Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 9, No. 4, November, Pp. 375-388

Layout generation: automated components placement for advertising poster using transformer-based model from layout graph

Aisyah Dliya Ramadhanti[1], Kemas Rahmat Saleh Wiharja[*1], Azmi Nurzakiah[2], Yoga Yustiawan[2]
School of Computing, Informatics, Telkom University, Bandung, Indonesia[1]
Digital BRIBRAIN Department, Digital Banking Development and Operation Division, Bank Rakyat Indonesia (BRI), Jakarta, Indonesia[2]

Article Info
Keywords:
Layout Generation, Transformer-Based Model, Layout Graph, Advertising Poster, Automated Components Positioning

Article history:
Received: June 05, 2024
Accepted: July 21, 2024
Published: November 30, 2024

Cite:
A. D. Ramadhanti, K. R. S. Wiharja, A. Nurzakiah, and Y. Yustiawan, "Layout Generation: Automated Components Placement for Advertising Poster using Transformer-based from Layout Graph", KINETIK, vol. 9, no. 4, Nov. 2024.
https://doi.org/10.22219/kinetik.v9i4.2035

*Corresponding author.
Kemas Rahmat Saleh Wiharja
E-mail address:
bagindokemas@telkomuniversity.ac.id

Abstract
In the digital era, graphic design plays an important role in a company's marketing strategy, especially advertising posters that can convey messages to the audience. However, the process of creating attractive and informative posters takes a long time, especially the component placement on the layout. This research aims to develop a layout generator system that automatically places components on the layout using one of the transformer-based models. The transformer-based model used is a Graph Transformer with edge features called SGTransformer, which accepts input data as a graph. SGTransformer consists of several graph transformer layers that will calculate the attention of node and edge features on the input layout graph. A layout graph describes the spatial relationship between components in a layout. The SGTransformer model was trained by using advertising poster datasets collected from social media. The performance of the model were evaluated using the evaluation metrics commonly used in the layout generation domain such as Alignment, Overlap, Max IoU, and FID. The scores obtained from each evaluation metric are 0.025, 1.274, 0.325, and 8.575 respectively. The model evaluation results show that SGTransformer can produce structured and more diverse layouts although there are still challenges such as overlap between components. Code and other materials will be released at https://github.com/syahdeee/Layout-Generator.

1. Introduction

The development of technology in the current digital era has made significant progress, which has an impact on increasing sophistication in various aspects of life. Currently, companies are striving for excellence to compete in an increasingly globalized marketplace. Graphic design plays a very important role in marketing strategies for many companies, one of which is in advertising. In the field of advertising, companies use advertising posters to promote a product. The poster acts as a medium to convey information to the audience [1]. Advertising posters are often considered a simple promotional medium. However, graphic designers need a lot of time to complete many poster designs.

There are several important aspects in creating a poster design, one of which is the layout [2]. The arrangement of information in a poster can influence the reader to navigate the message that the poster will convey. A cluttered layout makes it difficult for readers to navigate the information [2]. Therefore, the placement of poster components such as images, titles, subtitles, or other components is very important to be considered by graphic designers because it can affect the layout of a poster. A good layout will draw the audience's attention to the important information in a poster. However, manually placing the poster components takes a considerable amount of time. This poses a challenge for graphic designers to produce attractive poster layouts in large quantities. Therefore, this limitation requires a solution that can automate the layout design process.

Some companies that require advertising posters for promotion will need a system that can automate the layout design process, especially in terms of component placement. This system is used to produce large quantities of posters efficiently. In recent years, a growing number of generative modeling-based methods have become a solution to this challenge**.** Generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have successfully generated diverse and high-quality layouts [3], [4], [5], [6]. LayoutVAE and LayoutGAN are the main methods that utilize GAN and VAE to produce a graphic layout or scene [3], [4], [6]. LayoutGAN is the first approach that applies a generative model (GAN) to generate a layout [3]. The LayoutVAE method proposes an autoregressive method based on Conditional VAE by using Long Short-Term Memory (LSTM) to collect information related to the predicted bounding box [4]. LSTM cannot model the relationships of all components explicitly, so LayoutVAE has difficulty generating layouts with many components. In addition, the method used in the content-aware layout generator utilizes GAN to model complex layout distributions and proposes a semantic embedding network to encode multi-modal

contents and structural/categorical attributes in the design [5]. Although these approaches are successful in producing realistic layouts, the methods used have limitations in modeling the spatial relationships between components contained in a graphic design layout. This causes the method unable to place components in the layout based on spatial relationship information between components in the design.

The Neural Design Network (NDN) method is one of the successful methods in generating layouts in graphic design by placing components according to the component labels and relationships between components in graphic design layouts [7]. The NDN method represents the position relationship of components in a graph and uses a Graph Neural Network based on Conditional VAE to generate a layout [8]. The first step in the NDN method is to build a complete graph to represent the relationships between all the components in the layout. The distribution of relationships between these components is studied using VAE based on Graph Convolutional Network (GCN) [9]. The relationship labels between the components are extracted using heuristic rules to learn the layout distribution. This makes NDN susceptible to ambiguity in learning the layout distribution, hindering the model's performance in generating accurate layouts [9].

In addition to successfully generating layouts in graphic design, another generative model-based method was used to control the position of objects in a natural scene image [10], [11]. To generate a layout in a nature scene image, the graph can be used as a scene description to control the composition of the resulting image [11]. Previous research proposed a scene graph to describe the relationship between objects in the scene where nodes represent objects in the scene and edges represent spatial relationships between objects [10], [11]. The SG2IM model successfully performs layout generation for natural scene images by processing the scene graph as input [10]. In addition, SGTransformer model is also successful in generating a scene layout based on the generalization ability of the transformer to process the scene graph through a multi-head attention mechanism [11]. By using a scene graph as input, these models can generate a scene layout containing many objects and relationships between objects [10]. These models have been successful in organizing the position of objects in the scene layout but are still not widely explored for layout in graphic design.

This research aims to produce a layout generator system that focuses on the placement of components in the poster layout. The placement of these components will be organized by a layout graph. Layout graphs are used to explicitly describe the necessary components and positional relationships between components contained in a poster layout. Nodes represent the necessary components while edges represent the spatial relationship between components [7]. This research uses the SGTransformer model which will be trained on the advertising poster dataset. SGTransformer has a good ability to understand the structure and geometric relationships in the graph so that it can produce a structured layout [11]. This model generates a layout by calculating the attention on neighboring nodes and edge features in the graph.

Our contribution to this research is to collect advertising poster datasets from social media. Then, the dataset is used to train the SGTransformer model which was previously only used on natural scene image datasets [11]. In the previous research, SGTransformer was trained so that it could produce the layout of a natural scene image based on the input scene graph**.** In this research, we focus on producing layouts in graphic design specifically for advertising poster layouts so that we conduct SGTransformer training on the dataset that we have collected. This research consists of five stages, namely, data preparation, data preprocessing, building layout graphs, model training, and evaluation. The training stage is carried out to train the model used to produce a layout that can be organized using a layout graph. In the end, it will be evaluated how well the SGTransformer model performs in placing components to produce a structured and quality poster layout. The SGTransformer model evaluation results will be compared with SG2IM [10] to see how well the SGTransformer model performs in generating layouts.

