In [ ]:

Multimodal Retrieval Augmented Generation (RAG) using Gemini API in Vertex AI

Run in Colab

Run in Colab Enterprise

View on GitHub

Open in Vertex AI Workbench

Open in Cloud Skills Boost

Share to:

Author(s)	Lavi Nigam

⚠️ There is a new version of this notebook with new data and some modifications here: ⚠️

building_DIY_multimodal_qa_system_with_mRAG.ipynb

You can, however, still use this notebook as it is fully functional and has updated Gemini and text-embedding models.

Overview

Retrieval augmented generation (RAG) has become a popular paradigm for enabling LLMs to access external data and also as a mechanism for grounding to mitigate against hallucinations.

In this notebook, you will learn how to perform multimodal RAG where you will perform Q&A over a financial document filled with both text and images.

Gemini

Gemini is a family of generative AI models developed by Google DeepMind that is designed for multimodal use cases. The Gemini API gives you access to the Gemini models.

Comparing text-based and multimodal RAG

Multimodal RAG offers several advantages over text-based RAG:

1. **Enhanced knowledge access:** Multimodal RAG can access and process both textual and visual information, providing a richer and more comprehensive knowledge base for the LLM.
2. **Improved reasoning capabilities:** By incorporating visual cues, multimodal RAG can make better informed inferences across different types of data modalities.

This notebook shows you how to use RAG with Gemini API in Vertex AI, text embeddings, and multimodal embeddings, to build a document search engine.

Through hands-on examples, you will discover how to construct a multimedia-rich metadata repository of your document sources, enabling search, comparison, and reasoning across diverse information streams.

Objectives

This notebook provides a guide to building a document search engine using multimodal retrieval augmented generation (RAG), step by step:

1. Extract and store metadata of documents containing both text and images, and generate embeddings the documents
2. Search the metadata with text queries to find similar text or images
3. Search the metadata with image queries to find similar images
4. Using a text query as input, search for contextual answers using both text and images

Costs

This tutorial uses billable components of Google Cloud:

- Vertex AI

Learn about Vertex AI pricing and use the Pricing Calculator to generate a cost estimate based on your projected usage.

Getting Started

Install GenAI SDK for Python and other dependencies

In [1]:

```
%pip install --upgrade --quiet google-genai
%pip install --quiet pymupdf
```

```
Note: you may need to restart the kernel to use updated packages.
Note: you may need to restart the kernel to use updated packages.
```

Restart current runtime

To use the newly installed packages in this Jupyter runtime, you must restart the runtime. You can do this by running the cell below, which will restart the current kernel.

In [2]:

```
# Restart kernel after installs so that your environment can access the new
import IPython

app = IPython.Application.instance()
app.kernel.do_shutdown(True)
```

Out[2]:

```
{'status': 'ok', 'restart': True}
```

Define Google Cloud project information

In [1]:

```
from google import genai

PROJECT_ID = "qwiklabs-gcp-02-91a06e976252"  # @param {type:"string"}
LOCATION = "us-west1"  # @param {type:"string"}

client = genai.Client(vertexai=True, project=PROJECT_ID, location=LOCATION)
```

Import libraries

In [2]:

```
from IPython.display import Markdown, display
from rich.markdown import Markdown as rich_Markdown
from vertexai.generative_models import GenerationConfig, GenerativeModel, Im
```

Load the Gemini model

In [3]:

```
text_model = GenerativeModel("gemini-2.0-flash")
multimodal_model = text_model
multimodal_model_flash = text_model
```

Download custom Python utilities & required files

The cell below will download a helper functions needed for this notebook, to improve readability. It also downloads other required files. You can also view the code for the utils here: (`intro_multimodal_rag_utils.py`) directly on GitHub.

In [4]:

```
# download documents and images used in this notebook
!gsutil -m rsync -r gs://github-repo/rag/intro_multimodal_rag/intro_multimod
print("Download completed")
```

```
Building synchronization state...
Starting synchronization...
Copying gs://github-repo/rag/intro_multimodal_rag/intro_multimodal_rag_old_v
ersion/tac_table_revenue.png...
Copying gs://github-repo/rag/intro_multimodal_rag/intro_multimodal_rag_old_v
ersion/data/google-10k-sample-part2.pdf...
Copying gs://github-repo/rag/intro_multimodal_rag/intro_multimodal_rag_old_v
ersion/data/google-10k-sample-part1.pdf...
Copying gs://github-repo/rag/intro_multimodal_rag/intro_multimodal_rag_old_v
ersion/class_a_share.png...
Copying gs://github-repo/rag/intro_multimodal_rag/intro_multimodal_rag_old_v
ersion/intro_multimodal_rag_utils.py...
/ [5/5 files][882.3 KiB/882.3 KiB] 100% Done
Operation completed over 5 objects/882.3 KiB.
Download completed
```

Building metadata of documents containing text and images

The data

The source data that you will use in this notebook is a modified version of Google-10K which provides a comprehensive overview of the company's financial performance, business operations, management, and risk factors. As the original document is rather large, you will be using a modified version with only 14 pages, split into two parts - Part 1 and Part 2 instead. Although it's truncated, the sample document still contains text along with images such as tables, charts, and graphs.

Import helper functions to build metadata

Before building the multimodal RAG system, it's important to have metadata of all the text and images in the document. For references and citations purposes, the metadata should contain essential elements, including page number, file name, image counter, and so on. Hence, as a next step, you will generate embeddings from the metadata, which will is required to perform similarity search when querying the data.

In [5]:

```
from intro_multimodal_rag_utils import get_document_metadata
```

Extract and store metadata of text and images from a document

You just imported a function called `get_document_metadata()`. This function extracts text and image metadata from a document, and returns two dataframes, namely *text_metadata* and *image_metadata*, as outputs. If you want to find out more about how `get_document_metadata()` function is implemented using Gemini and the embedding models, you can take look at the source code directly.

The reason for extraction and storing both text metadata and image metadata is that just by using either of the two alone is not sufficient to come out with a relevent answer. For example, the relevant answers could be in visual form within a document, but text-based RAG won't be able to take into consideration of the visual images. You will also be exploring this example later in this notebook.

At the next step, you will use the function to extract and store metadata of text and images froma document. Please note that the following cell may take a few minutes to complete:

Note:

The current implementation works best:

- if your documents are a combination of text and images.
- if the tables in your documents are available as images.
- if the images in the document don't require too much context.

Additionally,

- If you want to run this on text-only documents, use normal RAG
- If your documents contain particular domain knowledge, pass that information in the prompt below.

⚠️ Do not send more than 50 pages in the logic below, its not degined to do that and you will get into quota issue. ⚠️

In [6]:

```
# Specify the PDF folder with multiple PDF

# pdf_folder_path = "/content/data/" # if running in Google Colab/Colab Ente
pdf_folder_path = "data/"  # if running in Vertex AI Workbench.

# Specify the image description prompt. Change it
image_description_prompt = """Explain what is going on in the image.
If it's a table, extract all elements of the table.
If it's a graph, explain the findings in the graph.
Do not include any numbers that are not mentioned in the image.
```

```
"""

# Extract text and image metadata from the PDF document
text_metadata_df, image_metadata_df = get_document_metadata(
    multimodal_model,  # we are passing Gemini 2.0 model
    pdf_folder_path,
    image_save_dir="images",
    image_description_prompt=image_description_prompt,
    embedding_size=1408,
    # add_sleep_after_page = True, # Uncomment this if you are running into
    # sleep_time_after_page = 5,
    # generation_config = # see next cell
    # safety_settings =  # see next cell
)

print("\n\n --- Completed processing. ---")
```

```
 Processing the file: --------------------------------- data/google-10k-samp
le-part1.pdf


Processing page: 1
Processing page: 2
Extracting image from page: 2, saved as: images/google-10k-sample-part1.pdf_
image_1_0_11.jpeg
Processing page: 3
Extracting image from page: 3, saved as: images/google-10k-sample-part1.pdf_
image_2_0_15.jpeg
Processing page: 4
Extracting image from page: 4, saved as: images/google-10k-sample-part1.pdf_
image_3_0_18.jpeg
Processing page: 5
Extracting image from page: 5, saved as: images/google-10k-sample-part1.pdf_
image_4_0_21.jpeg
Processing page: 6
Processing page: 7


 Processing the file: --------------------------------- data/google-10k-samp
le-part2.pdf


Processing page: 1
Extracting image from page: 1, saved as: images/google-10k-sample-part2.pdf_
image_0_0_6.jpeg
Extracting image from page: 1, saved as: images/google-10k-sample-part2.pdf_
image_0_1_8.jpeg
Processing page: 2
Extracting image from page: 2, saved as: images/google-10k-sample-part2.pdf_
image_1_0_13.jpeg
Processing page: 3
Processing page: 4
Extracting image from page: 4, saved as: images/google-10k-sample-part2.pdf_
image_3_0_19.jpeg
Processing page: 5
Extracting image from page: 5, saved as: images/google-10k-sample-part2.pdf_
image_4_0_22.jpeg
Extracting image from page: 5, saved as: images/google-10k-sample-part2.pdf_
image_4_1_23.jpeg
Processing page: 6
Extracting image from page: 6, saved as: images/google-10k-sample-part2.pdf_
image_5_0_26.jpeg
Processing page: 7


 --- Completed processing. ---
```

