Enhancing Circuit Diagram Understanding via Near Sight Correction Using VLMs

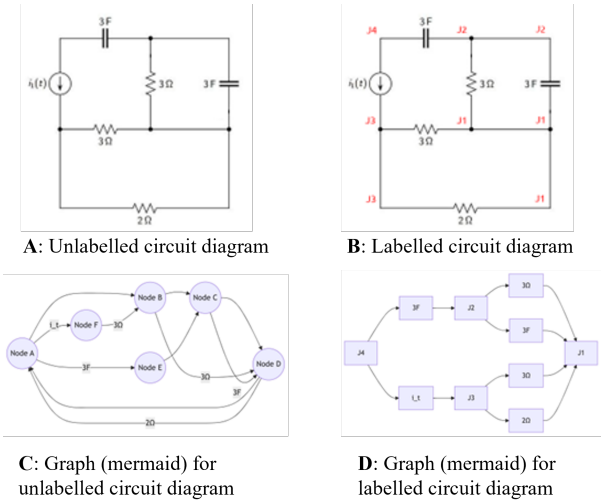
Shreyas Kulkarni, Vivek Kumar, Remish Leonard Minz, Munender Varshney, Thiruvengadam Samon, Abhishek Mitra, Nikhil Kulkarni, Nilanjan Chakravortty, Prateek Mital, Kingshuk Banerjee
Research & Development Centre, Hitachi India Pvt Ltd, Bangalore, India
{shreyas.kulkarni, vivek.kumar, remish.minz, munender.varshney, thiruvengadam.s, abhishek.mitra, nikhil.kulkarni, nilanjan.chakravortty, prateek.mital, kingshuk.banerjee}@hitachi.co.in

Abstract

Automated circuit diagram understanding is essential for circuit digitization, circuit design verification, education etc. Despite its practical importance, current methods struggle with interpreting circuit diagrams reliably. Recent advancements in Vision-Language Models (VLMs) have enabled significant progress in tasks such as Visual Question Answering (VQA), but VLMs still fail to capture basic visual relationships, such as line or circle intersections, that are essential in circuit schematics. In this work, We evaluate state-of-the-art VLMs for circuit diagram understanding and confirm that it face challenges in accurately identifying circuit connections. We propose Near Sight Correction (NSC), a pipeline to transform the circuit diagram into a more meaningful and enhanced circuit diagram by utilizing key elements. This pipeline automatically labels the connection key points in the original diagram. Thereafter we ingest it in a VLM for circuit understanding, either directly through graph generation or through additional VQA tasks. We evaluate our approach in three settings, (i) VQA on circuit components, (ii) VQA on circuit connections, and (iii) directly through graph generation. We name the three settings as Circuit VQA, Connection VQA and Connection Matrix Prediction task, respectively. All three settings are evaluated on adaptations of circuitvqa dataset [14]. Circuit VQA task achieves an accuracy of 87.38%. Connection VQA task and Connection Matrix Prediction task achieves the best F1 scores of 0.723 and 0.8735 respectively.

1. Introduction

Automated understanding of circuit diagrams is essential for various practical applications such as circuit diagram digitization, automated fault detection, circuit design verification, education and training, and circuit simulations. Conventional techniques struggle in achieve a good accuracy in circuit diagram understanding. Recent advances in AI, particularly Vision-Language Models (VLMs), offer new possibilities for automated interpretation of complex visual data across various tasks like visual grounding, visual reasoning and visual question answering. VQA has gained a lot from recent advancements in Vision Language Models (VLMs), enabling them to understand images and generate insightful responses. The combined ability of vision and language has opened paths for novel applications in the area of robotics, healthcare, autonomous systems and more. VLMs, such as CLIP [15] and Flamingo [2], have demonstrated impressive capabilities in a wide range of vision-language tasks, from object detection to image captioning, to even more complex multi modal reasoning tasks. However, despite their promising performance, VLMs continue to struggle with some simple artifacts in image understand-



A: Unlabelled circuit diagram

B: Labelled circuit diagram

C: Graph (mermaid) for unlabelled circuit diagram

D: Graph (mermaid) for labelled circuit diagram

Figure 1. Example input and outputs of our pipeline. 1A is the input 1B is the output of our key point labelling pipeline. 1C and 1D are the generate graph of 1A and 1B respectively.

ing. A recent study [16] highlights that VQA on simple objects, such as counting intersections in line diagrams or identifying geometric relationships in intersecting circles, remains challenging for state-of-the-art VLMs. This limitation raises concerns about their ability to handle tasks requiring fine-grained visual reasoning, such as interpreting schematic diagrams or understanding connectivity in circuit diagrams.