2. Research Method

The research was conducted through five stages, namely data preparation, data preprocessing, building layout graphs, model training, and evaluation. The research workflow can be seen in Figure 1. Research Workflow. Based on the research workflow in Figure 1. Research Workflow, the first stage is data preparation. In the data preparation stage, the process of collecting posters and poster annotation was carried out. We collected some advertising posters uploaded on some companies' social media. The posters were annotated to obtain data related to the bounding box of each component and the size of the poster image. The next stage is data preprocessing. At this stage, each bounding box was normalized. This is because each poster still had different size $(width, height)$ [12]. Then, the process of building a layout graph on the normalized bounding box data were carried out. At this stage, a layout graph was generated which would be input into the model [10], [11]. After the layout graph was successfully built, the data training process was carried out until the best model is obtained. At the end of the process, an evaluation was carried out using several evaluation metrics to measure the performance of the model.

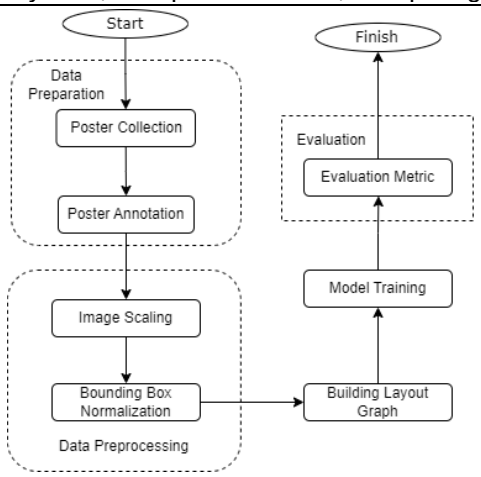

Figure 1. Research Workflow

2.1 Data Preparation

In the data preparation stage, the poster collection and poster annotation processes were carried out. The stages of the process carried out in data preparation can be seen in Figure 2.

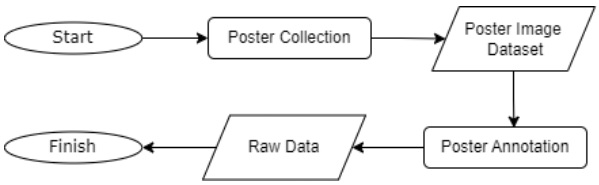

Figure 2. Data Preparation Workflow

In the poster collection stage, we collected 1413 poster images uploaded on social media to promote specific products. All the poster images were taken from Instagram and Facebook using one of the Chrome extensions, Esuit Extension Photos Downloader. Using this extension, we were able to download all the photos from Instagram or Facebook in bulk. Then, we prepared the collected poster images into a dataset using one of the tools that is often used to complete computer vision tasks, Roboflow. Roboflow is one of the popular tools used to annotate images [13]. This poster annotation process produced raw data. The raw data is the annotated data in the form of image size and bounding box information and labels for each component on the poster. The resulting bounding boxes have the format $[x, y, w, h]$ where (x, y) is the coordinate of the center point of the bounding box and (w, h) is the width and height of the bounding box.

2.2 Data Preprocessing

There are two stages of the process in data preprocessing, namely image scaling and normalization of bounding boxes. The process at this stage is described in Figure 3.

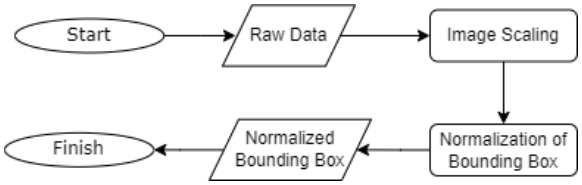

Figure 3. Data Preprocessing Workflow

At this stage, the raw data generated in section 2.1 were subjected to image scaling. Image scaling is one of the main processes that need to be done in the image preprocessing task [14]. Image scaling is used to resize the images [15]. This is because the poster sizes $(width, height)$ in the raw data were different [12]. The image scaling method used in this research was downscaling method. The downscaling method on images can reduce the size of the image to obtain a smaller number of pixels [16]. In this research, each poster is reduced in image size to $64 \times 64\ pixel$. The downscaling process on posters can be seen in Figure 4.

Figure 4. Downscaling Image Processing

Next, the normalization of the bounding box process was carried out. At this stage, each bounding box was normalized so that the bounding box coordinate value is in the range $[0,1]$. This stage is very important because most machine-learning algorithms require a consistent input size [12]. The normalization of the bounding box process in image processing has been done in previous research [17]. An illustration of the normalization of the bounding box process can be seen in Figure 5.

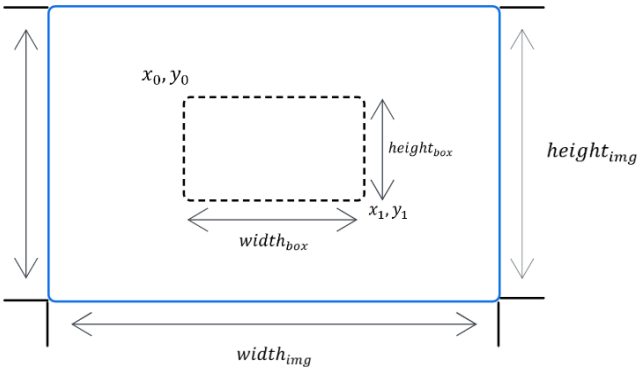

Figure 5. Normalization of Bounding Box

Based on Figure 5, the width and height of the image are expressed by $width_{img}$ and $height_{img}$. The dashed line represents the bounding box to be normalized. The width and height of the bounding box are expressed by $width_{box}$ and $height_{box}$. In the normalization process of the bounding box, we first normalized the width and height of the bounding box using Equation 1 and Equation 2. Then, we obtained the normalization result of each point in the bounding box $[x_0,\ y_0,\ x_1,\ y_1]$ using Equation 3 and Equation 4. By going through each process at this stage, we obtained the result in the form of normalized bounding box data.

$$\text{Normalized}_{\text{width}} = \frac{\text{width}_{\text{box}}}{\text{width}_{\text{img}}} \quad (1)$$

$$\text{Normalized}_{\text{height}} = \frac{\text{height}_{\text{box}}}{\text{height}_{\text{img}}} \quad (2)$$

$$\text{Normalized}_{\text{x}} = \text{x} \times \frac{\text{Normalized}_{\text{width}}}{\text{width}_{\text{img}}} \quad (3)$$

$$\text{Normalized}_{\text{y}} = \text{y} \times \frac{\text{Normalized}_{\text{height}}}{\text{height}_{\text{img}}} \quad (4)$$

2.3 Building Layout Graph

The layout graph created in this research is based on the principle of the scene graph found in the previous research [10]. The layout graph describes the spatial relationship between components in the poster as a directed graph, where nodes represent the required components while edges represent the spatial relationship between components in the poster [7]. The graph layout formation stage is described in Figure 6.