In [ ]:

```
# # Parameters for Gemini API call.
# # reference for parameters: https://cloud.google.com/vertex-ai/docs/genera

# generation_config=  GenerationConfig(temperature=0.2, max_output_tokens=20
# # Set the safety settings if Gemini is blocking your content or you are fa
# # ref for settings and thresholds: https://cloud.google.com/vertex-ai/docs

# safety_settings = {
#                   HarmCategory.HARM_CATEGORY_HARASSMENT: HarmBlockThreshol
#                   HarmCategory.HARM_CATEGORY_HATE_SPEECH: HarmBlockThresho
#                   HarmCategory.HARM_CATEGORY_SEXUALLY_EXPLICIT: HarmBlockT
#                   HarmCategory.HARM_CATEGORY_DANGEROUS_CONTENT: HarmBlockT
#                   }

# # You can also pass parameters and safety_setting to "get_gemini_response"
```

Inspect the processed text metadata

The following cell will produce a metadata table which describes the different parts of text metadata, including:

- **text**: the original text from the page
- **text_embedding_page**: the embedding of the original text from the page
- **chunk_text**: the original text divided into smaller chunks
- **chunk_number**: the index of each text chunk
- **text_embedding_chunk**: the embedding of each text chunk

In [7]:

```
text_metadata_df.head()
```

Out[7]:

	file_name	page_num	text	text_embedding_page
0	google-10k-sample-part1.pdf	1	source: https://abc.xyz/assets/investor/static...	[0.06604167073965073, 0.054615460336208344, -0...
1	google-10k-sample-part1.pdf	1	source: https://abc.xyz/assets/investor/static...	[0.06604167073965073, 0.054615460336208344, -0...
2	google-10k-sample-part1.pdf	2	Issuer Purchases of Equity Securities\nThe fol...	[0.035787057131528854, 0.008179700933396816, -...
3	google-10k-sample-part1.pdf	3	Stock Performance Graphs\nThe graph below matc...	[0.04338429123163223, 0.024151558056473732, -0...
4	google-10k-sample-part1.pdf	4	The graph below matches Alphabet Inc. Class A'...	[0.057393956929444527, 0.02924434281885624, -0....

Inspect the processed image metadata

The following cell will produce a metadata table which describes the different parts of image metadata, including:

- **img_desc**: Gemini-generated textual description of the image.
- **mm_embedding_from_text_desc_and_img**: Combined embedding of image and its description, capturing both visual and textual information.
- **mm_embedding_from_img_only**: Image embedding without description, for comparison with description-based analysis.
- **text_embedding_from_image_description**: Separate text embedding of the generated description, enabling textual analysis and comparison.

In [8]:

```
image_metadata_df.head()
```

Out[8]:

	file_name	page_num	img_num	img_path	img_desc	mm_embeddi
0	google-10k-sample-part1.pdf	2	1	images/google-10k-sample-part1.pdf_image_1_0_1...	The image is a table showing share repurchase ...	
1	google-10k-sample-part1.pdf	3	1	images/google-10k-sample-part1.pdf_image_2_0_1...	The image is a graph comparing the cumulative ...	[0.00688816 -0
2	google-10k-sample-part1.pdf	4	1	images/google-10k-sample-part1.pdf_image_3_0_1...	The image is a graph comparing the cumulative ...	[0.00658530 -
3	google-10k-sample-part1.pdf	5	1	images/google-10k-sample-part1.pdf_image_4_0_2...	The image is a table showing financial data fo...	[0.01665539 -
4	google-10k-sample-part2.pdf	1	1	images/google-10k-sample-part2.pdf_image_0_0_6...	The image is a table showing data for the year...	[0.0357287 (

Import the helper functions to implement RAG

You will be importing the following functions which will be used in the remainder of this notebook to implement RAG:

- **get_similar_text_from_query():** Given a text query, finds text from the document which are relevant, using cosine similarity algorithm. It uses text embeddings from the metadata to compute and the results can be filtered by top score, page/chunk number, or embedding size.
- **print_text_to_text_citation():** Prints the source (citation) and details of the retrieved text from the `get_similar_text_from_query()` function.
- **get_similar_image_from_query():** Given an image path or an image, finds images from the document which are relevant. It uses image embeddings from the metadata.
- **print_text_to_image_citation():** Prints the source (citation) and the details of retrieved images from the `get_similar_image_from_query()` function.
- **get_gemini_response():** Interacts with a Gemini model to answer questions based on a combination of text and image inputs.
- **display_images():** Displays a series of images provided as paths or PIL Image objects.

In [9]:

```
from intro_multimodal_rag_utils import (
    display_images,
    get_gemini_response,
    get_similar_image_from_query,
    get_similar_text_from_query,
    print_text_to_image_citation,
    print_text_to_text_citation,
)
```

Before implementing a multimodal RAG, let's take a step back and explore what you can achieve with just text or image embeddings alone. It will help to set the foundation for implementing a multimodal RAG, which you will be doing in the later part of the notebook. You can also use these essential elements together to build applications for multimodal use cases for extracting meaningful information from the document.

Text Search

Let's start the search with a simple question and see if the simple text search using text embeddings can answer it. The expected answer is to show the value of basic and diluted net income per share of Google for different share types.

In [10]:

```
query = "I need details for basic and diluted net income per share of Class
```

Search similar text with text query

In [11]:

```
# Matching user text query with "chunk_embedding" to find relevant chunks.
matching_results_text = get_similar_text_from_query(
    query,
    text_metadata_df,
    column_name="text_embedding_chunk",
    top_n=3,
    chunk_text=True,
)

# Print the matched text citations
print_text_to_text_citation(matching_results_text, print_top=False, chunk_te
Citation 1: Matched text:

score:  0.76
file_name:  google-10k-sample-part2.pdf
page_number:  4
chunk_number:  1
chunk_text:  liquidation and dividend rights are identical, the undistribute
d earnings are
allocated on a proportionate basis.
In the years ended December 31, 2019, 2020 and 2021, the net income per
share amounts are the same for Class A, Class B, and Class C stock because
the holders of each class are entitled to equal per share dividends or distr
ibutions
in liquidation in accordance with the Amended and Restated Certificate of
Incorporation of Alphabet Inc.
The following tables set forth the computation of basic and diluted net inco
me per
share of Class A, Class B, and Class C stock (in millions, except share amou
nts
which are reflected in thousands and per share amounts):

Citation 2: Matched text:

score:  0.7
file_name:  google-10k-sample-part2.pdf
page_number:  3
chunk_number:  1
chunk_text:  increases in content acquisition costs primarily for YouTube, d
ata center and
other operations costs, and hardware costs. The increase in data center and
Table of Contents Alphabet Inc. 36 other operations costs was partially offs
et by
a reduction in depreciation expense due to the change in the estimated usefu
l life
of our servers and certain network equipment beginning in the first quarter
of
2021.
Net Income Per Share
We compute net income per share of Class A, Class B, and Class C stock using
the two-class method. Basic net income per share is computed using the
weighted-average number of shares outstanding during the period. Diluted net
income per share is computed using the weighted-average number of shares and
the effect of potentially dilutive securities outstanding during the period.
Potentially dilutive securities consist of restricted stock units and other
contingently issuable shares. The dilutive effect of outstanding restricted
stock
units and other contingently issuable
Citation 3: Matched text:

score:  0.67
file_name:  google-10k-sample-part2.pdf
page_number:  3
chunk_number:  2
chunk_text:  e shares. The dilutive effect of outstanding restricted stock
units and other contingently issuable shares is reflected in diluted earning
s per
```

```
share by application of the treasury stock method. The computation of the di
luted
net income per share of Class A stock assumes the conversion of Class B stoc
k,
while the diluted net income per share of Class B stock does not assume the
conversion of those shares.
The rights, including the liquidation and dividend rights, of the holders of
our
Class A, Class B, and Class C stock are identical, except with respect to vo
ting.
Furthermore, there are a number of safeguards built into our certificate of
incorporation, as well as Delaware law, which preclude our Board of Director
s
from declaring or paying unequal per share dividends on our Class A, Class
B,
and Class C stock. Specifically, Delaware law provides that amendments to ou
r
certificate of incorporation which would have the effect of adversely alteri
ng the
rights, powers, or preferences of a
```

You can see that the first high score match does have what we are looking for, but upon closer inspection, it mentions that the information is available in the "following" table. The table data is available as an image rather than as text, and hence, the chances are you will miss the information unless you can find a way to process images and their data.

However, Let's feed the relevant text chunk across the data into the Gemini model and see if it can get your desired answer by considering all the chunks across the document. This is like basic text-based RAG implementation.

In [12]:

```
print("\n **** Result: ***** \n")

# All relevant text chunk found across documents based on user query
context = "\n".join(
    [value["chunk_text"] for key, value in matching_results_text.items()]
)

instruction = f"""Answer the question with the given context.
If the information is not available in the context, just return "not availab
Question: {query}
Context: {context}
Answer:
"""

# Prepare the model input
model_input = instruction

# Generate Gemini response with streaming output
get_gemini_response(
    text_model,  # we are passing Gemini
    model_input=model_input,
    stream=True,
    generation_config=GenerationConfig(temperature=0.2),
)
```

```
**** Result: *****
```

Out[12]:

```
'The net income per share amounts are the same for Class A, Class B, and Cl
ass C stock because the holders of each class are entitled to equal per sha
re dividends or distributions in liquidation. The computation of basic and
diluted net income per share of Class A, Class B, and Class C stock is avai
lable in the tables.\n'
```

You can see that it returned:

"The provided context does not include the details for basic and diluted net income per share of Class A, Class B, and Class C share for google. "

This is expected as discussed previously. No other text chunk (total 3) had the information you sought. This is because the information is only available in the images rather than in the text part of the document. Next, let's see if you can solve this problem by leveraging Gemini and Multimodal Embeddings.

Note: We handcrafted examples in our document to simulate real-world cases where information is often embedded in charts, table, graphs, and other image-based elements and unavailable as plain text.

