements into geometrical relations; this mathematical approach results in the qualities of proportion, circulation, acoustic, and structural properties. However, the modern architectural practice rarely uses that mathematical method to design the plan. Machine learning might bring new ways of utilizing the church plan design principles in modern architecture and generate spaces with good qualities.

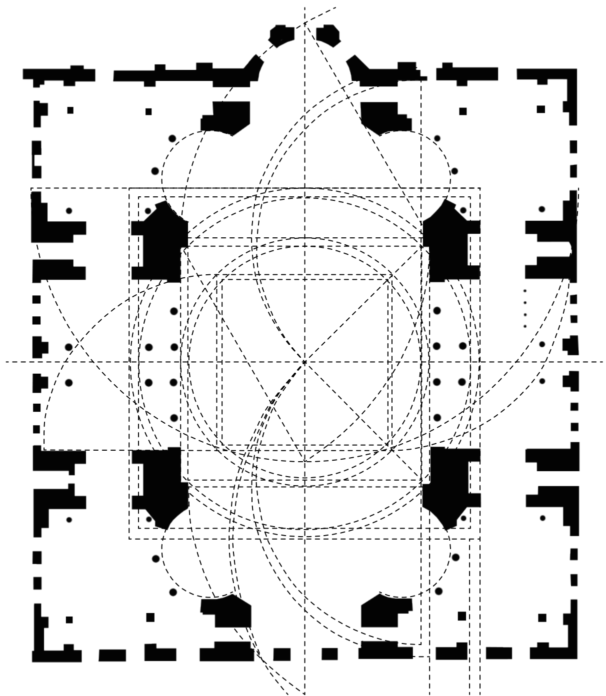


Fig18. Complex geometrucal relations on church plan
Geometry, Light, and Cosmology in the Church of Hagia Sophia
W. Jabi, Iakovos Potamianos

Dataset

The church plans used in this experiment are collected from the Internet. They mainly consisted of the cathedral, basilica, and temple in the Roman period. Since the scale of the temple is much smaller than the basilica and cathedral, it shows good resolution on 512*512 images. In contrast, due to the long and narrow geometrical features of the cathedral, interior elements such as columns do not have good resolution. To process the church plans into the dataset, the first thing that I have considered is which part should be decided by myself and which part should obligate to the computer. From the collected plans, information such as boundaries and entrance are apparent, I use them as labeled information on the input color map to let the computer find patterns. Other information such as ratio and structural layout is more elusive that I obligate them to computer to decide. By rotating and mirroring the asymmetric plan, this dataset is enlarged into about 1000 images.

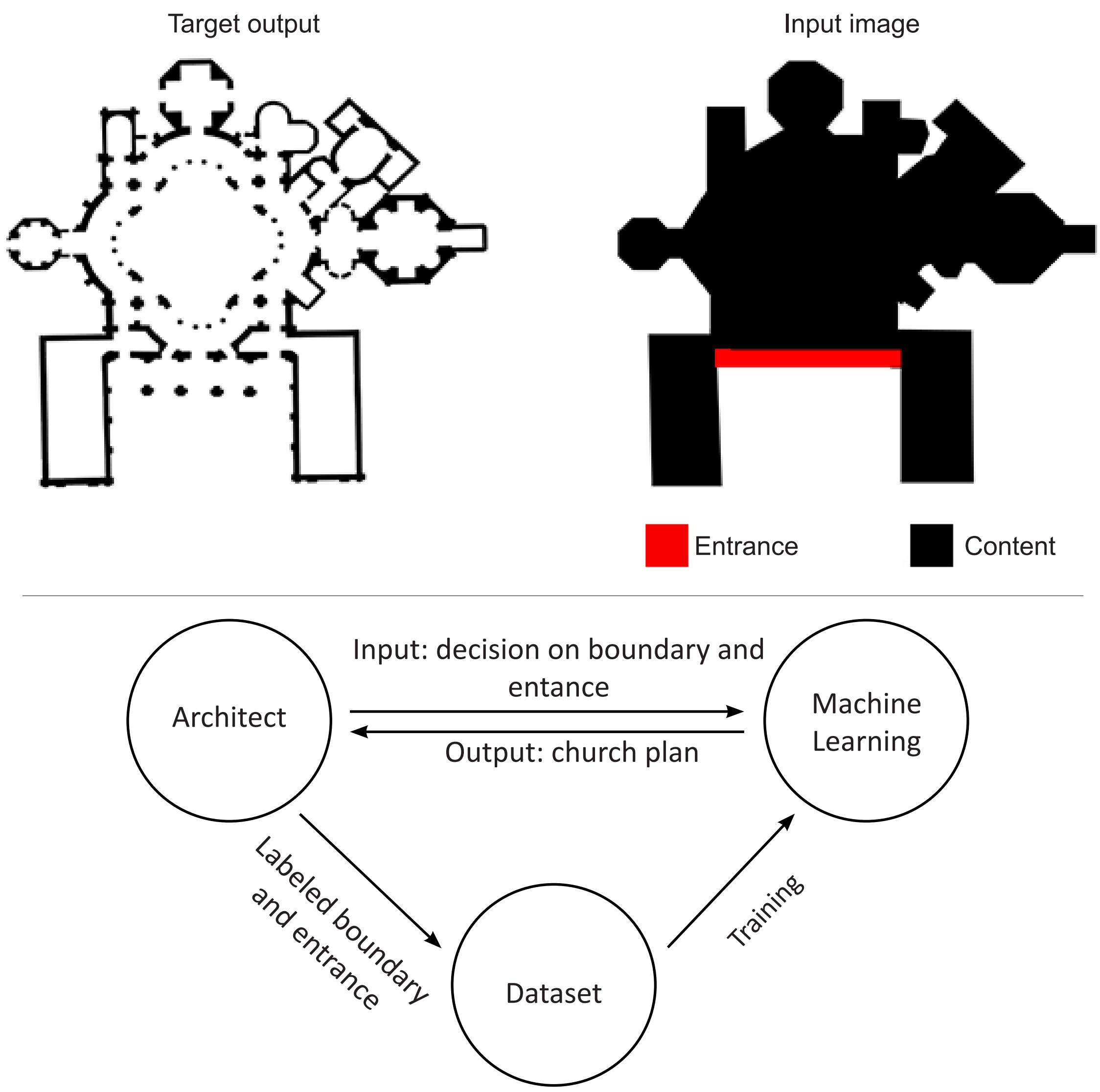


Fig19. The way of processing dataset

Training and testing

Through 50000 steps of training by pix2pix, the generator can translate labeled information into output. However, the unlabeled information such as walls and columns shows insufficient resolution and cannot be identified as the church plan.

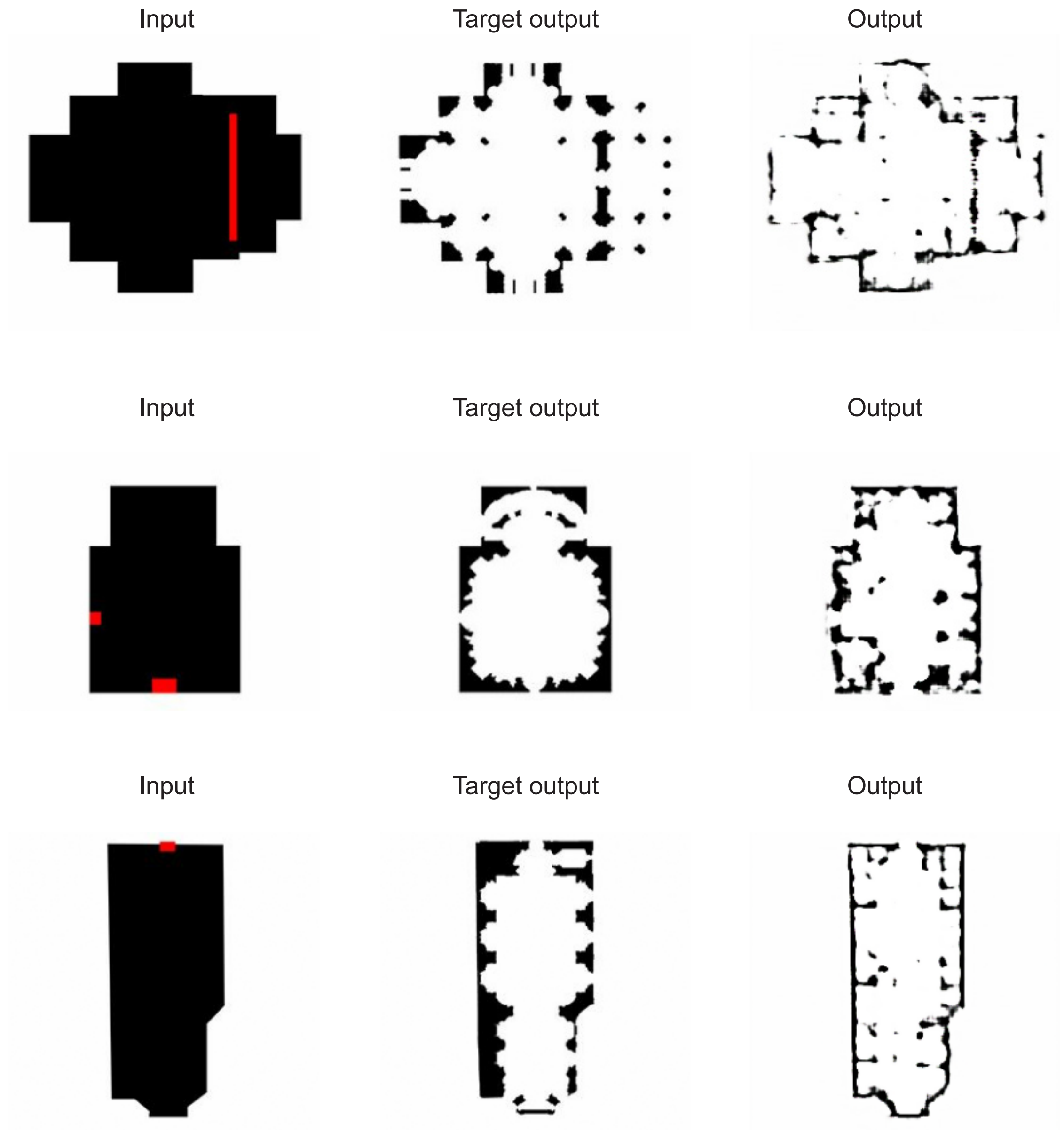


Fig20. Training process

Iteration 1

The church window is labeled on the input image to test if this can help the algorithm better recognize the data pattern between input and output. As for the result, the output in this iteration showed many similarities with ground truth, but the resolution of structural elements was low.

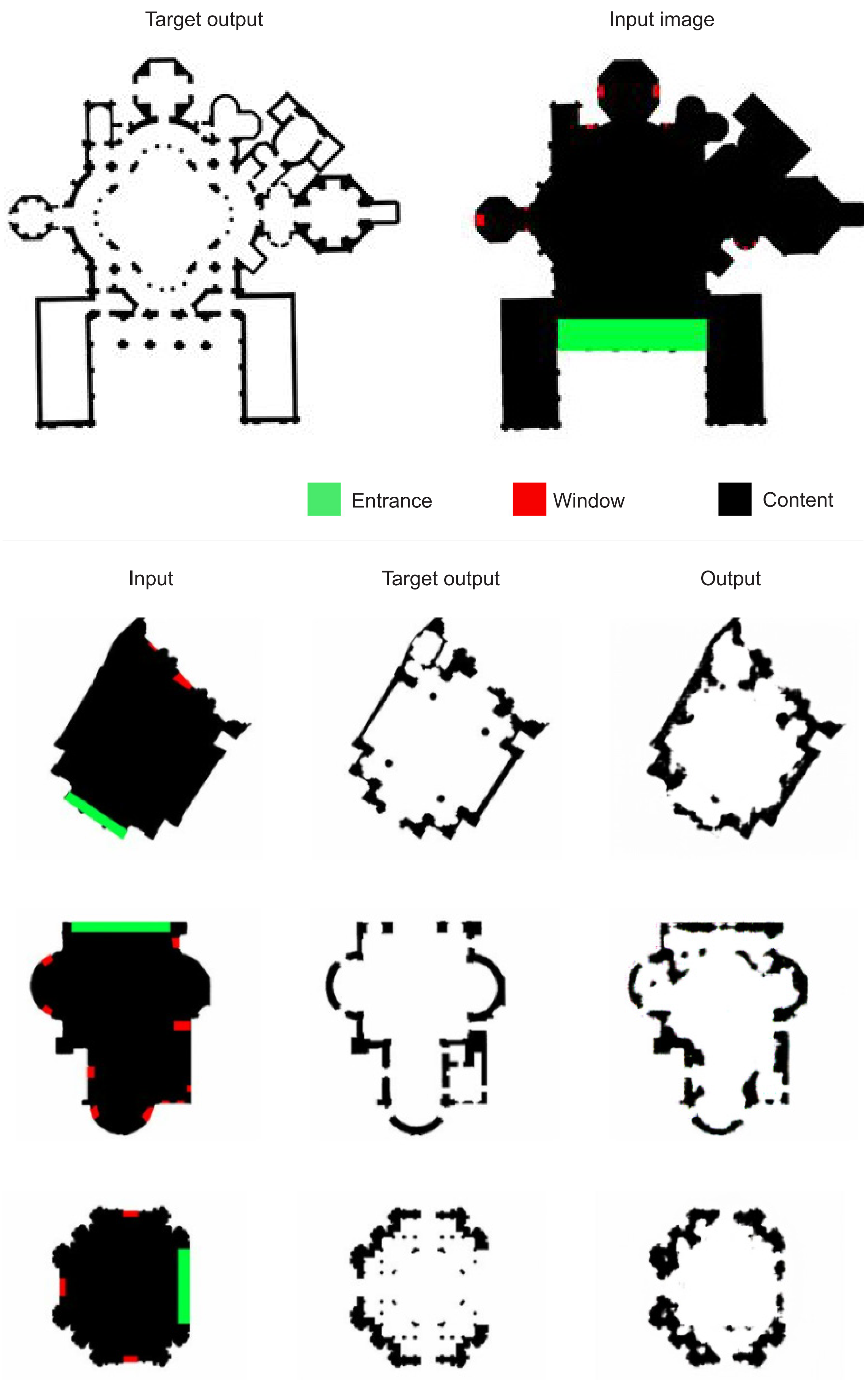


Fig21. Training process by updated dataset

Iteration 2

The nave on church plan normally has a large structural span and significantly influences the layout of structural elements. This iteration labeled nave on the input to improve the structural layout. The generated output in this iteration showed huge improvement that could be identified as church plan.

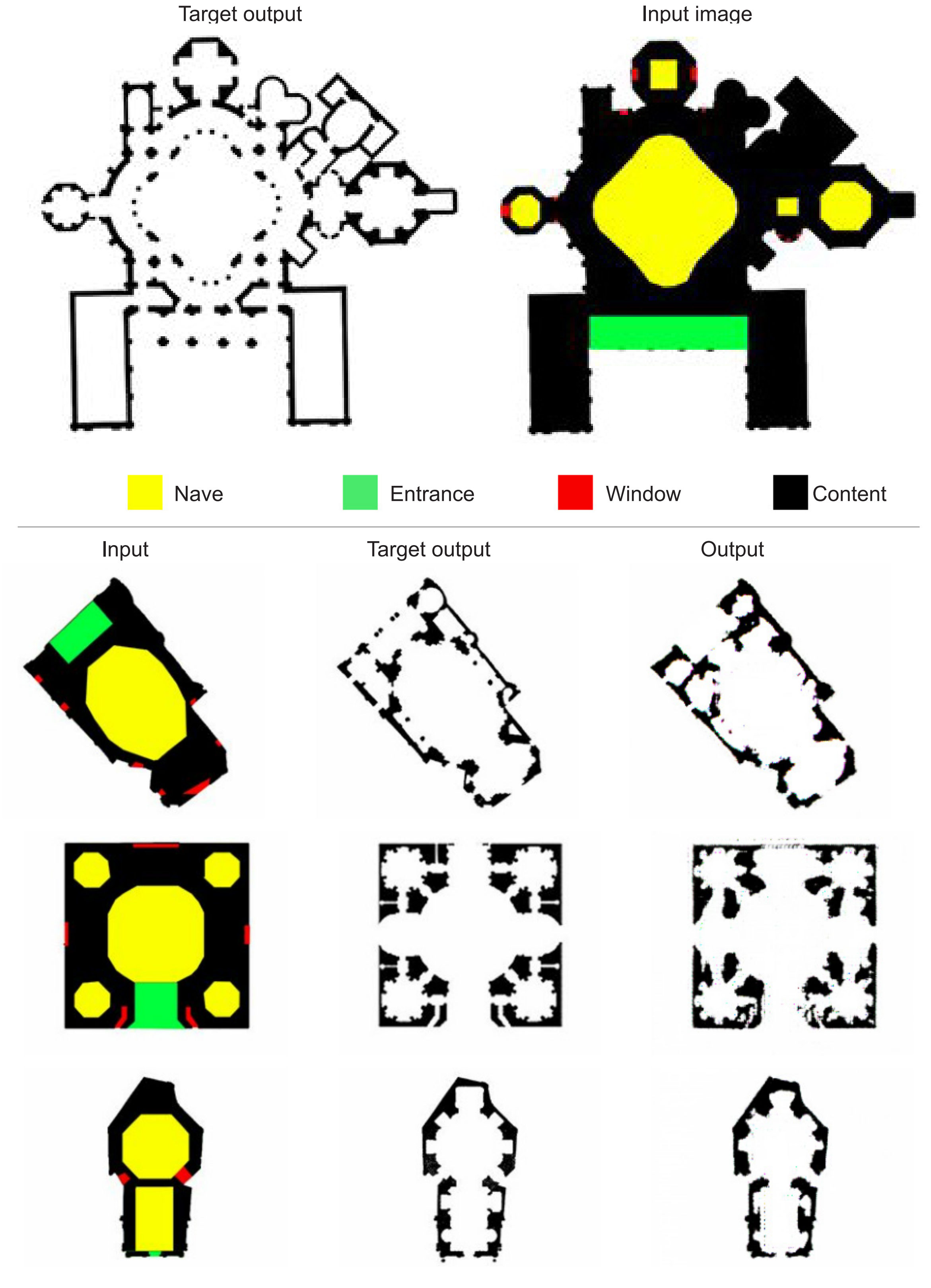


Fig22. Training process

Generation

This step used input out of the dataset to test if the algorithm could generate the appropriate result. The input image mainly includes the zoning maps of the temple and basilica but also added several randomly made images to test if the output could show church plan features. From the output image, many of the temple zonings were translated into high-resolution output, while the results by basilica zoning had insufficient resolution.

