

Vision language models are *blind*

Pooyan Rahmazadehgervi^{1*} Logan Bolton^{1*}
 pooyan.rmz@gmail.com logan.bolton@auburn.edu

Mohammad Reza Taesiri^{2*} Anh Totti Nguyen¹
 mtaesiri@gmail.com anh.ng8@gmail.com

¹ Auburn University, USA

² University of Alberta, Canada

Abstract. While large language models with vision capabilities (VLMs), *e.g.*, GPT-4o and Gemini-1.5 Pro, are powering various image-text applications and scoring high on many vision-understanding benchmarks, they are still surprisingly struggling with low-level vision tasks that are easy to humans. Specifically, on BlindTest, our suite of 7 very simple tasks such as identifying (a) whether two circles overlap; (b) how many times two lines intersect; (c) which letter is being circled in a word; and (d) the number of circles in an Olympic-like logo, four state-of-the-art VLMs are only 58.07% accurate on average. Sonnet-3.5 performs the best at 77.84% accuracy, but this is still far from the human expected accuracy of 100%. Across different image resolutions and line widths, VLMs consistently struggle with those tasks that require precise spatial information when geometric primitives overlap or are close together. Code and data are at: [vlmsareblind.github.io](https://github.com/vlmsareblind)

Keywords: Language models · Benchmarks · Geometric primitives

1 Introduction











In the last eight months, the advent of VLMs, starting with GPT-4V(ision) [33], has enabled numerous, unprecedented image-text processing applications [43]. VLMs can accurately identify objects in a scene [9, 34, 43] and perform complex tasks based on these detected objects, *e.g.*, calculating the cost of beers on a table from an image of the scene and an image of the menu [44]. Interestingly, VLMs advance so quickly that describing *unusual* activities in an image [38] (*e.g.*, a man ironing on a moving taxi) has become a standard sanity check [10].

Existing VLM benchmarks cover a wide range of tasks [14, 26, 47]. However, they often assess a high-level human-vs-machine performance gap conflating both visual and non-visual abilities. Interestingly, the input images in so many questions, *e.g.*, 42.9% of MMMU [47], are *not* even necessary [6] for determining the





* All authors contributed to conducting experiments, analyzing results, and writing the paper.

correct answer. Many answers (1) can be inferred from the textual question and choices alone [6, 11]; and (2) are memorized by VLMs from their Internet-scale training [6]. In contrast, it is important to exclusively **measure the visual capabilities** of VLMs, independent of their strong language abilities.

In this paper, we test VLMs’ ability to *see* (not reasoning) on low-level vision tasks inspired by the “visual acuity” tests [5] given to humans by optometrists. We test four state-of-the-art (SOTA) VLMs: GPT-4o [32], Gemini-1.5 Pro [35], Claude-3 Sonnet [4], and Claude-3.5 Sonnet [2] on our suite of 7 extremely simple visual tasks that involve only 2D geometric primitives (*e.g.*, lines and circles) [12] and require minimal to zero world knowledge. Our key findings are:

1. Despite excellent performance on chart and diagram benchmarks [4, 32], VLMs cannot reliably tell whether two lines (or two circles) are intersecting, especially when close together. Accuracy in detecting 0, 1 or 2 intersections in a line chart  of two 2-segment piecewise-linear functions ranges from ~41% to 76% (Sec. 4.1). For the two-circle  task, VLMs perform better (~75–93% accuracy), but still far from the expected 100% (Sec. 4.2).
2. VLMs can perfectly recognize a circle () and a word (Subdermatoglyphic) separately. Yet, when the circle is superimposed on the word (Subde)matoglyphic), models tend to struggle to identify which letter is being circled (Sec. 4.3).
3. VLMs can accurately count shapes, *e.g.*, circles () that are disjoint and far apart. However, all VLMs struggle to count intersecting circles  (like the Olympic logo), and, generally, primitive shapes (, , ) that are overlapping or nested (Sec. 4.4).
4. Tiling up squares into a grid , we find VLMs to fail to count the number of rows or columns in the grid, whether empty or containing text (Sec. 4.5). This is in stark contrast to VLM high performance ($\geq 90\%$ accuracy) [32, 35] on DocVQA [27], which includes many questions with tables.
5. When tasked with tracing colored paths in a simplified subway map of only 2 to 8 paths and a total of 4 stations, VLMs often fail to count the paths between two stations, *i.e.*, with an accuracy of ~31% to 58% (Sec. 4.6).
6. GPT-4o is better than Gemini-1.5 Pro on 7 existing complex VLM benchmarks [32, 35] but worse on BlindTest. On average across all 7 tasks, VLMs perform at 58.07% accuracy with Sonnet-3.5 being the best (77.84% accuracy), which is still far lower than the expected 100% accuracy of humans (see Tab. 1). In sum, BlindTest reveals some remarkable VLM limitations that are not measured in prior benchmarks.

2 Vision language models

Our goal is to study how SOTA VLMs perceive simple images composed of *interacting* geometric primitives. We test four SOTA models: GPT-4o () , Gemini-1.5 Pro () Gemini-1.5), Claude-3 Sonnet () Sonnet-3), and Claude-3.5 Sonnet () Sonnet-3.5) that are ranking highest on 7 recent multimodal vision benchmarks (see [32] and Table 10 in [35]), which cover multi-discipline, college-level

subjects in MMMU [47], science diagrams in AI2D [14], mathematics in MathVista [23], charts in ChartQA [26], documents in DocVQA [27], and videos in ActivityNet-QA [46] & EgoSchema [24]. We initially run experiments with Claude 3 Opus [3] but swap it with **Sonnet-3.5**, which performs more accurately on **BlindTest** and costs 5× less. All models tested are described in ??