Search similar images with text query

Since plain text search didn't provide the desired answer and the information may be visually represented in a table or another image format, you will use multimodal capability of Gemini model for the similar task. The goal here also is to find an image similar to the text query. You may also print the citations to verify.

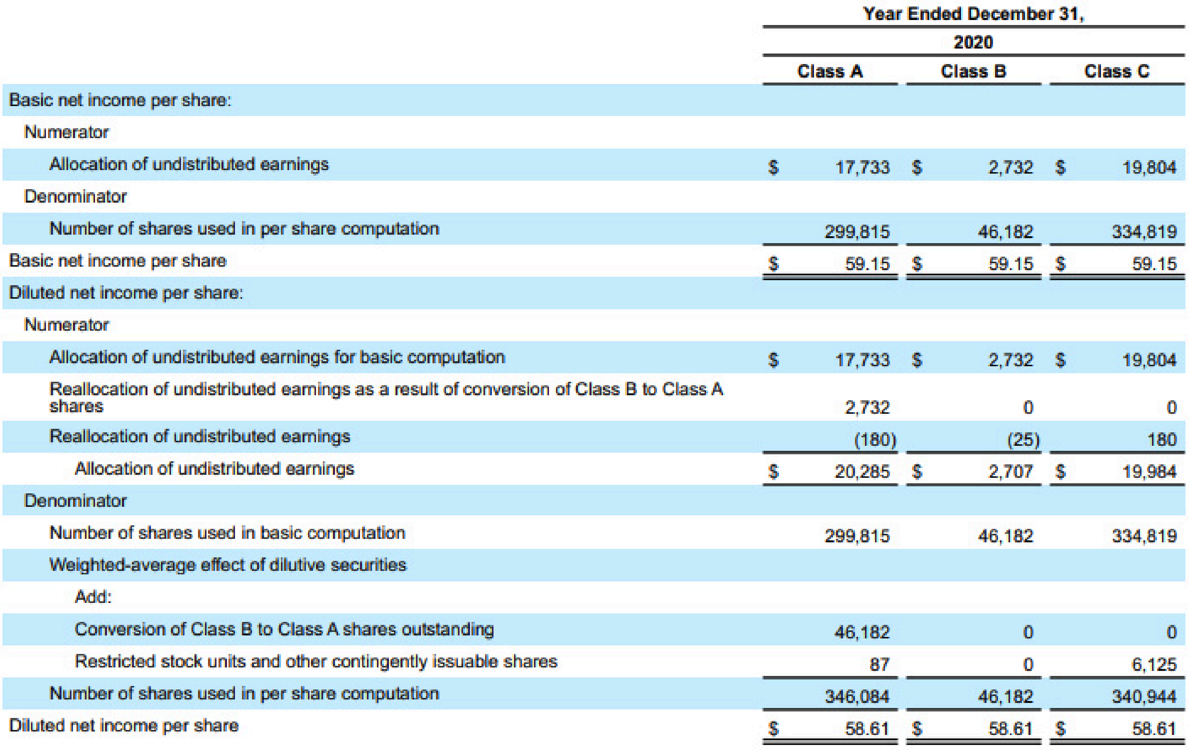
In [15]:

```
query = "I need details for basic and diluted net income per share of Class
```

In [16]:

```
matching_results_image = get_similar_image_from_query(
    text_metadata_df,
    image_metadata_df,
    query=query,
    column_name="text_embedding_from_image_description",  # Use image descri
    image_emb=False,  # Use text embedding instead of image embedding
    top_n=3,
    embedding_size=1408,
)

# Markdown(print_text_to_image_citation(matching_results_image, print_top=Tr
print("\n **** Result: ***** \n")

# Display the top matching image
display(matching_results_image[0]["image_object"])
```

```
**** Result: *****
```



	Year Ended December 31,		
	2020		
	Class A	Class B	Class C
Basic net income per share:			
Numerator			
Allocation of undistributed earnings	$ 17,733	$ 2,732	$ 19,804
Denominator			
Number of shares used in per share computation	299,815	46,182	334,819
Basic net income per share	$ 59.15	$ 59.15	$ 59.15
Diluted net income per share:			
Numerator			
Allocation of undistributed earnings for basic computation	$ 17,733	$ 2,732	$ 19,804
Reallocation of undistributed earnings as a result of conversion of Class B to Class A shares	2,732	0	0
Reallocation of undistributed earnings	(180)	(25)	180
Allocation of undistributed earnings	$ 20,285	$ 2,707	$ 19,984
Denominator			
Number of shares used in basic computation	299,815	46,182	334,819
Weighted-average effect of dilutive securities			
Add:			
Conversion of Class B to Class A shares outstanding	46,182	0	0
Restricted stock units and other contingently issuable shares	87	0	6,125
Number of shares used in per share computation	346,084	46,182	340,944
Diluted net income per share	$ 58.61	$ 58.61	$ 58.61

Bingo! It found exactly what you were looking for. You wanted the details on Google's Class A, B, and C shares' basic and diluted net income, and guess what? This image fits the bill perfectly thanks to its descriptive metadata using Gemini.

You can also send the image and its description to Gemini and get the answer as JSON:

In [17]:

```
print("\n **** Result: ***** \n")

# All relevant text chunk found across documents based on user query
context = f"""Image: {matching_results_image[0]['image_object']}
Description: {matching_results_image[0]['image_description']}
"""

instruction = f"""Answer the question in JSON format with the given context
Question: {query}
Context: {context}
Answer:
"""

# Prepare the model input
model_input = instruction

# Generate Gemini response with streaming output
Markdown(
    get_gemini_response(
        multimodal_model_flash,  # we are passing Gemini 2.0 Flash
        model_input=model_input,
        stream=True,
        generation_config=GenerationConfig(temperature=1),
    )
)
```

```
**** Result: *****
```

Out[17]:

```
{
"Basic net income per share": {
  "Numerator": "Allocation of undistributed earnings",
  "Denominator": "Number of shares used in per share computation"
},
"Diluted net income per share": {
  "Numerator": [
    "Allocation of undistributed earnings for basic computation",
    "Reallocation of undistributed earnings as a result of conversion of Class B to Class
A shares",
    "Reallocation of undistributed earnings",
    "Allocation of undistributed earnings"
  ],
  "Denominator": [
    "Number of shares used in basic computation",
    "Weighted-average effect of dilutive securities",
    "Conversion of Class B to Class A shares outstanding",
    "Restricted stock units and other contingently issuable shares"
  ]
}
}
```

In [ ]:

```
## you can check the citations to probe further.
## check the "image description:" which is a description extracted through G
Markdown(print_text_to_image_citation(matching_results_image, print_top=True
```

Image Search

Search similar image with image query

Imagine searching for images, but instead of typing words, you use an actual image as the clue. You have a table with numbers about the cost of revenue for two years, and you want to find other images that look like it, from the same document or across multiple documents.

Think of it like searching with a mini-map instead of a written address. It's a different way to ask, "Show me more stuff like this". So, instead of typing "cost of revenue 2020 2021 table", you show a picture of that table and say, "Find me more like this"

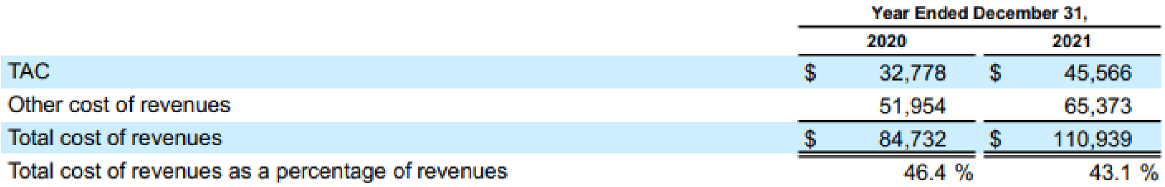
For demonstration purposes, we will only be finding similar images that show the cost of revenue or similar values in a single document below. However, you can scale this design pattern to match (find relevant images) across multiple documents.

In [18]:

```
# You can find a similar image as per the images you have in the metadata.
# In this case, you have a table (picked from the same document source) and
image_query_path = "tac_table_revenue.png"

# Print a message indicating the input image
print("***Input image from user:***")

# Display the input image
Image.load_from_file(image_query_path)
```

```
***Input image from user:***
```

Out[18]:



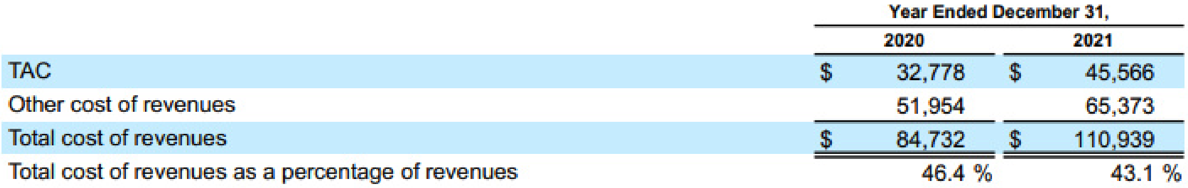
	Year Ended December 31,	
	2020	2021
TAC	$ 32,778	$ 45,566
Other cost of revenues	51,954	65,373
Total cost of revenues	$ 84,732	$ 110,939
Total cost of revenues as a percentage of revenues	46.4 %	43.1 %

You expect to find tables (as images) that are similar in terms of "Other/Total cost of revenues."