In this paper, we aim to address these challenges by exploring the application of VLMs in circuit diagram understanding. Since [16] discusses the challenges VLMs face interpreting intersections in line diagrams, We observe that while VLMs excel at object recognition, they struggle to identify and reason about complex spatial relationships and interconnections between electrical components in schematic circuit diagrams. This limitation hinders their ability to perform essential tasks in domains such as electrical engineering and circuit design. We utilize key elements to achieve our goal, which comprise of the circuit components and the connection key points. Figure 1 shows an example input and the corresponding outputs of our pipeline. Figure 1A shows the unlabelled input circuit diagram, while an intermediate output of our pipeline, an automatically labelled image is shown in Figure 1B.The graph representation of the unlabelled and the labelled circuit diagram are shown in Figure 1C and 1D respectively. The graph representation of the labelled circuit diagram correctly identifies the connections of the sample input circuit. We use Mermaid [21] to realize the graph representation. This example highlights that understanding unlabelled circuit image as shown in Figure 1 using VLMs has limitations. To overcome these limitation, we introduce a novel approach called Near Sight Correction (NSC). NSC is designed for schematic circuit diagrams and automatically labels connection key points in the images to enhance the VLM's ability to interpret component connectivity. By explicitly marking these key points, our method improves the VLM's capacity to understand the semantic connections between components, ultimately resulting in better performance in circuit diagram understanding and downstream tasks such as Vision Question Answering for these enhanced circuit diagrams. Our experiments demonstrate that the NSC approach significantly improves circuit diagram understanding. The results highlight NSC's effectiveness in enhancing the capabilities of VLMs, enabling them to more effectively reason about complex visual relationships in specialized domains when key points are provided in the input images. Our approach addresses key challenges in automated circuit understanding, paving the way for more accurate and reliable Vision-Language systems for specialized tasks. We created three datasets for the CircuitVQA, ConnectionVQA and Connection Matrix prediction task. We intend to make these datasets public.

2. Literature Survey

2.1. Circuti Diagram Understanding

The early work of automated circuit diagram understanding primarily focused on circuit recognition. One such early study from the pre-machine learning era [5], in which the authors partitioned the recognition problem into three sub-tasks: identifying nodes, components, and connections. All three sub-problems were addressed using pixel-based methods. Nodes and components were segmented by applying an appropriate threshold on a spatially varying object pixel density, while connection paths were traced using pixel stacks.

In the early 2000, with the widespread use of tablet PCs and other pen input devices, research toward automated circuit understanding focused on sketches on such devices. These works use techniques such as pen stroke, pixel gradient, and visual based features to recognize nodes, components and connections from such digitized sketches [6–8] .

Later on deep learning approaches were use to extract features directly form the input images. [13] was one of the early work credited for using convolution neural network to learn the definition of visual objects. Recent works such as [9] and [3] attempt to give a complete pipeline for understanding and digitizing analog circuits. Both approaches use YOLO for the detection of components and text, while Histogram of oriented gradients (HOG) based features are used to identify the connections.

[14] is closest to our work where the circuit understanding is done using Vision language models (VLM). It performs well in identifying circuit components, however, the emphasis on identifying connections is limited. The circuit components and connections understanding are done using VQA which uses vision modality and language to provide a natural language output for a given question. The authors release a dataset, which has a schematic and handdrawn electrical circuit. This dataset comprise of a collection of multiple small datasets and one hand drawn dataset, named CGHD [20]. The recent advancement of Transformers, enable us to utilize VLMs such as LLaVA [12], InstructBLIP [4] and GPT4V [1] on VQA datasets which shows remarkable performance. Still these work does not focus on the connections while AI2D for diagrams [11] which identify the structure and the semantics of its constituents and their relationships in the form of Diagram Parse Graphs (DPG).

