Infographics Wizard: Flexible Infographics Authoring and Design Exploration

1. Appendix A
Infographics Dataset Annotation with Visual Groups

1.1. Source Images
We used the study results by Lu et al. [LWL∗20] to sample 1000 infographics belonging to each of the 12 VIF categories (as proposed in their work [LWL∗20]) and in the ratio as they appear in the original dataset (Figure 8 in [LWL∗20]). Because of our framework’s design aspect, since we aimed to relate the VG designs with VIF layouts, sampling infographic images based on the VIF layout was crucial for consistency with real-world designs.

1.2. Experiment Setup
We annotated the infographics using Amazon Mechanical Turk [Ama21]. The workers were first introduced to the concept of VGs and their individual components with some example images. The task was to segment individual VGs, the components inside the VGs (refer to Figure 3B of the paper), and annotate VGs with a bounding box in each of the infographic images. For consistency, each image was annotated by three different workers, and we chose the best annotated VGs for each image manually. An example user annotated image is shown in Figure 3C in the paper. Some example user annotations are provided in the supplementary material.

1.3. Processing the Segmentation Maps
Human-generated segmentation maps are coarse with ill-defined shapes. To generate a well-defined VG segmentation map, we employed the automated GrabCut [RKB04] algorithm, similar to [CWW∗19]. The human annotations were used as input to the algorithm to generate high-quality segmentation masks for VGs. This mask was then passed as input to Solaris [Sol21] which generated SVG paths for segmented VG designs. Finally, we added additional components to the SVG based on bounding box annotations of individual VGs, provided with the dataset by Lu et al. [LWL∗20].

2. Appendix B
Design Component Extraction
In this section, we discuss the process of extracting VIF, VGs and Connection designs from the infographics dataset.

2.1. Visual Information Flow (VIF) Extraction
VIF is the first component of our framework (D1), representing the backbone layout inside an infographic. As shown in Figure 3A of the paper, VIF represents the direction of information flow following the positions of repeated VGs inside an infographic. It is represented as a series of 2D points where each point represents a VG position in two dimensions corresponding to a design area, and the number of points represents the number of VGs in that particular flow. The sequence of points represents the sequence of visual groups based on the information flow. To calculate the VIF for a given infographic, we used the algorithm discussed in [LWL∗20].

2.2. Visual Group Design Extraction
Since VG designs are the basic building block for infographic generation in our framework (D2), we chose multiple sources to collect VG designs to attain maximum coverage. Out of the total three sources we used, the first one is Adobe Stock [Ado21], which contains SVG designs of several human-generated infographics, from which we manually separated the VG designs; the second source is our human segmented VG dataset which we curated for this work, explained in Appendix A; the third source is taken from the work by Chen et al. [CWW∗19] where we extracted VGs from timeline infographics using a Mask R-CNN [HGDG17], explained in the following text.

These sources were carefully chosen so that we get the VG designs created directly by humans and also using automated deep learning networks. We chose the SVG format to store the VG designs because of various benefits like scaling, color, and widespread usability. Each VG is further divided into four components as shown in Figure 3B in the paper, namely: Label, Text, Title, and Image. Each VG can contain any number of these elements, and each SVG design contains separate markings for each component, which the user’s content can then replace as per our design pipeline.

2.2.1. Mask R-CNN for Initial Element Segmentation
In addition to the VG designs generated from the datasets available at Adobe Stock [Ado21], we modified the timeline infographics work by Chen et al. [CWW∗19] to extract more VG designs. The timeline infographics dataset released with this work contains 9592 manually created timeline infographics from Timeline Storyteller [BLR∗19] and 1138 infographics from online sources. It includes 4689 images with manually labeled infographics components.
Figure 1: Process of generating VG designs from the Mask R-CNN network. After generating the network’s annotations, the bounding box and mask values are sorted in the direction perpendicular to the predicted orientation. Then the first values of every bounding box and mask are grouped to get a VG.

Table 1: Relations of VG Elements with the Timeline Infographics prediction from Mask-RCNN

<table>
<thead>
<tr>
<th>Timeline Component</th>
<th>Visual Group Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Text</td>
<td>Text and Title</td>
</tr>
<tr>
<td>Annotation Icon</td>
<td>Icon</td>
</tr>
<tr>
<td>Annotation Mark</td>
<td>Visual Group Design</td>
</tr>
<tr>
<td>Event Mark</td>
<td>Connecting Point</td>
</tr>
<tr>
<td>Event Text</td>
<td>Label</td>
</tr>
</tbody>
</table>

2.2.2. Visual Group SVG Generation

Creating reusable VG designs from the Mask R-CNN predictions requires binding several predicted bounding boxes based on their respective VGs. Since the Mask R-CNN output does not predict which VG a particular bounding box belongs to, we developed a simple scheme to bind the bounding boxes based on their respective VG positions. Also, since the purpose here is to get the VG design, we only created one VG component from each infographic. We assume that the design of the VGs remains consistent within the infographic, which was found to be true in all the timeline infographics from the dataset by Lu et al.