In [19]:

```
# Search for Similar Images Based on Input Image and Image Embedding

matching_results_image = get_similar_image_from_query(
    text_metadata_df,
    image_metadata_df,
    query=query,  # Use query text for additional filtering (optional)
    column_name="mm_embedding_from_img_only",  # Use image embedding for sim
    image_emb=True,
    image_query_path=image_query_path,  # Use input image for similarity cal
    top_n=3,  # Retrieve top 3 matching images
    embedding_size=1408,  # Use embedding size of 1408
)

print("\n **** Result: ***** \n")

# Display the Top Matching Image
display(
    matching_results_image[0]["image_object"]
)  # Display the top matching image object (Pillow Image)
```

```
 **** Result: *****
```



	Year Ended December 31,	
	2020	2021
TAC	$ 32,778	$ 45,566
Other cost of revenues	51,954	65,373
Total cost of revenues	$ 84,732	$ 110,939
Total cost of revenues as a percentage of revenues	46.4 %	43.1 %

It did find a similar-looking image (table), which gives more detail about different revenue, expenses, income, and a few more details based on the given image. More importantly, both tables show numbers related to the "cost of revenue."

You can also print the citation to see what it has matched.

In [20]:

```python
# Display citation details for the top matching image
print_text_to_image_citation(
    matching_results_image, print_top=True
)  # Print citation details for the top matching image
```

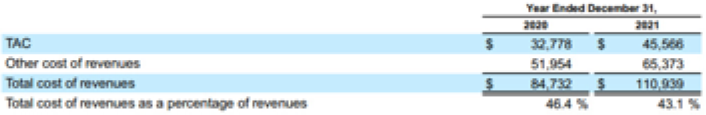
```
Citation 1: Matched image path, page number and page text:

score:  0.99
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_1_0_13.jpeg
page number:  2
page text:  APAC revenue growth from 2020 to 2021 was favorably affected by
foreign
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Australian dollar, partially offset by the U.S. dollar strengthening rel
ative to
the Japanese yen.
Other Americas growth change from 2020 to 2021 was favorably affected by
changes in foreign currency exchange rates, primarily due to the U.S. dollar
weakening relative to the Canadian dollar, partially offset by the U.S. doll
ar
strengthening relative to the Argentine peso and the Brazilian real.
Costs and Expenses
Cost of Revenues
The following tables present cost of revenues, including TAC (in millions, e
xcept
percentages):
Cost of revenues increased $26.2 billion from 2020 to 2021. The increase was
due to an increase in other cost of revenues and TAC of $13.4 billion and $1
2.8
billion, respectively.
The increase in TAC from 2020 to 2021 was due to an increase in TAC paid to
distribution partners and to Google Network partners, primarily driven by gr
owth
in revenues subject to TAC. The TAC rate decreased from 22.3% to 21.8% from
2020 to 2021 primarily due to a revenue mix shift from Google Network
properties to Google Search & other properties.
The TAC rate on Google Search & other properties revenues and the TAC rate
on Google Network revenues were both substantially consistent from 2020 to
2021. The increase in other cost of revenues from 2020 to 2021 was driven by

image description:  The image shows a table with financial data for the year
s ended December 31, 2020 and 2021. The table includes the following element
s:

| Item                                         | 2020    | 2021    |
| -------------------------------------------- | ------- | ------- |
| TAC                                          | 32,778  | 45,566  |
| Other cost of revenues                       | 51,954  | 65,373  |
| Total cost of revenues                       | 84,732  | 110,939 |
| Total cost of revenues as a percentage of revenues | 46.4%   | 43.1%   |

The table shows an increase in all cost of revenues categories from 2020 to
2021. The total cost of revenues as a percentage of revenues decreased from
46.4% to 43.1%.
```

In [21]:

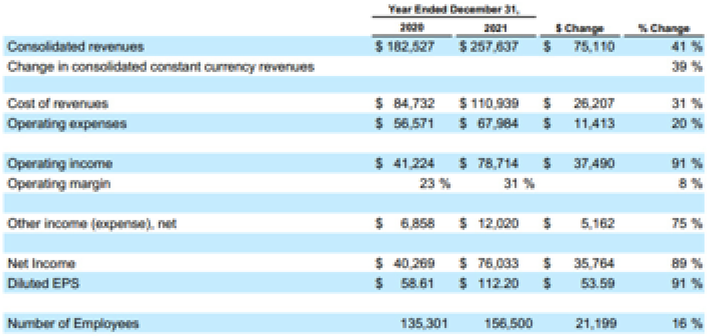
```
# Check Other Matched Images (Optional)
# You can access the other two matched images using:
```

```
print("---------------Matched Images------------------\n")
display_images(
    [
        matching_results_image[0]["img_path"],
        matching_results_image[1]["img_path"],
    ],
    resize_ratio=0.5,
)
```

```
---------------Matched Images------------------
```



	Year Ended December 31,	
	2020	2021
TAC	$ 32,778	$ 45,566
Other cost of revenues	51,954	65,373
Total cost of revenues	$ 84,732	$ 110,939
Total cost of revenues as a percentage of revenues	46.4 %	43.1 %



	Year Ended December 31,			
	2020	2021	$ Change	% Change
Consolidated revenues	$ 182,527	$ 257,637	$ 75,110	41 %
Change in consolidated constant currency revenues				39 %
Cost of revenues	$ 84,732	$ 110,939	$ 26,207	31 %
Operating expenses	$ 56,571	$ 67,984	$ 11,413	20 %
Operating income	$ 41,224	$ 78,714	$ 37,490	91 %
Operating margin	23 %	31 %		8 %
Other income (expense), net	$ 6,858	$ 12,020	$ 5,162	75 %
Net Income	$ 40,269	$ 76,033	$ 35,764	89 %
Diluted EPS	$ 58.61	$ 112.20	$ 53.59	91 %
Number of Employees	135,301	156,500	21,199	16 %

The ability to identify similar text and images based on user input, using Gemini and embeddings, forms a crucial foundation for development of multimodal RAG systems, which you explore in the next section.

Comparative reasoning

Next, let's apply what you have done so far to doing comparative reasoning.

For this example:

Step 1: You will search all the images for a specific query

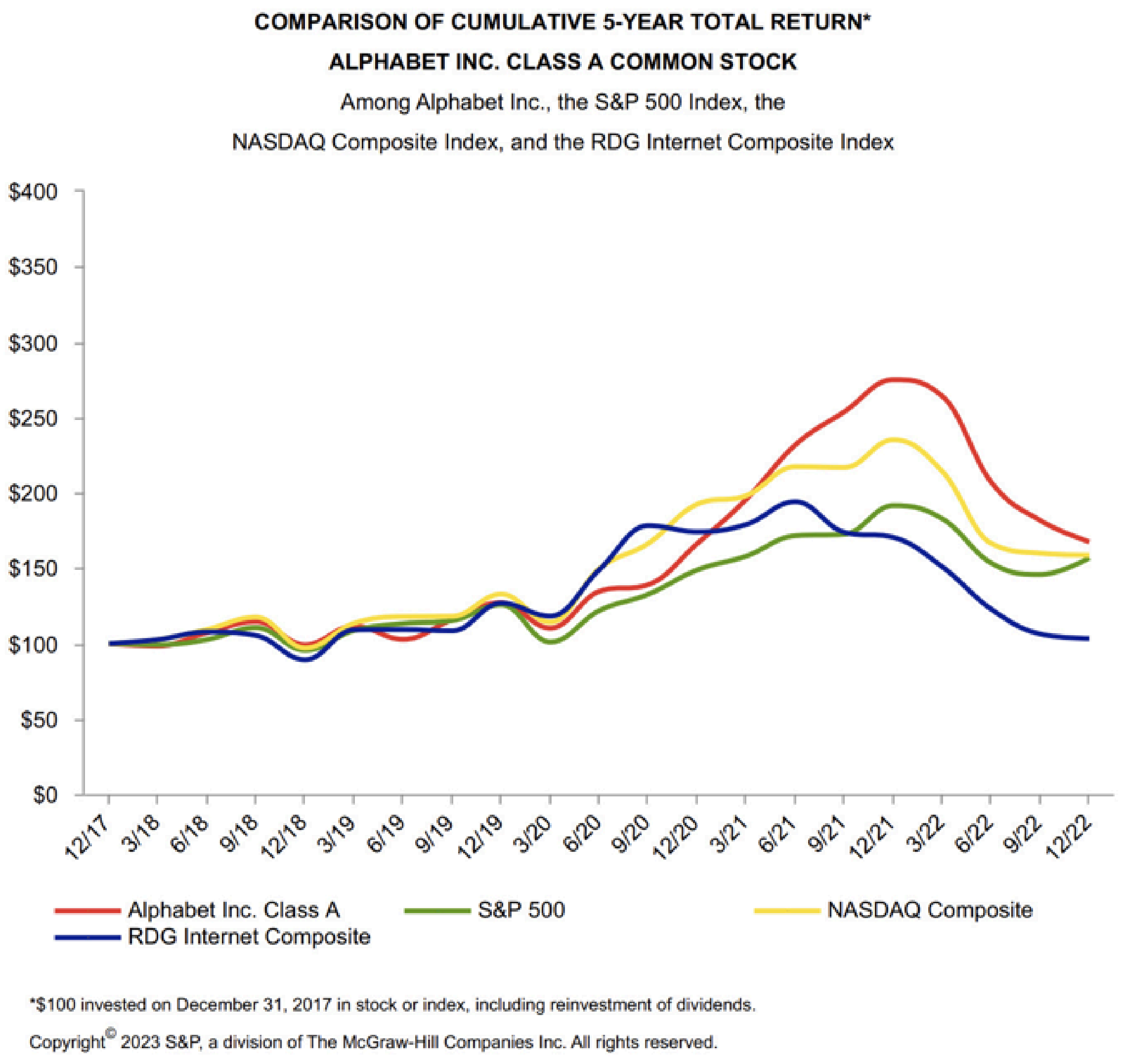
Step 2: Send those images to Gemini to ask multiple questions, where it has to compare and provide you with answers.