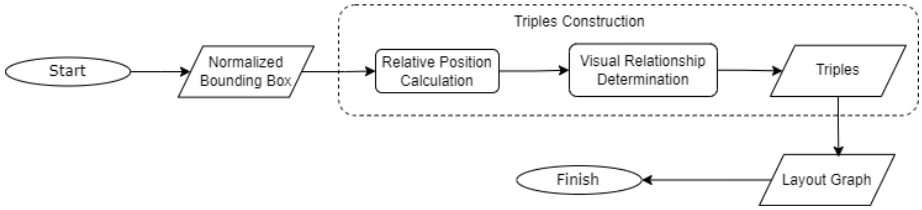

Figure 6. Building Layout Graph Workflow

At this stage, the dataset used to build the layout graph is the normalized bounding box value of each component. Then, the bounding box information can be used to determine the spatial relationship between two components in a layout. The spatial relationship was represented by a triple consisting of $< subject, relation, object >$ [18]. Based on the basic principles of the scene graph, a set of successfully formed triples will build a layout graph [19]. In triples format, the first component is represented as $subject$ and the second component is represented as $object$. While the $relation$ in the triples will represent the spatial relationship between objects [15].

In the triple formation stage, the relative position is the position between the first component $subject$ and the second component $object$. To determine the relative position, it is necessary to calculate the distance vector between the bounding box center of the $subject$ components and the bounding box center of the $object$ component. Then, the relative position that has been determined can be used to determine spatial relationships such as surrounding, inside, left of, above, right of, or below [10]. The information obtained in the spatial relationship determination stage can be used to build triples with the format $< subject, relation, object >$. Then, the triples that have been formed will be used to build the layout graph [19]. An illustration of the triple construction process can be seen in Figure 7.

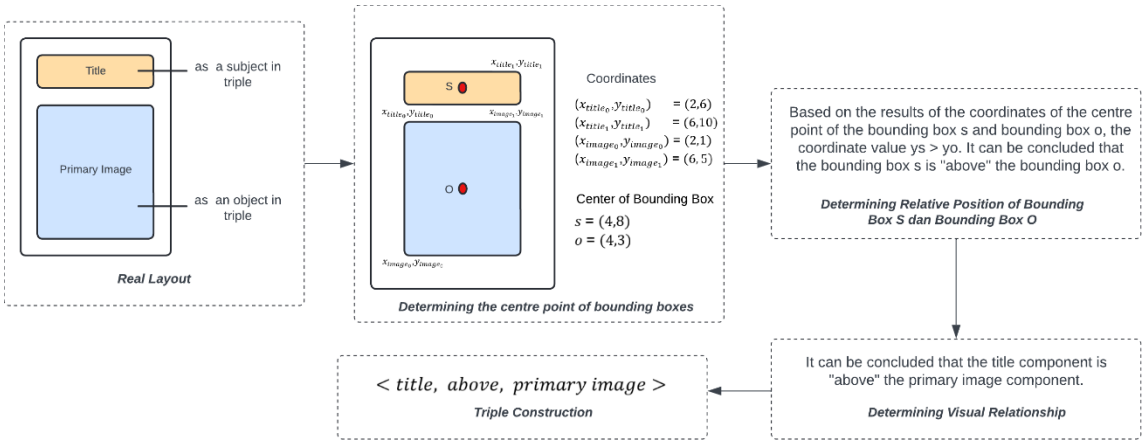

Figure 7. Illustration of Triples Construction

2.4 Model Training

The dataset used at this stage is the layout graph generated in section 2.3. In this research, we use the Deep Graph Library (DGL) to help process graph-structured data [20]. Before the training process is carried out, data splitting is required which divides the data into training datasets, validation datasets, and testing datasets. The model training process can be seen in Figure 8.

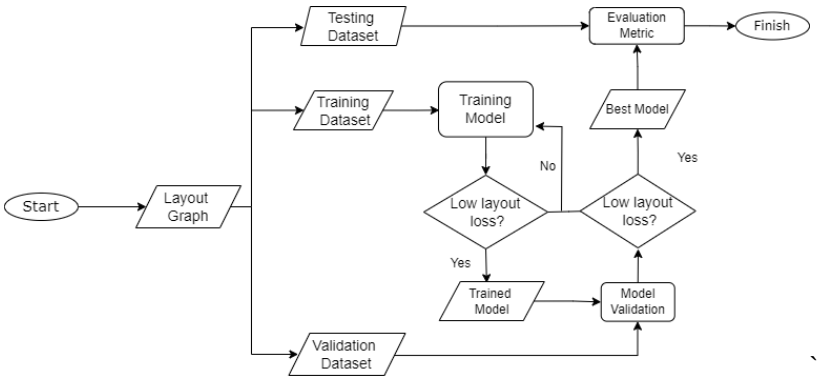

Figure 8. Model Training Workflow

Based on Figure 8, the model was trained using the training dataset. In this research, the models used to train the training dataset are SGTransformer [11] as the main model and SG2IM [10] as the comparison model. Then, the best training model will be evaluated using several evaluation metrics that will be discussed in section 2.5.

2.4.1 Layout Graph to Layout with SGTransformer

We used SGTransformer architecture which is an implementation of the graph transformer model [21]. In the previous research, SGTransformer succeeded in generating scene layouts based on scene graph input by calculating attention on neighboring nodes in the graph and then predicting the bounding box coordinates of each object in the scene layout [11]. SGTransformer receives input in the form of a graph. This graph is represented by the layout graph described in section 2.3. The layout graph describes the spatial relationship between components in the layout in the form of a directed graph $G(V, E)$. The components in the graph are represented as nodes and the spatial relationships are represented as edges. Nodes and edges will be projected into a higher dimensional space so that each node i is represented with a feature vector $h_i \in \mathbb{R}^{dn}$ and its neighboring node h_j. The edge between nodes h_i and h_j is represented by the feature vector e_{ij}.