3. NSC Framework

We propose a Near Sight Correction(NSC) framework that takes a digital image of a schematic circuit diagram as input, i.e., an analog electrical circuit, and then automatically annotates the key points in the schematic diagram. Our framework comprise of seven step process, as described in this section. In an abstract view, steps one through four achieve

the automated key point awareness. Step six performs the accurate schematic diagram understanding with the help of the key point aware circuit images and represents that understanding in a graph form using Mermaid [21], which is expressive enough to describe any analog schematic diagram and simple enough to be accurately generated by the VLMs. Figure 2 shows all five steps, where steps one and two are abstracted as one block. The final step, step seven, performs the VQA task using the graph representation of the circuit diagram and the key point aware analog circuit image, which is the output of step two.

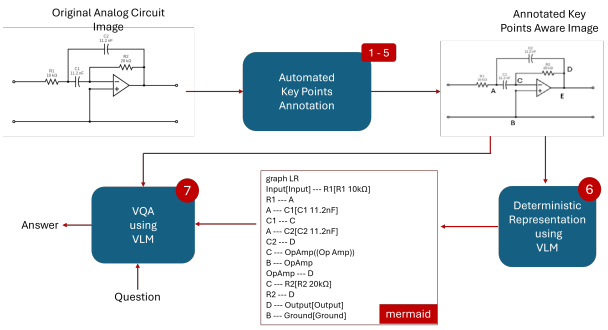


Figure 2. Proposed Framework NSC

Steps one through five are the key contributions of this work. Step one performs the object detection task on the schematic diagram, where the class objects are broadly classified into circuit components and connection key points. Step two performs the edge detection, label localization, and font size normalization. Step three removes the components and texts from the image and identify the line segments using Hough transformation. Step four identified equipotential junctions by identifying junctions on connected segments such that segments where there is no component between junctions. Step five does the equipotential label refinement where the equipotential points are assigned same label. Figure 3 shows the flow diagram of the above five steps.

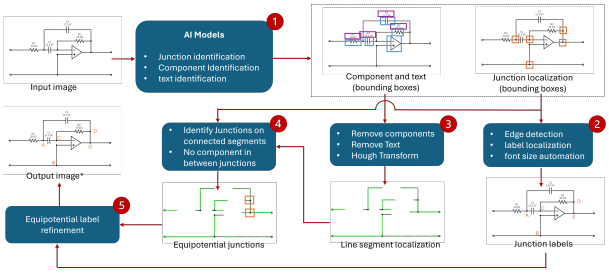


Figure 3. Proposed automated annotation pipeline.

3.1. Key elements localization

Our proposed pipeline uses an automated labelling process to help the vision language model (VLM) for semantic understanding of the diagram to give a relevant answer to the question asked. The key element localization is achieved using the widely used YOLOv8 object detection model [17, 19].

3.1.1. Annotation of key elements

To understand the connections and spatial correlation in a circuit image, we have annotated all the schematic circuit images in the CircuitVQA dataset [14] for 3-way and 4-way Junctions, Cross-over, and terminals. At the same time, the remaining annotations of components are already available in the dataset. All these annotation classes are key elements in the comprehensive understanding of circuits, while the prior four classes are key points of the circuit representing the connection details among the components. The total number of classes thus formed is 63 for schematic analog circuits and 70 for the complete set of schematic circuit images comprising both analog and digital. We use the CVAT tool [18] to annotate all these schematic circuit images. After manually annotating the images using the CVAT platform, we obtained the annotations in the YOLOv8 Detection 1.0 format.

3.2. Fine-Tuning Yolov8 for Localizing key elements

The preprocessed manual annotations of these key elements are used for training the Yolov8 object detection model to localize all key elements in the circuit using a bounding box. We have taken a pre-trained YOLOv8 model, which has been fine-tuned for schematic analog circuit images, a complete set of schematic images, a hand-drawn CGHD dataset, and a circuitvqa dataset. The number of classes varies based on the dataset used as the many components in digital circuit are not present in analog dataset such as OR, AND, NAND and XOR. So, the set of classes is in the range of 63 or 70, as discussed above. The trained model can now localize various components and key points for the semantic and spatial understanding of the diagram. The classes localized by the model include resistors, capacitors, inductors, current sources, and voltage sources, to name a few. Since our main research focus is a comprehensive understanding of circuits by identifying the connections between the components, we decided to go ahead with a simple set of analog circuit components. We have split the 75% dataset for training, 10% validation, and the remaining 15% for testing. We reckon the VLMs lack the global view of the image while answering any given question in a VQA task. We test this by asking questions about closely connected components in a smaller region of the entire input circuit image against the components that are sparsely connected. We observe a better answer for the closely connected compo-