Open-source For completeness, we also test 8 *open-source* models of varying sizes (from 0.5B to 72B parameters) across three different families: LLaVA OneVision-qwen2 [16], Phi-3.5-vision-instruct [1], and InternVL-2 [7]. Yet, they underperform the four closed-source models described above (see their results in ??).

3 BlindTest benchmark of 7 tasks

Eye exams Like humans’ visual acuity tests [5], we design a set of 7 very simple, yet novel tasks that are composed of common geometric primitives. We do not use the existing tests designed for human-eye exams for two reasons. First, we avoid using the questions that exist on the Internet, which may provide an inflated measure of vision capabilities [6, 11, 45]. Second, our preliminary experiments show that **GPT-4o** *already performs very well* on humans’ eye exams, which typically contain single, separate symbols—*e.g.*, the Snellen chart [5], tumbling E [5], and contrast sensitivity charts [13, 25].

Motivation Our **BlindTest** benchmark tests VLMs on identifying known geometric primitives when they are close together, overlapping, or intersecting. We hypothesize that VLMs will struggle because they mostly rely on “late fusion” [21, 39], *i.e.*, first extracting visual representations *without* considering the textual question, and then feeding them to a large language model (LLM) for processing. Therefore, while geometric primitives in **BlindTest** are well known, their exact spatial information on a white canvas (*e.g.*, the size and position of a \bigcirc) is typically not describable in natural language, even for humans, and may not be captured by the vision encoders trained mostly on natural images.

Controls For each test image, we prompt VLMs using **two** different, yet semantically equivalent questions. Furthermore, we test VLMs on multiple versions of each task across **three** different image sizes (Secs. 3.1, 3.2, 3.4, 3.6 and 3.7) and **two** to **three** line thickness values (Secs. 3.1 and 3.4 to 3.7).

3.1 Task 1: Counting line intersections

Given the impressive accuracy of VLMs on answering questions on diagrams and charts (*e.g.*, **Sonnet-3.5** scoring 94.7% on AI2D and 90.8% on ChartQA) [2], a reasonable hypothesis is that VLMs must be able to see if two graphs intersect in a chart. Here, we test this hypothesis by asking VLMs to count the number of intersections (0, 1 or 2) between two 2-segment piece-wise linear functions.

Images We create 1,800 images (??) of 2D line plots drawn on an image of size of $C \times C$, where $C \in \{384, 768, 1152\}$. Each line plot consists of two line segments, defined by three points whose x-coordinates are fixed at $\{0, \frac{C}{2}, C\}$ px

(see ??). The y-coordinates are randomly sampled from a pre-defined, invisible 12×12 grid to ensure there is sufficient spacing between two plots and that there are exactly 0, 1 or 2 intersections. See ?? for more details.

3.2 Task 2: Two circles ●●

In the task of counting line intersections (Sec. 3.1), each image contains two long, thin colored lines on a large white canvas. Here, we test models in a complementary setting where the two interacting objects (here, two same-sized filled circles ●●) are larger while their gap is smaller. This task evaluates VLM ability in detecting (1) a small gap between two circles; and (2) that two circles are overlapping, *i.e.*, no gaps. We vary circle and gap sizes and ask VLMs if two circles are (a) overlapping or (b) touching each other.

Images Given a blank image of size $C \times C$, we draw two same-sized circles of diameter $\phi \in \{\frac{C}{4}, \frac{C}{5}, \frac{C}{6}, \frac{C}{7}\}$ with a boundary-to-boundary distance = $\phi \times d$ where $d \in \{-0.25, -0.2, -0.15, -0.1, -0.05, 0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$ to cover all three cases: overlapping, tangent, and disjoint (see ??a). The two circles are arranged in four different orientations, making a 90° , 0° , -45° , and 45° angle with the x-axis (??b). The whole grid sampling generates 224 images per image size. We replicate the procedure for 3 image sizes, *i.e.*, $C = 384, 769, 1155$ px to create a total of $3 \times 224 = 768$ images. See ?? for more details.




3.3 Task 3: The circled letter Subdermatoglyphic



Consistent with prior reports [36, 43, 44], we find that VLMs can 100% accurately identify a primitive shape (*e.g.*, a red circle ○) [36] and can perfectly read an English word (*e.g.*, Subdermatoglyphic) alone. Here, we superimpose the red circle on every letter, one at a time, in the word, and ask VLMs to identify which letter is being circled. While the task is easy to humans, our hypothesis is that if a VLM’s vision is “blurry”, it might not be able to identify the exact letter being circled since there is tiny spacing between the adjacent letters.

Images We choose three strings Acknowledgement, Subdermatoglyphic, and tHyUiKaRbNqWeOpXcZvM because they contain characters of variable widths and heights. Furthermore, all four tested VLMs can read out all characters in these strings when they are input to the models as an image. While Acknowledgement is a common English word, Subdermatoglyphic is the longest word without repetitive letters. We also test VLMs on the random string tHyUiKaRbNqWeOpXcZvM to estimate how much model accuracy is due to its familiarity with the word.


For each (string, circled-letter) pair, we render a 512×512 image by choosing among 3 red oval line-thickness levels, 2 font families, and 4 different values of image padding for a total of 24 images. That is, we generate 360, 408, and 480 images for Acknowledgement (15 letters), Subdermatoglyphic (17 letters), and tHyUiKaRbNqWeOpXcZvM (20 letters), respectively. We ensure each letter to be circled fits completely the oval ○ (see ??). See ?? for more details.