Hence, after predicting various elements for a given infographic image, we used the Orientation as predicted by the Mask R-CNN to generate the VG design. We assumed that VGs are spread in the direction perpendicular to the infographic orientation. For example, in Figure 1, the VGs extend in a vertical direction when the infographic is horizontal. Using this assumption, after generating the marks from Mask R-CNN for each of the infographic design elements, we sorted every element based on the bounding box centers in the direction perpendicular to the predicted orientation. After, we assigned the first element of each predict mark to the same VG.

For simplicity, to solve the purpose of extracting meaningful VG designs, we only used the infographics with Horizontal and Vertical orientations. Also, we confirmed the extracted VG designs by a human analyst for their correctness and discarded the incorrectly generated VGs. The final bounding boxes and masks obtained from this process were converted to the SVG-based VG designs using Solaris [Sol21].

We used Mask R-CNN [HGDG17] with this dataset to generate the segmentation maps for the infographic components. Besides predicting the segmentation masks, a modified network predicts the local and global information about a timeline infographic image. The global information predicted by the network is the infographic Representation, Scale, Layout, and Orientation. For example, an infographic can have the global properties Linear, Sequential, Unified, and Horizontal, as shown in Figure 1. Besides the global information, the network also predicts the bounding boxes for local information like Text, Label, Title and Icon along with the segmentation masks for the designs containing these elements. Figure 1 shows the network and sample annotations, respectively. The complete details of these global and local properties can be found in [CWW∗19].

As we can see, the output of the above model provides us with the annotation labels of Text, Icon and Mark and event labels of Mark and Text along with Main Body labels. We can convert each of these labels to the corresponding VG elements based on the relations shown in Table 1. The entire pipeline is shown in Figure 1.
Following up from Equation 1 from the paper, \( E \) is an indicator function that is 1 if and only if none of the VIF flow elements lie in the boundaries of the bounding box of the pivot element, and 0 otherwise. Another component of the layout scoring energy function is \( E_C \), which calculates the score based on the coverage a VIF layout provides. More the coverage, the higher the value of \( E_C \), as shown in Equation (1). Here \( Hull(VIF) \) means the convex hull of the points in a particular VIF layout.

\[
E_C = \frac{\text{Area of Hull(VIF)}}{\text{Area of Canvas}} \tag{1}
\]

The last component of the layout scoring function is the uniformity in the distance of the VIF elements wrt to the center of the pivot elements. To calculate this, we first calculate the center of the bounding box of the pivot element, denoted as \( P_{XC} \) and \( P_{YC} \), shown in Equation (2) and (3).

\[
P_{XC} = P_X + \frac{P_L}{2} \tag{2}
\]

\[
P_{YC} = P_Y + \frac{P_W}{2} \tag{3}
\]

Then, we calculate the variance of the distance between each VIF element and the pivot center, which gives a score of uniformity in the spread of the layout, shown in Equation (4). Lesser the value of \( E_U \), the better the uniformity and vice versa. \( \bar{D} \) is the mean Euclidean distance between the VIF elements and the pivot center.

\[
E_U = \frac{\sum_{X,Y \in \text{VIF}} \sqrt{(X_i - P_{XC})^2 + (Y_i - P_{YC})^2 - \bar{D}}}{N} \tag{4}
\]

### 3.0.1. VG-VIF Index

The VG-VIF index aims to capture a global relationship between the VG designs and the VIF layouts to support a ranking system that allows accurate recall of the suitable VG designs given a VIF layout. To develop the VG-VIF index, we first clustered all the VIF flows in our dataset into 12 categories as mentioned in [LWL+20]. We first generated the VIF images for each infographic in our dataset to generate these clusters. These images were then reduced to 50 dimensions using PCA [JM92] to get the principal components, which were then further reduced to 2D using t-SNE [VdMH08]. This 2D space was divided into clusters with centers around the high-density regions with DBSCAN [Liu06] and then iteratively clustering all the points into one of the centers. We repeated this procedure until there was no further improvement in the forming clusters. We used the same 12 classes discussed in [LWL+20] to form the basis of the VIF layouts (denoted as CL-X in Table 2).

Following this, we extracted the VG designs from each of the infographics. Table 2 shows an example of 4 Visual Groups (denoted as VG-\( x \)) that can occur in multiple VIF flow categories (denoted as VIF-\( x \)). This example shows the one-to-many relationship between the VG designs and the VIF layouts. We can treat these relations as “words” occurring in some “documents”, where the VGs are the documents and the corresponding cluster centers are the words in each document. Using this analogy, we obtain the ranking of VGs given a VIF cluster ID by calculating the TF-IDF score [R03] for each VG. The TF-IDF score has been extensively used in information retrieval and has properties that we can use to rank the VGs. Firstly, the TF-IDF score down-weights the most commonly appearing words; in this case, it results in higher rankings of less frequent VGs. Thus, our recommendation engine generates infographics using the VGs, which are less common. Also, the TF-IDF scores are generated covering the entire domain, in this case, the domain of VIFs, which is a desirable property while generating infographic designs.