In [22]:

```
matching_results_image_query_1 = get_similar_image_from_query(
    text_metadata_df,
    image_metadata_df,
    query="Show me all the graphs that shows Google Class A cumulative 5-yea
    column_name="text_embedding_from_image_description",  # Use image descri
    image_emb=False,  # Use text embedding instead of image embedding
    top_n=3,
    embedding_size=1408,
)
```

```
In [23]: # Check Matched Images
         # You can access the other two matched images using:

         print("---------------Matched Images------------------\n")
         display_images(
             [
                 matching_results_image_query_1[0]["img_path"],
                 matching_results_image_query_1[1]["img_path"],
             ],
             resize_ratio=0.5,
         )
```

```
---------------Matched Images------------------
```



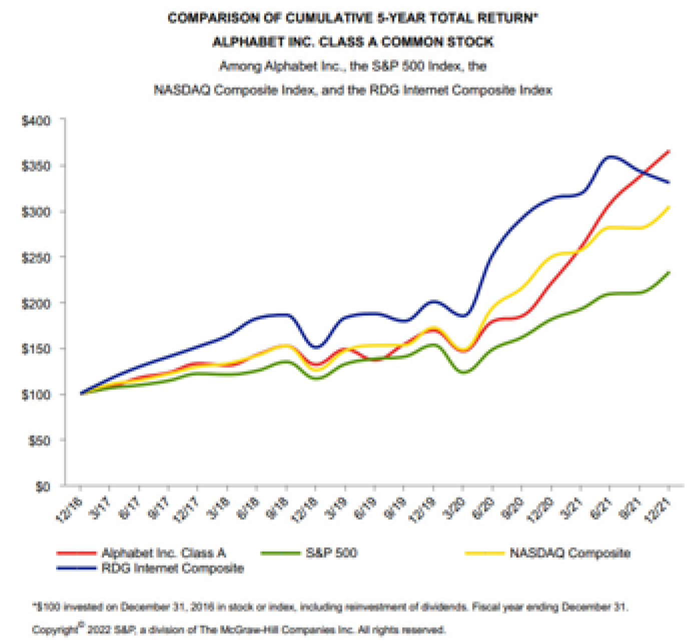
COMPARISON OF CUMULATIVE 5-YEAR TOTAL RETURN*

ALPHABET INC. CLASS A COMMON STOCK

Among Alphabet Inc., the S&P 500 Index, the NASDAQ Composite Index, and the RDG Internet Composite Index

*$100 invested on December 31, 2017 in stock or index, including reinvestment of dividends.

Copyright© 2023 S&P, a division of The McGraw-Hill Companies Inc. All rights reserved.


COMPARISON OF CUMULATIVE 5-YEAR TOTAL RETURN*
ALPHABET INC. CLASS A COMMON STOCK
Among Alphabet Inc., the S&P 500 Index, the
NASDAQ Composite Index, and the RDG Internet Composite Index

*$100 invested on December 31, 2016 in stock or index, including reinvestment of dividends. Fiscal year ending December 31.

In [24]:

```
prompt = f""" Instructions: Compare the images and the Gemini extracted text
Make sure to think thoroughly before answering the question and put the nece

Context:
Image_1: {matching_results_image_query_1[0]["image_object"]}
gemini_extracted_text_1: {matching_results_image_query_1[0]['image_descripti
Image_2: {matching_results_image_query_1[1]["image_object"]}
gemini_extracted_text_2: {matching_results_image_query_1[2]['image_descripti

Question:
 - Key findings of Class A share?
 - What are the critical differences between the graphs for Class A Share?
 - What are the key findings of Class A shares concerning the S&P 500?
 - Which index best matches Class A share performance closely where Google i
 - Identify key chart patterns in both graphs.
 - Which index best matches Class A share performance closely where Google i
"""

# Generate Gemini response with streaming output
rich_Markdown(
    get_gemini_response(
        multimodal_model,  # we are passing Gemini 2.0
        model_input=[prompt],
        stream=True,
        generation_config=GenerationConfig(temperature=1),
    )
)
```

Out[24]: Here's an analysis of the provided information, addressing each quest

Key Findings of Class A Share (Based on Image 1)

- **Outperformance:** The red line (Alphabet Inc. Class A) generally tre NASDAQ Composite, RDG Internet Composite) over the 5-year period. outperformed these indices in terms of cumulative total return.
- **Volatility:** While outperforming, the Alphabet Class A line likely potentially steeper declines compared to the broader indices, indi the exact chart, this is an inference based on typical stock perfo

Critical Differences Between the Graphs for Class A Share

- Image 1 focuses on the *performance* of Alphabet Class A shares comp cumulative total return.
- Image 2 provides data on the *movement of unvested Alphabet RSUs*. I weighted-average grant-date fair value as they are granted, vestec
- **Essentially, Image 1 shows how the stock price performed, while In over a period.**

Key Findings of Class A Shares Concerning the S&P 500

- **Outperformance (Likely):** The description states that the red line green line (S&P 500). This means Alphabet Class A shares provided 500 over the 5-year period.
- **Potentially Higher Volatility:** As mentioned earlier, Alphabet's st price swings (both up and down) than the broader S&P 500 index.

Which Index Best Matches Class A Share Performance Closely Where Goog reasoning.

- **RDG Internet Composite:** The RDG Internet Composite is likely to be basket of internet-based companies and the Nasdaq Composite, which of which Google is part. Therefore, the RDG Internet Composite wou performance.

Identify key chart patterns in both graphs.

- **Image 1:**
  - **Uptrend:** A general upward trend suggests positive long-term gro
  - **Volatility:** Look for steeper inclines (rapid growth) and declir
- **Image 2:** No specific chart patterns can be identified in a tabular

Which index best matches Class A share performance closely where Goog reasoning.

- **RDG Internet Composite:** The RDG Internet Composite is likely to be basket of internet-based companies and the Nasdaq Composite, which of which Google is part. Therefore, the RDG Internet Composite wou performance.

⚠️ **Disclaimer: This is not a real investment advise and should not be taken seriously!!** ⚠️

Multimodal retrieval augmented generation (RAG)

Let's bring everything together to implement multimodal RAG. You will use all the elements that you've explored in previous sections to implement the multimodal RAG. These are the steps:

- **Step 1:** The user gives a query in text format where the expected information is available in the document and is embedded in images and text.
- **Step 2:** Find all text chunks from the pages in the documents using a method similar to the one you explored in `Text Search`.
- **Step 3:** Find all similar images from the pages based on the user query matched with `image_description` using a method identical to the one you explored in `Image Search`.
- **Step 4:** Combine all similar text and images found in steps 2 and 3 as `context_text` and `context_images`.
- **Step 5:** With the help of Gemini, we can pass the user query with text and image context found in steps 2 & 3. You can also add a specific instruction the model should remember while answering the user query.
- **Step 6:** Gemini produces the answer, and you can print the citations to check all relevant text and images used to address the query.

Step 1: User query

In [25]:

```
# this time we are not passing any images, but just a simple text query.

query = """Questions:
 - What are the critical difference between various graphs for Class A Share
 - Which index best matches Class A share performance closely where Google i
 - Identify key chart patterns for Google Class A shares.
 - What is cost of revenues, operating expenses and net income for 2020. Do
 - What was the effect of Covid in the 2020 financial year?
 - What are the total revenues for APAC and USA for 2021?
 - What is deferred income taxes?
 - How do you compute net income per share?
 - What drove percentage change in the consolidated revenue and cost of reve
 - What is the cause of 41% increase in revenue from 2020 to 2021 and how mu
 """
```

Step 2: Get all relevant text chunks

In [26]:

```
# Retrieve relevant chunks of text based on the query
matching_results_chunks_data = get_similar_text_from_query(
    query,
    text_metadata_df,
    column_name="text_embedding_chunk",
    top_n=10,
    chunk_text=True,
)
```

Step 3: Get all relevant images

In [27]:

```
# Get all relevant images based on user query
matching_results_image_fromdescription_data = get_similar_image_from_query(
    text_metadata_df,
    image_metadata_df,
    query=query,
    column_name="text_embedding_from_image_description",
    image_emb=False,
    top_n=10,
    embedding_size=1408,
)
```

Step 4: Create context_text and context_images

In [28]:

```
# combine all the selected relevant text chunks
context_text = []
for key, value in matching_results_chunks_data.items():
    context_text.append(value["chunk_text"])
final_context_text = "\n".join(context_text)

# combine all the relevant images and their description generated by Gemini
context_images = []
for key, value in matching_results_image_fromdescription_data.items():
    context_images.extend(
        ["Image: ", value["image_object"], "Caption: ", value["image_descrip
    )
```

Step 5: Pass context to Gemini

In [29]:

```
prompt = f""" Instructions: Compare the images and the text provided as Cont
Make sure to think thoroughly before answering the question and put the nece
If unsure, respond, "Not enough context to answer".

Context:
 - Text Context:
 {final_context_text}
 - Image Context:
 {context_images}

{query}

Answer:
"""

# Generate Gemini response with streaming output
rich_Markdown(
    get_gemini_response(
        multimodal_model,
        model_input=[prompt],
        stream=True,
        generation_config=GenerationConfig(temperature=1),
    )
)
```

Out[29]: Here's an analysis of the questions based on the provided context:

- **What are the critical difference between various graphs for Class**
  - The graphs compare the cumulative 5-year total stockholder retu against various indices (S&P 500, NASDAQ Composite, RDG Interne the period covered by each graph: one graph covers Dec 31, 2016 31, 2017 to Dec 31, 2022. The more recent graph shows more rece
- **Which index best matches Class A share performance closely where G reasoning.**
  - Based on the provided graphs, the RDG Internet Composite Index performance more closely than the S&P 500. The S&P 500 is a bro weighted in technology stocks and the RDG Internet Composite In thus is the most suitable index to make the comparison
  - To determine if Alphabet (Google) is part of these indices it w the index provider websites
  - This may still be flawed if Alphabet dominates the RDG Internet
- **Identify key chart patterns for Google Class A shares.**
  - The provided context focuses on numerical comparisons and summa (e.g., head and shoulders, double top) requires a detailed visu which is beyond the scope of the provided information. We can r shares compared to benchmark indices based on visual assessment
- **What is cost of revenues, operating expenses and net income for 20**
  - **Cost of Revenues (2020):** $84,732 million
    - Percentage change not applicable as it is data from a single
  - **Operating Expenses (2020):** $56,571 million
    - Percentage change not applicable as it is data from a single
  - **Net Income (2020):** $40,269 million
    - Percentage change not applicable as it is data from a single
- **What was the effect of Covid in the 2020 financial year?**
  - The text indicates that in March 2020, despite an increase in u declined compared to the prior year. This was attributed to a s commercial topics and reduced spending by advertisers. For the revenues declined due to the continued effects of COVID-19.
- **What are the total revenues for APAC and USA for 2021?**
  - **APAC Revenues (2021):** $46,123 million
  - **USA Revenues (2021):** $117,854 million
- **What is deferred income taxes?**
  - The provided table shows deferred tax assets and liabilities, r temporary differences between the book (accounting) and tax bas not further clarify, and understanding deferred income taxes in
- **How do you compute net income per share?**
  - The text describes that Alphabet Inc. uses the two-class methoc Class B, and Class C stock.
  - **Basic net income per share:** Computed using the weighted-average period.
  - **Diluted net income per share:** Computed using the weighted-avera potentially dilutive securities outstanding during the period (issuable shares).
  - Undistributed earnings are allocated on a proportionate basis b identical.
- **What drove percentage change in the consolidated revenue and cost effect of Covid?**
  - **Consolidated Revenue:** The 41% increase in revenues was primaril The adverse effect of COVID-19 on 2020 advertising revenues als
  - **Cost of Revenue:** The 31% increase was primarily driven by incre content acquisition costs.
  - **COVID-19 Effect:** The adverse impact of COVID-19 in 2020 made th contributing to the percentage increase.
- **What is the cause of 41% increase in revenue from 2020 to 2021 and**
  - The primary drivers were:

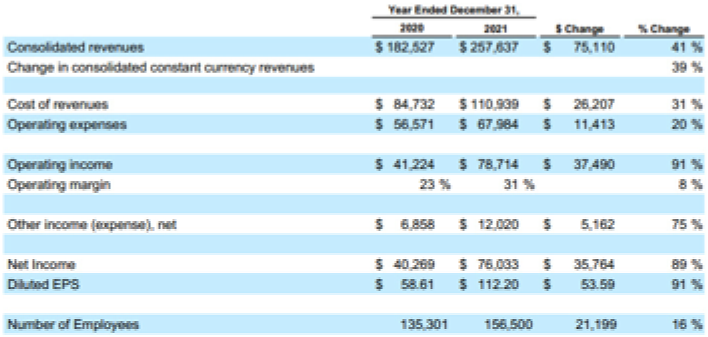
    - Growth in Google Services and Google Cloud.
    - Comparison against lower 2020 revenues due to the adverse ef
- The dollar change in revenues from 2020 to 2021 was $75,110 mil

Step 6: Print citations and references

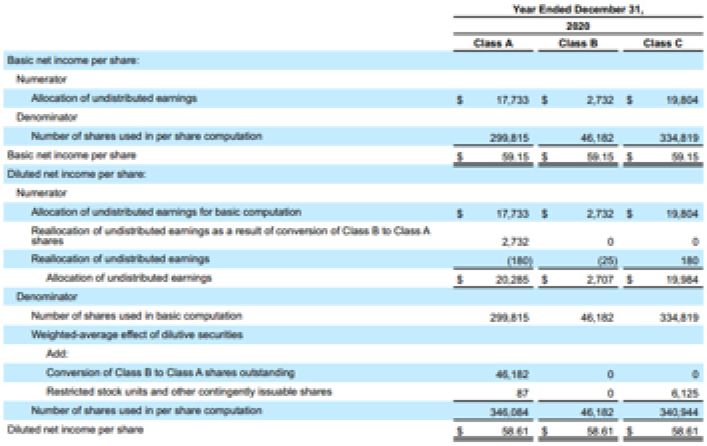
In [30]:

```
print("---------------Matched Images------------------\n")
display_images(
    [
        matching_results_image_fromdescription_data[0]["img_path"],
        matching_results_image_fromdescription_data[1]["img_path"],
        matching_results_image_fromdescription_data[2]["img_path"],
        matching_results_image_fromdescription_data[3]["img_path"],
    ],
    resize_ratio=0.5,
)
```

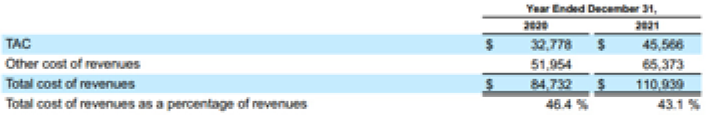
```
---------------Matched Images------------------
```



	Year Ended December 31,			
	2020	2021	$ Change	% Change
Consolidated revenues	$ 182,527	$ 257,637	$ 75,110	41 %
Change in consolidated constant currency revenues				39 %
Cost of revenues	$ 84,732	$ 110,939	$ 26,207	31 %
Operating expenses	$ 56,571	$ 67,984	$ 11,413	20 %
Operating income	$ 41,224	$ 78,714	$ 37,490	91 %
Operating margin	23 %	31 %		8 %
Other income (expense), net	$ 6,858	$ 12,020	$ 5,162	75 %
Net Income	$ 40,269	$ 76,033	$ 35,764	89 %
Diluted EPS	$ 58.61	$ 112.20	$ 53.59	91 %
Number of Employees	135,301	156,500	21,199	16 %



	Year Ended December 31,		
	2020		
	Class A	Class B	Class C
Basic net income per share:			
Numerator			
Allocation of undistributed earnings	$ 17,733	$ 2,732	$ 19,804
Denominator			
Number of shares used in per share computation	299,815	46,182	334,819
Basic net income per share	$ 59.15	$ 59.15	$ 59.15
Diluted net income per share:			
Numerator			
Allocation of undistributed earnings for basic computation	$ 17,733	$ 2,732	$ 19,804
Reallocation of undistributed earnings as a result of conversion of Class B to Class A shares	2,732	0	0
Reallocation of undistributed earnings	(180)	(25)	180
Allocation of undistributed earnings	$ 20,285	$ 2,707	$ 19,984
Denominator			
Number of shares used in basic computation	299,815	46,182	334,819
Weighted-average effect of dilutive securities			
Add:			
Conversion of Class B to Class A shares outstanding	46,182	0	0
Restricted stock units and other contingently issuable shares	87	0	6,125
Number of shares used in per share computation	346,084	46,182	340,944
Diluted net income per share	$ 58.61	$ 58.61	$ 58.61



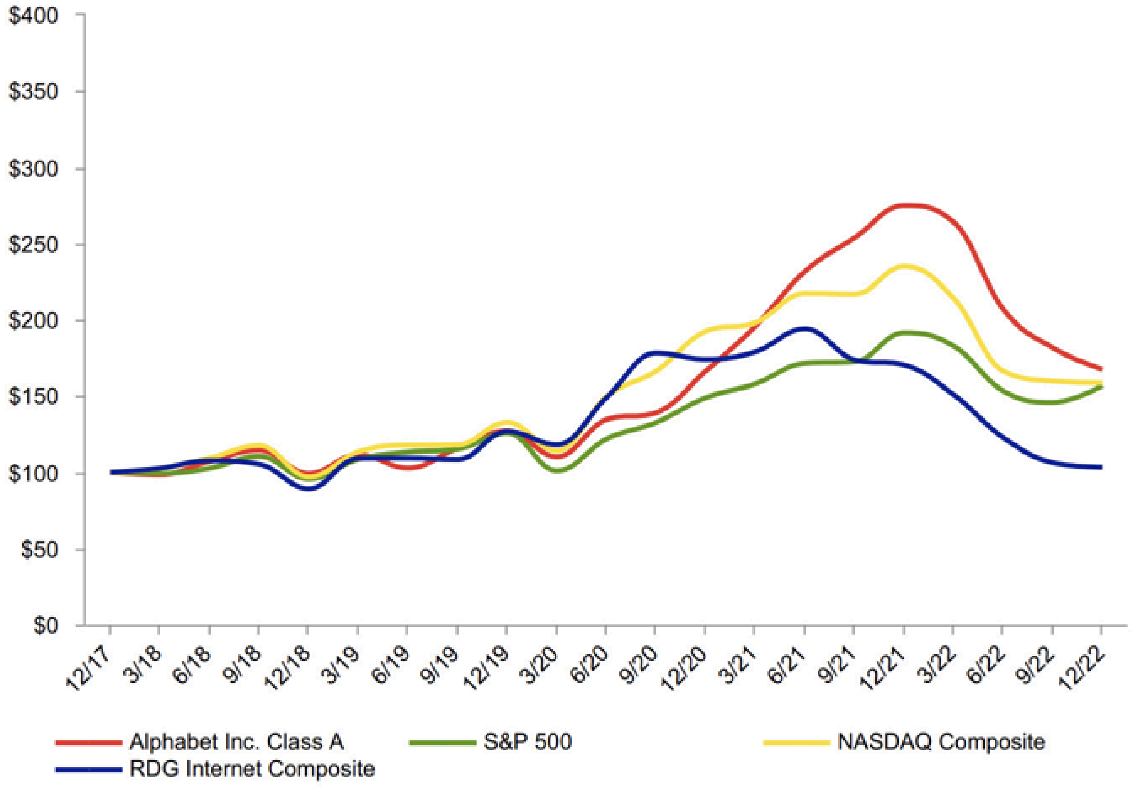
	Year Ended December 31,	
	2020	2021
TAC	$ 32,778	$ 45,566
Other cost of revenues	51,954	65,373
Total cost of revenues	$ 84,732	$ 110,939
Total cost of revenues as a percentage of revenues	46.4 %	43.1 %

COMPARISON OF CUMULATIVE 5-YEAR TOTAL RETURN*

ALPHABET INC. CLASS A COMMON STOCK

Among Alphabet Inc., the S&P 500 Index, the

NASDAQ Composite Index, and the RDG Internet Composite Index



*$100 invested on December 31, 2017 in stock or index, including reinvestment of dividends.