SGTransformer acts as a graph encoder, then the output of SGTransformer will proceed to the multilayer perceptron (MLP) architecture to predict the bounding box coordinates of each component in the layout. The SGTransformer architecture consists of several block stacks where each block will perform multi-head attention calculations on each node h_i and its neighboring nodes h_j. In the attention calculation, edge features e_{ij} are also considered because edge features contain important information about the relationship between components. For each node h_i, the multi-head attention mechanism is defined in Equation 5.

$$\text{Attention (Q, K, V, E)} = \text{soft max}\left(\frac{\text{h}_\text{i}\text{W}^\text{Q} \,.\, \text{h}_\text{j}\text{W}^\text{K}}{\sqrt{\text{d}_\text{k}}}\right) .\, \text{h}_\text{j}\text{W}^\text{V} .\, \text{e}_\text{ij}\text{W}^\text{E} \tag{5}$$

Based on Equation 5, W_k^Q, W_k^K, W_k^V, $W_k^E \in \mathbb{R}^{d_k \times d}$ are learnable projection matrices, $k \in \{0, \ldots, N_{heads}\}$ and $d_k = \frac{d}{N_{heads}}$ [11]. Since SGTransformer is composed of several graph transformer layers, each layer will calculate the attention value on each node using Equation 5. In Equation 5, the node h_i is used as a query (Q). While node h_j is the neighbor of the node h_j which will be mapped as a key-value pair. Then, the edge features e_{ij} will be multiplied by the product of the query (Q) and key (K) before the softmax operation [21]. The result of this operation can be used to update edge features.

Since SGTransformer acts as a graph encoder, the output of SGTransformer will be continued to the two MLP heads architecture which will be able to predict the bounding box value for each component. We trained the SGTransformer model using the layout loss L_{layout} defined by Equation 6.

$$\text{L}_\text{layout} = \text{L}_\text{box} + \text{L}_\text{iou} \tag{6}$$

Based on Equation 6, the layout loss L_{layout} is the total of L_{box} and L_{iou} , where L_{box} is the L_2 at the bounding box coordinates [22] dan L_{iou} is the distance IoU loss [23] .

2.4.2 Layout Graph to Layout with SG2IM

As the comparison, we applied a model that has been proposed in the previous study [10] which used the SG2IM model based on a graph convolutional network consisting of several graph convolutional layers. In the graph convolutional layer, given a graph input with vectors of dimension D_{in} at each node and edge, it will produce a new output vector of dimension D_{out} for each node and edge. The resulting output vector depends on the local information from the vicinity of the connected input vectors [24]. The working principle of the graph convolutional layer [10] is illustrated in Figure 9.

Based on Figure 9, the model receives input vectors $v_i, v_r \in \mathbb{R}^{D_{in}}$ for each object $o_i \in O$ and edge $(o_i, r, o_j) \in E$. Then, three functions g_s, g_p and g_o will be used to calculate the output vectors v_i', $v_r' \in \mathbb{R}^{D_{out}}$. The functions g_s, g_p dan g_o accept input in the form of triples of vectors (v_i, v_r, v_j) and produce the output subject o_i, predicate r , and object o_j. To calculate the output vector v_r' for each edge where $v_r' = g_p(v_i, v_r, v_j)$. The output vector v_i for object o_i is calculated by $v_i = h(V_s^i, V_o^i)$, h is a symmetric function that combines a set of input vectors into a single output vector. Then, the output of the process will be forwarded to the multilayer perceptron (MLP) which will produce bounding box coordinates for each component [10].

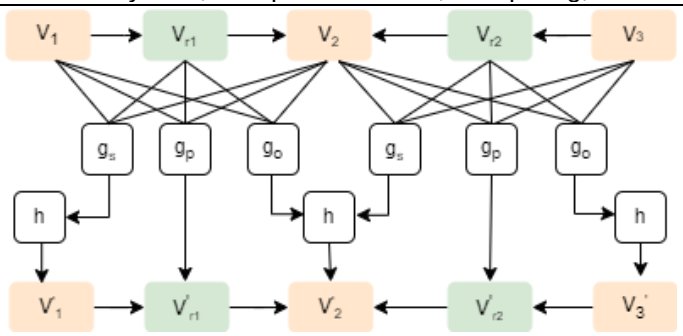

Figure 9. Computational Processing of a Single Graph Convolutional Layer

2.5 Evaluation Metrics

We performed an evaluation using four metrics namely, alignment, overlap, maximum IoU, and FID to measure the quality of the generated layout. For alignment, overlap, and FID score metrics, the lower the score, the better [7], [26]. Whereas for maximum iou, the higher the score, the better [28].

2.5.1 Alignment

Alignment is one of the important principles in creating a design because it affects the audience's perception of the layout [7]. The components in a good design must be in center alignment or edge alignment [25]. Adjacent elements usually have six alignment types, namely left, x-center, right, top, y-center, and bottom aligned. Therefore, it is necessary to measure the alignment between components explicitly [26]. The alignment measurement is defined by Equation 7.

$$\text{alignment} = \frac{1}{\mathrm{N}} \sum_{\mathrm{k}} \sum_{\mathrm{i}} \min_{\mathrm{j,i} \neq \mathrm{j}} \left\{ \min \left(\mathrm{l}\left(\mathrm{e}_{\mathrm{i}}^{\mathrm{k}},\ \mathrm{e}_{\mathrm{j}}^{\mathrm{k}}\right), \mathrm{m}\left(\mathrm{e}_{\mathrm{i}}^{\mathrm{k}},\ \mathrm{e}_{\mathrm{j}}^{\mathrm{k}}\right), \mathrm{r}\left(\mathrm{e}_{\mathrm{i}}^{\mathrm{k}},\ \mathrm{e}_{\mathrm{j}}^{\mathrm{k}}\right) \right) \right\} \tag{7}$$

Based on Equation 7, *N* is the number of generated layouts, e_i^k and e_j^k are the i_{th} component and j_{th} component of the d_{th} layout. Meanwhile $l, m,$ and r are alignment functions where the distances between the left, center, and right components are considered.

2.5.2 Overlap

Based on the design principles, a good layout will avoid overlapping between elements [26]. The overlap metric will calculate the total area of overlap between pairs of bounding boxes in a layout [8], [27].The overlap calculation will be defined by Equation 8.

$$\text{alignment} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{\forall \mathrm{j} \neq \mathrm{i}} \frac{\mathrm{s_i} \cap \mathrm{s_j}}{\mathrm{s_i}} \tag{8}$$

Based on Equation 8, $s_i \cap s_j$ represents the overlapping area between elements i and j. N is the number of elements in the layout.

2.5.3 Maximum IoU

Maximum IoU (Max. IoU) is used to calculate the similarity between the original set of layouts $B = \{b_i\}_{i=1}^N$ dan and the generated layouts $B' = \{b_i'\}_{i=1}^N$ [28]. The similarity calculation between the two layouts $B = \{b_i\}_{i=1}^N$ and $B' = \{b_i'\}_{i=1}^N$ is calculated using Intersection Over Union (IoU After calculating the similarity, optimal matching between B and B' will be performed and the average IoU of the bounding boxes will be calculated based on Equation 9.

$$gIoU(B, B', L) = \max_{\pi \in S_N} \frac{1}{N} \sum_{i=1}^{N} IoU\left(b_i\ ,\ b'_{\pi(i)}\right) \tag{9}$$

Based on Equation 9, IoU (. , .) is used to calculate the IoU between bounding boxes. To evaluate the similarity between the generated layouts $B = \{B_m\}_{m=1}^M$ and real layouts $B' = \{B'_m\}_{m=1}^M$ requires calculating the average similarity with optimal matching based on Equation 10.

$$\mathrm{MaxIoU}(\mathrm{B},\ \mathrm{B}',\ \mathrm{L}) = \max_{\pi \in S_M} \frac{1}{M} \sum_{m=1}^{M} \mathrm{gIoU}\left(\mathrm{B}_m,\ \mathrm{B}'_{\pi(m)},\ \mathrm{L}_m\right) \quad (10)$$

In Equation 10, only matches between two layouts with identical label sets are considered.