3.4 Task 4: Counting overlapping shapes

Aligned with prior research [44], we also find VLMs to be able to count disjoint circles (). Yet, here, we test VLMs on counting circles that are *intersecting* () like in the Olympic logo—a common cognitive development exercise for preschoolers [31, 37]. Our hypothesis is that a “blurry” vision may not see the intersection between two circles clearly and therefore unable to trace circles and count them. For generalization of our findings, we repeat the experiment with pentagons () as well (instead of circles).





Images In an image of size $C \times C$, where $C \in \{384, 769, 1155\}$ px. We draw $N \in \{5, 6, 7, 8, 9\}$ overlapping, same-sized circles arranged in two rows like the Olympic logo (see ??). A circle diameter $\phi \in \{\frac{C}{7}, \frac{C}{10}\}$. We repeat the images with two different line thicknesses for rendering circles. This procedure renders 3 resolutions \times 5 values of $N \times 2$ diameters \times 2 line widths \times 2 color options = 120 images. We also vertically flip all 120 images, resulting in a total of 240 images. We repeat for pentagons () in addition to circles () , resulting in 240×2 shapes = 480 images in total. For pentagons, their side length $d \in \{\frac{C}{7}, \frac{C}{10}\}$. See ?? for more details.

3.5 Task 5: Counting the nested squares

In addition to testing VLMs on counting the **intersecting** circles (Sec. 3.4), here, we test a complementary setting by arranging the shapes so that their edges do *not* intersect. That is, each shape is **nested** entirely inside another (see ??). For completeness, we test squares () in this task.

Images In an image of size 1000×1000 px, we render $N \in \{2, 3, 4, 5\}$ nested squares one at a time from the largest to the smallest. First, the outermost square is rendered using a random edge length d . And each subsequent smaller square is placed randomly inside the previous one and has an edge length of 75% of that of the outer square. We render squares using a line width of $\{3, 4, 6\}$ px and ensure no squares touch by edges. For each line width, we generate 10 images (where squares have different, random locations) to create $3 \times 10 = 30$ images. Repeating the process for all N values results in $4 \times 30 = 120$ images. See ?? for more details.

3.6 Task 6: Counting the rows and columns of a grid

The results from prior tasks show VLMs cannot always count shapes that are overlapping (Sec. 3.4) or nested  (Sec. 3.5). What about adjacent shapes  ? Here, we tile up shapes (specifically, ) into a grid and challenge VLMs to count—a task that is supposedly simple to VLMs given their remarkable performance ($\geq 90\%$ accuracy) [32, 35] on DocVQA [27], which includes many questions with tables. To simplify the task, we ask models to count the number of rows and columns in a given table (either empty or text-containing).

Images A grid may have $N \times N$, $N \times N'$, or $N' \times N$ cells, where $N \in \{3, 4, 5, 6, 7, 8, 9\}$ and $N' = N + 1$. We also include grids of size 10×10 to balance the benchmark with the row and column sizes. Each grid is rendered with two different line

widths on a canvas of size $C \times C$ where $C \in \{500, 1250, 2000\}$ px. Besides empty grids, we also replicate the procedure to make grids contain text (which is more common in real-world tables) where each cell contains a single random English word (see ??). Both versions (empty and text-containing) combined have $2 \times 132 = 264$ images. See ?? for more details.













3.7 Task 7: Following single-colored paths

It is important for VLMs to be able to follow paths in order to read maps or charts [26], interpret graphs [15], and user annotations (*e.g.*, arrows) in input images [44]. To assess path-following capability, this task asks models to count the unique-color paths between two given stations in a simplified subway map.

Images We create each subway map on an image of size $C \times C$, where $C \in \{512, 1024\}$ px (see ??). We write 4 station names (A, B, C, D) at 4 fixed coordinates $\in \{(\frac{C}{2}, C), (C, \frac{C}{2}), (\frac{C}{2}, 0), (0, \frac{C}{2})\}$, respectively. We divide the canvas into an invisible grid of 18×18 cells and initialize 3 path-starting points $\frac{C}{18}$ px away from each station. We draw a path, using the depth-first search algorithm starting from a random station and a random starting point, where a valid move is one cell in any direction: North, south, east or west. We repeat the process so that each station has exactly $N \in \{1, 2, 3\}$ outgoing paths, for a total of 180 maps. See ?? for details.

4 Results

Table 1: Accuracy (%) of models over 7 tasks in **BlindTest**. The mean accuracy over all four models is 58.07% substantially better than random chance (24%), which is computed considering each task as a single-label, N -way classification problem. **Sonnet-3.5** is the best (77.84% accuracy) but still far from the 100% expected accuracy. Note that the best performing open-source model (??) is only on par with **Sonnet-3** here.

Model	a. 	b. 	c. 	d. 	e. 	f. 	g. 	h. 	i.	Task mean
Random	33.33	50.00	5.77	20.00	20.00	25.00	4.55	33.33	24.00	
 GPT-4o	41.61	75.91	74.23	41.25	20.21	55.83	39.58	53.19	50.23	
 Gemini-1.5	66.94	93.62	83.29	20.25	24.17	87.08	39.39	53.13	58.48	
 Sonnet-3	43.41	86.46	72.06	29.79	1.87	65.00	36.17	31.11	45.73	
 Sonnet-3.5	75.36	90.82	87.88	66.46	77.71	92.08	74.26	58.19	77.84	
Mean	56.84	86.70	79.36	39.44	30.99	74.99	47.35	48.90	58.07	