nents. To provide a general solution in this setting, we use our Yolov8 to get an enhanced image with localized components and key points using a bounding box, which improves the VLM's comprehensive understanding. This fine-tuned model achieves a map50 accuracy of 0.87074 with a precision and recall of 0.87464 and 0.8091, respectively.

3.3. Automatic Key Point Labelling

In Automatic key point labelling step we iterate through each of the connection key point inferred in the key point localization step, such as the junctions, cross-overs, terminals and even corners. Each of these inferred objects has a bounding box. To automatically label a connection key point, we pick one inference at a time. The bounding box of this inference is measured. Another box, called the label box is used to place the label. The label box is of the same size as the inferred bounding box of the key point. This label box is convolved around the inferred bounding box with the criteria, that if there is any edge inside the label box, then the space spawned by the label box is not fit for automatic labelling. In such a situation, we go ahead with the convolution step per pixel and check the criteria again. Only, at the first instance, when there is no edge withing the label box, we go ahead and place an alphanumeric label within the label box.

3.4. Equipotential Label Refinement

Once we label the junctions with labels. There are three sub steps called the line segment localization, equipotential junction identification and equipotential label refinement. In line segment localization, we remove the components and the text from the image and apply Hough transformation to identify the line segment in the input image. In the next step we identify the junctions on the connected segments. We check whether there is any line identified by Hough transformation connecting two junctions and no components between them. If we find such junction pairs we mark them as equipotential. In the equipotential label refinement step we redistribute the labels assigned to the junctions such that equipotential junctions have the same label. After each key points are covered, we supplement additional steps of normalization on the labels based on size. For instance, Out of all the annotation box containing alphabetic annotation of different sizes, we scale all of them to the one having the largest size. This ensures that all the annotations are of same size. We ensure that the font size of the label is maximum with respect to the label box size.

3.5. Graph Understanding through VLM

In this step we provide the image with key point labelling to the VLM and prompt it to generate a graph representation of the schematic diagram. The graph representation we choose is mermaid code representation [21]. This is because, mermaid is very simple to represent connected components of any domain, yet is expressive enough to represent any circuit. We use one shot prompting technique to query VLMs to generate mermaid code. Figure 1C and Figure 1D shows the graph visualization of the mermaid code generated for the circuit diagram shown in Figure 1A and Figure 1B respectively.

3.6. VQA Task

This is the forth and the final step of our pipeline. In this step we perform the VQA task on the Key point-aware analog circuit images. Hear also we use one shot prompting technique to query VLMs to get the answer to the VQA question by providing the image in both labelled and unlabelled setting along with the corresponding mermaid code generated in step 3.

4. Experimental Results

In this section we present the datasets and the experimental results of the various stages of our research. Starting from preliminary experimentation to test the ability of VLMs to perform VQA on circuit images. The experimental results of automatically labelling the circuit images to make it key point aware, followed by the improvement achieved by our NSC approach results.

4.1. Datasets

We have four datasets, all are adaptations of circuitvqa [14], Table 1 shows all the four datasets. Circuit-VQA1 dataset is the complete circuitvqa's [14] test dataset which consists of 1145 images (hand drawn + schematic) of both analog and digital circuits. Over these images total 23313 datapoints exist in this dataset. We show samples of two datapoints in table 2. While performing manual analysis over randomly sampled datapoints in Circuit-VQA1 dataset, we observed the mismatch between 'true answer' of question and ground truth values mentioned in the Circuit-VQA1. Therefore, to get accuracy results, we decided to manually correct the ground truth values of a small subset and rerun the experiments on it. We sampled 175 images from Circuit-VQA1 (schematic + analog) at random and manually rectified corresponding QA pairs from Circuit-VQA1. This led to Circuit-VQA2 dataset of 1615 datapoints on 175 images. The Circuit-VQA1 dataset does not contain questions (schematic circuits) for evaluating the VLM's ability to identify component interconnections within a circuit diagram. Therefore, we created a new dataset for the connections. We did this by sampling 275 images from Circuit-VQA1 (schematic + analog) at random and manually construct the QA pairs on those 275 images, thus creating Connection-VQA dataset, which has 1025 datapoints. The last dataset we constructed is the Connection Matrix dataset.