2.5.4 FID Score

The FID (Frechet Inception Distance) metric can be used to evaluate visual quality by measuring the distribution distance between real and generated layouts [8]. FID can also measure realism and diversity [7]. To calculate the FID, representative features of the layout are required. Following [29], this research trained a neural network to classify between ground truth layouts and noise-added layouts. Then, it was combined with an additional decoder network so that the training can realize both alignment and positioning [29].

3. Results and Discussion

In this section, we will discuss several things such as the dataset used and the performance of the SGTransformer model in placing components on the layout based on the input layout graph. The evaluation metrics used are alignment, overlap, maximum iou, and FID score. The performance of the SGTransformer model will be analyzed and compared with SG2IM. Subsequently, a visualization of the layout results generated by the SGTransformer model is performed.

3.1 Dataset

The dataset used in this research is a poster consisting of five components, namely title, subtitle, info, primary image, and logo. The process of preparing poster data into bounding boxes has been described in section 2.1. In the method used, input data is required in the form of a graph composed of several triples in the format of $< subject, relation, object >$ where $subject$ and $object$ is the component contained in the poster while $relation$ is the spatial relationship between components as explained in section 2.3. The distribution of components and relations in our triple dataset can be seen in Figure 10 and Figure 11.

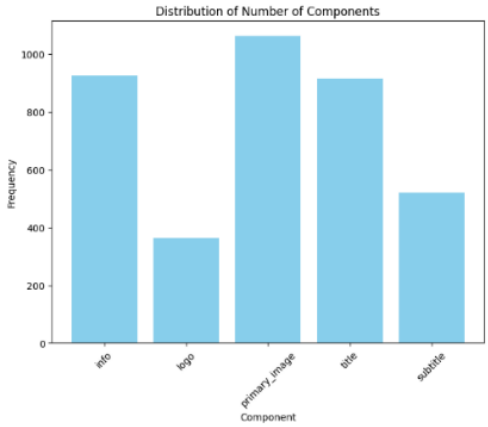

Figure 10. Distribution of Component Counts

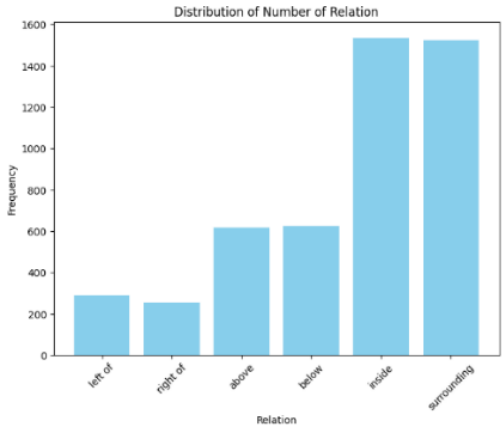

Figure 11. Distribution of Relation Counts

Based on Figure 11, the most common spatial relationship categories in triples are inside and surrounding. This shows that the tripe dataset that will be built into a layout graph is dominated by spatial relationships with the inside and surrounding categories. However, the right and left categories are still rarely found in the layout graph. In addition, the distribution of components can be seen in Figure 10, where components such as primary image, title, and info are pretty much contained in the layout graph.

3.2 Result and Model Evaluation

In this section, we will analyze the performance results of the SGTransformer model. We will explore the contribution of edge features and Laplacian positional encoding (LapPE) to the performance of the SGTransformer model. Then, we will discuss the impact of using spatial relationships in the input on the performance of the SGTransformer model.

3.2.1 Analysis of Laplacian Positional Encoding (LapPE) and Edge Features in SGTransformer Model

In the SGTransformer architecture, two important components can affect the model's ability to place components in the layout. These components are edge features and Laplacian positional encoding (LapPE) [11]. Exploration of the contribution of edge features and lapPE to the model is divided into four scenarios. In the first scenario, the model is trained without using edge features and lapPE. In the second scenario, the model is trained using edge features but without using lapPE. In the third scenario, the model is trained with LapPE but without using edge features. Lastly, the model is trained using edge features and lapPE. The results of the model performance evaluation based on the contribution of edge features and LapPE can be seen in

Table 1.
Based on

Table 1, the fourth scenario provides the best evaluation results on the overlap, max iou, and FID score metrics. In that scenario, the model excels on the FID score metric indicating that the model can produce better quality and more diverse layouts compared to the other scenarios. This is because the FID score measures diversity and fidelity [8]. In the max IoU metric, the model also managed to outperform other scenarios. This shows that the layout produced by the model is more like the ground truth layout compared to other scenarios. In addition, the model is better in minimizing the overlap compared to models in other scenarios. This is because, in the fourth scenario, the model is trained using edge features and lapPE. Edge features store additional information about the type of relationship between nodes [11] while lapPE utilizes laplacian eigenvectors [30] to express the relative position between nodes in the graph to enrich the knowledge representation in the graph and produce a more structured layout.

Table 1. Evaluation Result Based on Contribution of Edge Features and LapPE in SGTransformer

Scenario	Result of Evaluation			
	Alignment	Overlap	Max IoU	FID Score
SGTransformer (without EF & LapPE)	0.029	1.474	0.242	10.349
SGTransformer (with EF & without LapPE)	**0.020**	1.523	0.310	10.684
SGTransformer (with LapPE & without EF)	0.026	1.430	0.266	9.202
SGTransformer (with EF & with LapPE)	0.025	**1.274**	**0.325**	**8.575**

As for the alignment metric, the model in the second scenario gives better results compared to scenario 4 and other scenarios. In the second scenario, the model is trained without using LapPE and only relying on the information contained in edge features. Without LapPE, there are limitations for the model to capture the global structure of the graph and it is difficult to understand the positional relationship between nodes in the graph [21]. This causes the model to tend to generate components at relatively fixed positions because the model only relies on edge features. Therefore, the resulting alignment value is better than the other scenarios. However, this model lacks of layout variation as the resulting positions tend to be fixed. This is shown by the lowest FID score in scenario 2 compared to the other scenarios because the model is not good at generating diversity in the resulting layout.

3.2.2 Impact of Using Spatial Relationships

This research will discuss the impact of using spatial relationships (SR) on inputs. Spatial relationship describes the relative position between components [10]. In this section, the exploration is divided into two parts, namely

SGTransformer with spatial relationships (scenario 1) and SGTransformer without spatial relationships (scenario 2). The results of the model performance evaluation based on the use of spatial relationships can be seen in Table 2.