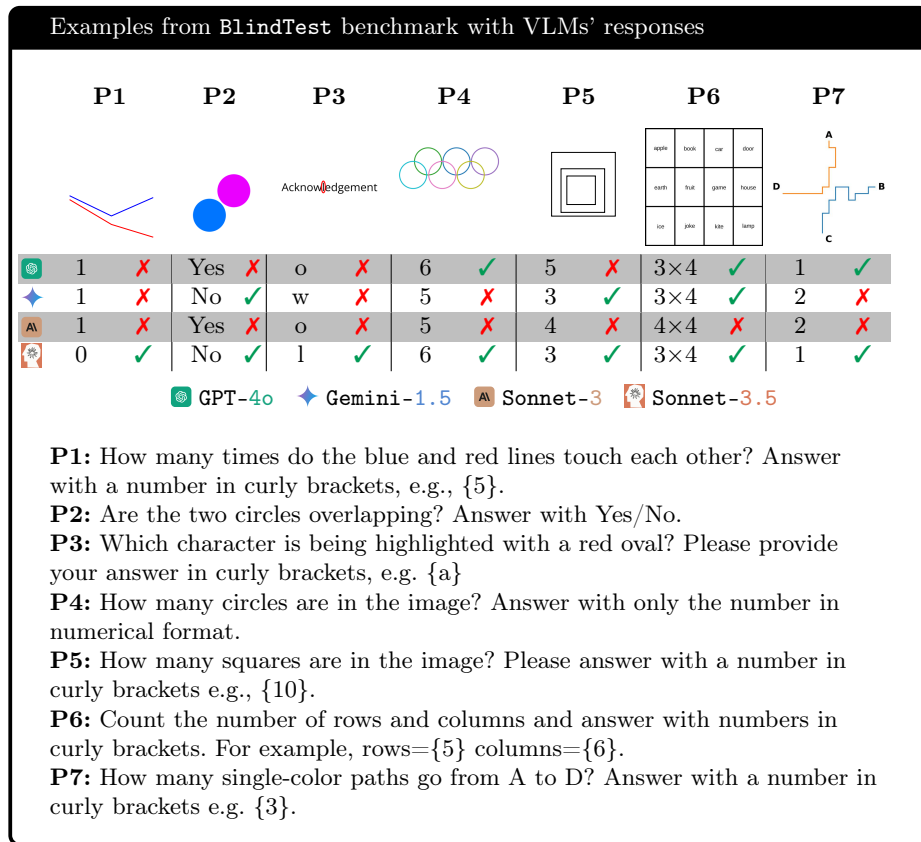






Fig. 1: VLMs fail on the simple tasks of BlindTest.

4.1 VLMs cannot reliably count line intersections

Experiment We parse every model’s response to extract the final answer and then compare it to the groundtruth. We report the mean accuracy of every model on two prompts and analyze how accuracy changes as we vary hyperparameters (e.g., line widths and image sizes).

Results First, across two prompts and two line widths, all VLMs are 56.84% accurate (Tab. 1a), far from the expected 100% accuracy on this easy task (??). The best accuracy is only 75.36% (Sonnet-3.5) (Tab. 2). Specifically, **VLMs tend to perform worse when the distance between two plots narrows** (Fig. 2). As each line plot is composed of three key points, the distance between two plots is computed as the mean distance over three corresponding point pairs. See Fig. 1 and ?? for more samples of VLM predictions. VLMs perform similarly across three image sizes (??).

Table 2: The accuracy breakdown by line width in pixels (where C = image width), averaged over two prompts, shows that VLMs cannot reliably count the intersections between two simple 2D line plots.

Line width				
$0.005 \times C$	45.00	67.55	45.22	75.83
$0.010 \times C$	38.22	66.33	41.61	74.88
Mean	41.61	66.94	43.41	75.36

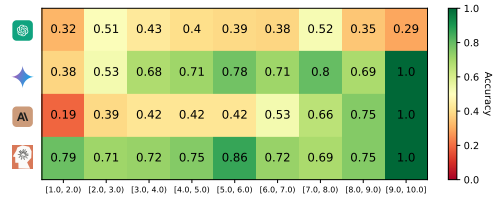




Fig. 2: As a line-plot image is divided into a 12×12 grid, the x-axis shows the mean **distance** (in grid cells) over 3 pairs of points of two 2-segment plots. VLMs are often more confused when two plots are closer together (left) than when they are further apart (right).

Our findings are in stark contrast to the high accuracy of VLMs on ChartQA [32, 35], suggesting that VLMs can recognize the overall trend of a line plot but unable to “zoom in” to see fine details, *e.g.*, which lines are intersecting.

4.2 VLMs cannot clearly see if two circles overlap or not

Motivated by VLM poor performance in counting line intersections (Sec. 4.1), here, we replace lines by large, filled circles and ask VLMs explicitly if the two circles are touching (or overlapping).

Experiment Since we instruct VLMs to output a binary answer (Yes/No), we use Python to extract VLMs’ formatted answer from their responses for comparing with groundtruth.

Results Surprisingly, even on this task where objects ( ) are large and clearly visible to humans, no VLMs are able to solve it perfectly with a mean accuracy of 86.70% (Tab. 1b). The best accuracy is 93.62% (**Gemini-1.5**) over all images and two prompts (Tab. 3). A common trend is **when two circles are close together, VLMs tend to perform poorly**, making educated guesses, *e.g.*, **Sonnet-3.5** often answers “No” conservatively (??). **GPT-4o** performs the worst and shockingly is not 100% accurate even when the distance between two circles is as large as one radius (Fig. 3; $d = 0.5$). That is, consistent with the results from Sec. 4.1, VLMs seem to be unable to always detect the gap or intersection between two filled circles (Fig. 1 and ??).

An explanation is that due to the late-fusion mechanism [39], VLMs extract visual features from the image *before* even looking at the question, causing this “blindness”. In contrast, if a model first knows that the question asks it to focus on the area between the two circles, it then might be able to extract accurate visual information to answer such simple questions.