This dataset targets the VLM's capability to understand the presence or absence of a connection in a circuit. Connection Matrix dataset comprise of 3457 possible datapoints, or connections between any two components over 50 circuit images.

Table 1. Datasets

Dataset Name	Description
Circuit-VQA 1	Complete circuitvqa [14] test dataset
Circuit-VQA 2	175 images containing 1615 corrected data points
Connection-VQA	275 images containing 1025 connection data points
Connection-Matrix	50 images containing 3457 possible connection data points

Table 2. Connection-VQA Dataset Samples

Question	Answer
Do 3ohm and 3ohm make direct electrical connection with each other?	Yes
Do 3F, 3F and 2ohm make direct electrical connection with each other?	No

4.2. Experiment Pipelines

We have six experiment pipelines. Each experiment pipeline is derived from the proposed framework shown in Figure 2. Figure 4 shows six derived pipelines. The first pipeline shown in Figure 4(1) is the one where VQA task is performed on unlabelled images directly. The second pipeline shown in Figure 4(2) is the one, where first mermaid code of unlabeled circuit diagram is generated through VLM which acts as additional prompt while performing VQA task.

4.3. VLM Prompts

Depending upon the task and experimental pipeline, variety of prompts are used. To generate mermaid code through VLM for unlabeled (original) images, the prompt used is:

```
"You are expert of electrical circuit
diagrams.  Generate mermaid code for
this electrical circuit diagram.  Do
not give any additional information
in the output.  The objective of
the mermaid code is to understand
all electrical components such as
resistors, capacitor, current sources
etc.  and to understand how these
components are connected to each
other."
```

To generate mermaid code through VLM for labeled images, the prompt used is as follows.

```
"You are expert of electrical circuit
diagrams.  Generate mermaid code for
this electrical circuit diagram.  Do
not give any additional information
in the output.  The objective of
the mermaid code is to understand
all electrical components such as
resistors, capacitor, current sources
etc.  and to understand how these
components are connected to each other.
To help you in understanding of these
connections, we have labeled some of
locations or connection nodes as 'J1',
'J2', 'J3' etc.  in red color.  The
nodes with same label are electrically
identical."
```

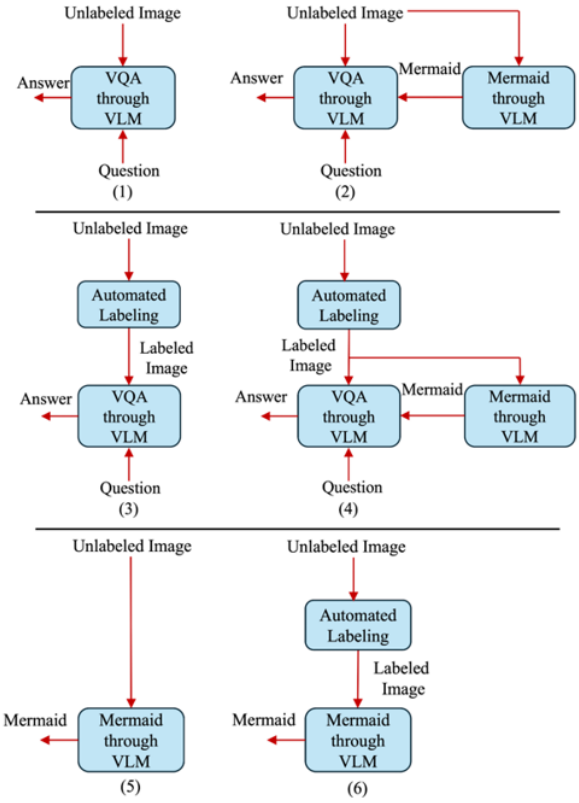


Figure 4. Proposed automated annotation pipeline.