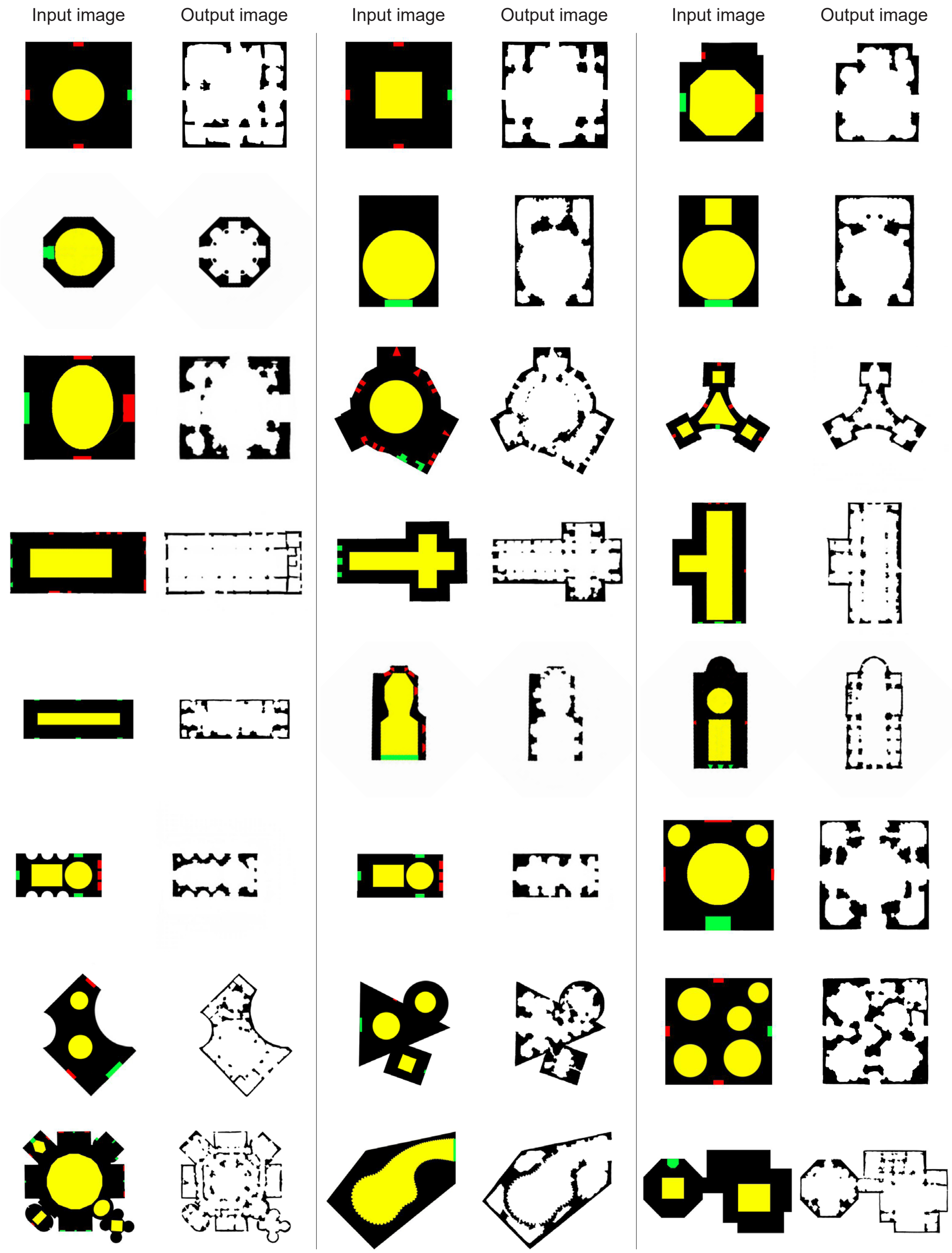


Fig23. Generation process

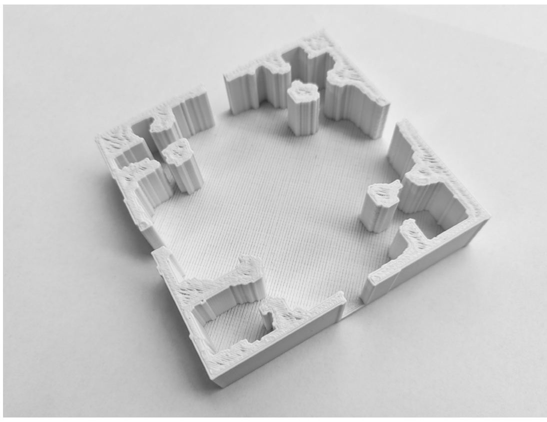
Evaluation
The input images on the right page are similar to the temple, in which nave takes up large central areas. The output had good resolution and presented features of the temple plan.

Case1.1 did not show the geometric symmetry, but the spatial proportions of sub-space could be coordinated by a grid diagram. The input image of **case1.1** labeled a circular nave at the center, while the output did not show the configuration of a circular nave. As for the structural elements, the layout of columns and walls did not obey the structural principles of the dorm.

The result of **case 1.2** showed many features of the temple. The nave space from the input was translated into output, and the arrangement of the interior wall could resist the thrust force from the nave. In addition, the corner space on the right side had several chapels, which is similar to the temple, and the grid line shows the symmetry of spatial proportions.

Both **case 1.3** and **case 1.4** showed geometrical symmetry on the plan. The chapels were arranged on four concerns around the nave, which reflected the design principles of the temple.

The input of **case 1.5** was produced based on Louis Kahn's project. The input color map does not show any geometric symmetry. As for the output, the edge showed good resolution while the interior elements were unclear.



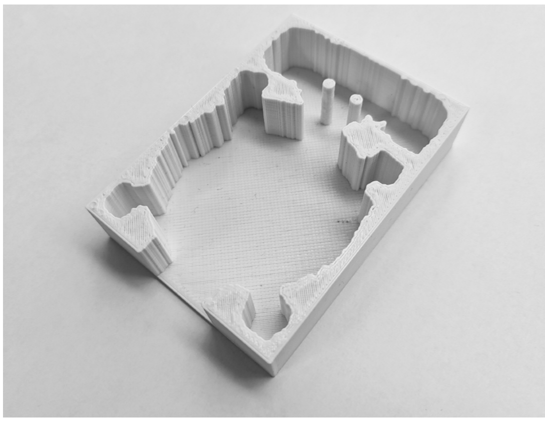


Fig24. Model of case 1.2 and 1.4

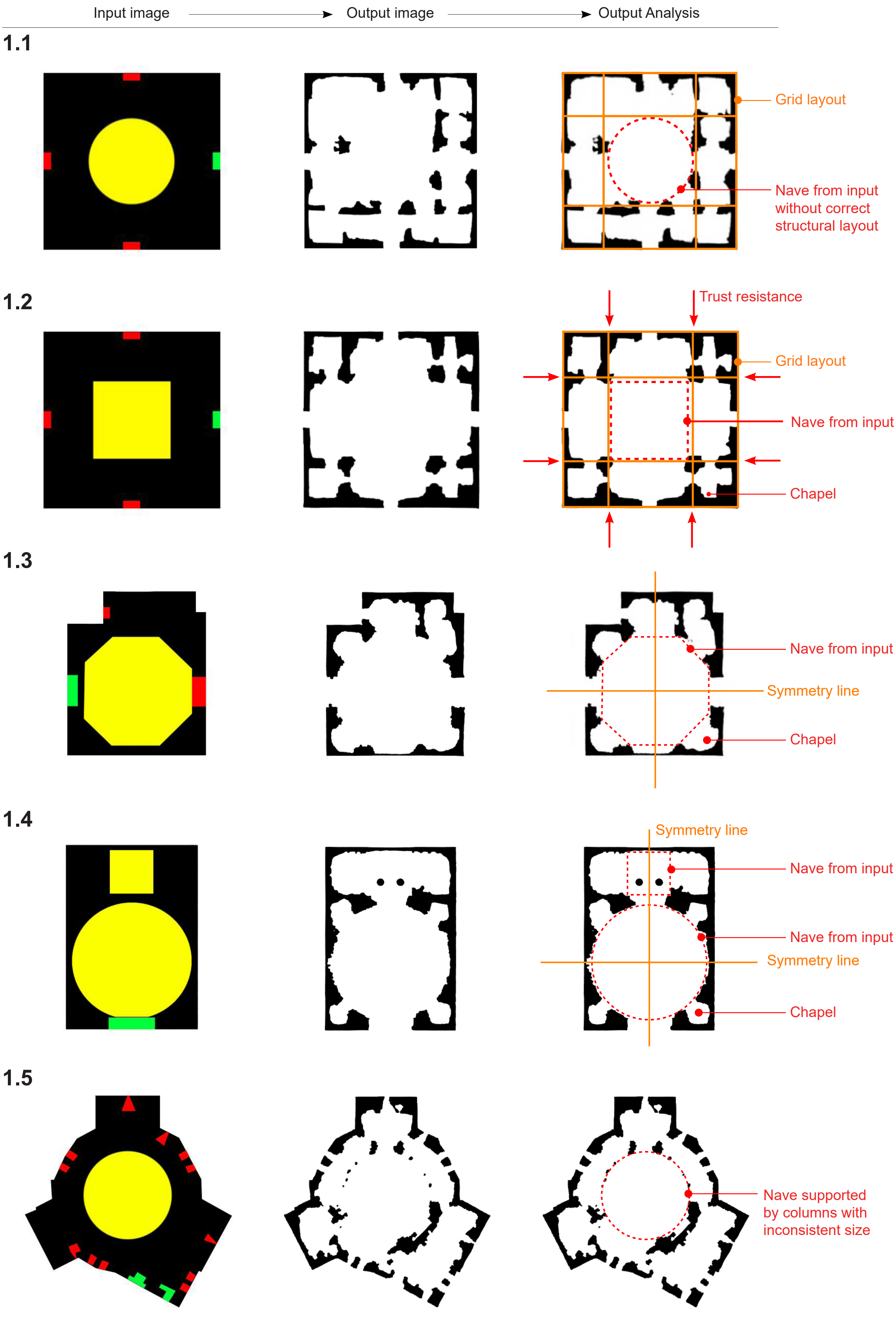


Fig25. Temple set with big central nave

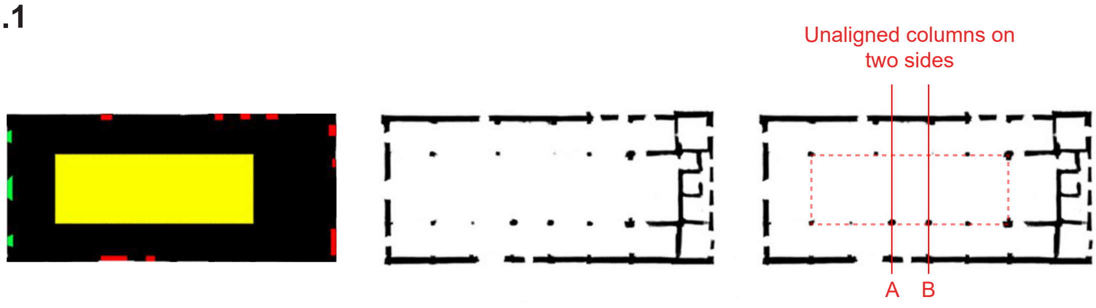
The input color map in this testset refers to the basilica which has a long nave. The columns on all the outputs had bad resolution. This might be due to the large scale of the basilica, which makes the columns have limited pixel information in the dataset. The machine learning algorithm might not be capable of recognizing the data pattern of columns.

The output of **cases 2.1, 2.2 and 2.3** showed plan features of the basilica in which the columns were arranged through the long nave. However, the columns (e.g. see A and B) were not aligned on two sides and had the inconsistent scales.

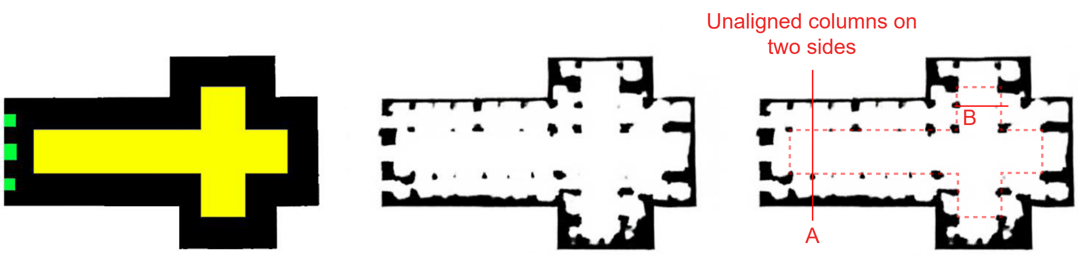
Case 2.4 and 2.5 had a relatively higher resolution than the others. The columns of **case 2.4** correctly aligned on both sides of the nave. In contrast, the nave of **case 2.5** is overhanded at the center which is impossible by stone structure.

Input image → Output image → Output Analysis

2.1

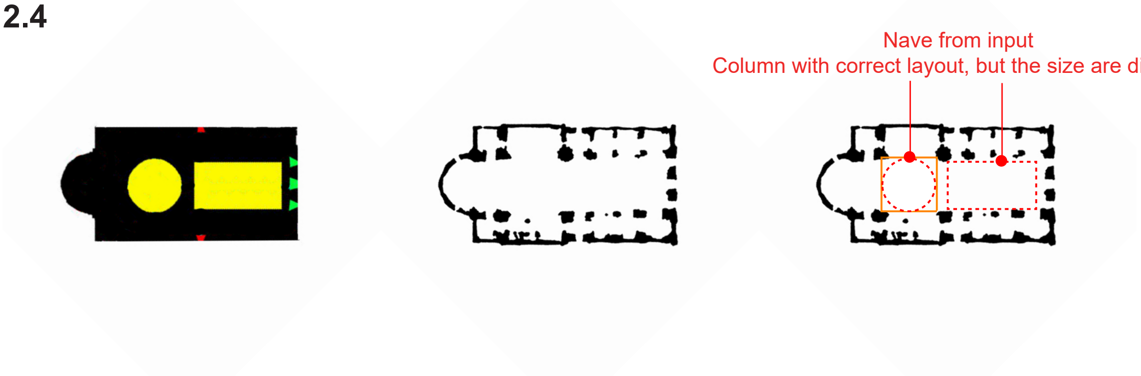


2.2



2.3

2.4



2.5

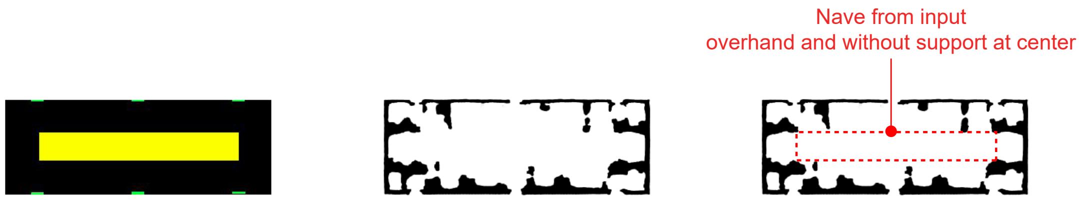
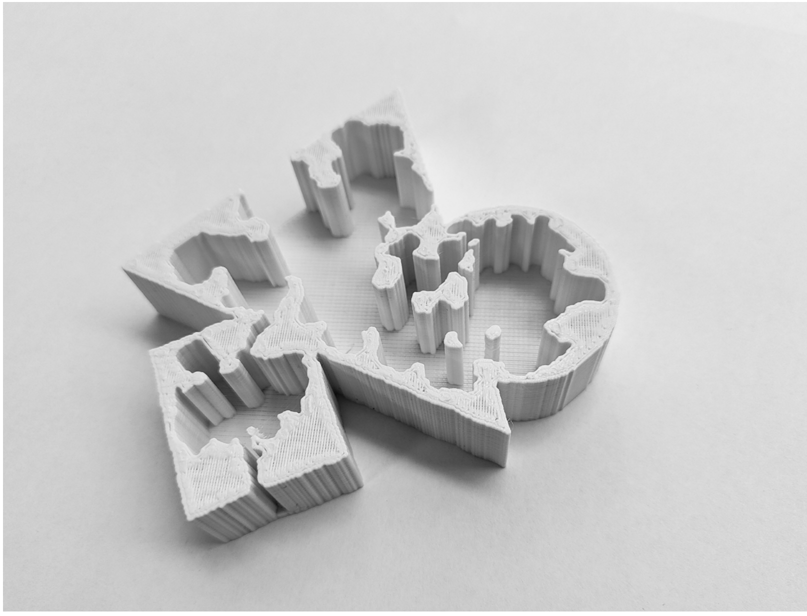


Fig26. Basilica set with long nave

The input of **cases 3.1, 3.2, and 3.3** were made in a random manner. Although all the output showed complicated spatial layouts, church plan features such as nave, chapels are still identifiable. Due to without typological reference and scale, it is hard to evaluate the quality of output.

The input of **cases 3.4 and 3.5** were based on Louis Kahn and Peter Zumthor's project. Compared to the ratios of door and window, the scale of the generated result in **case 3.4** is smaller than the original plan, while the scale of **case 3.5** is much larger than the original plan due to the subdivided spaces. This seems the scale of the generated plan is uncontrollable in the input images.



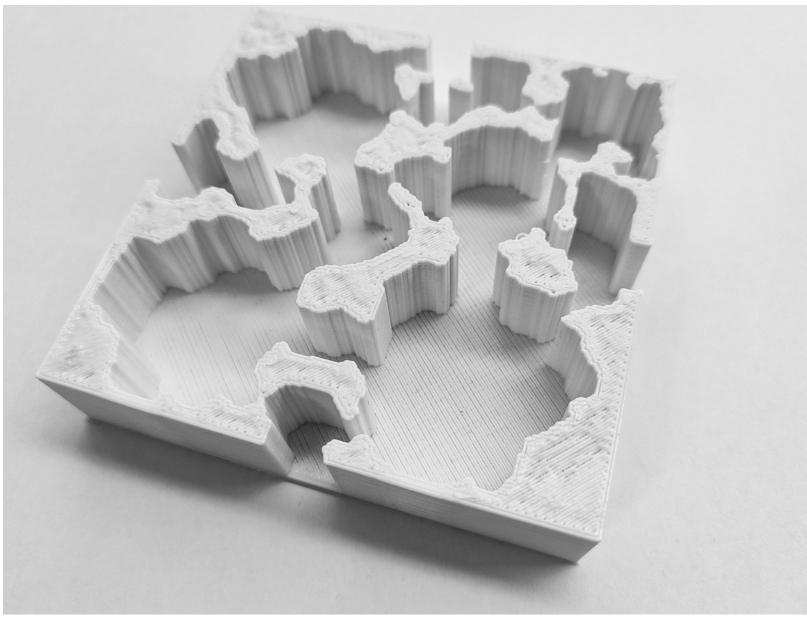


Fig27. Model of case 3.2 and 3.3

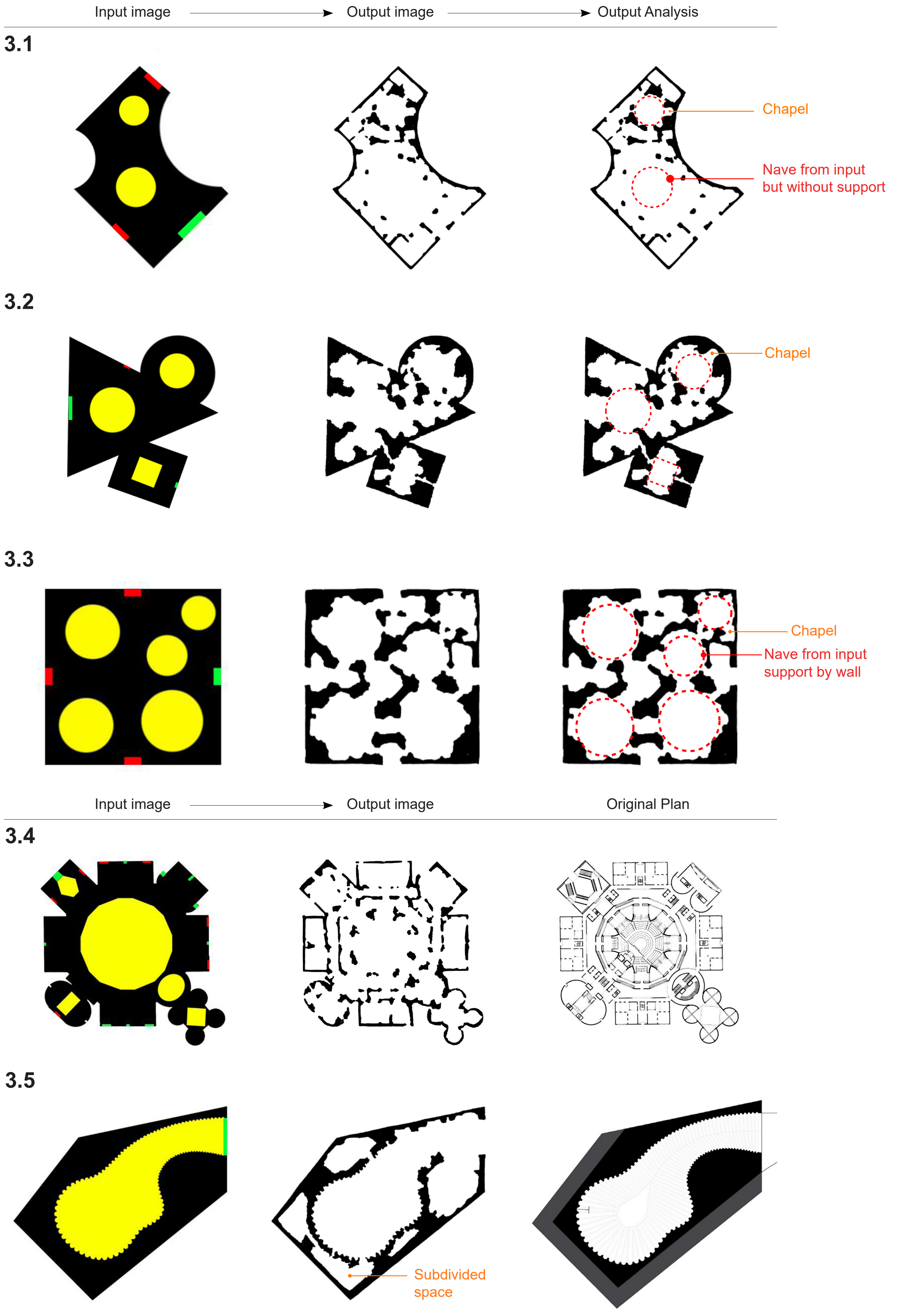


Fig28. Random made set (3.1, 3.2, 3.3), From exisiting building(3.4, 3.5)

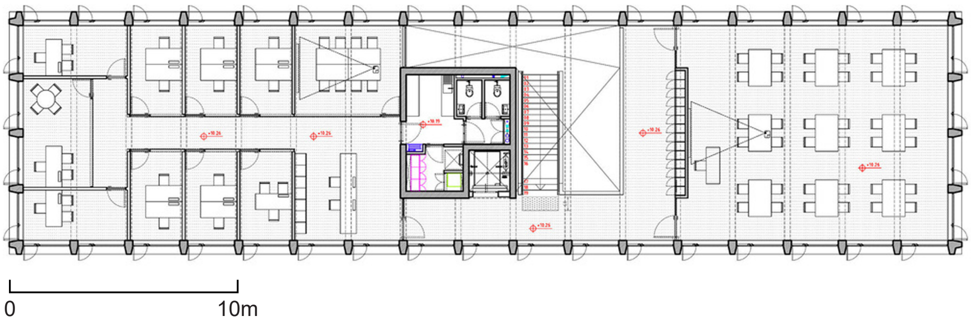
Transformation

This test took a transformative approach that apply the learned data pattern on the other building types and evaluate the quality. In the early stage of the design process, many architects use zoning maps to guide the plan layout. The input color map in my dataset is similar to the simplified zoning map. In this test, two plans from existing building projects were processed into zoning map and used the trained church plan generation tool to add the computer decision for the plan layouts.