While VLMs perform consistently across three image resolutions (??), every model performs the best at a specific circle orientation (??). Moreover, VLMs’ performance does not change substantially ($\pm 5.79\%$ for **Sonnet-3.5** and

$\pm 10.81\%$ for GPT-4o) when tested against different colors (??), ruling out the impact of color on their inability to see the overlapping or touching circles. More examples of VLMs’ answers are in ??.

Table 3: GPT-4o and Gemini-1.5 perform more consistently over the two different prompts (“overlapping” and “touching”) than Sonnet-3 and Sonnet-3.5.





Model	Overlapping	Touching	Mean
	74.74	77.08	75.91
	94.01	93.23	93.62
	89.58	83.33	86.46
	86.46	95.18	90.82



Fig. 3: VLMs perform poorly when two circles are tangent ($d = 0.0$) or close together ($d = 0.05, 0.10$). Yet, Sonnet-3.5 is better at $d \geq 0.0$. (perhaps due to its tendency to answer “No”).

4.3 VLMs do not always see the letter inside the red circle

Experiment To evaluate the models’ ability to recognize individual characters in an image, we place a red circle over one character in a word. We then prompt the models to put their prediction in {curly braces} and then we compare the lower case version of this character to the lower case version of the ground truth character.

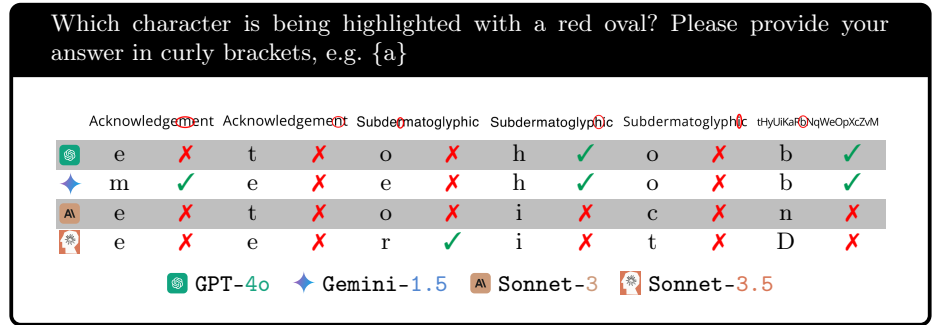



Fig. 4: Identifying the letter being circled is non-trivial for VLMs across both English words (Acknowledgement & Subdermatoglyphic) and a random string (tHyUiKaRbNqWeOpXcZvM). When making mistakes, VLMs tend to predict letters adjacent to the circled one.

Results All VLMs can accurately spell out the string when there is a red oval  superimposed on the image. Yet, interestingly, reading out which letter is being circled turns out to be a challenge (mean model accuracy: 79.7%; Tab. 1c).

When the letters are close together, VLMs often predict letters adjacent to the one being circled (see the confusion matrix in ?? and more results in Fig. 1 and ??). Sometimes models hallucinate, *e.g.*, coming up with characters non-existent in Subdermatoglyphic (*e.g.*, “9”, “n”, “©”) despite having the ability to accurately spell out the word (see ??). We also observe that VLMs, on average, fail to see the circled letter across various common English words (mean accuracy is 86.43% in ??). However, as the words get shorter in length and there is no repetitive letters in them, VLMs tend to perform better. More failure cases are reported in ????.

On average, models perform better (+0.46 to +13 points) on the two English words compared to the random string (??), suggesting that **knowing the word help VLMs make better educated guesses**, slightly improving accuracy.

Sonnet-3.5 and **Gemini-1.5** are the top-2 models (87.88% and 83.29%) and are better than **GPT-4o** and **Sonnet-3** by a large margin of nearly +15 points (??). VLMs perform similarly across two prompts (??) and two font families (??). See also ?? for an example of **GPT-4o** and **Gemini-1.5** making educated guesses on the color of the overlapping area between two overlapping circles (Task 1).

4.4 VLMs struggle to count overlapped and nested shapes

Experiment We run all VLMs on all images of overlapping circles and pentagons (Sec. 3.4) and nested squares (Sec. 3.5). We prompt models to output the predicted number of shapes in a formatted answer. We compare extracted answers with groundtruth. For each shape (circles, pentagons and squares), we run two different prompts.

Results On counting overlapping circles, pentagons, and nested squares, VLM mean accuracy is 39.44%, 30.99%, and 74.99%, respectively (Tab. 1d–f). That is, counting shapes is not easy to models regardless of whether the shapes are overlapped or nested, *i.e.*, their edges intersect or not (Fig. 1 and ??). On nested squares, model accuracies vary widely—**GPT-4o** (55.83%) and **Sonnet-3** (65.00%) are at least -25 points behind **Gemini-1.5** (87.08%) and **Sonnet-3.5** (92.08%). This gap is even larger on counting overlapped circles and pentagons—**Sonnet-3.5** is better than the other models by multiple times (*e.g.*, 77.71% vs. 1.87% of **Sonnet-3**; Tab. 1).

All four models are at least 83% accurate in counting 5 circles. Yet, surprisingly increasing the number of circles by one is sufficient to cause accuracy to dip substantially to near zero for all models, except **Sonnet-3.5** (Fig. 5; column 6–9). In counting pentagons, all VLMs (except **Sonnet-3.5**) perform poorly even at 5 pentagons. Overall, **counting from 6 to 9 shapes (both circles and pentagons) is hard for all models**.