Table 2. Evaluation Results Based on Spatial Relationships in SGTransformer Input

Scenario	Result of Evaluation			
	Alignment	Overlap	Max IoU	FID Score
SGTransformer with SR	**0.025**	**1.274**	**0.325**	**8.575**
SGTransformer without SR	0.026	1.430	0.266	9.202

In the SGTransformer architecture, spatial relationships will be represented by edge features [11]. Based on Table 2, the SGTransformer model scenario with spatial relationship input (scenario 1) produces the best model performance on all metrics used. The model was trained using edge features. This experiment shows that considering edge features when calculating attention in the graph can improve the performance of the model in generating a more structured layout [11]. The model in the first scenario produces better alignment and overlap compared to the second scenario. This shows that the model in the first scenario can produce a good layout in alignment and can minimize overlap. In addition, the model also obtains higher max IoU and FID score. This shows that the model can produce more diverse layouts compared to the second scenario. A visualization of the layouts generated by the model can be seen in section 3.4.

3.3 Model Comparison

In this section, we will discuss the performance comparison between SGTransformer and SG2IM. SG2IM is the first model that can generate layouts based on the input graph [10]. The results of the evaluation of the two models can be seen in Table 3.

Table 3. Model Performance

Method	Result of Evaluation			
	Alignment	Overlap	Max IoU	FID Score
SGTransformer	0.025	**1.274**	**0.325**	**8.575**
SG2IM	**0.012**	1.901	0.286	9.391

Based on Table 3, SGTransformer successfully outperforms the SG2IM model on the overlap, max iou, and FID score metrics. Whereas in the alignment metric, SG2IM obtains higher score than SGTransformer. This is because the SG2IM model is based on a Graph Convolutional Network (GCN) where GCN can maintain strong local relationships and has limitations in capturing the global context [31], thus ensuring that the spatial relationship between related nodes will remain consistent. This causes SG2IM to produce component placement in a more consistent position so that the resulting layout has better alignment and is more structured. This can be seen from the lower alignment score compared to SGTransformer. However, because SG2IM can produce more consistent relative positions between components, the model becomes less capable of producing varied layouts. This is shown by the lower FID score compared to SGTransformer.

The SGTransformer model calculates the relationship between nodes globally [10], [21] so that the model can produce higher variability in position between components [11], [21]. This causes the SGTransformer model to be able to produce component placement in a more varied layout compared to SG2IM. Therefore, the FID score of SGTransformer is higher than SG2IM. In addition, SGTransformer also excels in the overlap and max iou metrics. The lower overlap score indicates that the placement of components generated by the SGTransformer model has less overlap than SG2IM. Meanwhile, the max IoU metric means that the SGTransformer model is more capable of producing a layout that is like the ground truth compared to SG2IM.

In addition to being compared with SG2IM, SGTransformer was shown to provide quantitative results that are superior to previous studies [7, 29]. Previous research conducted experiments using the Neural Design Network (NDN) method [7]. This NDN method was used to automatically generate layouts on several datasets namely Magazine, RICO, and Image Banner Ads. Compared to our study, SGTransformer provides better FID score and alignment results than the research. Then, there is previous research that uses the LayoutGAN++ method to automatically generate layouts on several datasets such as RICO, PubLayNet, and Magazine [29]. Compared to our study, SGTransformer gives better results for alignment, overlap, and FID score metrics.

3.4 Experiment Result

In this section, a layout visualization of the placement of components generated by the SGTransformer model will be shown. The resulting layout visualization comes from inputs that do not provide spatial relationship information and inputs that provide spatial relationships on the layout graph. The visualization results can be seen in Figure 12 and Figure 13.