Copyright© 2023 S&P, a division of The McGraw-Hill Companies Inc. All rights reserved.

In [31]:

```
# Image citations. You can check how Gemini generated metadata helped in gro

print_text_to_image_citation(
    matching_results_image_fromdescription_data, print_top=False
)
```

```
Citation 1: Matched image path, page number and page text:

score:  0.66
file_name:  google-10k-sample-part1.pdf
path:  images/google-10k-sample-part1.pdf_image_4_0_21.jpeg
page number:  5
page text:  Executive Overview
The following table summarizes consolidated financial results for the years
ended
December 31, 2020 and 2021 unless otherwise specified (in millions, except f
or
per share information and percentages):
Revenues were $257.6 billion, an increase of 41%. The increase in
revenues was primarily driven by Google Services and Google Cloud. The
adverse effect of COVID-19 on 2020 advertising revenues also contributed
to the year-over-year growth.
Cost of revenues was $110.9 billion, an increase of 31%, primarily driven
by increases in TAC and content acquisition costs.
An overall increase in data centers and other operations costs was partially
offset by a reduction in depreciation expense due to the change in the
estimated useful life of our servers and certain network equipment.
Operating expenses were $68.0 billion, an increase of 20%, primarily
driven by headcount growth, increases in advertising and promotional
expenses and charges related to legal matters.

image description:  The image is a table showing financial data for the year
s ended December 31, 2020 and 2021, along with the dollar and percentage cha
nge between the two years.

Here are the elements of the table:

*   **Consolidated revenues:** 2020: $182,527, 2021: $257,637, $ Change: $7
5,110, % Change: 41%
*   **Change in consolidated constant currency revenues:** 39%
*   **Cost of revenues:** 2020: $84,732, 2021: $110,939, $ Change: $26,207,
% Change: 31%
*   **Operating expenses:** 2020: $56,571, 2021: $67,984, $ Change: $11,413,
% Change: 20%
*   **Operating income:** 2020: $41,224, 2021: $78,714, $ Change: $37,490, %
Change: 91%
*   **Operating margin:** 2020: 23%, 2021: 31%, % Change: 8%
*   **Other income (expense), net:** 2020: $6,858, 2021: $12,020, $ Change:
$5,162, % Change: 75%
*   **Net Income:** 2020: $40,269, 2021: $76,033, $ Change: $35,764, % Chang
e: 89%
*   **Diluted EPS:** 2020: $58.61, 2021: $112.20, $ Change: $53.59, % Chang
e: 91%
*   **Number of Employees:** 2020: 135,301, 2021: 156,500, $ Change: 21,199,
% Change: 16%
Citation 2: Matched image path, page number and page text:

score:  0.66
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_4_0_22.jpeg
page number:  5
page text:  Stock-Based Award Activities
```

```
The weighted-average grant-date fair value of RSUs granted during the years
ended December 31, 2019 and 2020 was $1,092.36 and $1,407.97, respectively.
Total fair value of RSUs, as of their respective vesting dates, during the y
ears
ended December 31, 2019, 2020, and 2021 were $15.2 billion, $17.8 billion, a
nd
$28.8 billion, respectively. As of December 31, 2021, there was $25.8 billio
n of
unrecognized compensation cost related to unvested employee RSUs. This
amount is expected to be recognized over a weighted-average period of 2.5
years. 401(k) Plans We have two 401(k) Savings Plans that qualify as deferre
d

image description:  The image shows a table titled "Year Ended December 31,
2020" with columns for Class A, Class B, and Class C. The table presents cal
culations for basic and diluted net income per share.

Here are the elements of the table:

*   **Basic net income per share:**
    *   Numerator: Allocation of undistributed earnings
    *   Denominator: Number of shares used in per share computation
    *   Basic net income per share

*   **Diluted net income per share:**
    *   Numerator:
        *   Allocation of undistributed earnings for basic computation
        *   Reallocation of undistributed earnings as a result of conversion
of Class B to Class A shares
        *   Reallocation of undistributed earnings
        *   Allocation of undistributed earnings
    *   Denominator:
        *   Number of shares used in basic computation
        *   Weighted-average effect of dilutive securities
        *   Add:
            *   Conversion of Class B to Class A shares outstanding
            *   Restricted stock units and other contingently issuable share
s
        *   Number of shares used in per share computation
    *   Diluted net income per share

The table includes dollar amounts for allocation of undistributed earnings a
nd share counts for the number of shares used in computations. The basic net
income per share is the same for all classes. The diluted net income per sha
re is also the same for all classes.
Citation 3: Matched image path, page number and page text:

score:  0.64
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_1_0_13.jpeg
page number:  2
page text:  APAC revenue growth from 2020 to 2021 was favorably affected by
foreign
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Australian dollar, partially offset by the U.S. dollar strengthening rel
```

```
ative to
the Japanese yen.
Other Americas growth change from 2020 to 2021 was favorably affected by
changes in foreign currency exchange rates, primarily due to the U.S. dollar
weakening relative to the Canadian dollar, partially offset by the U.S. doll
ar
strengthening relative to the Argentine peso and the Brazilian real.
Costs and Expenses
Cost of Revenues
The following tables present cost of revenues, including TAC (in millions, e
xcept
percentages):
Cost of revenues increased $26.2 billion from 2020 to 2021. The increase was
due to an increase in other cost of revenues and TAC of $13.4 billion and $1
2.8
billion, respectively.
The increase in TAC from 2020 to 2021 was due to an increase in TAC paid to
distribution partners and to Google Network partners, primarily driven by gr
owth
in revenues subject to TAC. The TAC rate decreased from 22.3% to 21.8% from
2020 to 2021 primarily due to a revenue mix shift from Google Network
properties to Google Search & other properties.
The TAC rate on Google Search & other properties revenues and the TAC rate
on Google Network revenues were both substantially consistent from 2020 to
2021. The increase in other cost of revenues from 2020 to 2021 was driven by

image description:  The image shows a table with financial data for the year
s ended December 31, 2020 and 2021. The table includes the following element
s:

| Item                                        | 2020    | 2021    |
| ------------------------------------------- | ------- | ------- |
| TAC                                         | 32,778  | 45,566  |
| Other cost of revenues                      | 51,954  | 65,373  |
| Total cost of revenues                      | 84,732  | 110,939 |
| Total cost of revenues as a percentage of revenues | 46.4%   | 43.1%   |

The table shows an increase in all cost of revenues categories from 2020 to
2021. The total cost of revenues as a percentage of revenues decreased from
46.4% to 43.1%.

Citation 4: Matched image path, page number and page text:

score:  0.63
file_name:  google-10k-sample-part1.pdf
path:  images/google-10k-sample-part1.pdf_image_3_0_18.jpeg
page number:  4
page text:  The graph below matches Alphabet Inc. Class A's cumulative 5-yea
r total
stockholder return on common stock with the cumulative total returns of the
S&P
500 index, the NASDAQ Composite index, and the RDG Internet Composite
index. The graph tracks the performance of a $100 investment in our common
stock and in each index (with the reinvestment of all dividends) from Decemb
er
31, 2017 to December 31, 2022. The returns shown are based on historical
```

```
results and are not intended to suggest future performance.

image description:  The image is a graph comparing the cumulative 5-year tot
al return of Alphabet Inc. Class A common stock, the S&P 500 Index, the NASD
AQ Composite Index, and the RDG Internet Composite Index. The graph shows th
e performance of each investment over time, starting from 12/17 and ending i
n 12/22. The red line represents Alphabet Inc. Class A, the green line repre
sents the S&P 500, the yellow line represents the NASDAQ Composite, and the
blue line represents the RDG Internet Composite.
Citation 5: Matched image path, page number and page text:

score:  0.63
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_3_0_19.jpeg
page number:  4
page text:  liquidation and dividend rights are identical, the undistributed
earnings are
allocated on a proportionate basis.
In the years ended December 31, 2019, 2020 and 2021, the net income per
share amounts are the same for Class A, Class B, and Class C stock because
the holders of each class are entitled to equal per share dividends or distr
ibutions
in liquidation in accordance with the Amended and Restated Certificate of
Incorporation of Alphabet Inc.
The following tables set forth the computation of basic and diluted net inco
me per
share of Class A, Class B, and Class C stock (in millions, except share amou
nts
which are reflected in thousands and per share amounts):

image description:  The image is a table showing the basic and diluted net i
ncome per share for the year ended December 31, 2019. The table is broken do
wn by Class A, Class B, and Class C.

Here are the elements of the table:

*   **Year Ended December 31, 2019**
*   **Class A**
*   **Class B**
*   **Class C**
*   **Basic net income per share:**
    *   Numerator
        *   Allocation of undistributed earnings: Class A is $14,846, Class
B is $2,307, and Class C is $17,190.
    *   Denominator
        *   Number of shares used in per share computation: Class A is 299,4
02, Class B is 46,527, and Class C is 346,667.
    *   Basic net income per share: Class A is $49.59, Class B is $49.59, an
d Class C is $49.59.
*   **Diluted net income per share:**
    *   Numerator
        *   Allocation of undistributed earnings for basic computation: Clas
s A is $14,846, Class B is $2,307, and Class C is $17,190.
        *   Reallocation of undistributed earnings as a result of conversion
of Class B to Class A shares: Class A is $2,307, Class B is $0, and Class C
is $0.
```

```
        *   Reallocation of undistributed earnings: Class A is $(126), Class
B is $(20), and Class C is $126.
        *   Allocation of undistributed earnings: Class A is $17,027, Class
B is $2,287, and Class C is $17,316.
    *   Denominator
        *   Number of shares used in basic computation: Class A is 299,402,
Class B is 46,527, and Class C is 346,667.
        *   Weighted-average effect of dilutive securities
            *   Add:
                *   Conversion of Class B to Class A shares outstanding: Cla
ss A is 46,527, Class B is $0, and Class C is $0.
                *   Restricted stock units and other contingently issuable s
hares: Class A is 413, Class B is $0, and Class C is 5,547.
        *   Number of shares used in per share computation: Class A is 346,3
42, Class B is 46,527, and Class C is 352,214.
    *   Diluted net income per share: Class A is $49.16, Class B is $49.16,
and Class C is $49.16.
Citation 6: Matched image path, page number and page text:

score:  0.62
file_name:  google-10k-sample-part1.pdf
path:  images/google-10k-sample-part1.pdf_image_2_0_15.jpeg
page number:  3
page text:  Stock Performance Graphs
The graph below matches Alphabet Inc. Class A's cumulative 5-year total
stockholder return on common stock with the cumulative total returns of the
S&P
500 index, the NASDAQ Composite index, and the RDG Internet Composite
index. The graph tracks the performance of a $100 investment in our common
stock and in each index (with the reinvestment of all dividends) from Decemb
er
31, 2016 to December 31, 2021. The returns shown are based on historical
results and are not intended to suggest future performance.

image description:  The image is a graph comparing the cumulative 5-year tot
al return of Alphabet Inc. Class A Common Stock, the S&P 500 Index, the NASD
AQ Composite Index, and the RDG Internet Composite Index. The graph shows th
e performance of each investment over time, starting from 12/16 and ending o
n 12/21.
Citation 7: Matched image path, page number and page text:

score:  0.61
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_0_1_8.jpeg
page number:  1
page text:  source: https://abc.xyz/assets/investor/static/pdf/20220202_alph
abet_10K.pdf
source: https://abc.xyz/assets/9a/bd/838c917c4b4ab21f94e84c3c2c65/goog-10-k-
q4-2022.pdf
Note: Tables and figures are converted to images for demonstration purposes.
Revenues by Geography
The following table presents revenues by geography as a percentage of
revenues, determined based on the addresses of our customers:
The following table presents the foreign exchange effect on international
revenues and total revenues (in millions, except percentages):
EMEA revenue growth from 2020 to 2021 was favorably affected by foreign
```

```
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Euro and British pound.

image description:  The image is a table showing revenues for the years ende
d December 31, 2020 and 2021, along with the percentage change from the prio
r year.

Here's the table data:

| Item                                 | 2020    | 2021    | % Change from
Prior Year |
| ------------------------------------ | ------- | ------- | --------------
---------- |
| EMEA revenues                        | $55,370 | $79,107 | 43%
|
| EMEA constant currency revenues      |         | 76,321  | 38%
|
| APAC revenues                        | 32,550  | 46,123  | 42%
|
| APAC constant currency revenues      |         | 45,666  | 40%
|
| Other Americas revenues              | 9,417   | 14,404  | 53%
|
| Other Americas constant currency revenues |         | 14,317  | 52%
|
| United States revenues               | 85,014  | 117,854 | 39%
|
| Hedging gains (losses)               | 176     | 149     |
|
| Total revenues                       | $182,527| $257,637| 41%
|
| Revenues, excluding hedging effect   | $182,351| $257,488|
|
| Exchange rate effect                 |         | (3,330) |
|
| Total constant currency revenues     |         | $254,158| 39%
|

The table shows revenue breakdowns by region (EMEA, APAC, Other Americas, Un
ited States) and includes constant currency revenue figures for 2021. It als
o shows hedging gains/losses, total revenues, revenues excluding hedging eff
ect, exchange rate effect, and total constant currency revenues.

Citation 8: Matched image path, page number and page text:

score:  0.59
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_5_0_26.jpeg
page number:  6
page text:  salary arrangements under Section 401(k) of the Internal Revenue
Code. Under
these 401(k) Plans, matching contributions are based upon the amount of the
employees contributions subject to certain limitations. We recognized expens
e of
approximately $724 million, $855 million, and $916 million for the years end
```