To perform CircuitVQA task over images without any mermaid, the prompt used is as follows.

```
"You are expert of electrical circuit
diagrams.  The task is to answer the
following questions based on this
image."
```

To perform CircuitVQA task over images with corresponding mermaid code, the prompt used is as follows.

"You are expert of electrical circuit diagrams. The approximate mermaid code for attached electrical/digital circuit diagram image is: {}. The task is to answer the following questions based on this mermaid code and image."

To perform ConnectionVQA over images without any mermaid, the prompt used is:

"You are expert of electrical circuit diagrams. The task is to answer the electrical connection related questions based on this image. You can consider following guidelines while answering the questions: 1. There exists a direct electrical connection between 2 components if they are connected to each other by ONLY wire (no in between component). 2. There exists a direct electrical connection between 2 or more components if they all are connected to one common junction point or node. The question is as follows:"

To perform ConnectionVQA over images with mermaid code, the prompt used is:

"You are an expert of electrical circuit diagrams. The approximate mermaid code for attached electrical/digital circuit diagram image is: {}. The task is to answer the electrical connections related questions based on this image and the mermaid code. You can consider following guidelines while answering the questions: "There exists a direct electrical connection between 2 or more components if they all are connected to one common junction point or node in the mermaid code." The question is as follows:"

4.4. Experiments

This section lists out all the experiments.

4.4.1. Sample Outputs for sub tasks

Our proposed approach generates automatically labeled images from unlabeled (original) images. Figure 1B shows the labeled version of 1A. The mermaid code generated for images through VLM can be visualized through online mermaid editor tool. The mermaid visualization for circuit in 1A is shown in 1C. Similarly, the mermaid visualization for circuit in 1B is shown in 1D. One can observe that mermaid generated for labeled images is relatively accurate as compared to mermaid for unlabeled image. For this research work, we have ignored the arrow directions in mermaid.

4.4.2. Experiment 1 (CircuitVQA task)

As a task to replicate the results mentioned in CircuitVQA paper, we used Circuit-VQA1 dataset and experimented with Pipeline1 using top performing VLMs as per VLM leaderboard dated Aug 2024. The objective of this experiment was to track the capability enhancement of off-the-shelf VLMs as they are continuously being updated. The results are shown in Table 3 with GPT4o being the standout winner.

Table 3. CircuitVQA task results
Dataset1, Pipeline 1

VLM	VQA Accuracy
Llava7b-1.6-vic-q8 (Open Source)	34.66%
InternVL2-4B (Open Source)	38.92%
GPT-4o (Closed Source)	**49.87%**

4.4.3. Experiment 2 (CircuitVQA task)

Although the proposed approach of automated labeling is targeted towards improving the connection interpretation ability of VLMs, we decided to check its side effect while answering other types of questions such as count, position, values mentioned in CircuitVQA. This time we used the Circuit-VQA2 and experimented with four experimental pipelines. The corresponding results are shown in Table 4. With change of experimental pipeline, the accuracy results vary within the range of $\pm$ 3% and this trend is common across top 3 closed-source VLMs (Feb 2025) however no experimental pipeline turns out to be clear winner.

Table 4. CircuitVQA task results
Dataset2; VQA Accuracy %

Experiment Pipeline	Claude Sonnet 3.5	GPT4o Feb 2025	Gemini 2.0 flash
Pipeline 1: No labels +No mermaid	**78.50**	80.79	84.48
Pipeline 2: No labels + mermaid	78.35	**81.14**	84.71
Pipeline 3: labels + No mermaid	76.96	78.60	86.93
Pipeline 4: labels + mermaid	76.76	79.25	**87.38**

4.4.4. Experiment 3 (ConnectionVQA task)

Now we turn to evaluation of our proposed approach for connection-based dataset i.e. Connection-VQA. We Experimented with four pipelines on Connection-VQA with eval-

uation metric as F1 score since answers are binary in nature as shown in Table 2. The corresponding results are shown in Table 5.