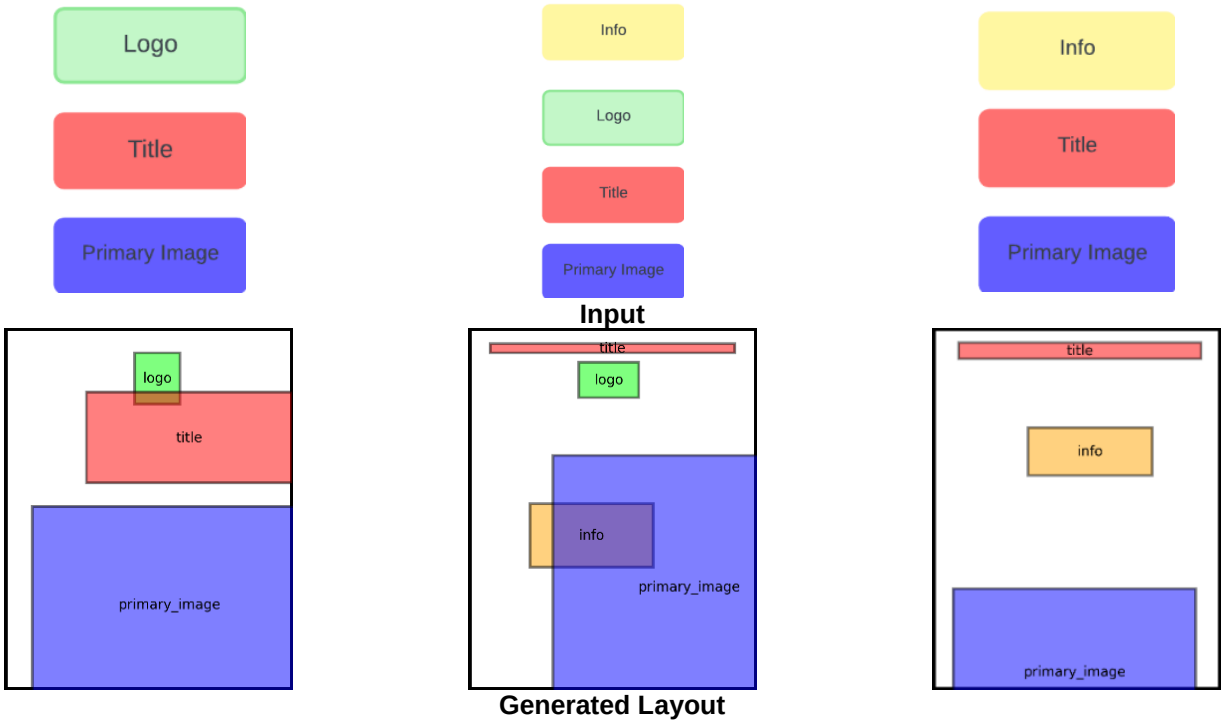

Figure 12. Layout Generation without Spatial Relationship Input

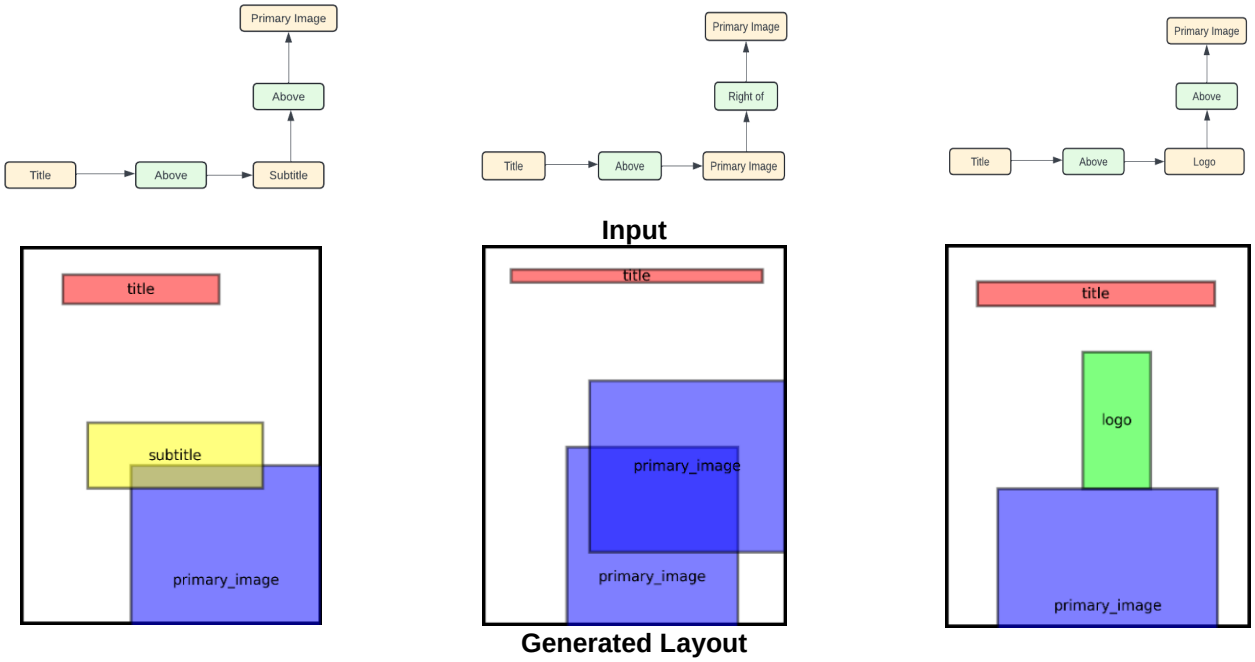

Figure 13. Layout Generation with Spatial Relationship Input

In Figure 12, the visualization shown comes from inputs that do not have spatial relationship information. Judging from the resulting layout, it shows that the SGTransformer model is also able to produce component placement on layouts with inputs that do not have spatial relationship information (only component information is needed). In Figure

13, the model can generate the placement of components based on the description of spatial relationships in the input represented by the layout graph. The placement of these components is by using the relative position information between components described in the input.

The qualitative results prove the ability of the SGTransformer model to generate layouts based on the input layout graph of component labels and relationships between components which can be seen in Figure 13. This implies that the SGTransformer model has proficiency in learning the spatial relationships between components represented by the input layout graph [11]. In other words, this model shows skill in determining the placement of components in a layout based on component information and relationships between components in the input. Unlike the generative models based on Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) in previous studies [3, 4, 5, 6], these models cannot generate layouts based on input in the form of component labels and relationships between components required by the user. This is because GAN and VAE-based models generate outputs in an unconditional manner (i.e.,generation from sampled noise vectors) [7].

3.5 Limitations

Although the SGTransformer model proved successful in locating components based on the input layout graph, we recognize that there are still some limitations to this research. The model, which we successfully trained on the advertising poster dataset, is only able to place components from input layout graphs that are not too complex (containing only a few components and relationships between components). There are still failures in the model when receiving input layout graphs that are quite complex. If the model receives input in the form of a complex layout graph, there will be an overlap in the resulting components. This model failure can be seen in Figure 14.

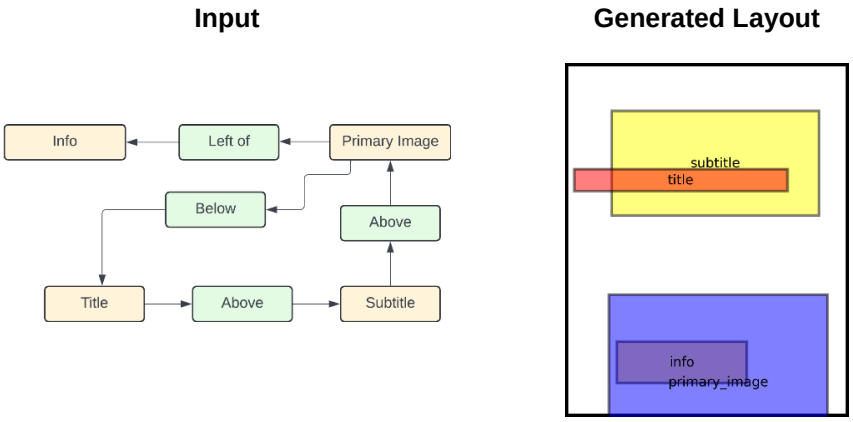

Figure 14. Failure Case

Then, the size of the bounding box on some components generated by the model is still not the required bounding box size. This can be seen from the bounding box size of each component generated by the model in Figure 13 and Figure 14. In Figure 13, the bounding box width of the title component is too small. This is because the SGTransformer model cannot learn the required bounding box size for each component. SGTransformer only learns the spatial relationship between existing components, so the model still fails to generate the appropriate bounding box size for each component or the required bounding box size [11].

3.6 Future Directions

Based on some limitations in our research, we hope that future research can expand the diversity of datasets. Future research development is expected to be able to provide datasets with more layout variations, especially in component placement in the poster layout. Datasets that have more diverse component layouts are expected to make the model able to place components according to the design principles. Then, it is hoped that further research can increase the training data so that the model can improve its generalization ability. In addition, further model development can use hyperparameter tuning to improve model performance.

By considering the limitations of the model in this study, we hope that future research can overcome the shortcomings of our model which cannot produce a bounding box size that suits the user needs. Future research can add other methods that can learn the size of the bounding box of each component so that it can meet the user needs.

4. Conclusion

This research aims to develop a layout generator system that automatically places components in advertising poster layouts using a transformer-based model (SGTransformer). The SGTransformer model developed in this

research can place components based on information in the form of component labels and spatial relationships between components represented by the layout graph. Based on the experiments conducted, the SGTransformer model that has been trained provides evaluation results, namely the alignment of 0.025, the overlap of 1.274, Max IoU of 0.325, and FID of 8.575. These evaluation results show better scores compared to the previous studies in the Layout Generation domain [7, 29]. The use of Laplacian positional encoding (LapPE) and edge features in the SGTransformer model is proven to improve the model performance. The spatial relationships used in the input also have a significant contribution to improving the quality of the generated layouts. Compared to SG2IM, evaluation metrics such as overlap, maximum IoU, and FID scores show that the SGTransformer model can produce better layouts, especially in terms of layout variation, although there are still problems with overlapping components.