We further investigate why VLMs are nearly perfect at counting 5 circles but struggle to count 5 pentagons or more than 5 shapes in general. When there are

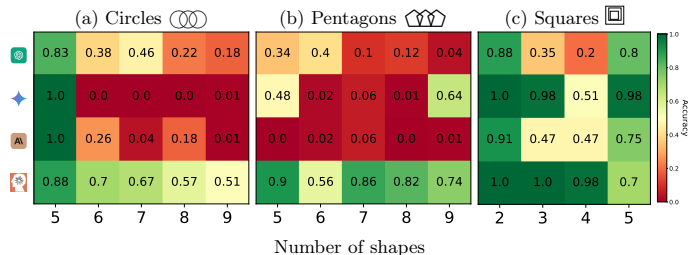


Fig. 5: All four VLMs can count 5 circles well (leftmost; **0.83**), but only **Sonnet-3.5** can count 5 overlapping pentagons well at 0.9 accuracy (b). Counting from 6–9 shapes (either \circ or \pentagon) is **challenging** to VLMs. Interestingly, **GPT-4o** (G) and **Sonnet-3** (A) are unable to count **two** nested squares reliably, *i.e.* 0.88 and 0.91 accuracy (c).

more than five circles (\circ) and VLMs predict an incorrect count, **Gemini-1.5** predicts “5” 99.74% of the time regardless of the actual number of circles (??). For other models, this frequency is also much higher than that in the case of pentagons. Our results show strong evidence that **VLMs are biased towards the well-known 5-circle Olympic logo** (more results on this bias in ??).

GPT-4o performs better on colored shapes than on black shapes and **Sonnet-3.5** is increasingly better as the image size increases. However, the accuracy of all other models only change marginally as we change colors (??) and image resolutions (??).

Note that there are only 2 to 5 squares in each image in the task of counting nested squares and these squares do not intersect (??). Surprisingly, **GPT-4o** and **Sonnet-3** are still unable to perfectly count two and three nested squares (Fig. 5c). When the count increases to four and five, all models are far from 100% accurate (Fig. 5c). Our results show that it is not easy for VLMs to extract accurate representation of shapes even when their edges do not intersect.

4.5 VLMs cannot easily count the rows and columns in a grid

Since VLMs struggle in counting the number of simple shapes when the shape edges intersect (Sec. 3.4) or separate (Sec. 3.5), here, we test the remaining case where these shapes are placed adjacently sharing edges, specifically, tiling up multiple rectangles into a single grid. Given the impressive accuracy of VLMs [2, 32, 35] on questions involving tables and spreadsheets in DocVQA [27], we hypothesize that VLMs must be able to count the rows and columns in a grid.

Experiment We run all VLMs on the images of empty grids and text-containing grids (Sec. 3.6) and analyze their formatted answers.

Results First, VLMs surprisingly performs poorly (34.37% accuracy) in counting the rows and columns in an **empty** grid (see ??). Specifically, they are often off by one or two (*e.g.*, **GPT-4o** predicts 4×4 and **Gemini-1.5** predicts 5×5 for a 4×5 grid; ?? and Fig. 1). This finding suggests that VLMs can extract important content from a table to answer table-related questions in DocVQA [27] but do not clearly “see” a table cell-by-cell as a human does.

This might be because tables in documents are mostly non-empty and VLMs are not used to them. Aligned with that hypothesis, after **adding a single word to each cell**, we observe the accuracy of all VLMs to increase **almost twice** (e.g., from 26.13% to 53.03% for GPT-4o) (??). Yet, no models can solve this task with the best model (Sonnet-3.5) performing at 88.68% on text-containing grids and 59.84% on empty grids (Fig. 6a vs. b).

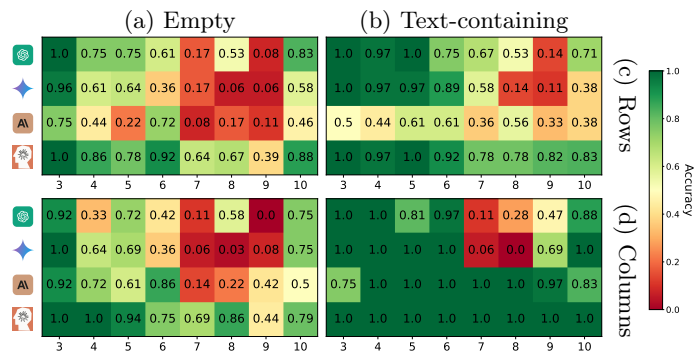


Fig. 6: Accuracy of counting rows (c) vs. columns (d) (here, analyzed separately) when the grids are empty (a) vs. contain text (b). VLMs (especially, \star and AI) generally count much more accurately when grids contain text vs. empty grids (b vs. a). Interestingly, columns are also easier for VLMs to count than rows (d vs. c).

Interestingly, **VLMs are better at counting columns than rows**—70.53% vs. 60.83% accuracy (Fig. 6c vs. d). However, these numbers are still far from 100% showing that VLMs currently cannot count neither rows or columns in a table reliably. See ???? for more results.

4.6 VLMs struggle to count single-colored paths

This path-counting task tests a VLM’s ability in recognizing a path of a unique color and *trace* it from a given starting station to the destination, an important task in reading maps and graphs in general [26].

Experiment From a subway map (Sec. 3.7), we randomly sample 2 connected stations and prompt every model to count the single-colored paths that connect them. We extract numbers from VLM templated responses and compare them with the groundtruth.

Results Overall, VLMs perform poorly at a mean accuracy of 48.90% (Tab. 1h). Even when there is only *one path* between two stations, no models can reach 100% accuracy (the best is Sonnet-3.5 at 93.33% and the worst is 20%; ??). VLM predicted counts are often off by 1 to 3 paths (??). VLM accuracy reduces substantially, e.g., Sonnet-3.5 from 93.33% to 58.33% and 22.91% as the complexity of the maps increases from 1, 2 to 3 paths, respectively (??). More samples of VLM responses are in Fig. 1 and ??.