Table 5. ConnectionVQA task results
Dataset3; VQA F1 score

Experiment Pipeline	Claude Sonnet 3.5	GPT4o Feb 2025	Gemini 2.0 flash
Pipeline 1: No labels +No mermaid	0.634	0.610	**0.647**
Pipeline 2: No labels + mermaid	0.582	**0.678**	0.632
Pipeline 3: labels + No mermaid	0.588	0.537	0.557
Pipeline 4: labels + mermaid	**0.723**	0.635	0.534

The Claude Sonnet 3.5 achieves the overall best numerical result with an F1 score of 0.723 using Pipeline 4, which is significantly higher than the F1 score of 0.634 obtained with Pipeline 1 for the same model. This result points to conclusion that proposed approach of automated labeling along with use of VLM generated mermaid significantly improves VLMs ability to interpret the connections. However, this observation is not consistent across other VLMs. Therefore, we decided to perform deeper analysis of the images, their mermaid codes generated through VLMs and VLMs ability to interpret the image and mermaid together as explained in Experiment 4.

4.4.5. Experiment 4 (Connection matrix prediction task)

The inconsistency observed in experiment 3 results lead to the intermediate output analysis of our experiments. There could be 2 possible sources of error coming from VLMs. First, VLMs are not able to produce highly accurate mermaid codes. Second, VLMs are not able to analyze, and reason based on mermaid code for given VQA query.

Let us consider Pipeline 5 and Pipeline 6 for investigation where VQA part is removed and only the mermaid generation part of unlabelled and labelled images is considered.

To evaluate this setup, we created the connection matrix for given circuit image manually. We followed systematic nomenclature to deal with terminal positions and same value components while writing the connection matrix. The sample connection matrix for circuit shown in Fig.1 is shown in Fig.5. If terminals of two different components are connected to each other, then that matrix position is filled with 1. We further created the connection matrix for VLM predicted mermaid codes manually. The predicted connection matrices are compared with true connection matrix to quantify the accuracy of mermaid generation.

This experiment was performed on Connection Matrix dataset with Pipeline5 and Pipeline6 and results are shown in Table 6 and 7. Table 6 shows the percentage of the number of images in which all the connections are identified correctly, against total number of images in dataset. First

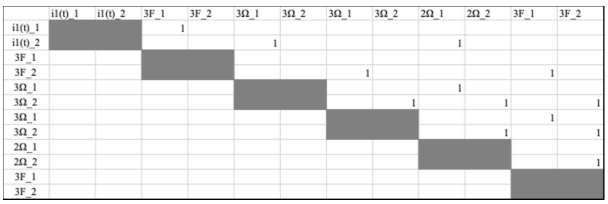


Figure 5. Samaple Connection Matrix

row shows the percentage for pipeline 5, where the input images have no label and second row shows the percentage for pipeline 6, where input images have labels.

The connection prediction performance through the mermaid code is drastically high for labeled images compared to unlabeled images across all VLMs. Therefore we present a deeper metric for the results of labelled images.

We calculate the F1 score for the connection matrix using pipeline 6 for labelled images, by considering all the possible connections in a circuit. A generated mermaid code, when represented as a connection matrix, looks like Figure 5. All the connection space in the upper triangular matrix of the figure are the possible connections available, for example the possible connections in Figure 5 is 60. If the generated mermaid's connects match with the one's in the ground truth matrix, we count it as true positive, if the generated mermaid's connection does not match with the one's in the ground truth matrix, we count it as false positive. We sum up the true positives, false positives, true negatives and false negatives to get the F1 Score for each of the three VLMs. The cumulatively total possible connections in the dataset are 3457. Since Connection-Matrix is subset of Connection-VQA, let us observe the results of Experiment 3 over Connection-Matrix as shown in Table 8.

Table 6. Connection matrix results
Dataset4;Mermaid, Percentage

Experiment Pipeline	Claude Sonnet 3.5	GPT4o Feb 2025	Gemini 2.0 flash
Pipeline5: No labels + mermaid	2%	4%	2%
Pipeline6: labels + mermaid	54%,	60%	62%

Table 7. Connection matrix results
Dataset4,Mermaid, F1 score

Experiment Pipeline	Claude Sonnet 3.5	GPT4o Feb 2025	Gemini 2.0 flash
Pipeline6: labels + mermaid	**0.8735**	0.872	0.77

By observing the Table 6 and table 8 together one can conclude Tables 6, 7 and 8 show that, VLMs can generate the mermaid codes with high accuracy when input images are appropriately labelled. However, VLMs have the limi-

tations in analyzing this connection information in mermaid codes while performing VQA task.