This research contributes to the efficiency of the advertising poster design process and provides practical solutions to the creation of advertising posters in the digital marketing industry. The ability of the SGTransformer model to learn the spatial relationship between components in the layout graph causes this model to be able to place components based on component information and relationships between components in the input. This is different from GAN and VAE-based generative models that cannot generate layouts based on input spatial relationships between components. The construction of this system is expected to make the design process more efficient, especially when the placement of components manually is quite time-consuming. Future research is expected to develop models that can produce layouts that are more in line with the design principles.

References

[1] E. Setiawan Nababan, "Implementation of Advertising Poster As A Promotional Media For MSME."

[2] K. J. Murchie and D. Diomede, "Fundamentals of Graphic Design-essential tools for effective visual science communication," *Facets*, vol. 5, no. 1. Canadian Science Publishing, pp. 409–422, Jun. 11, 2020. https://doi.org/10.1139/facets-2018-0049

[3] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "LayoutGAN: Generating Graphic Layouts with Wireframe Discriminators," Jan. 2019. https://doi.org/10.48550/arXiv.1901.06767

[4] A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "LayoutVAE: Stochastic Scene Layout Generation From a Label Set," Jul. 2019. https://doi.org/10.48550/arXiv.1907.10719

[5] X. Zheng, X. Qiao, Y. Cao, and R. W. H. Lau, "Content-aware generative modeling of graphic design layouts," *ACM Trans Graph*, vol. 38, no. 4, Jul. 2019. https://doi.org/10.1145/3306346.3322971

[6] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "LayoutGAN: Synthesizing Graphic Layouts with Vector-Wireframe Adversarial Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 7, pp. 2388–2399, Jul. 2021. https://doi.org/10.1109/TPAMI.2019.2963663

[7] H.-Y. Lee *et al.*, "Neural Design Network: Graphic Layout Generation with Constraints," Dec. 2019. https://doi.org/10.48550/arXiv.1912.09421

[8] S. Chai, L. Zhuang, and F. Yan, "LayoutDM: Transformer-based Diffusion Model for Layout Generation," May 2023. https://doi.org/10.48550/arXiv.2305.02567

[9] D. M. Arroyo, J. Postels, and F. Tombari, "Variational Transformer Networks for Layout Generation.". https://doi.org/10.48550/arXiv.2104.02416

[10] J. Johnson, A. Gupta, and L. Fei-Fei, "Image Generation from Scene Graphs," Apr. 2018. https://doi.org/10.48550/arXiv.1804.01622

[11] R. Sortino, S. Palazzo, and C. Spampinato, "Transformer-based Image Generation from Scene Graphs," Mar. 2023. https://doi.org/10.48550/arXiv.2303.04634

[12] E. Quiring, A. Müller, and K. Rieck, "On the Detection of Image-Scaling Attacks in Machine Learning," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2023, pp. 506–520. https://doi.org/10.1145/3627106.3627134

[13] "Automated Catalog Generation using Deep Learning," *International Research Journal of Modernization in Engineering Technology and Science*, Aug. 2023. https://doi.org/10.56726/irjmets44010

[14] P. S. Parsania and P. V Virparia, "International Journal on Recent and Innovation Trends in Computing and Communication Performance Analysis of Image Scaling Algorithms". https://doi.org/10.17762/ijritcc.v4i6.2359

[15] O. I. Khalaf, C. A. T. Romero, A. Azhagu Jaisudhan Pazhani, and G. Vinuja, "VLSI Implementation of a High-Performance Nonlinear Image Scaling Algorithm," *J Healthc Eng*, vol. 2021, 2021. https://doi.org/10.1155/2021/6297856

[16] E. Quiring and K. Rieck, "Backdooring and Poisoning Neural Networks with Image-Scaling Attacks," Mar. 2020. https://doi.org/10.48550/arXiv.2003.08633

[17] J. Shi, S. Sun, Z. Shi, C. Zheng, and B. She, "Water Column Detection Method at Impact Point Based on Improved YOLOv4 Algorithm," *Sustainability (Switzerland)*, vol. 14, no. 22, Nov. 2022. https://doi.org/10.3390/su142215329

[18] G. Zhu *et al.*, "Scene Graph Generation: A Comprehensive Survey," Jan. 2022. https://doi.org/10.48550/arXiv.2201.00443

[19] S. Khandelwal and L. Sigal, "Iterative Scene Graph Generation," Jul. 2022. https://doi.org/10.48550/arXiv.2207.13440

[20] M. Wang *et al.*, "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks," Sep. 2019. https://doi.org/10.48550/arXiv.1909.01315

[21] V. P. Dwivedi and X. Bresson, "A Generalization of Transformer Networks to Graphs," Dec. 2020. https://doi.org/10.48550/arXiv.2012.09699

[22] Z. Chen *et al.*, "PIoU Loss: Towards Accurate Oriented Object Detection in Complex Environments," Jul. 2020. https://doi.org/10.48550/arXiv.2007.09584

[23] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," Nov. 2019. https://doi.org/10.48550/arXiv.1911.08287

[24] I. Ullah, M. Manzo, M. Shah, and M. Madden, "Graph Convolutional Networks: analysis, improvements and results," Dec. 2019. https://doi.org/10.48550/arXiv.1912.09592

[25] A. Sobolevsky, G.-A. Bilodeau, J. Cheng, and J. L. C. Guo, "GUILGET: GUI Layout GEneration with Transformer," Apr. 2023. https://doi.org/10.48550/arXiv.2304.09012

[26] J. Li, J. Yang, J. Zhang, C. Liu, C. Wang, and T. Xu, "Attribute-conditioned Layout GAN for Automatic Graphic Design," Sep. 2020. https://doi.org/10.48550/arXiv.2009.05284

[27] R. Carletto, H. Cardot, and N. Ragot, "Deep Learning for Document Layout Generation: A First Reproducible Quantitative Evaluation and a Baseline Model," pp. 20–35, 2021. https://doi.org/10.1007/978-3-030-86334-0_2

[28] Q. Jing *et al.*, "Layout Generation for Various Scenarios in Mobile Shopping Applications," in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, Apr. 2023. https://doi.org/10.1145/3544548.3581446

[29] K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, “Constrained Graphic Layout Generation via Latent Optimization,” in *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, Association for Computing Machinery, Inc, Oct. 2021, pp. 88–96. https://doi.org/10.1145/3474085.3475497
[30] E. Min *et al.*, “Transformer for Graphs: An Overview from Architecture Perspective,” Feb. 2022. https://doi.org/10.48550/arXiv.2202.08455
[31] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, “Graph Transformer Networks,” Nov. 2019. https://doi.org/10.48550/arXiv.1911.06455