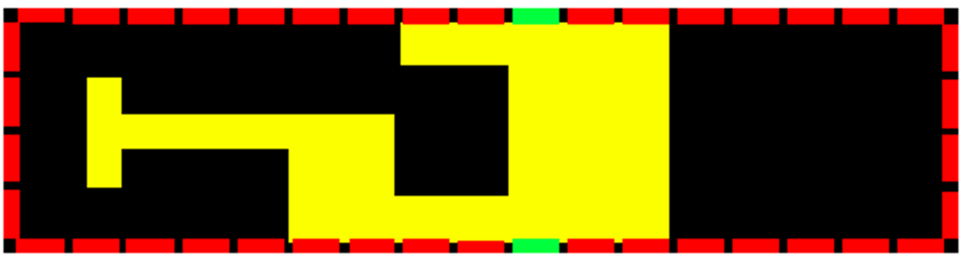
The first case selected a school plan that mainly consists of class room, service space, office, and corridor. As for the input color map, the corridor was labeled as 'nave' to prevent the algorithm adding walls and columns to block the circulation. Other contents such as spatial layout and division are obligated to the computer decision.

As for the output, it showed a circulation that allows for walking around. The arrangement of the interior wall and column is not aligned with the nave. Through overlapping with the original plan, it shows several conflicts that the added heavy wall and columns led the school plan lost floor area, and meanwhile, some of the walls blocked the original functions.

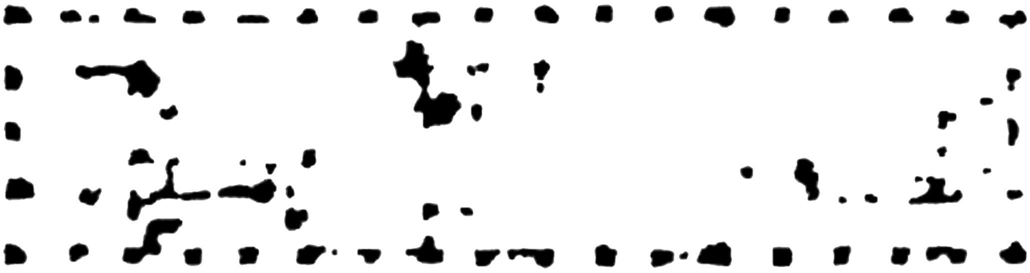
After adjusting and refurnishing the generated plan, it lost some office and classroom space, but the added wall brings quality when moving through the space.



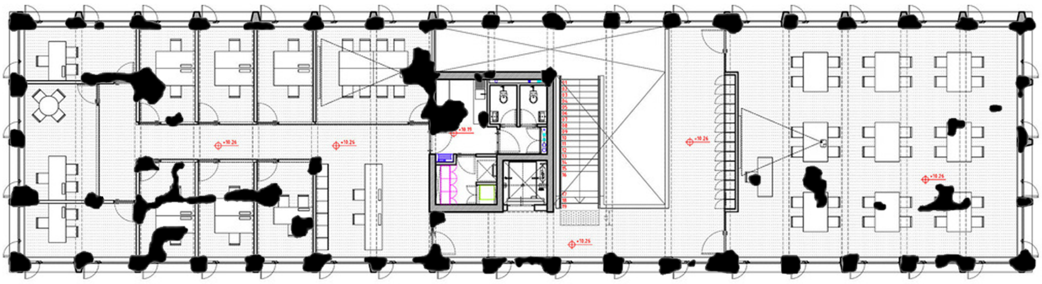
Original plan



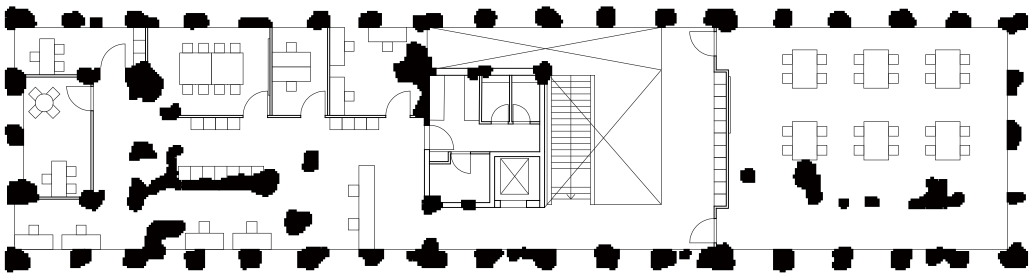
Input color map



Generated plan



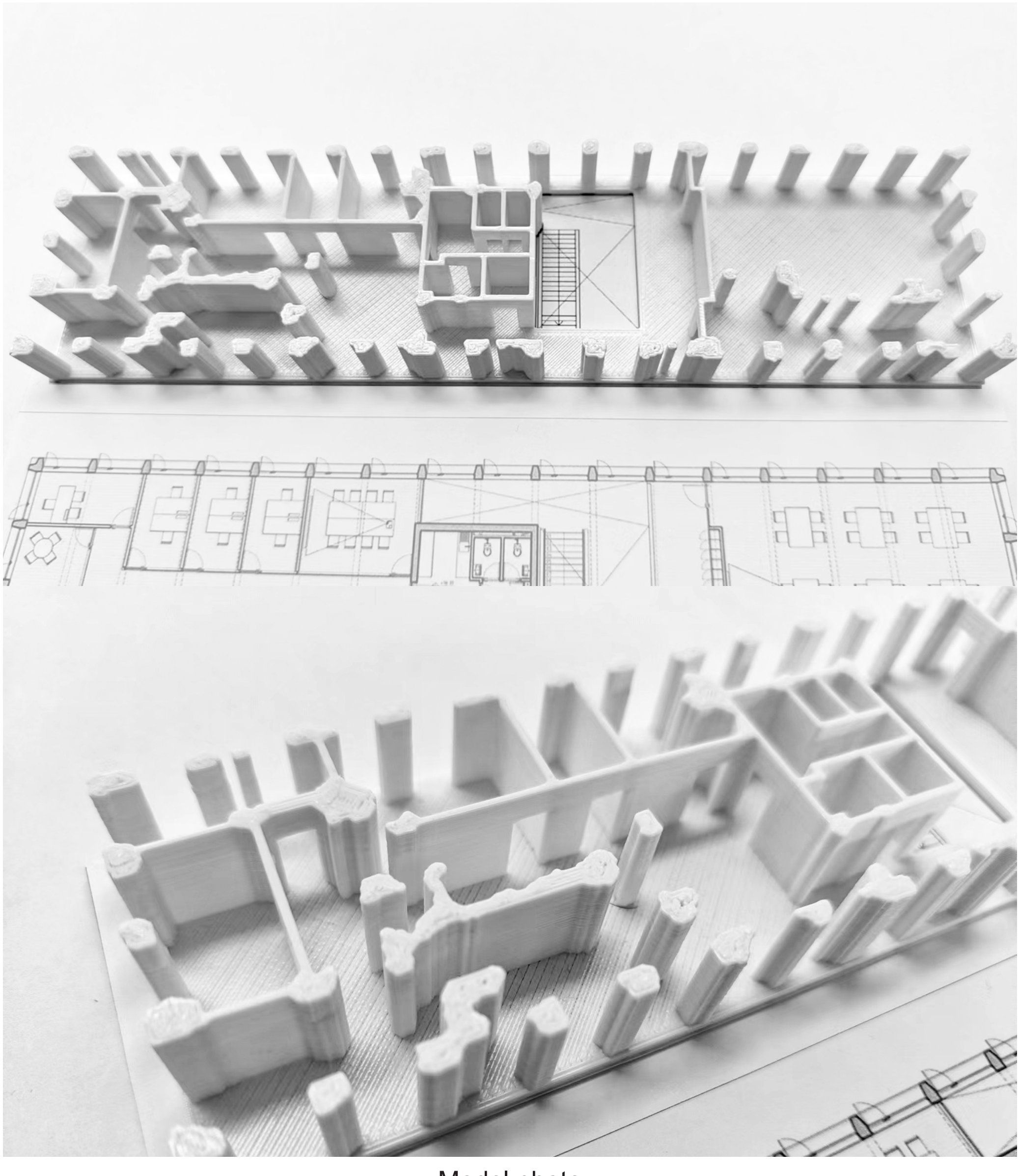
Overlap with original plan



Refurnished plan

Render of office space

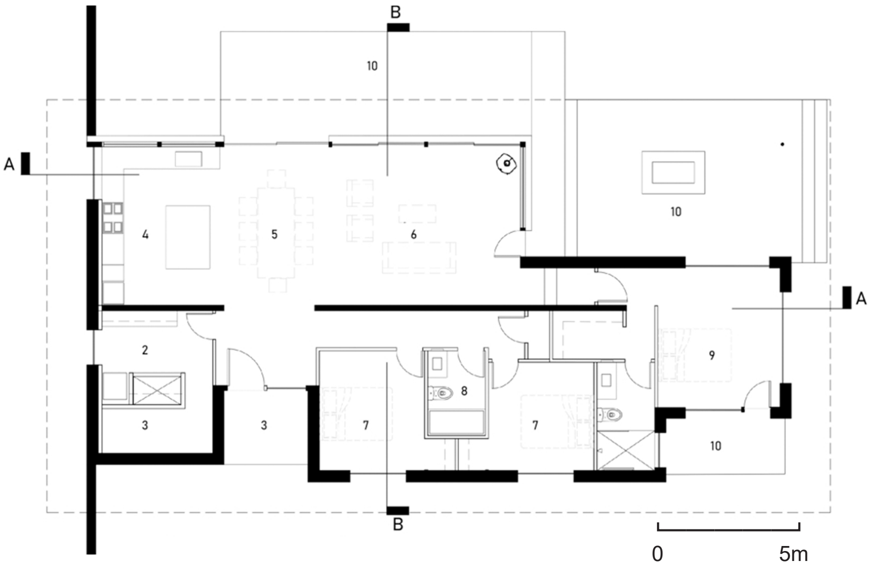
Render of the class room



Model photo

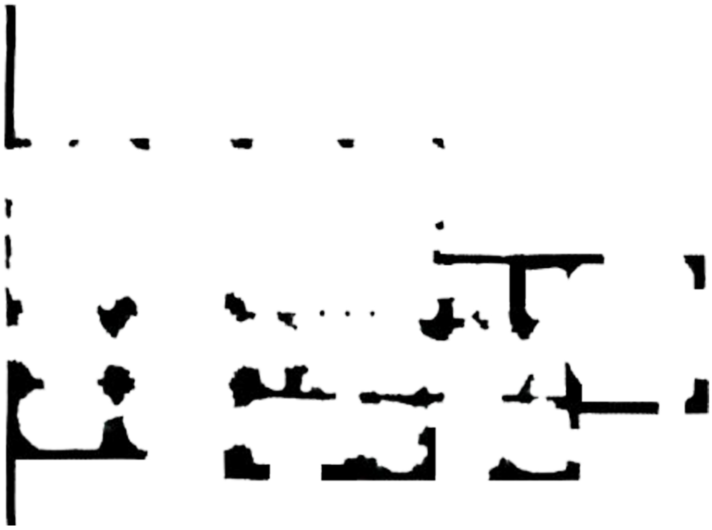
Fig29. Transformation of exisiting school plan

The second test was based on a big house plan which might bring more freedom to adding thick walls and columns. The input color map maintained the original features that labels the living room and corridor as the nave. The output image of this test showed many church features such as chapels and void geometries. The overlapping result had the conflict in the toilet and bedroom.

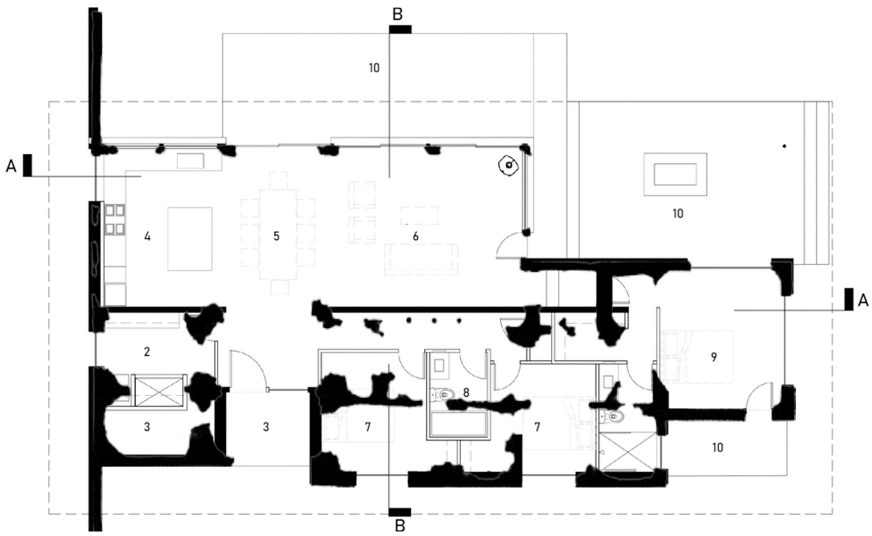


Original plan

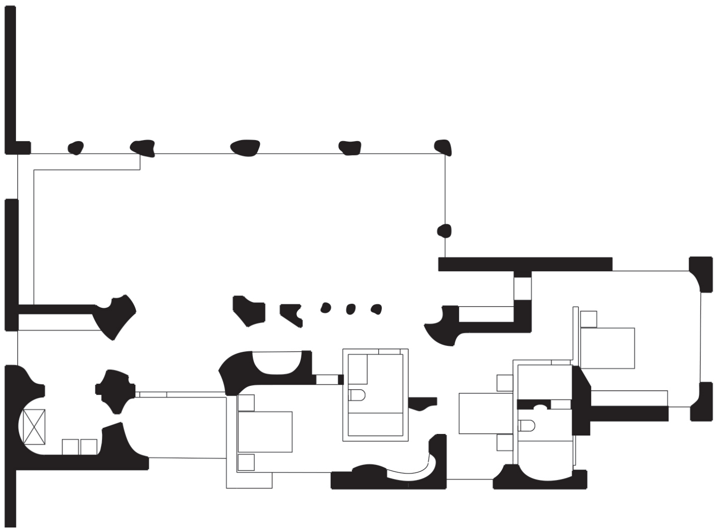
Input color map



Generated plan



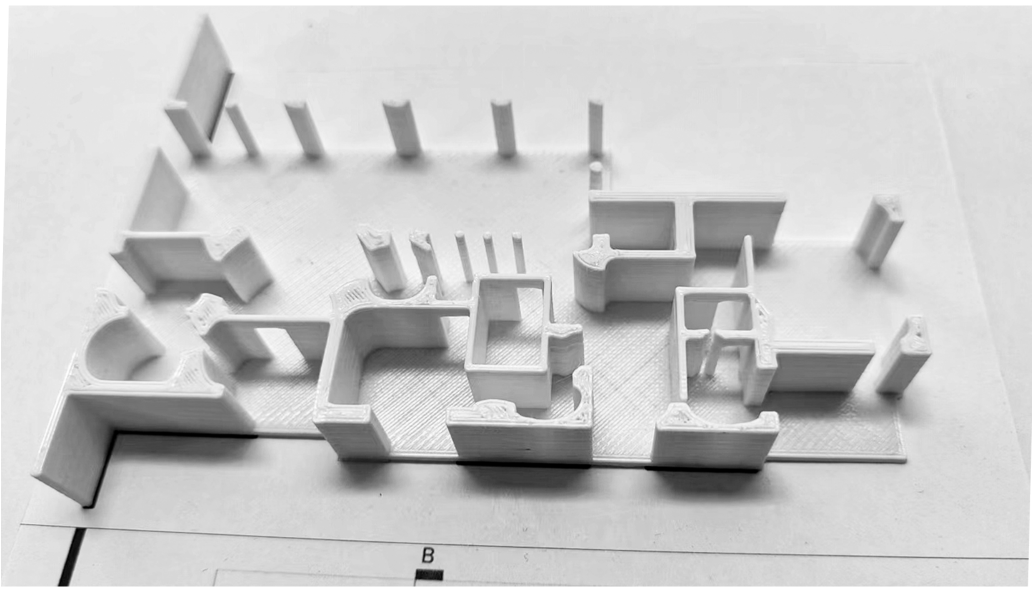
Overlap with original plan



Refurnished plan

Render of the corridor

Render of the living room



Model photo

Fig30. Transformation of exisiting house plan

3.3 From two-dimension to three-dimension

The previous experiments only focused on the two-dimensional qualities of church plans. It is difficult to understand the right scale without the correct measurement. This experiment projected the section ratios onto the church plan and tested if the algorithm could recognize the data pattern between plan and section ratios.

Projection

Many church plans use the projection line to illustrate the shape and structural typologies, but it's difficult to determine the height without section drawings. The previous experiment proves that the algorithm may not be capable of detecting data patterns from the limited pixel information. So I avoid directly draw the numbers on the plan to label the section mesurement. Instead, I utilized the HSB color block to draw the roof projection. The S and B data record the ratio of beam height and the highest point. For H = 255, it displays a blue color that could be filtered by the B channel, this could be helpful when creating a 3D model. In the refined dataset, target outputs also use the red block to present the projection of the arch in order to display how structural elements support the nave. The input image maintains the same information as previous experiments.