Literature [10] backs the relative performance of Claude model against GPT4o and Gemini2.0. Claude model is considered superior over GPT4o and Gemini2.0 for analytical reasoning tasks. It is evident here since Claude shows relatively best performance on Connection VQA even though it preforms relatively poor in accurate mermaid generation.

Table 8. ConnectionVQA task results
Dataset4,VQA, F1 score

Experiment Pipeline	**Claude Sonnet 3.5**	**GPT4o Feb 2025**	**Gemini 2.0 flash**
Pipeline4: labels + mermaid	**0.818**	0.656	0.273

5. Conclusions

In this work, we address the diagram understanding from a holistic point of view, such as relative spatial localization and various connection joints. We have proposed an NSC pipeline to enhance the circuit image, which labels all the key elements such as components, junctions, terminals, etc. This enhancement is crucial for the analysis of analog electric circuits specifically in connection identification and visualizing them using mermaid. We show a better comprehension ability of VLM using our NSC pipeline approach with the VQA task. We also derived multiple datasets for various experimental setups to quantify the performance related to connections and component labelling. To assert the NSC based quantitative performance, we have also proposed the Connection VQA task and the Connection matrix prediction task, which achieve the best F1 scores of 0.723 and 0.8735, respectively.

6. Acknowledgement

We would like to acknowledge the valuable assistance of three interns who contributed significantly to the manual data annotation for most of the experiments. Their work included annotating additional key elements such as 3-way and 4-way junctions. Furthermore, they played a crucial role in the creation of datasets for Circuit VQA, Connection VQA, and the Connection Matrix. The interns—Tanmay Kukreja, Sambit Kumar Mitra, and Prachi Parakh—worked with our organization during the course of this study.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 23716–23736. Curran Associates, Inc., 2022. 1

[3] Bharat Bohara and Harish S Krishnamoorthy. Deep learning-based framework for power converter circuit identification and analysis. *IEEE Access*, 2024. 2

[4] Wenliang Dai, Junnan Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*, 2, 2023. 2

[5] Chandran V Edwards B. Machine recognition of hand-drawn circuit diagrams. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, page 3618–3621, 2000. 2

[6] Leslie Gennari, Levent Burak Kara, Thomas F. Stahovich, and Kenji Shimada. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers Graphics*, 29(4):547–562, 2005. 2

[7] Leslie Gennari, Levent Burak Kara, Thomas F Stahovich, and Kenji Shimada. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers & Graphics*, 29(4):547–562, 2005.

[8] Frans CA Groen, Arthur C Sanderson, and John F Schlag. Symbol recognition in electrical diagrams using probabilistic graph matching. *Pattern Recognition Letters*, 3(5):343–350, 1985. 2

[9] D. Hemker, J. Maalouly, H. Mathis, R. Klos, and E. Ravanan. From schematics to netlists – electrical circuit analysis using deep-learning methods. *Advances in Radio Science*, 22:61–75, 2024. 2

[10] Cornelia Caragea Eduard Dragut Longin Jan Latecki Huitong Pan, Qi Zhang. Flowlearn: Evaluating large vision-language models on flowchart understanding. *arXiv preprint arXiv:2407.05183v1*, 2024. 8

[11] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016. 2

[12] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[13] Levent Burak Kara Luoting Fu. Recognizing network-like hand-drawn sketches: A convolutional neural network approach. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 671–681, 2010. 2

[14] Rahul Mehta, Bhavyajeet Singh, Vasudeva Varma, and Manish Gupta. Circuitvqa: A visual question answering dataset for electrical circuit images. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 440–460. Springer, 2024. 1, 2, 3, 4, 5

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[16] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 18–34, 2024. 2

[17] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023. 3

[18] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang, lizhming, and Tritin Truong. opencv/cvat: v1.1.0. 2020. 3

[19] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5(4):1680–1716, 2023. 3

[20] Felix Thoma, Johannes Bayer, Yakun Li, and Andreas Dengel. A public ground-truth dataset for handwritten circuit diagram images. In *International Conference on Document Analysis and Recognition*, pages 20–27. Springer, 2021. 2

[21] J.M. B Yoshi G, Areal T. Mermaid ai. https://www.mermaidchart.com/mermaid-ai?gad_source=1, 2025. 2, 3, 4