5 Related Work

Benchmarking VLM vision understanding College-level topics [47], charts [26], documents [27] or videos [46] are among the common benchmarks for assessing VLM vision understanding [2, 4, 32, 35] and are witnessing VLMs’ recent rapid progress—*e.g.*, **Sonnet-3.5** is reaching 95.2% on DocVQA, 90.8% on ChartQA, and 94.7% on AI2D [2]. However, most of vision benchmarks attempt to evaluate VLMs on real-world, topic-specific data that require extensive prior knowledge [6, 17, 41], which has a “data leakage” problem, *i.e.*, VLMs many times can answer accurately without even the input image [6]. Furthermore, most benchmarks test VLMs on the data that humans have to deal with to provide a high-level sense of the human-machine intelligence gap [22, 45]. In contrast, our **BlindTest** benchmark differs significantly from prior benchmarks because it is (1) **extremely easy to humans and can be solved by a 5-year-old** (unlike [26, 27, 47]); (2) the first low-level, visual sanity check for VLMs; (3) requiring minimal to zero prior knowledge; (4) requiring minimal commonsense or complex reasoning (unlike [8, 48])—*i.e.*, **a strong language model is of little use here when it is non-natural for humans to describe BlindTest images in language.**

The ARC benchmark [8, 30] also contains abstract images made up of simple shapes; however, it challenges VLMs to understand and reason based on those patterns. That is, ARC assumes VLMs can identify the abstract shapes in order to reason. In contrast, our **BlindTest** directly evaluates VLM capabilities in recognizing these primitive shapes.

Improving VLM vision capabilities Most recent recipes for improving SOTA VLMs involve finetuning a pretrained LLM coupled with vision encoders to solve high-level vision tasks [20]. Such late-fusion approaches fuse visual representations learned from the tokenized image with a powerful thinking brain [18, 19, 28]. However, current vision approaches for VLMs are facing challenges as models sometimes are “blind”—unable to see natural objects exist in a real photo [40]. In contrast, we are showing VLMs are visually impaired at low-level abstract images, *e.g.*, inability to count 6 overlapping circles or 3 nested squares.

Our circled-letter task (Sec. 3.3) is inspired by VLM abilities in recognizing content inside a red circle over real objects in natural images [36, 43, 44]. In contrast, we show that VLMs can fail at a low-level, optical character recognition as opposed to recognizing real objects. To the best of our knowledge, no prior attempts have been made to address the exact limitations raised in our paper: (1) identifying and counting simple lines, shapes and geometric primitives when they interact (Sec. 4.1 to Sec. 4.5); (2) following colored paths (Sec. 4.6). Solving these limitations may be the foundation for VLMs to progress on some existing vision benchmarks on graphs, *e.g.*, [15], visual math [23] and some existing blind-spots in natural images (*e.g.*, understanding the direction an object is facing [40]).

6 Discussion and Conclusion

We propose **BlindTest**, a benchmark of seven novel low-level visual tasks for testing VLM ability to “see” simple geometric primitives (such as line, circles, squares, intersections) that are the basic building blocks for many image tasks. The tasks are designed from scratch and require minimal to zero knowledge. As the tasks did not exist on the Internet before and require minimal world knowledge, there is minimal chance that VLMs can solve **BlindTest** by memorization or by not using the input image—an issue in some prior benchmarks [6, 11].

Furthermore, we also test common prompting techniques (??) including 2-shot, chain-of-thought [42], and meta-prompting [29] but do not obtain better accuracy, which (1) suggests that VLMs understand **BlindTest** questions and (2) confirms that these visual tasks do not benefit from thinking aloud [42].

The poor performance of VLMs on **BlindTest** suggests that models will perform poorly on the real-world visual tasks that require them to follow arrow directions or paths, (*e.g.*, reading subway maps in ??, street maps or directed graphs in ??), perceive lines and intersections (*e.g.*, reading music sheets; ??), identify and counts objects in a crowded scene.

Acknowledgement

We thank Hung H. Nguyen, Thang Pham, Ali Yildirim, Giang Nguyen, and Tin Nguyen at Auburn University for feedback and discussions of the earlier results. We are also thankful for the API research credits from Anthropic and `together.ai` to MRT. AN was supported by the NSF Grant No. 1850117 & 2145767, and donations from NaphCare Foundation & Adobe Research.

References

1. Abdin, M., Jacobs, S.A., Awan, A.A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H., et al.: Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint arXiv:2404.14219 (2024)
2. Anthropic: Introducing claude 3.5 sonnet \ anthropic. <https://www.anthropic.com/news/claude-3-5-sonnet>, (Accessed on 07/03/2024)
3. Anthropic: Introducing the next generation of claude \ anthropic. <https://www.anthropic.com/news/claude-3-family>, (Accessed on 07/23/2024)
4. Anthropic, A.: The claude 3 model family: Opus, sonnet, haiku. Claude-3 Model Card (2024)
5. Bailey, I.L., Lovie-Kitchin, J.E.: Visual acuity testing. from the laboratory to the clinic. *Vision research* **90**, 2–9 (2013)
6. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? arXiv preprint arXiv:2403.20330 (2024)
7. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)