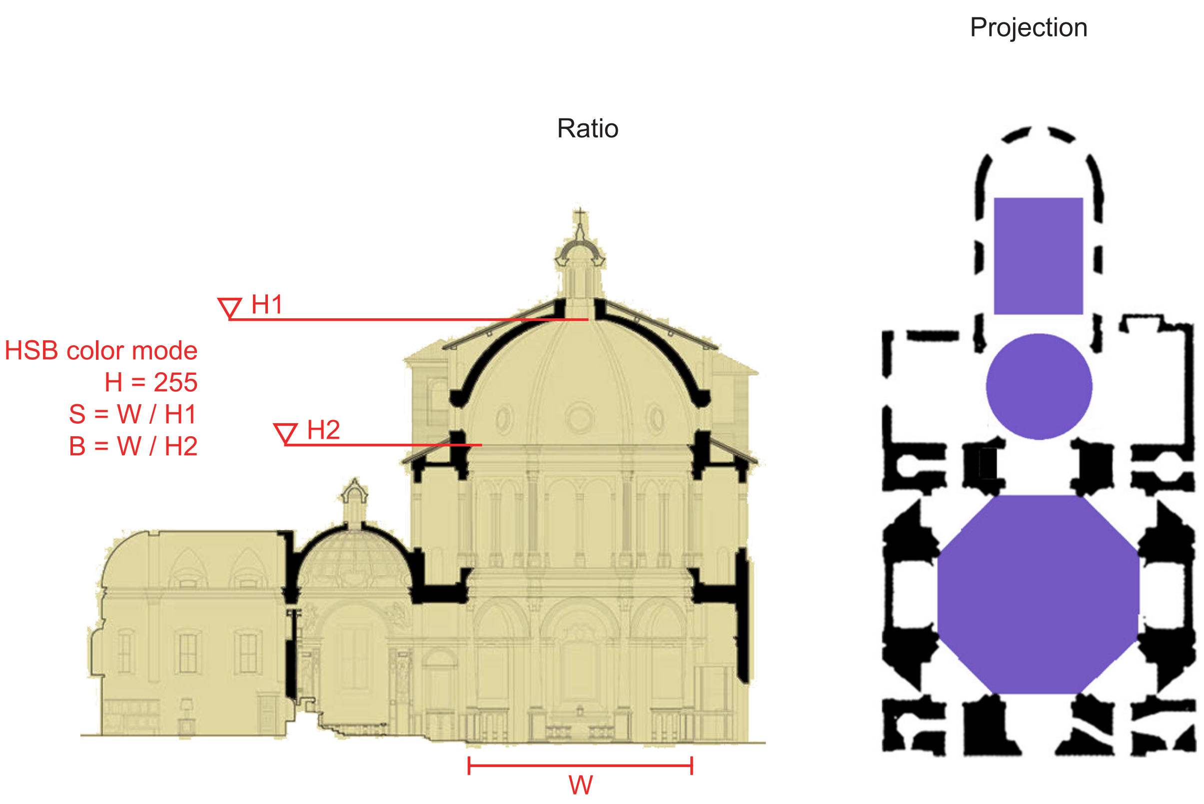


Fig31. Roof projection method

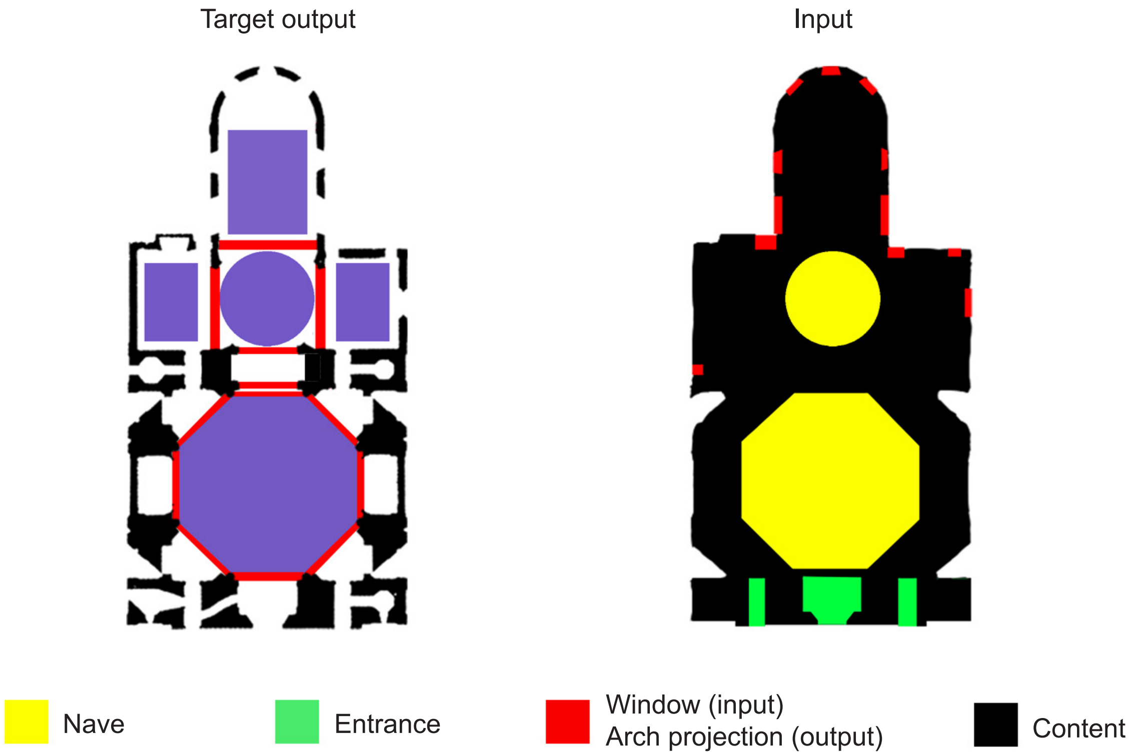


Fig32: Dataset with ratio projection

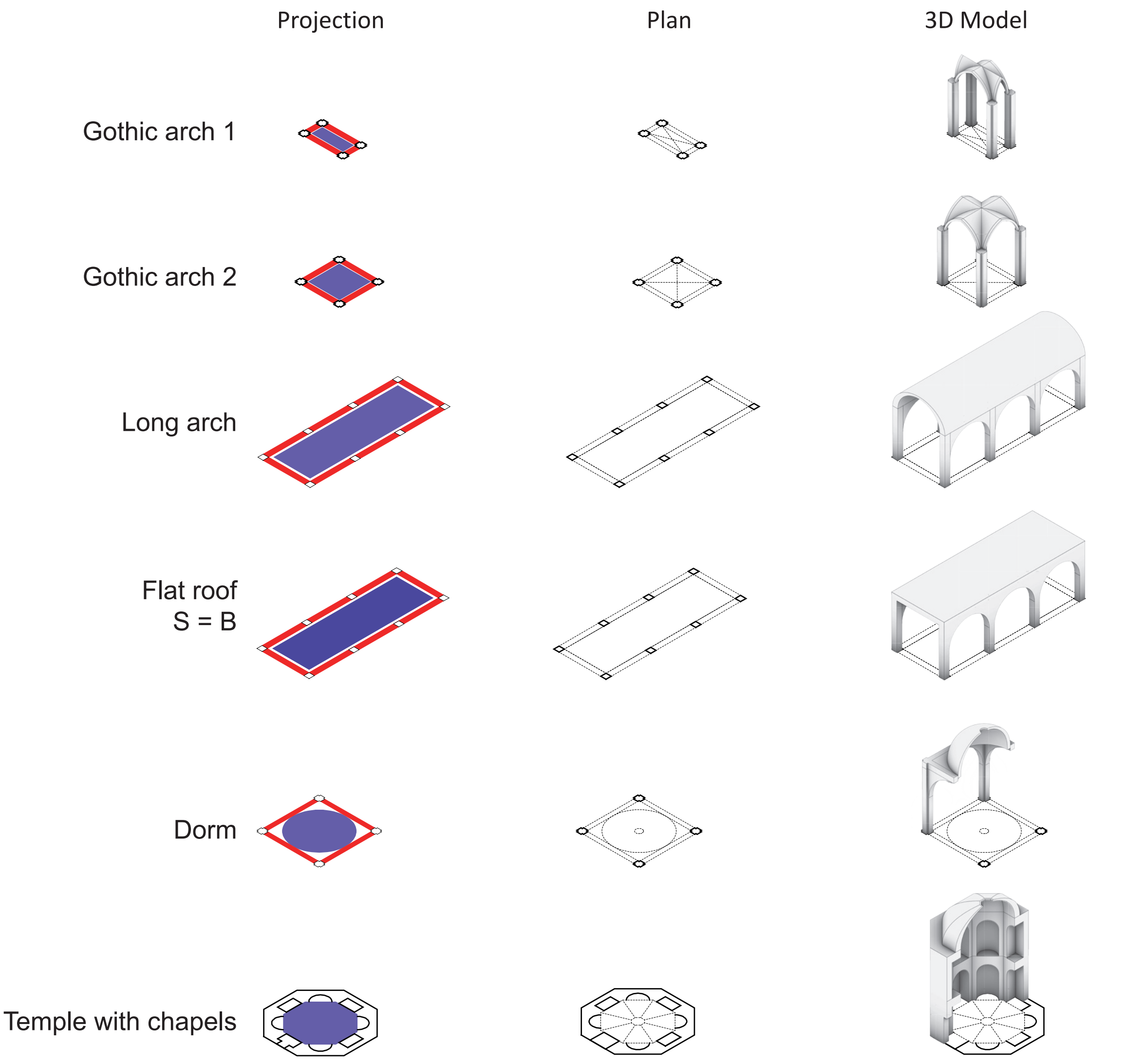


Fig33: Catalog for 3D modeling

Training and testing

After 200,000 steps of training, most of the output showed similarities with the target output. As for the typologies, the result generated by temple had good resolution while the interior element of basilica had low resolution on output. As for the HSB data, all the output had the same HSB color with the target output, this might shows that algorithm recognized data pattern of section ratios.

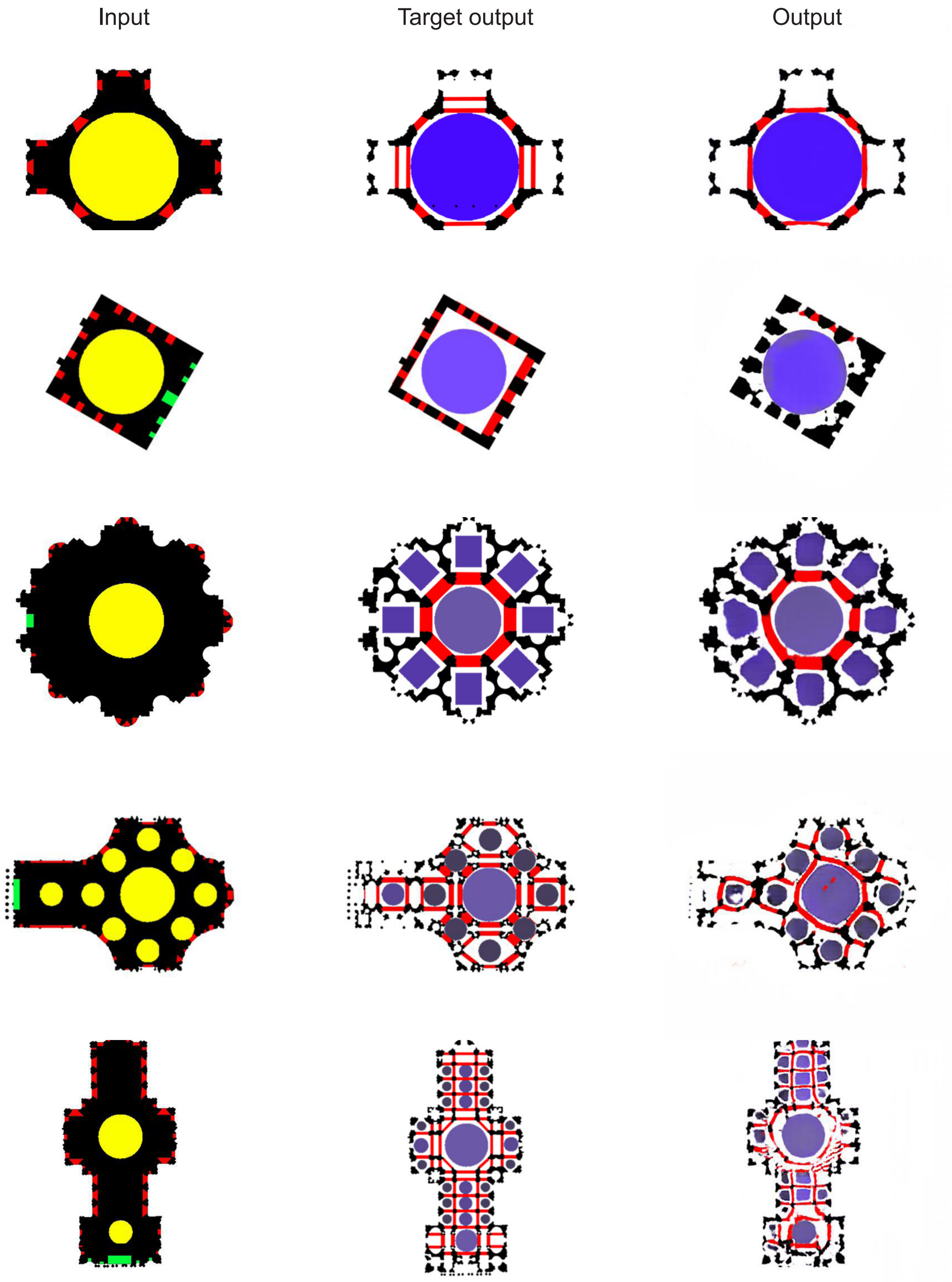


Fig34. Training process

Evaluate the training result

This target output is a gothic temple where buttresses support the roof. However, in the generated results, the walls are increased thickness and added the chapels while retaining the buttresses. This modification is not necessary in terms of the structural princples. As for the section ratios, the generated result shows the almost the same HSB color as the original case.

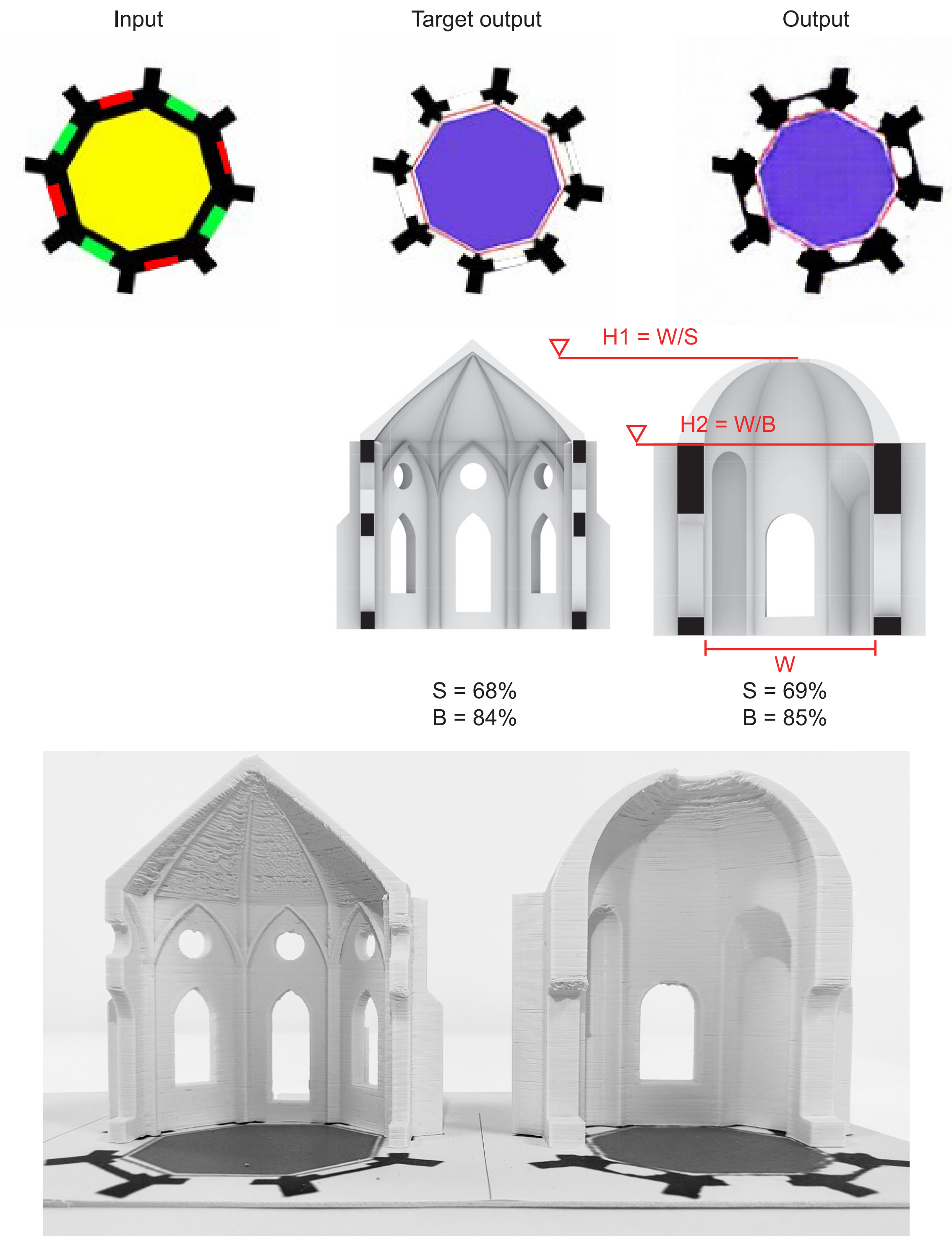
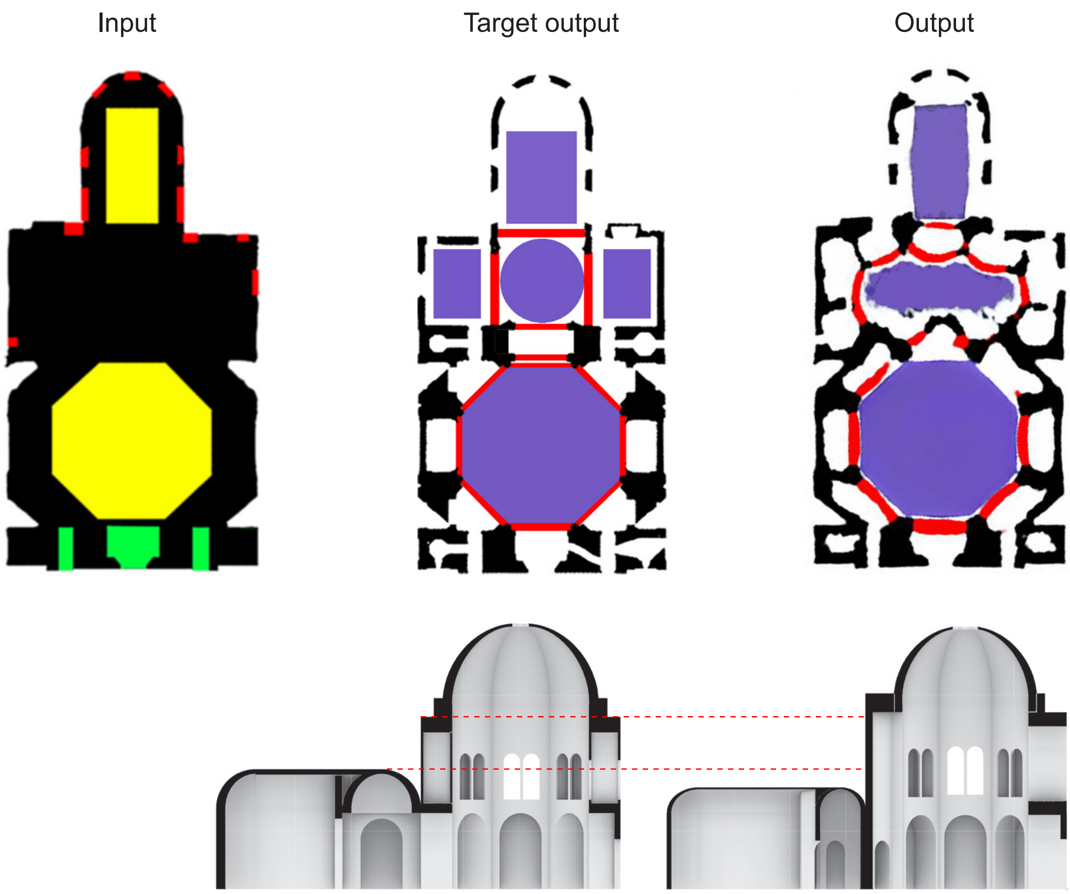


Fig35. Result of gothic temple

The target output of this case has three different types of roofs. The generated result reduced the number of roofs and changed the circular nave into a long arch. Although the corner and arch are modified into a curvy shape, they still structurally support the nave. As for the section, the generated result looks close to the target output, but the changed ratio made the height slightly different from the original case.