To summarize the VG ranking process, we choose the best VIF layout vectors to place the VGs based on the canvas elements. The VG scores are obtained from the VG-VIF index, which shows how well-fitting a VG is for a given layout. The VG-VIF index scores are sorted, and a subset of high-scoring VGs are selected, which match the user’s markdown input (i.e., the number and type of components inside a VG). In some cases, users might want to design their own VG designs, an example of it can be a VG with some chart information. Since generating VGs with multidimensional data information is not yet supported in Infographics Wizard, users can use tools like Text-2-Viz [CZW+19] to design custom VGs and upload into Infographics Wizard to generate infographics.

### 3.0.2. Visual Group Placement and Information Embedding

The ranking algorithm discussed above provides a list of VGs, which can be used with a selected VIF layout. However, placing the VG on the VIF layout requires proper rotation and scaling. The rotation of a VG is defined based on whether or not the infographic has a pivot graphic. In case the pivot graphic is present, each VG should face towards the pivot graphic, and hence the rotation angle is calculated by measuring the angle of the arc connecting the VG from the center of the pivot graphic on the circle centered at the pivot graphic. After the rotation angle for each VG is fixed, the input information can be embedded inside each VG.

<table>
<thead>
<tr>
<th>Visual Group ID</th>
<th>VIF Layout Cluster IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>VG-1</td>
<td>CL-1, CL-3, CL-4, CL-7</td>
</tr>
<tr>
<td>VG-2</td>
<td>CL-1, CL-2</td>
</tr>
<tr>
<td>VG-3</td>
<td>CL-4, CL-5, CL-6</td>
</tr>
<tr>
<td>VG-4</td>
<td>CL-1, CL-2, CL-3, CL-4, CL-5, CL-6</td>
</tr>
</tbody>
</table>

Table 2: A VG-VIF index stores the information of VGs, and the corresponding VIF cluster centers CL-X in which they appear. Each VIF layout is clustered into one of the total twelve classes, shown as CL-X.
their respective SVG files for placing individual components (i.e., image, text, title, and label). Text’s font size and dimensions of the image to be embedded are calculated based on the size of these placeholders and the content’s size. Also, embedded text and images are invariant to the VG rotation and are always placed at the same angle in each of the VGs.

4. Appendix D
Exceptional Cases

Referring to the exceptional Infographic designs not covered under our framework from Section 3.1 from the paper, some examples are shown in Figure 2. These cases include designs where the VGs are scaled following the VIF layout in images A, B, and C. Also, in images A and C, the pivot graphics are composed of multiple objects and contain additional icons. Sometimes, the VG design is not fixed, as in the case of image B. Such designs are hard to generalize with our tool, which was developed to support general-purpose infographic design and exploration.

5. Appendix E
Evaluating Design Experience with User Study

5.1. Tasks and Procedures

We initially familiarized the participants with the concepts of Infographics and related terminologies, such as the definition of VG, VIF, Pivot Graphic, and Connections for our framework. Next, we showed a few examples of infographics chosen from our dataset to familiarize them with infographic designs. The participants were then allowed to experiment with Infographics Wizard and ask clarifying questions regarding the tool. The task was to generate an infographic design from the content and design input of choice and evaluate the design support features of Infographics Wizard.

We also provided sample markdown inputs to users, and they had an option to either generate infographics from the content of their choice or use the provided examples.

After the participants were satisfied with their generated or filtered infographic designs, we conducted a short semi-structured interview to collect qualitative feedback. We asked the participants to rank their experience with Infographics Wizard on a five-point Likert scale. The questionnaire was based on a total of six factors: Enjoyment, Exploration, Expressiveness, Results worth the effort, Ease of use, and Workflow. Four of these factors were taken from the work by Cherry et al. [CL14] for quantifying the creativity support for design tools. Two specific factors, Effort and Workflow, were added in the questionnaire to specifically evaluate Infographics Wizard for its support in infographics design and exploration tasks. The whole study took about 45 minutes for each participant.

References


Figure 2: Infographic designs out of the scope of our framework. Some of these features include VG scaling (A, B, and C), inconsistent VG designs (A and B), complex pivot graphics composed of multiple SVG paths (A and D), and complex connection designs (E).

Figure 3: Example infographics generated using our framework using the example cases. We evaluated our framework with three example inputs, each of which had different content and design, shown on the left of each row. For each input, our framework recommended infographics, where we show some recommended infographics in each row for similar inputs.
Figure 4: Sample infographic designs generated with our framework collected during a user study.