8. Chollet, F.: On the measure of intelligence. arXiv preprint arXiv:1911.01547 (2019)
9. Custer, G.: Gemini spatial example. <https://gemini-spatial-example.grantcuster.com/>, (Accessed on 05/31/2024)
10. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning (2023)
11. Hegde, N., Paul, S., Madan, G., Aggarwal, G.: Analyzing the efficacy of an llm-only approach for image-based document question answering. arXiv preprint arXiv:2309.14389 (2023)
12. Hughes, J.F.: Computer graphics: principles and practice. Pearson Education (2014)
13. Inc., C.Z.V.: Vision screening. <https://www.zeiss.com/vision-care/us/eye-health-and-care/vision-screening.html>, (Accessed on 07/03/2024)
14. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: 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. pp. 235–251. Springer (2016)
15. yunxin li, Hu, B., Shi, H., Wang, W., Wang, L., Zhang, M.: Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. In: Forty-first International Conference on Machine Learning (2024), <https://openreview.net/forum?id=gjoUXwuZdy>
16. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
17. Liang, Z., Guo, K., Liu, G., Guo, T., Zhou, Y., Yang, T., Jiao, J., Pi, R., Zhang, J., Zhang, X.: Scemqa: A scientific college entrance level multimodal question answering benchmark. arXiv preprint arXiv:2402.05138 (2024)
18. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
19. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), <https://llava-vl.github.io/blog/2024-01-30-llava-next/>
20. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
21. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36** (2024)
22. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
23. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In: International Conference on Learning Representations (ICLR) (2024)
24. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. Advances in Neural Information Processing Systems **36** (2024)
25. Mäntyjärvi, M., Laitinen, T.: Normal values for the pelli-robson contrast sensitivity test. Journal of Cataract & Refractive Surgery **27**(2), 261–266 (2001)
26. Masry, A., Do, X.L., Tan, J.Q., Joty, S., Hoque, E.: ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In: Muresan, S.,

- Nakov, P., Villavicencio, A. (eds.) Findings of the Association for Computational Linguistics: ACL 2022. pp. 2263–2279. Association for Computational Linguistics, Dublin, Ireland (May 2022). <https://doi.org/10.18653/v1/2022.findings-acl.177>, <https://aclanthology.org/2022.findings-acl.177>
27. Mathew, M., Karatzas, D., Jawahar, C.: Docvqa: A dataset for vqa on document images. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2200–2209 (2021)
 28. McKinzie, B., Gan, Z., Fauconnier, J.P., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., Belyi, A., Zhang, H., Singh, K., Kang, D., Jain, A., He, H., Schwarzer, M., Gunter, T., Kong, X., Zhang, A., Wang, J., Wang, C., Du, N., Lei, T., Wiseman, S., Yin, G., Lee, M., Wang, Z., Pang, R., Grasch, P., Toshev, A., Yang, Y.: Mml: Methods, analysis & insights from multimodal llm pre-training. ArXiv [abs/2403.09611](https://arxiv.org/abs/2403.09611) (2024), <https://api.semanticscholar.org/CorpusID:268384865>
 29. Mirza, M.J., Karlinsky, L., Lin, W., Doveh, S., Micorek, J., Kozinski, M., Kuhene, H., Possegger, H.: Meta-prompting for automating zero-shot visual recognition with llms. arXiv preprint [arXiv:2403.11755](https://arxiv.org/abs/2403.11755) (2024)
 30. Mitchell, M., Palmarini, A.B., Moskvichev, A.K.: Comparing humans, GPT-4, and GPT-4v on abstraction and reasoning tasks. In: AAAI 2024 Workshop on "Are Large Language Models Simply Causal Parrots?" (2023), <https://openreview.net/forum?id=3rGT50kzpC>
 31. Olya.by@mail.ru: How many counting game with color simple geometric shapes for kids, educational maths task for the development of logical thinking, preschool worksheet activity, count and write the result, vector stock vector by @olya.by@mail.ru 266096226. <https://depositphotos.com/vector/how-many-counting-game-with-color-simple-geometric-shapes-for-kids-educational-maths-task-for-266096226.html>, (Accessed on 07/05/2024)
 32. OpenAI: Hello gpt-4o | openai. <https://openai.com/index/hello-gpt-4o/>, (Accessed on 05/31/2024)
 33. OpenAI: Gpt-4 technical report (2023)
 34. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: Glamm: Pixel grounding large multimodal model. The IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
 35. Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint [arXiv:2403.05530](https://arxiv.org/abs/2403.05530) (2024)
 36. Shtedritski, A., Rupprecht, C., Vedaldi, A.: What does clip know about a red circle? visual prompt engineering for vlms. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11953–11963. IEEE Computer Society, Los Alamitos, CA, USA (oct 2023). <https://doi.org/10.1109/ICCV51070.2023.01101>, <https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01101>
 37. Station, M.T.: Count shapes printables | myteachingstation.com. <https://www.myteachingstation.com/preschool/math/numbers/count-shapes-printables>, (Accessed on 07/05/2024)
 38. Taesiri, M.R., Feng, T., Bezemer, C.P., Nguyen, A.: Glitchbench: Can large multimodal models detect video game glitches? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22444–22455 (2024)

39. Team, C.: Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint arXiv:2405.09818 (2024)
40. Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., Xie, S.: Eyes wide shut? exploring the visual shortcomings of multimodal llms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9568–9578 (June 2024)
41. Wang, K., Pan, J., Shi, W., Lu, Z., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset (2024)
42. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **35**, 24824–24837 (2022)
43. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v (2023), <https://arxiv.org/abs/2310.11441>
44. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C., Liu, Z., Wang, L.: The dawn of llms: Preliminary explorations with gpt-4v(ision). *CoRR* **abs/2309.17421** (2023). <https://doi.org/10.48550/ARXIV.2309.17421>, <https://doi.org/10.48550/arXiv.2309.17421>
45. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. In: International conference on machine learning. PMLR (2024)
46. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: a dataset for understanding complex web videos via question answering. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. pp. 9127–9134 (2019)
47. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In: Proceedings of CVPR (2024)
48. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6720–6731 (2019)