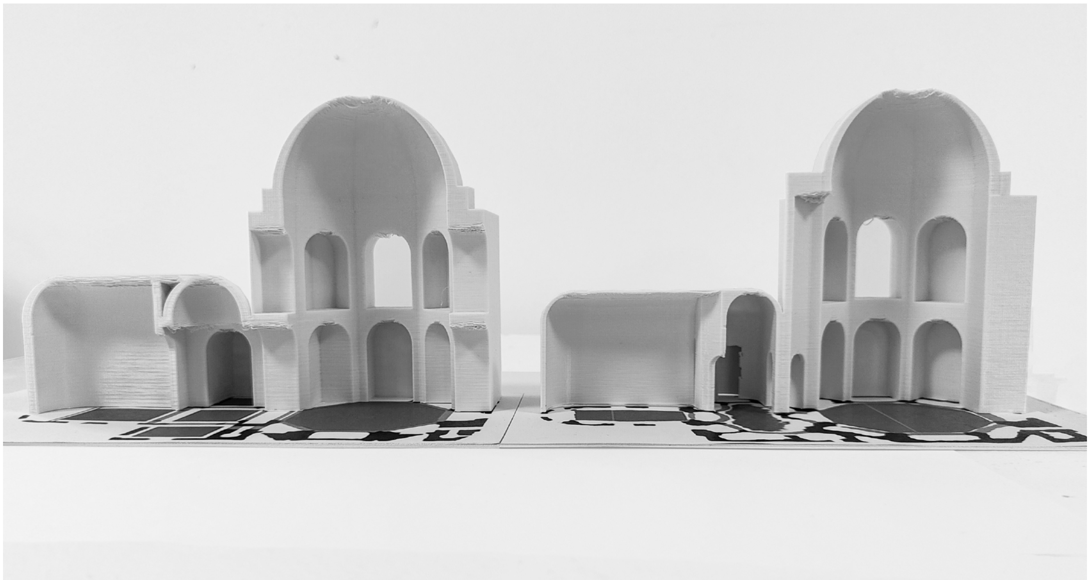


Fig36. Result during the training process

Generation

The trained algorithm was tested with inputs out from the dataset. The output generated by the basilica color map showed low resolution of columns and roof projections, while the output generated by the temple color map had the correct structural layout and higher resolution. As for the randomly produced inputs, it generated the output had irregular roof shapes.

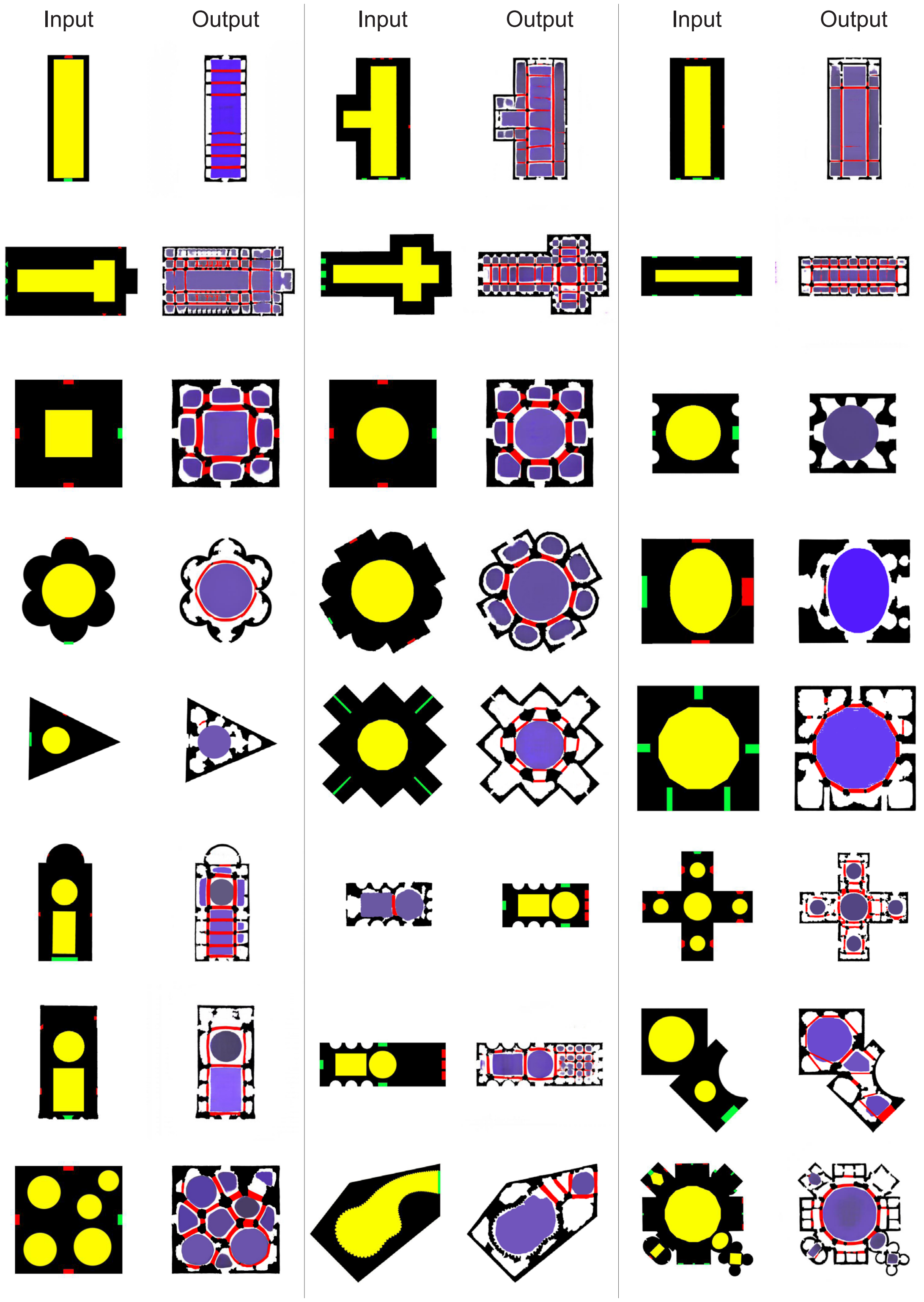


Fig37. Generated result

Evaluation

A distribution chart evaluated the output in this phase. The black dot on the chart represents the section ratios from the dataset. The red dot represents the section ratios of the generated result. If the red dot is close to the main distribution line, the algorithm may have correctly decided the section ratio.

The generated result of **case 1.1** showed high resolution on structural elements. The roof projection (A) on the central area is long and considered as a long arch supported by wall. The roof projection (B) at the front and rear are small rectangles, which are considered to be the gothic arch. This case combined two different roof types and the interior space might look similar to an actual basilica.

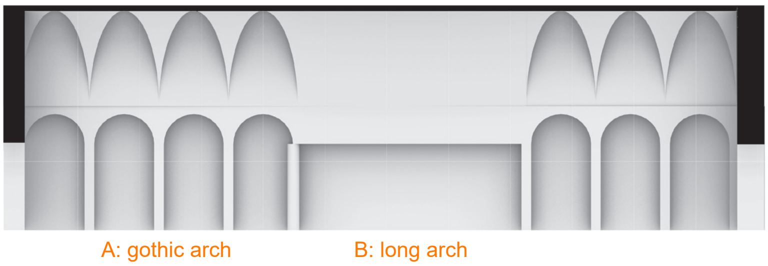


Fig38. Section and render of case 1.1

Case 1.2 and 1.3 had insufficient resolutions on interior space. Part of the structural elements are invisible, and the arch projection is completely overhanded. However, the trained algorithm still made a good decision on the section ratio which is close to the main distribution line.

Case 1.4 and 1.5 had the correct layout of columns, but they showed low resolution and inconsistent sizes. In contrast, section ratios are close to the main distribution. The basilica sets(with long nave) showed that the trained algorithm could decide the appropriate ratio and layout, but it needs manual refinement to improve the output due to the low resolution.

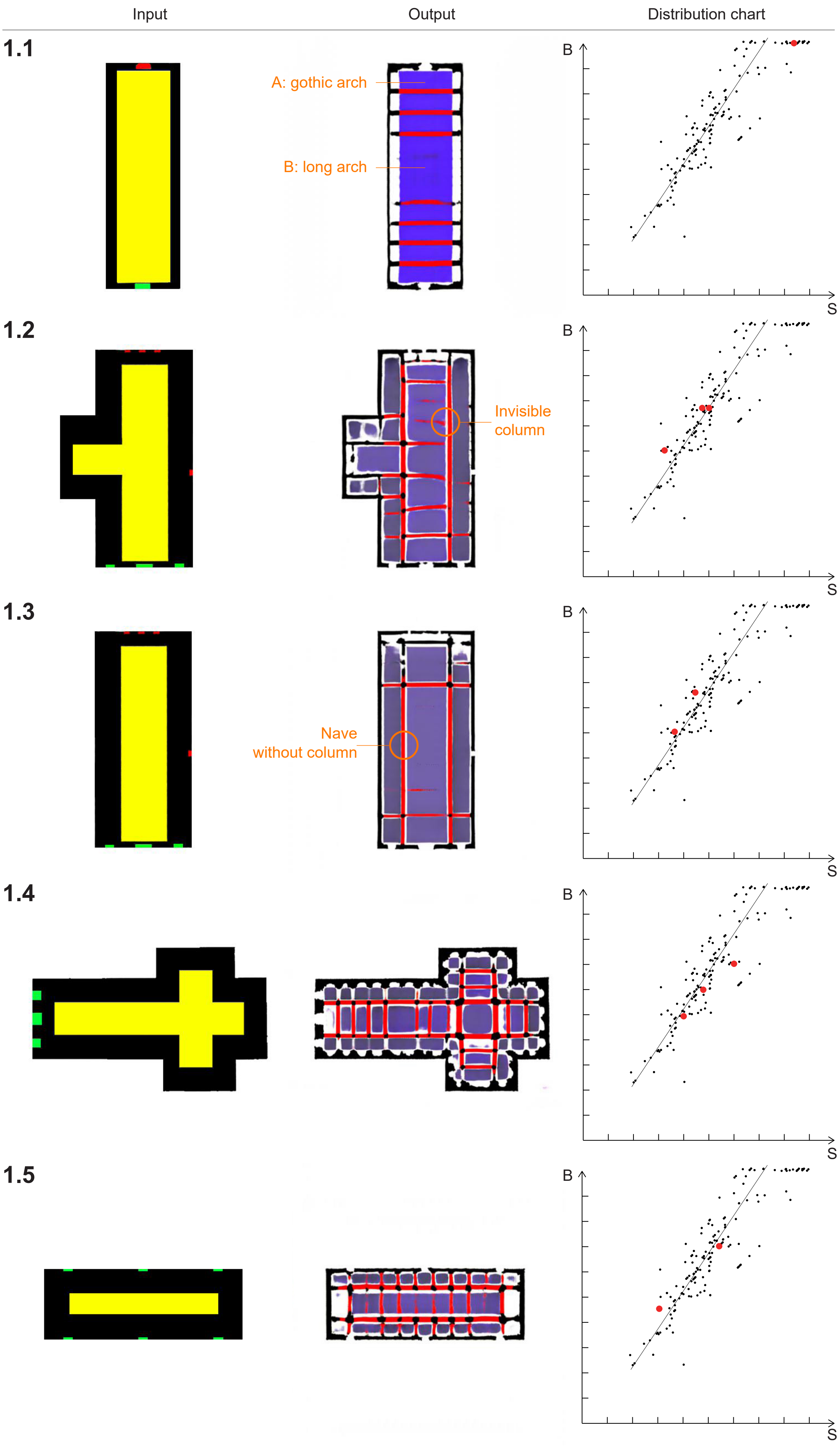
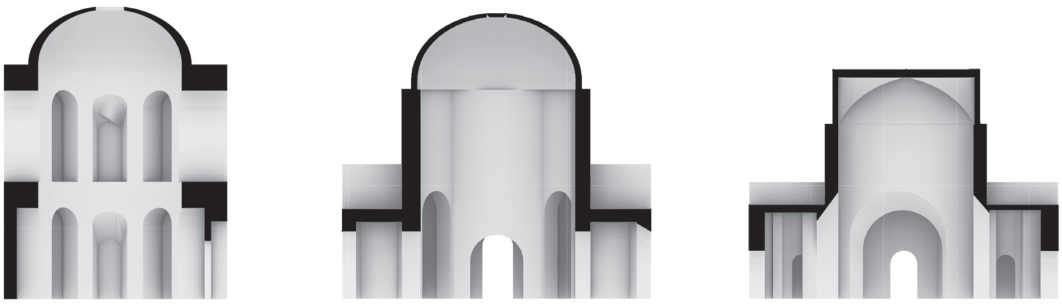


Fig39. Basilica set with long nave

The results of the temple sets(with a big central nave) had higher resolution than the basilica sets(with long nave). The input of **case 2.1** labeled a large nave on the central area and removed some hemispherical space on the boundary to test whether the algorithm could cope with this unconventional boundary. The generated output has a high resolution and shows the features of the temple where the chapel surrounds the central nave.

Case 2.2 and 2.3 reduced the area of nave and generated several subspaces around the central nave. The ratio of central nave in both cases is close to the main distribution, but the subspace had unconventional ratios. The S and B data were equal on those subspaces and considered flat roofs.



Case 2.1 Case 2.2 Case 2.3

Fig40. Section Images

Case 2.4 used an unconventional shape from the temple to test the algorithm. The layout on the generated plan shows correct structural principles, but the chapels had irregular shapes.

Case 2.5 used an oval nave to test the trained algorithm. The generated result has a high resolution, but the roof is too low and did not follow the correct structural principle.

Case 2.4 Case 2.5

Fig41. Section Images

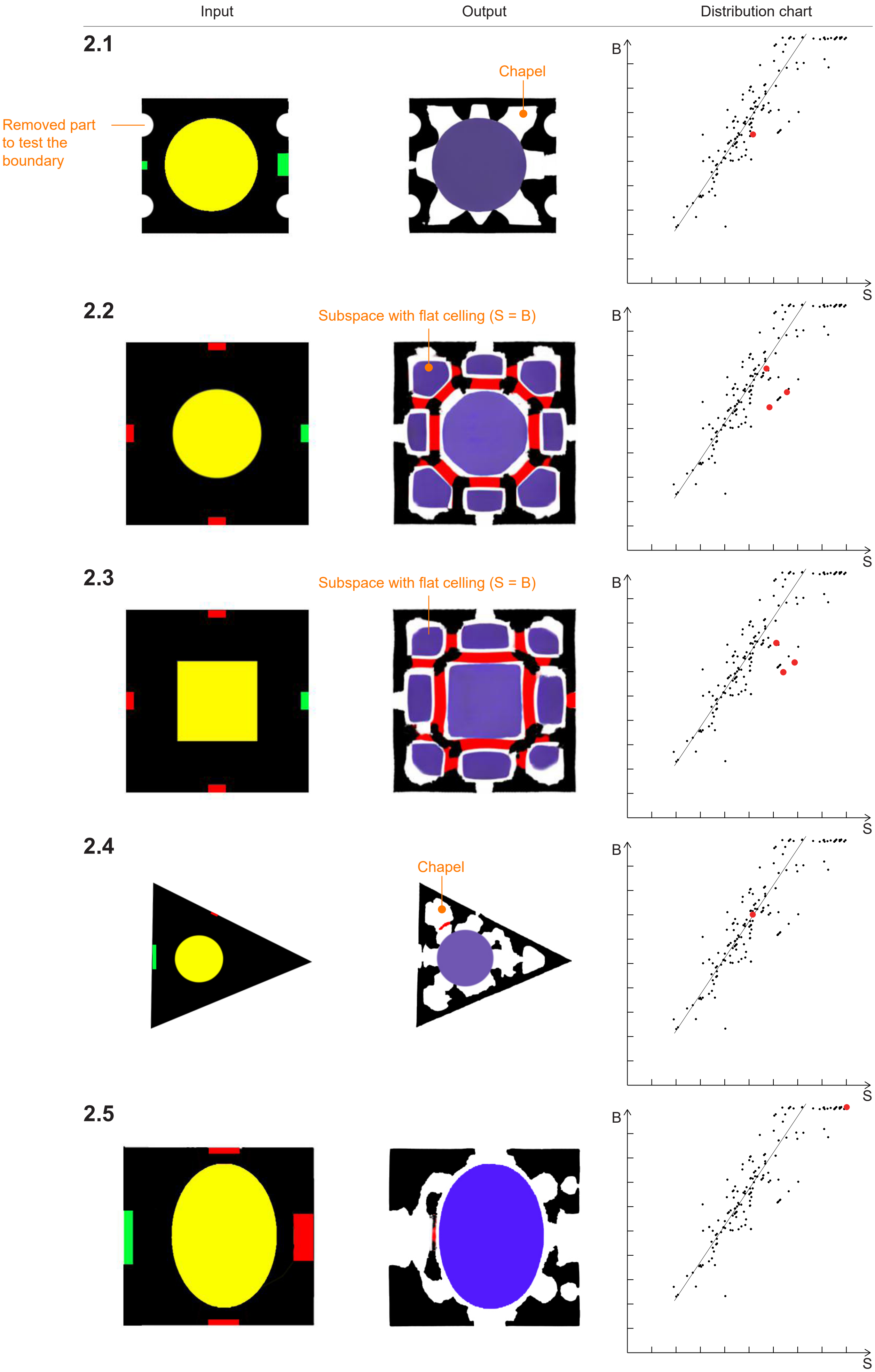
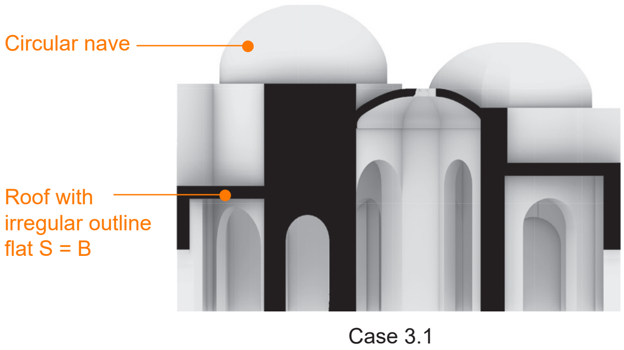


Fig42. Temple set with big central nave

The input of **case 3.1 and 3.2** were randomly made. Both of the generated results showed several irregular roof outlines. As for **case 3.1**, the irregular outline roof is flat because the S data is the same as the B data. The ratio of the circular nave is slightly separated from the main distribution. As for **case 3.2**, the ratio of the irregular roof is close to the main distribution. It shows an unconventional shape on the 3D model.



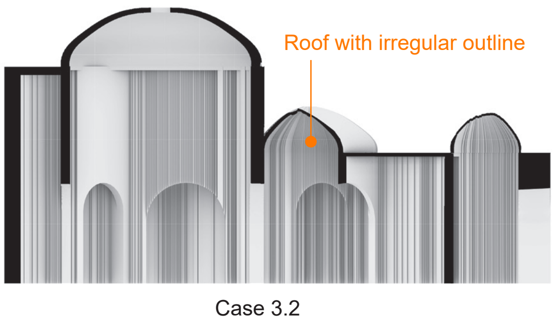


Fig43. Section of case 3.1 (left) and 3.2 (right)
Render of 3.1

The input of **case 3.3** increased the complexity of the outline and labeled long entrance, which is unconventional for the church plan. In the generated result, the labeled long entrance does not influence the spatial layout, and the central nave had a good ratio which close to the main distribution.

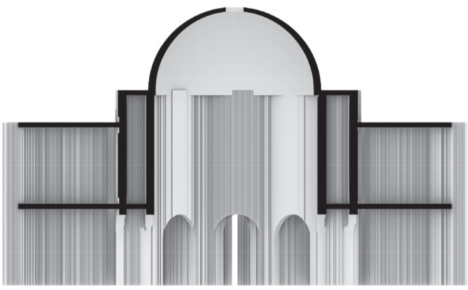


Fig44. Section of 3.3

The input of **case 3.4 and 3.5** are based on existing projects. Although the section ratio on output is close to the main distribution, the scale is much different from the original design. **As for case 3.4**, the original design by Zumthor is a small chapel, but the generated result is sub-divided into several spaces and looks much larger than the original design. **As for case 3.5**, the central nave is much higher than the other nave and shows a strange volume.

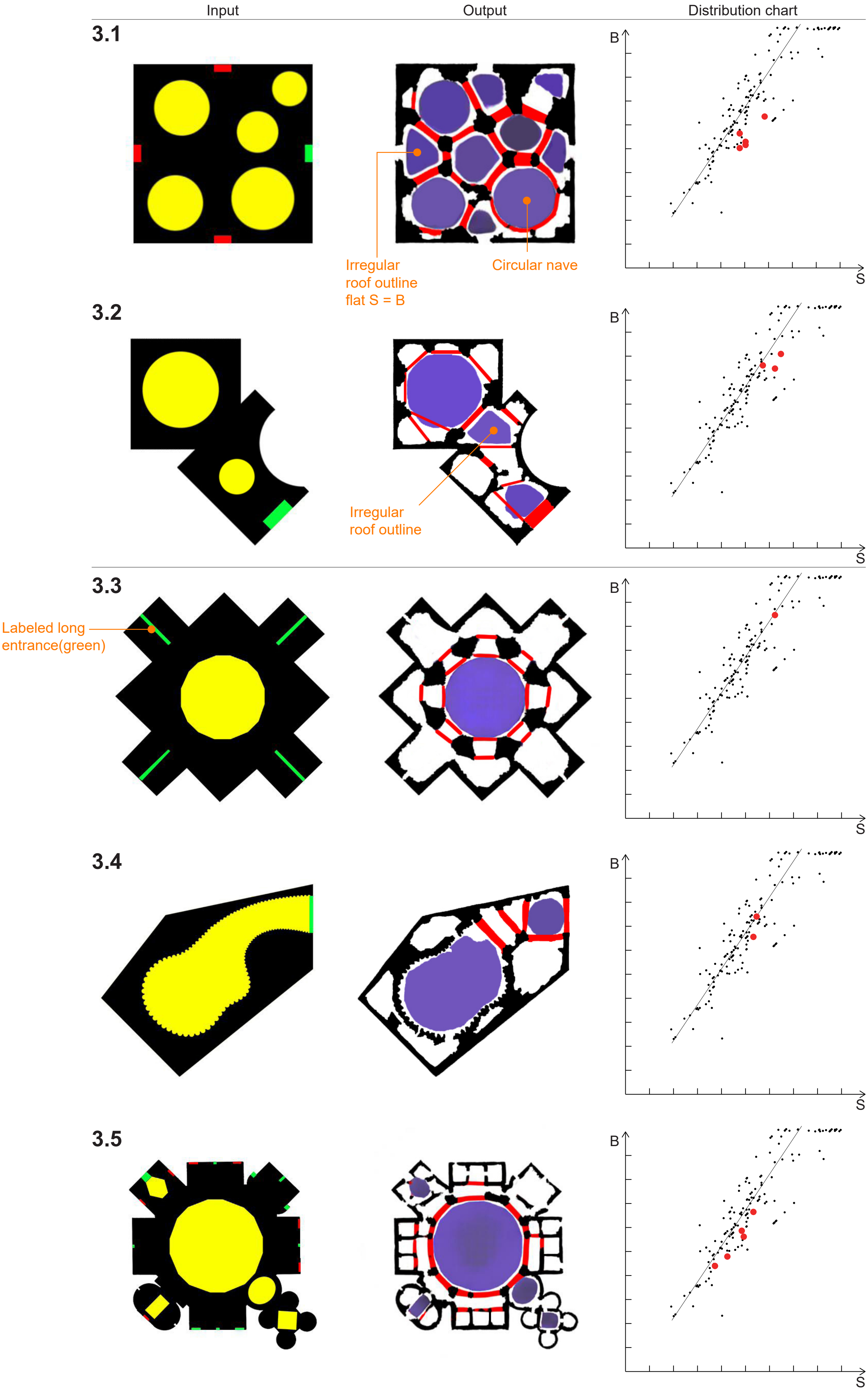
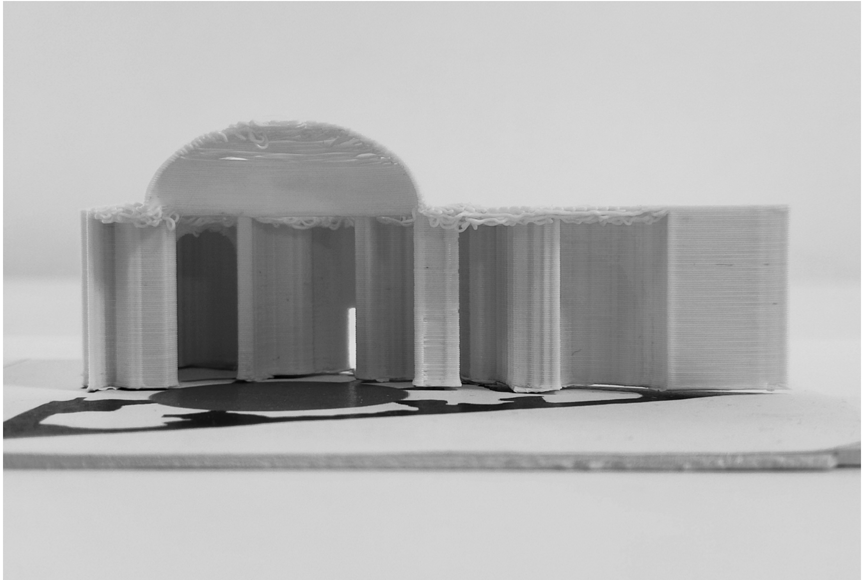
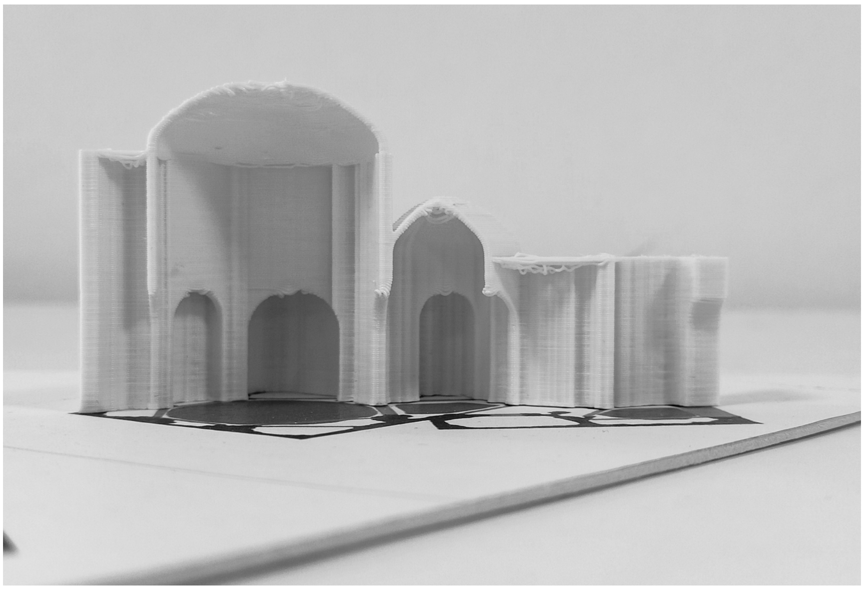


Fig45. Random set(3.1, 3.2) and exsisting building set(3.3, 3.4, 3.5)





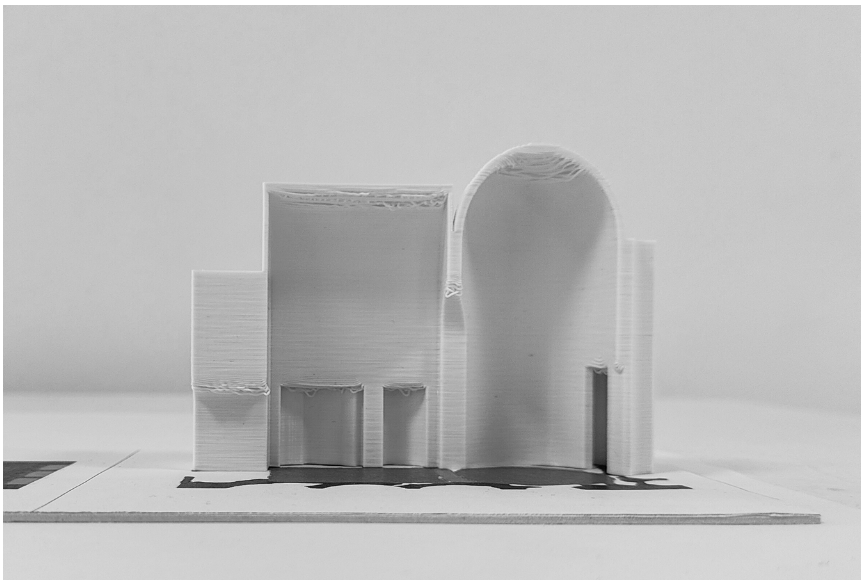
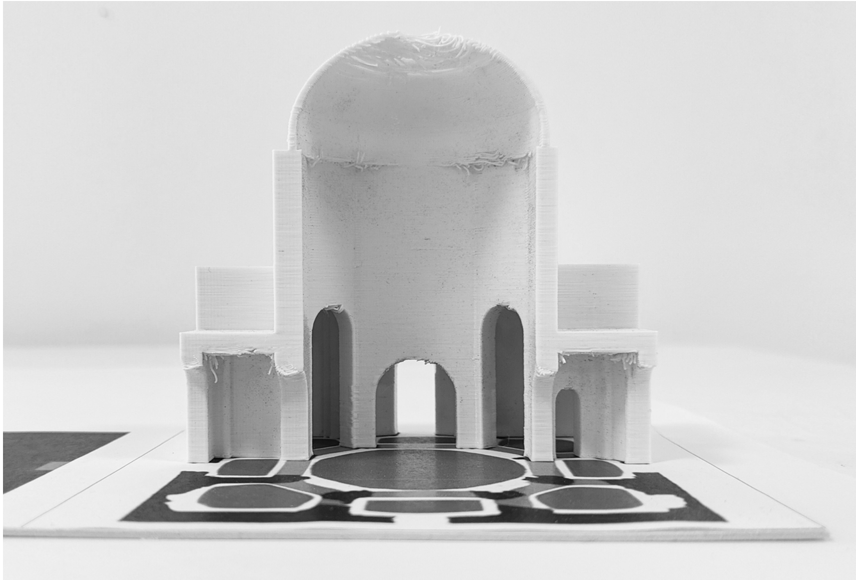
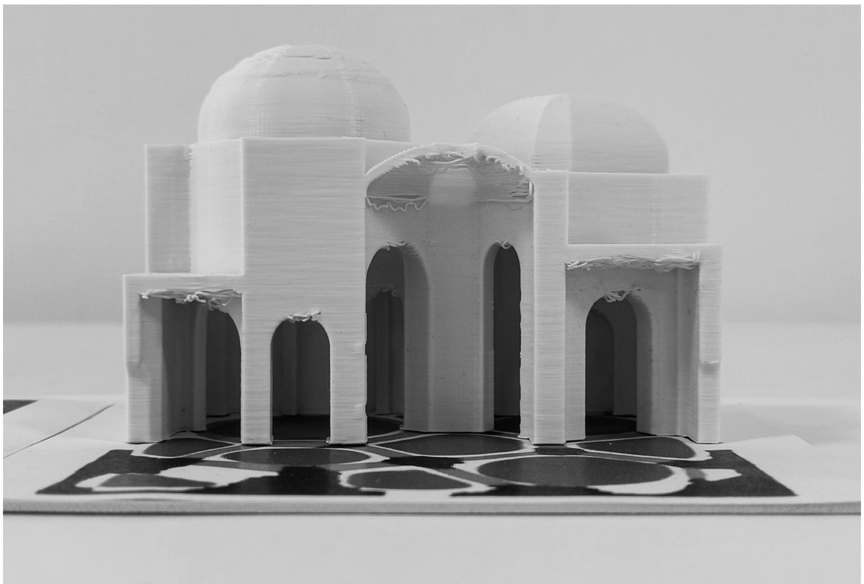


Fig46: Section mo





3.4 Feature transfer by NST

This experiment took a different approach from the previous experiments, aiming to use the Neural style transfer(NST) algorithm to transfer the geometrical features from one object into another. Inspired by the CT scan, the complex geometrical form could be sliced into image sequences to present geometrical features on each layer. This experiment sliced content and style objects into sequence images and transferred the feature layer by layer. In the end, through sampling and voxelizing of each pixel from outputs, the 3D model can be reconstructed from the output image sequence.

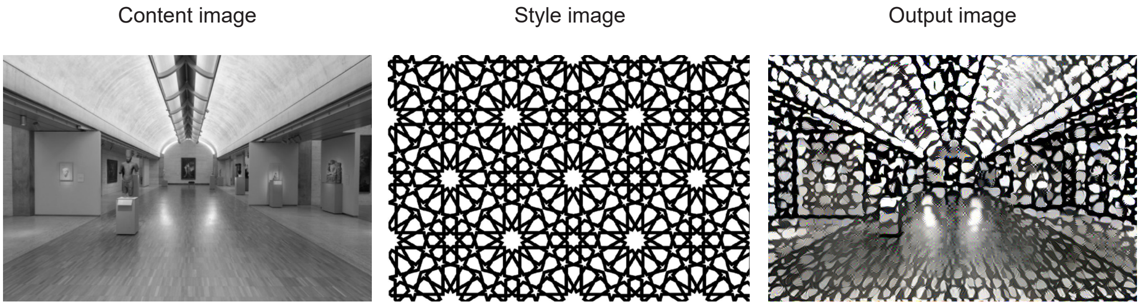


Fig47: NST algorithm test

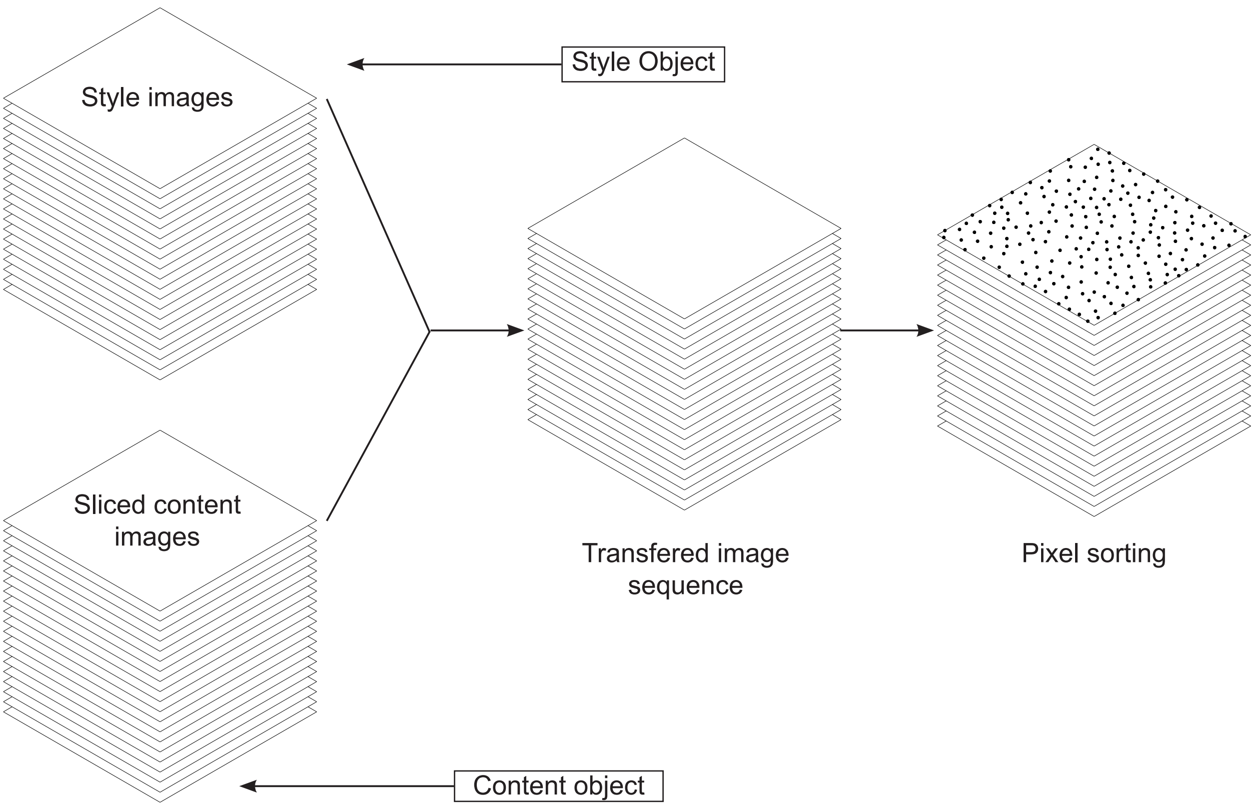


Fig48: Experiment method

3.4.1 Pattern transfer

In the industry and biological world, many complex patterns are integrated with different functionalities that significantly improve physical performance. To achieve functional integrated design, some designers use the parametrical approach to find the mathematical logic behind complex patterns and then use the learned logic to generate a pattern for the design targets. This approach normally takes a significant amount of time, but the ability of the NST algorithm might give a new approach to using neural networks to separate the pattern from one object and synthesis this pattern to the target object.

This experiment transferred the metallic foam pattern into a cylindrical-shaped model. Metallic foam is an industrial material that integrates thermal insulation, noise reduction, and air filtration functions. This functional integration is achieved by a complex geometrical pattern. The fabrication method of the metallic foam pattern is unsustainable due to the requirements of high temperature and pressure. NST might bring the possibility of transferring the metallic foam pattern into building components and 3D printing with sustainable material. This new process might improve the physical performance of the building component and reduce environmental compliance during the fabrication process.

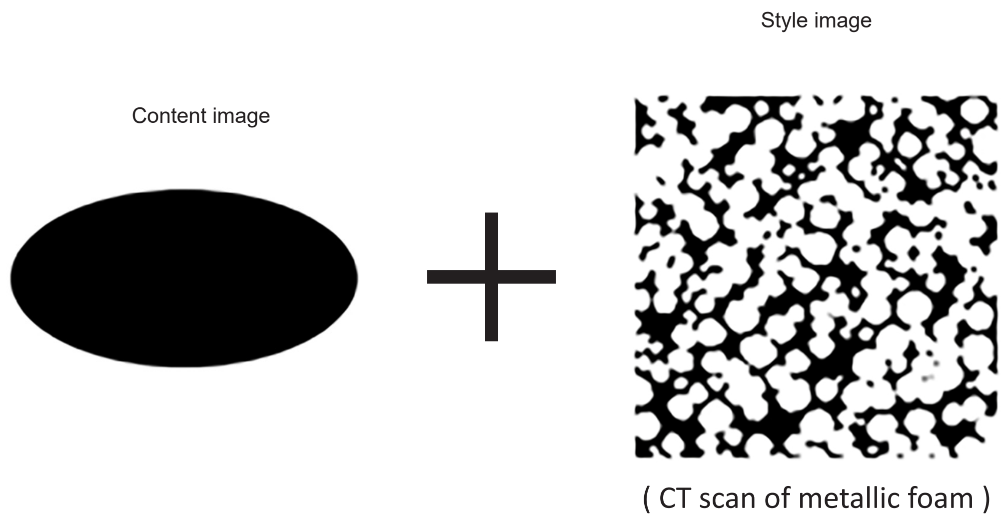
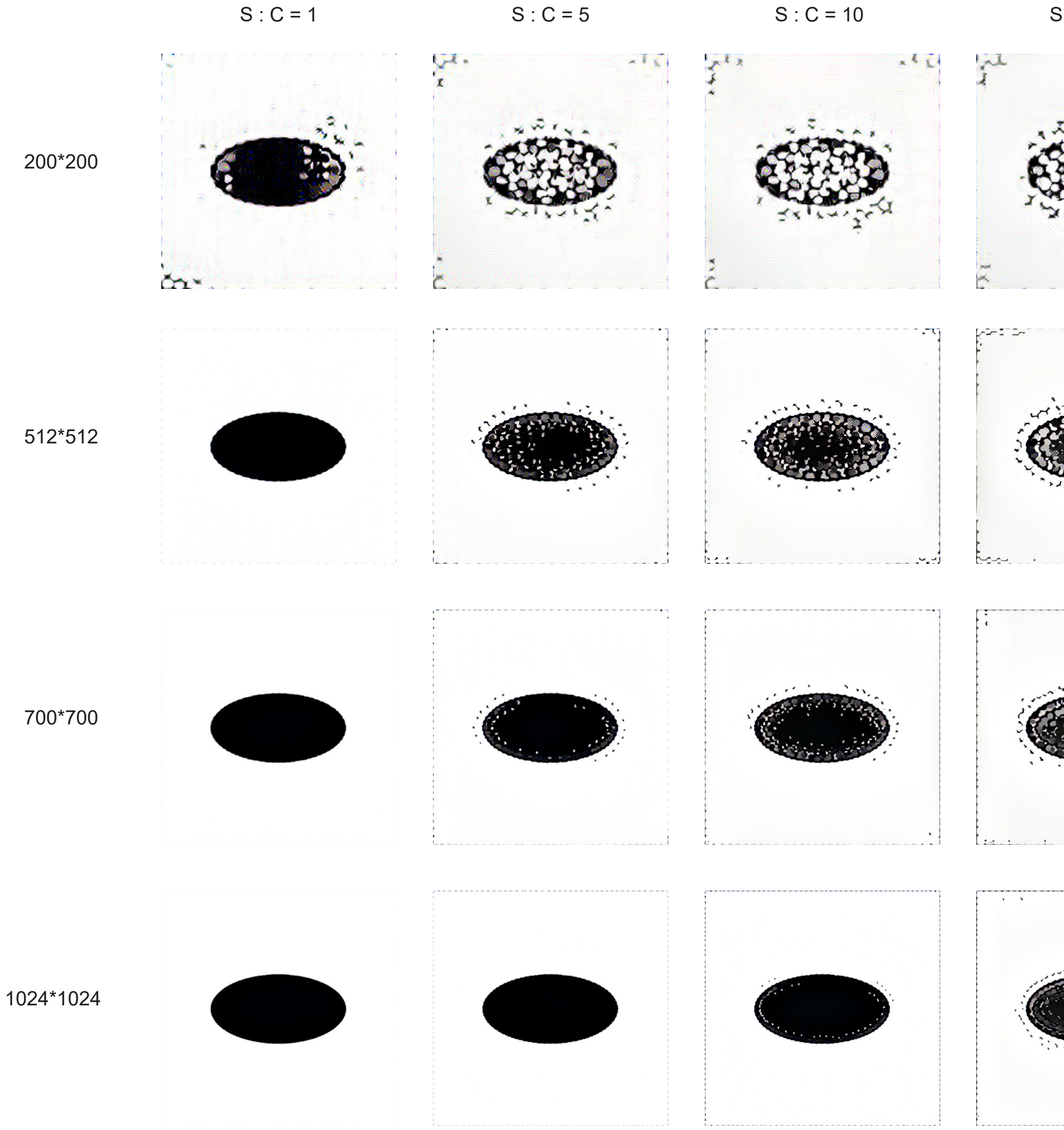


Fig49: Transfer the metalic foam pattern to content image

Several input parameters are required in the NST process, and they have a significant influence on the outcome. To find the best combination of input parameters, I conducted a control test by NST algorithm. From the result, the 200 pixels resolution with 1000 style weight achieved the closest result to the style image.



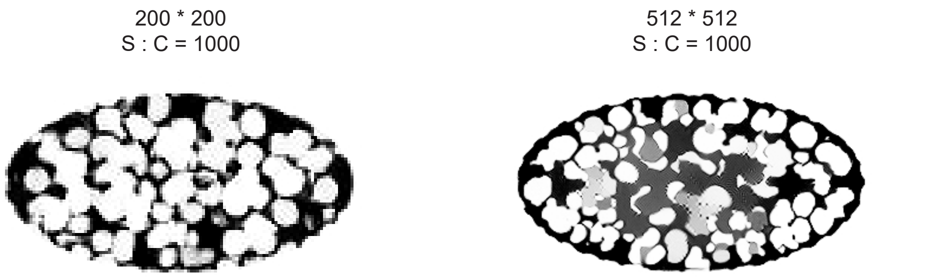
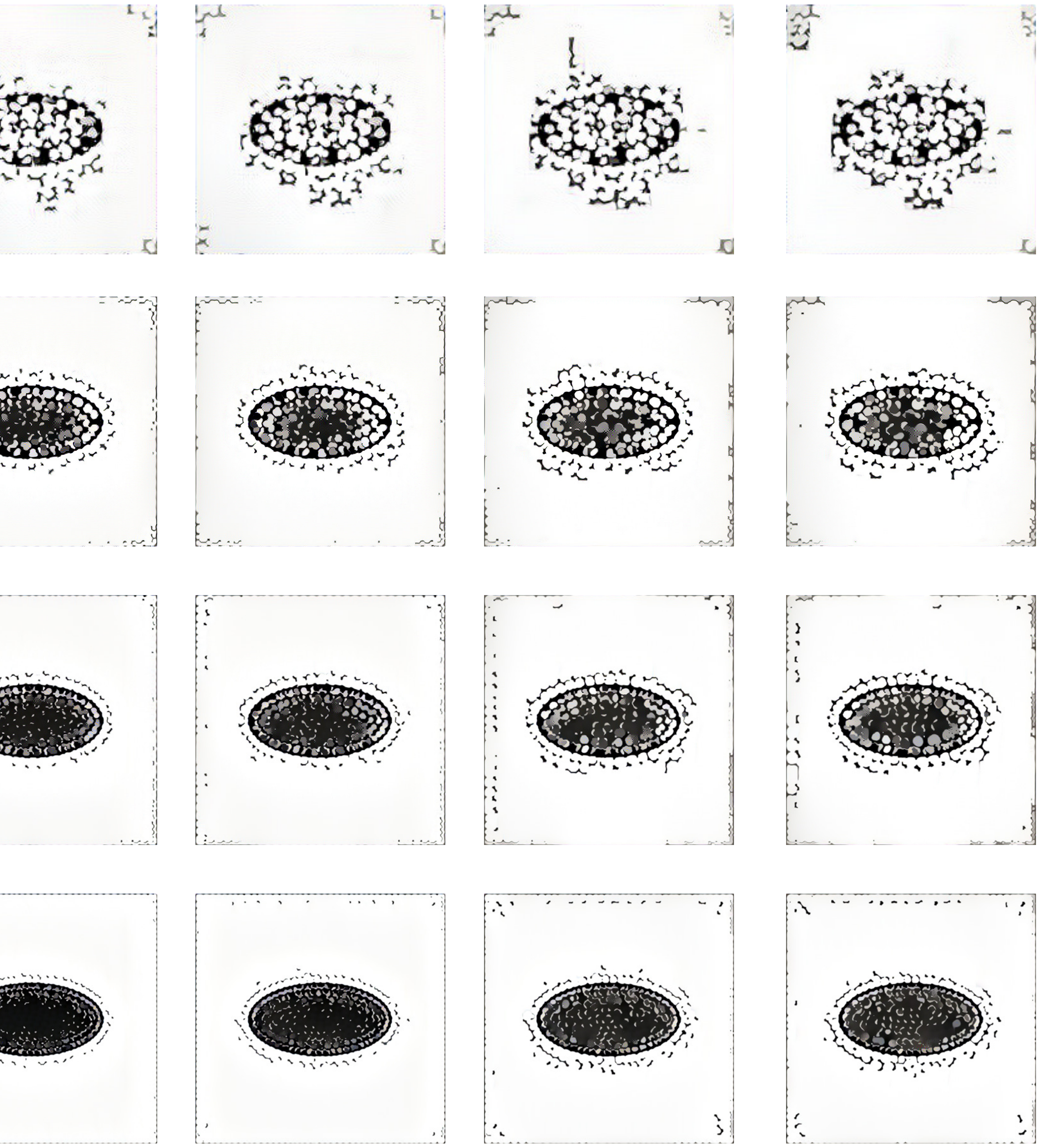


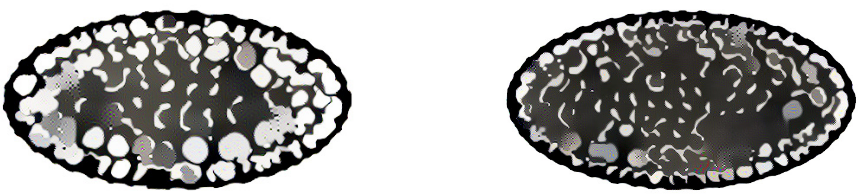
Fig50: Control test search

: C = 50 S : C = 100 S : C = 500 S : C = 1000



700 * 700
S : C = 1000

1024 * 1024
S : C = 1000



n for the input parameters

Result

The transferred pattern blured the boundary on the output images. By listing the index of the white pixel on the input image and change the output pixel with the same index into white, the boundary were fixed. In the 3D reconstruction process, all the black pixels on the output images were voxelized into 3D model. However, the 3D reconstructed model showed a big difference with metallic foam and no longer holds the original geometrical features both on the surface and inside. This experiment did not achieve the hypothesis.

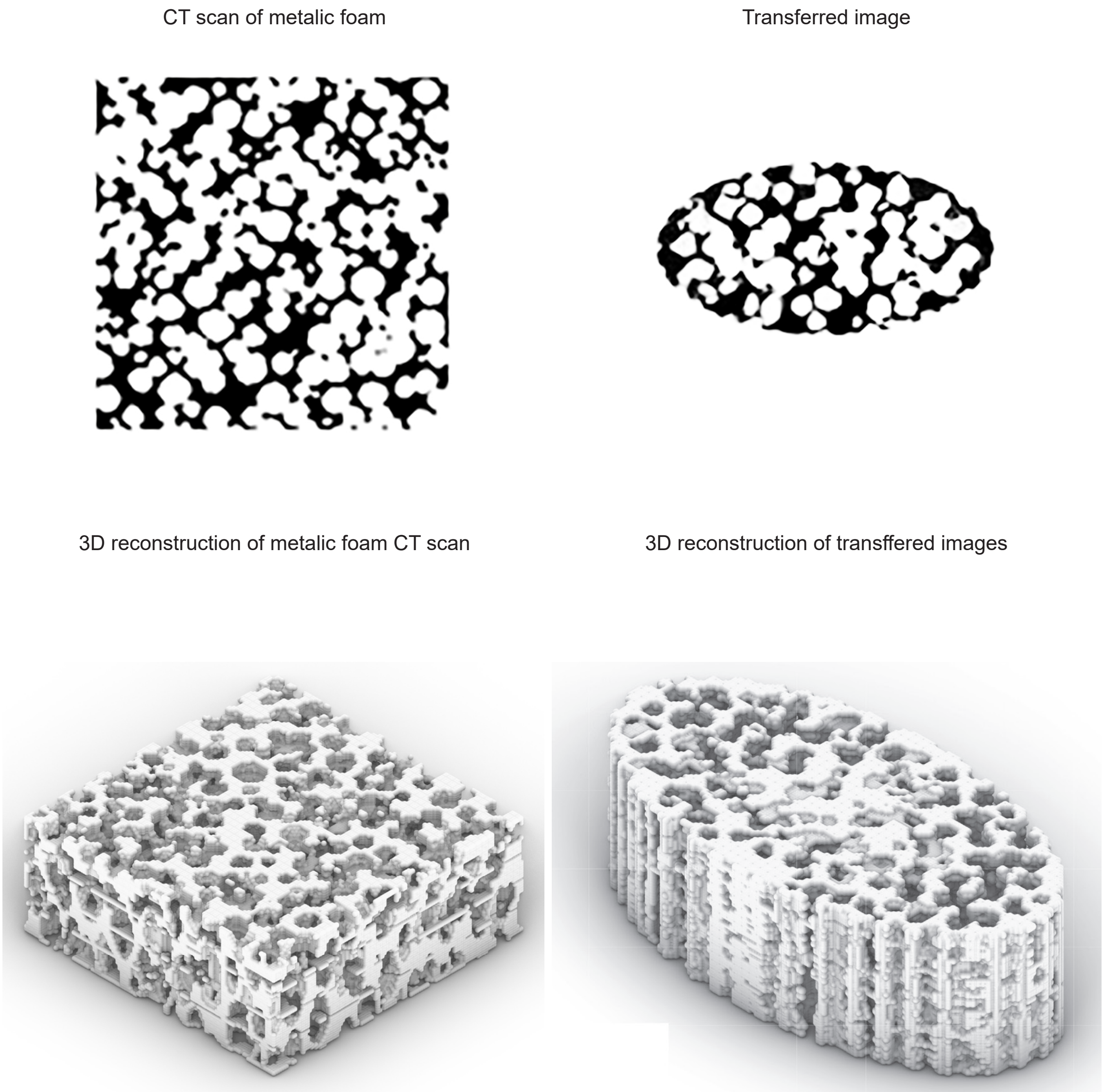


Fig51: Result of the pattern transfer test

3.5.2 Geometry transfer

The second experiment used NST to transfer the church facade features into a curvy wall. The selected church facade has complex geometrical principles, which refine the design outcome with a good ratio and visual experience. Meanwhile, a large amount of details on the church facade shape void and shadows to enhance the outer expression.

The concept of this experiment did not focus on ornamentation and cultural aspects. Instead, focusing on the geometrical features that bring quality to the church facade. The experiment tested whether NST could use the neural network to transfer the church facade features into a new object.

Method

This experiment sliced the facade of Basilica di Santa Maria Della Salute into 100 images and transferred the features of each style image into the content. The final result uses the same method in 3.5.1 to reconstruct the output image sequence into a 3D model.

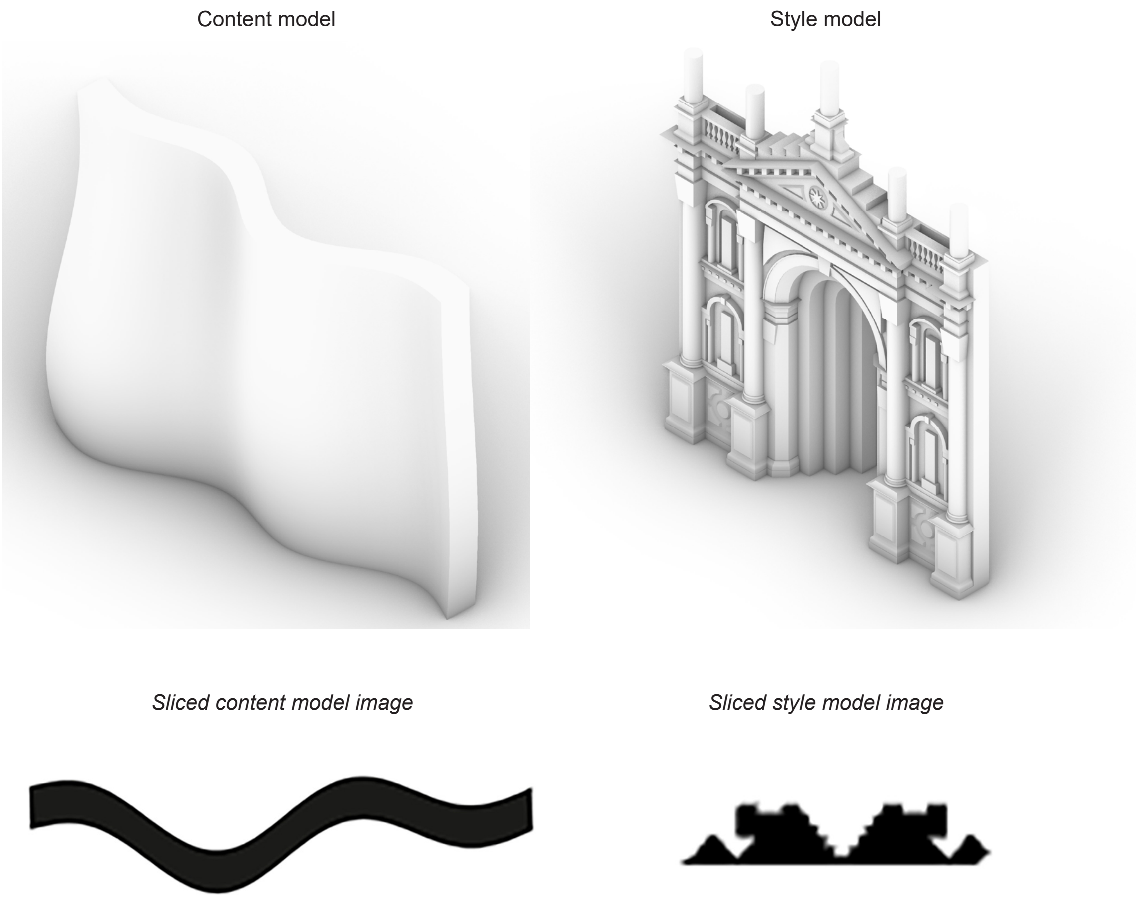


Fig52: Sliced section image of content and style model

Result

The output image showed NST-transferred zigzag form features from Basilica di Santa Maria Della Salute into the content image. However, all the visual elements on the basilica's facade were no longer to be identifiable in the reconstructed model. The increased form complexity from the reconstructed model is very hard to be evaluated.

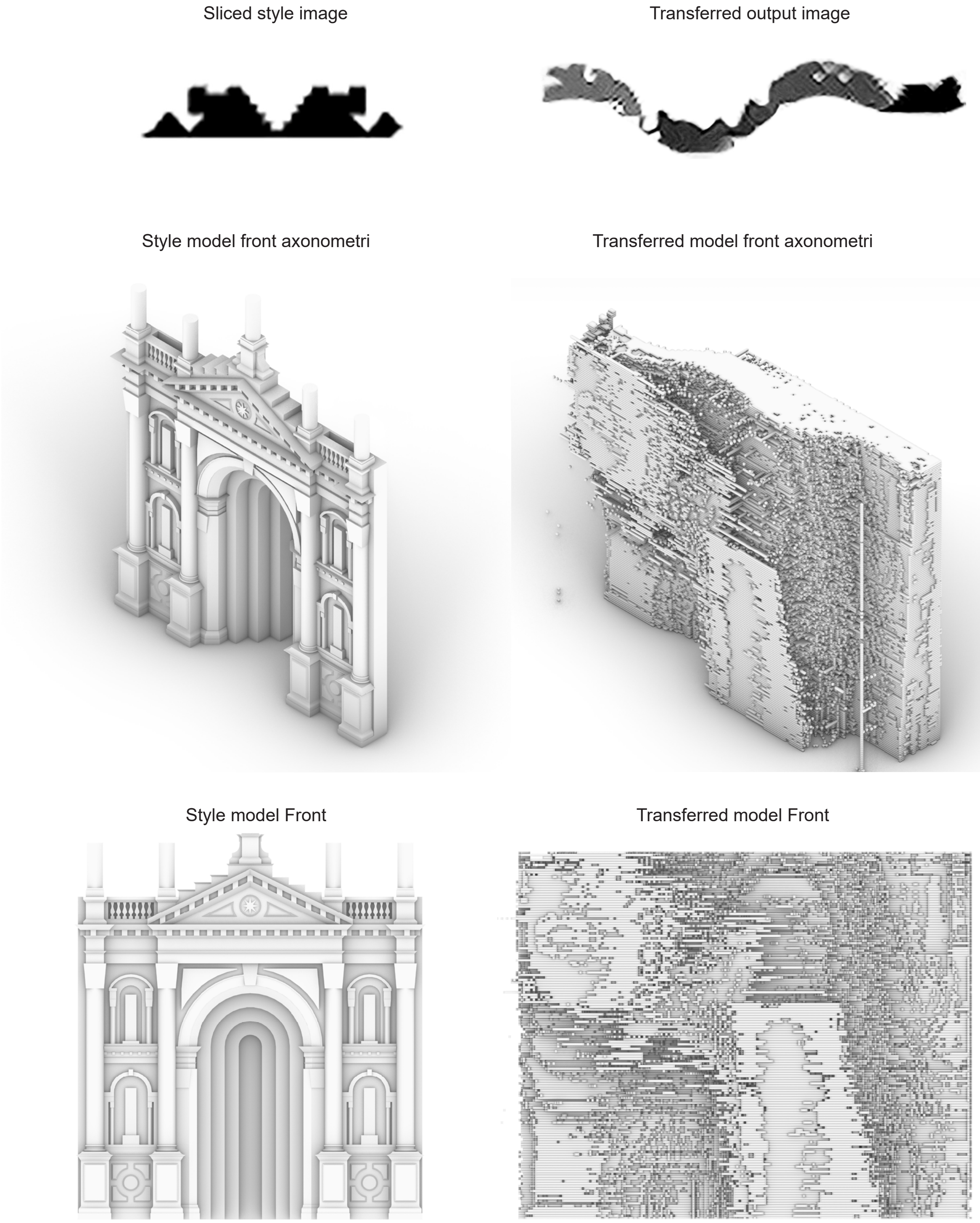


Fig53: Comparison of 3D reconstructed model and church facade

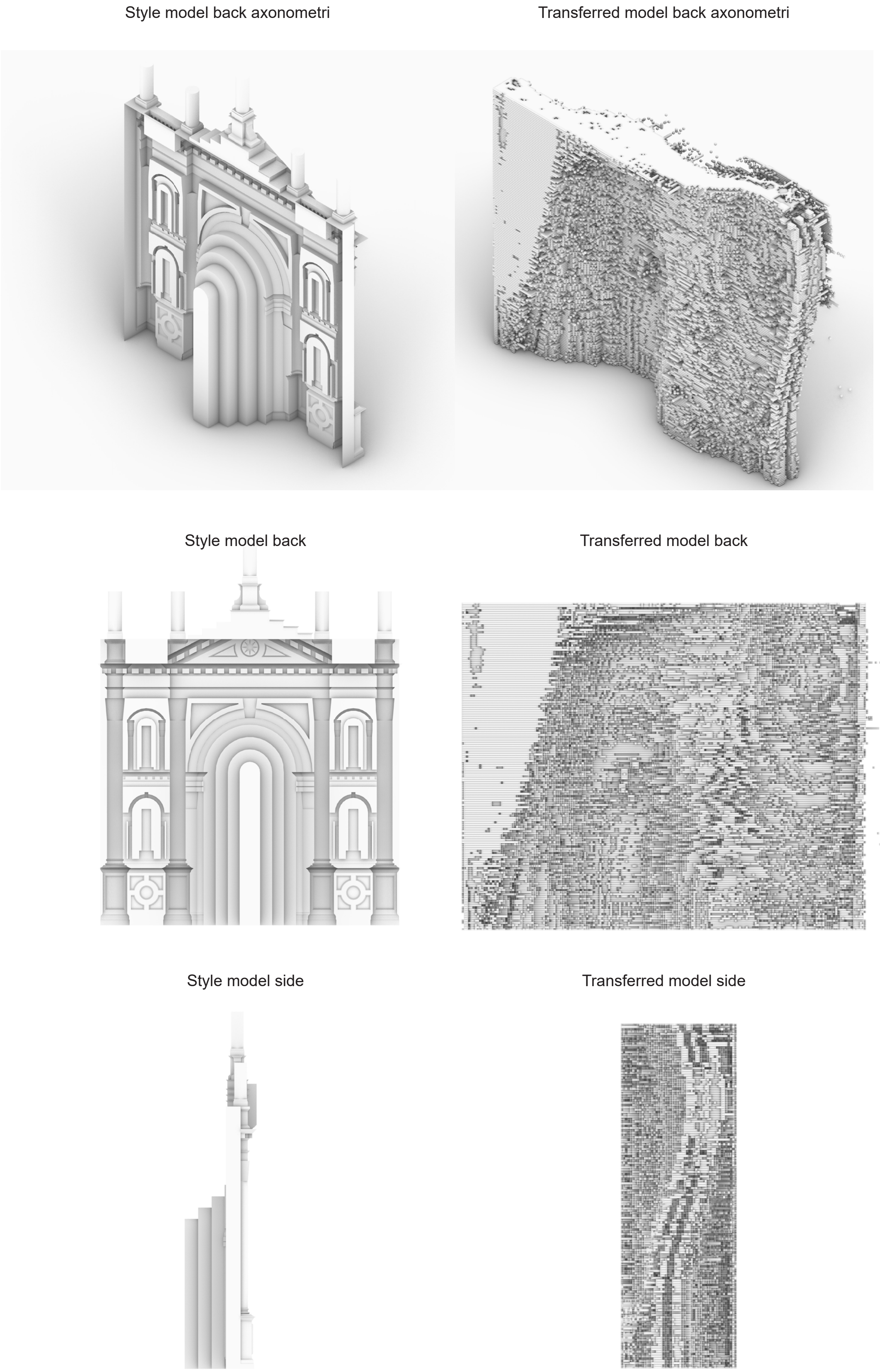


Fig54: Comparison of 3D reconstructed model and church facade

4. Conclusions:

4.1 The new relationship between architect, image, and tool.
The process of my experiment has shown that image-based machine learning brings a new relationship that architects use images to train algorithms for specific tasks. My workflow in this process consists of three phases: preparation, training and generation.

In the preparation stage, my first role as an architect is to define the machine learning function and consider which parts are decided by humans and which parts are obligated to the computer. The second role is to classify the image that matches the desired function and qualities. The processed dataset from the classified images help the algorithm recognize the data pattern.

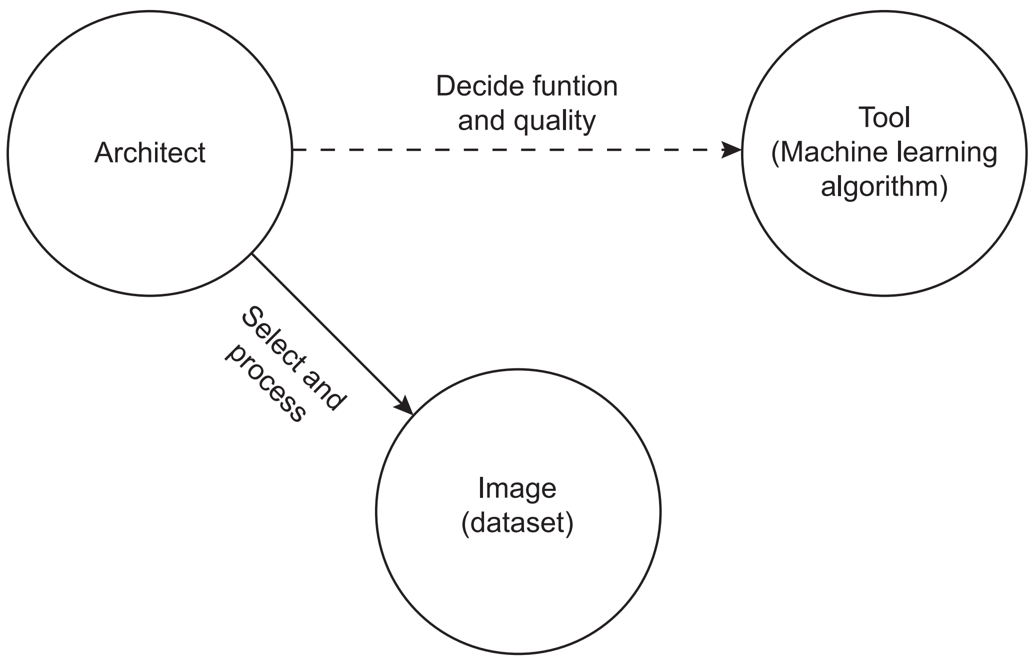


Fig55: Preparation stage

The training process of the machine learning algorithm is automated, but it might not achieve the expected results at once. The architect needs to use expertise to briefly find the pattern from the dataset and use that pattern to label the images to help the algorithm better understand the relationship between input and output. In my experiment, I also found that reducing the weight of the computer decision could significantly improve the training result.

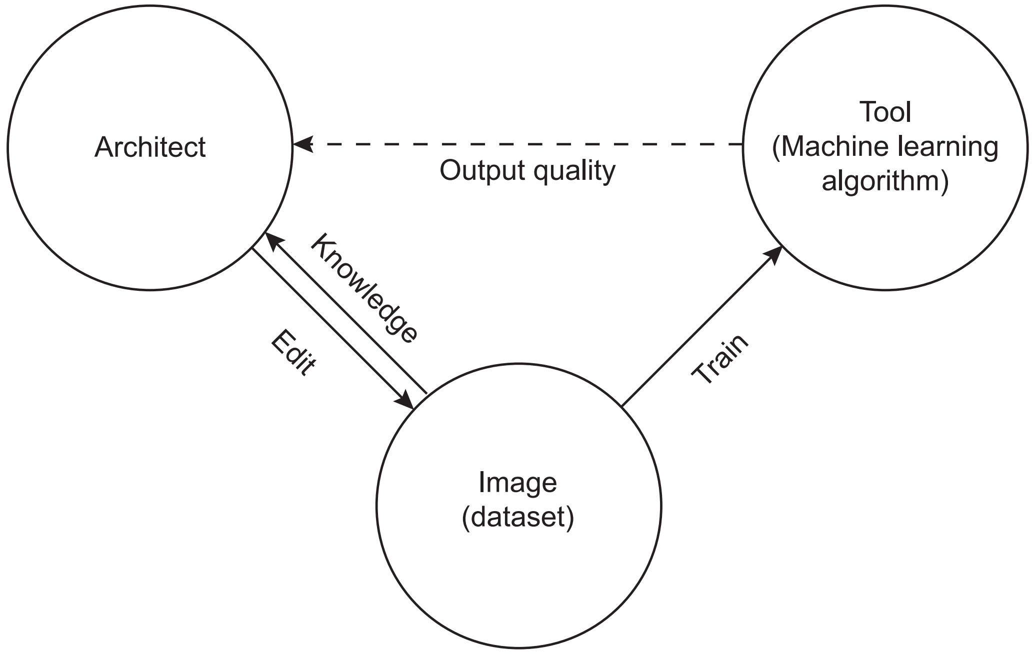


Fig56: Training stage

The generation stage is a cooperation between human and computer decisions. The role of the architect is to process the design decision into an input image and let the algorithm use learned data patterns to translate the input image into an output. Although the output image might reflect a specific design solution, it is still hard to use it as a final design outcome due to insufficient resolution and the conflict with the desired result. In my experiment, I treated the generated output as a suggestion, then refined configurations of wall and column to improve the quality.

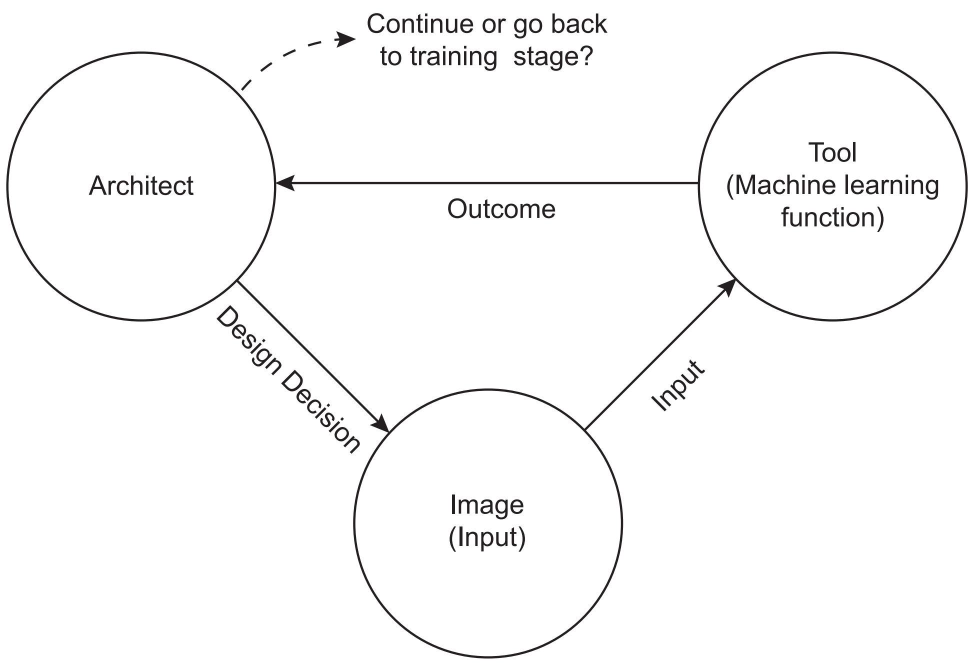


Fig57: Generation stage

4.2 Cooperation between human and computer decision

The trained plan generation tool in my thesis only considered the relation between zoning and spatial layout. Other architectural factors, such as urban context are not involved in this trained function. It needs the participation of human agents in those factors when applying this tool in a full architectural process. Currently, it seems impossible to obligate the entire design process to the computer, and machine learning workflow needs cooperation between human and computer decisions.

Weight of human and computer decision

The early stage of my experiments demonstrated the importance of defining the weight of the computer and human decisions. This weight is influenced by the dataset. In the facade and church generation experiment, the unlabeled information represents the computer decision that cannot be directly controlled by human. The labeled information on input presents the human decision, and the human agent takes responsibility for arranging those labels based on design decisions. The iteration in my experiments showed that increasing the weight of human decisions could significantly increase the learning outcome and make the result more controllable.

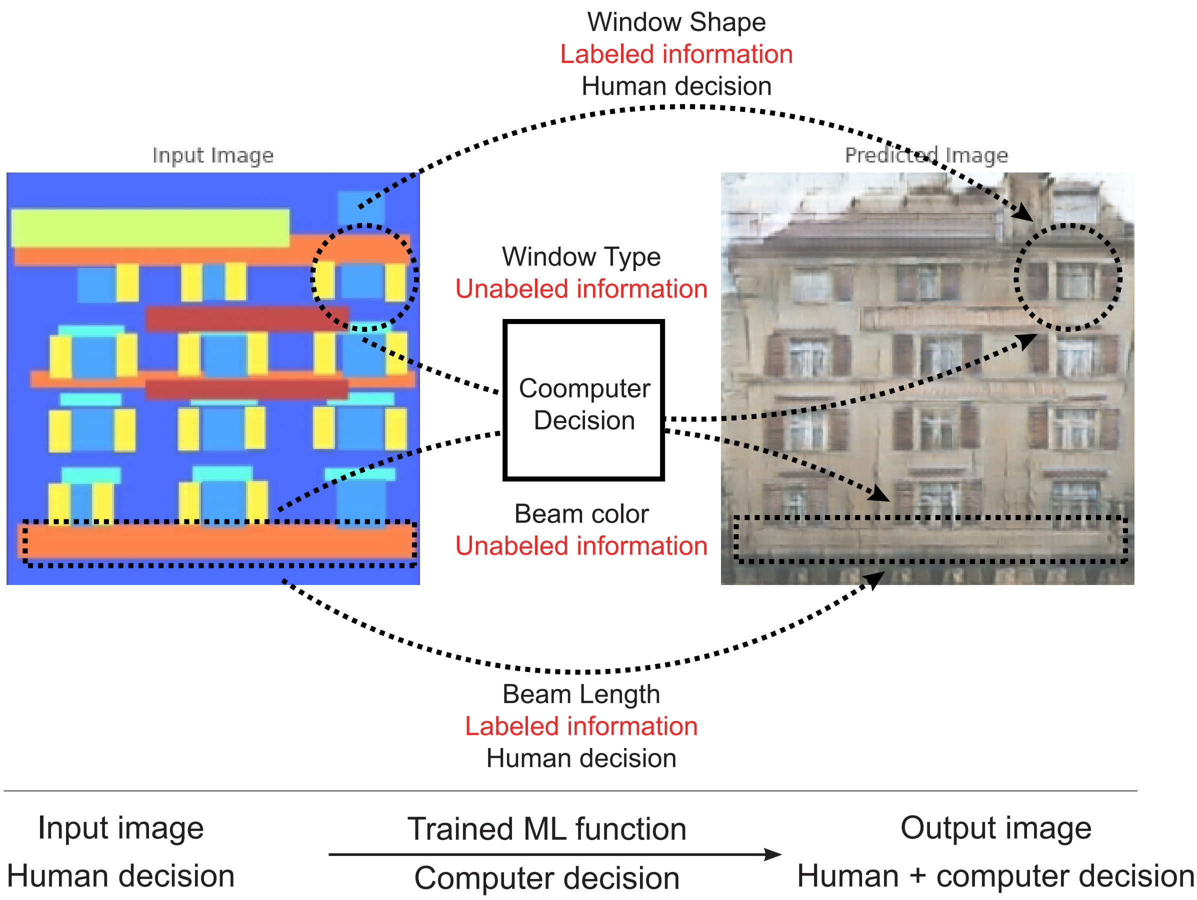


Fig58: Dataset influence the rule of human and computer decision

Accurate solution or a general suggestion

The generated results in my experiments reflected the black-box nature of machine learning that might produce unexpected results or elusive solutions. The trained algorithm might be incapable in some of the situations. In addition, some results in my experiment had insufficient resolutions. It became impossible to directly consider the output image as an accurate church plan, but instead, use it as a suggestion or inspiration tool to guide the plan drawing.

4.3 The future development of machine learning in architecture

Classification

The countless architectural images on Internet has potentional to be used for training the machine functions. To achieve this, the architect must classify the image well.

In the church plan generation experiment, I used church typology to classify the image. However, this classification is very broad and could be further subdivided into temple and basilica. This broad classification in my dataset led the certain types of input generating acceptable results, while some other types of input led to insufficient resolution. The dataset needs to be classified precisely to improve machine learning performance on specific tasks

In addition, the selected old church plan has consistency in the design principles. Those principles result in great spatial qualities. The consistency of the church plan principles helps the algorithm better recognize the data pattern and generate good results. However, the design principle of modern architecture is much more diverse, which results in very different solutions. When using the image of modern architecture as the dataset, it might be imprecise to use typology to make the classification. In the future development of image-based machine learning in architecture, the architect needs to take an essential role in better evaluating and classifying the image.

Furthermore, it is also possible to train machine learning into image classification tools. The dataset of my church experiments used the color map as input to generate the church plan. This could be reversed by using church plan as input and letting the algorithm label the information. Architect could then use the script statistically evaluate the output and find the pattern to better classify the image.

Enlarge the dataset

The experiments in my thesis show that image-based machine learning algorithms typically require more than 1000 images as the dataset. However, some classification methods may have difficulty collecting 1000 images. In the game and media industries, machine learning engineers rotate and erase images to increase the dataset. However, these methods may not be suitable for architectural images as changing the orientation can significantly impact solar radiation. So architects need to work with computer scientists to develop new algorithms to enlarge the dataset. Alternatively, a database could be created to encourage architects to not only upload their works or collected images according to classification methods, but also to evaluate the images in the database to ensure the quality of design solutions.

Scale issues

The dataset for the church experiment was not coordinated the scale. The huge scale differences in some cases can led limited pixel information . This makes it difficult to make accurate measurements when making 3D models.

As for the results, the temple set has good training and the output shows high resolution. In contrast, due to the basilica set having limited pixel information on interior elements, the machine learning algorithm did not fully recognize the data pattern and led to the result with low resolution. Computer scientists might need to improve the maximum resolution of dataset images and optimize the training speed with large pixel information.

5. Reflections:

In the early stage of my thesis, I read an article called Problem-solving and Problem-worrying by Stanford Anderson in 1966. This article discussed the problem-solving and problem-worrying attitudes of the architect. I think the idea in this article could be a good reference to discuss machine learning in architecture.

In the building process, architect normally needs to face specific problems, such as designing an apartment floor plan that meets the building regulations. To solve those problems, architects need much experience from building practice and spending time making design strategies. The ongoing machine learning technology could be a problem-solving tool that quickly learns from past experience and apply the learned principle to generate solutions. This problem-solving ability of machine learning relays on a well-defined problem which could guide the architect to classify the dataset accurately. However, there are also unwell-defined problems or concepts in architecture, such as designing a library that contributes to the community. Apparently, the machine learning algorithm cannot use sociological and architectural knowledge to consider what is a good community and how to use a library to improve the community. It is even hard for architect to find the appropriate way to classify the dataset. To work with the unwell-defined problem or concept, Anderson (1966) mentioned the problem-worrying attitude that architects research the problem and raise a better problem. This problem-worrying ability is exactly what the machine learning algorithm lack of. So I do not believe machine learning will replace architects.

In addition, despite the current machine learning algorithm cannot problem-worrying, I believe the problem-solving ability of machine learning could assist architects in problem-worrying. For example, it could be used as a research tool that finds the principle from architectural-related data and allows the architect to interact with learned principles to discover the new potential. Like my church experiment, although the generated result could not treat as a solution in architectural practice, it helped me to learn the church design method and test the potential. In the transformation experiment, the generated result suggested circulation and wall geometry. This may raise a problem-worrying about the potential of transferring the features of religious buildings into modern architecture. Perhaps Louis kahn's work already reflected the potentional of this concept, but how could we do better with the current machine learning technology, and how could we explore more hidden principles through machine learning technology.

Overall, machine learning technology could be a problem-solving tool for the architect to tackle specific and well-defined problems. Meanwhile, it also brings the value of helping architects take the problem-worrying attitude in architectural practice.

6. Acknowledgments:

I would like to thank my supervisor Elin Daun for helping me to organize my thoughts and giving me great suggestions to improve my thesis. With Elin’s patient guidance, I garbed practical research and experiment skills. Thank my examiner David Andréen for guiding the direction of my experiments and giving crucial advice on my report.

I would also like to thank Pablo Miranda for encouraging me to take the problem-worrying attitude to explore machine learning in architecture and give me great advice to finalize my works. Also thank my critics Anton Tetov Johansson and Petra Jenning for leading the great discussion in my final presentation, it helped me to consider the future improvement of image-based machine learning functions in architecture.

7. References:

Phil Bernstein(2022) Machine Learning Architecture in the age of Artificial Intelligence

July Shao(2022), Solution Generation, Cloud-based Editing, Automatic Drawing... the Junior AI Assistant for Architects
https://www.archdaily.com/974602/

Ian J. Goodfellow(2014), Generative Adversarial Networks

Brian Caulfield(2017), Vincent AI Sketch Demo Draws In Throngs at GTC Europe
https://blogs.nvidia.com/blog/2017/10/11/vincent-ai-sketch-demo-draws-in-throngs-at-gtc-europe/

Hao Zheng(2020), Apartment floor plans generation via generative adversarial networks

Leon A. Gatys(2015), A Neural Algorithm of Artistic Style

Matias del Campo(2020), A Question of Style

Phillip Isola(2017), Image-to-Image Translation with Conditional Adversarial Networks

Wassim Jabi(2017), Complex geometrucal relations on church plan: Geometry, Light, and Cosmology in the Church of Hagia Sophia

Steven Fleming(2006), Timely Timelessness: Traditional Proportions and Modern Practice in Kahn’s Kimbell Museum

Stanford Anderson(1966), Problem-Solving and Problem-Worrying