ed
December 31, 2019, 2020, and 2021, respectively. Note 14. Income Taxes
Income from continuing operations before income taxes consisted of the
following (in millions):
Deferred Income Taxes
Deferred income taxes reflect the net effects of temporary differences between
the carrying amounts of assets and liabilities for financial reporting purposes and
the amounts used for income tax purposes. Significant components of our
deferred tax assets and liabilities were as follows (in millions):
As of December 31, 2021, our federal, state, and foreign net operating loss
carryforwards for income tax purposes were approximately $5.6 billion, $4.6
billion, and $1.7 billion respectively. If not utilized, the federal net operating loss
carryforwards will begin to expire in 2023, foreign net operating loss
carryforwards will begin to expire in 2025 and the state net operating loss
carryforwards will begin to expire in 2028. It is more likely than not that
certain

image description: The image is a table showing deferred tax assets and deferred tax liabilities as of December 31, for the years 2020 and 2021.

Here are the elements of the table:

Deferred tax assets:
*   Accrued employee benefits: $580 (2020), $549 (2021)
*   Accruals and reserves not currently deductible: 1,049 (2020), 1,816 (2021)
*   Tax credits: 3,723 (2020), 5,179 (2021)
*   Net operating losses: 1,085 (2020), 1,790 (2021)
*   Operating leases: 2,620 (2020), 2,503 (2021)
*   Intangible assets: 1,525 (2020), 2,034 (2021)
*   Other: 981 (2020), 925 (2021)
*   Total deferred tax assets: 11,563 (2020), 14,796 (2021)
*   Valuation allowance: (4,823) (2020), (7,129) (2021)
*   Total deferred tax assets net of valuation allowance: 6,740 (2020), 7,667 (2021)

Deferred tax liabilities:
*   Property and equipment, net: (3,382) (2020), (5,237) (2021)
*   Net investment gains: (1,901) (2020), (3,229) (2021)
*   Operating leases: (2,354) (2020), (2,228) (2021)
*   Other: (1,580) (2020), (946) (2021)
*   Total deferred tax liabilities: (9,217) (2020), (11,640) (2021)

Net deferred tax assets (liabilities):
*   (2,477) (2020), (3,973) (2021)

Citation 9: Matched image path, page number and page text:

score: 0.58
file_name: google-10k-sample-part2.pdf
path: images/google-10k-sample-part2.pdf_image_0_0_6.jpeg
page number: 1
page text: source: https://abc.xyz/assets/investor/static/pdf/20220202_alphabet_10K.pdf

```
source: https://abc.xyz/assets/9a/bd/838c917c4b4ab21f94e84c3c2c65/goog-10-k-
q4-2022.pdf
Note: Tables and figures are converted to images for demonstration purposes.
Revenues by Geography
The following table presents revenues by geography as a percentage of
revenues, determined based on the addresses of our customers:
The following table presents the foreign exchange effect on international
revenues and total revenues (in millions, except percentages):
EMEA revenue growth from 2020 to 2021 was favorably affected by foreign
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Euro and British pound.

image description:  The image is a table showing data for the years ended De
cember 31, 2020 and 2021. The table includes the following elements:

|             | 2020 | 2021 |
| ----------- | ----------- | ----------- |
| United States      | 47%       | 46%        |
| EMEA   | 30%        | 31%        |
| APAC      | 18%       | 18%        |
| Other Americas   | 5%        | 5%        |
Citation 10: Matched image path, page number and page text:

score:  0.58
file_name:  google-10k-sample-part2.pdf
path:  images/google-10k-sample-part2.pdf_image_4_1_23.jpeg
page number:  5
page text:  Stock-Based Award Activities
The weighted-average grant-date fair value of RSUs granted during the years
ended December 31, 2019 and 2020 was $1,092.36 and $1,407.97, respectively.
Total fair value of RSUs, as of their respective vesting dates, during the y
ears
ended December 31, 2019, 2020, and 2021 were $15.2 billion, $17.8 billion, a
nd
$28.8 billion, respectively. As of December 31, 2021, there was $25.8 billio
n of
unrecognized compensation cost related to unvested employee RSUs. This
amount is expected to be recognized over a weighted-average period of 2.5
years. 401(k) Plans We have two 401(k) Savings Plans that qualify as deferre
d

image description:  The image is a table summarizing the activities for unve
sted Alphabet RSUs for the year ended December 31, 2021. The table has two c
olumns: Number of Shares and Weighted-Average Grant-Date Fair Value.

Here are the elements of the table:
*   Unvested as of December 31, 2020: Number of Shares is 19,288,793 and Wei
ghted-Average Grant-Date Fair Value is $1,262.13.
*   Granted: Number of Shares is 10,582,700 and Weighted-Average Grant-Date
Fair Value is $1,949.16.
*   Vested: Number of Shares is (11,209,486) and Weighted-Average Grant-Date
Fair Value is $1,345.98.
*   Forfeited/canceled: Number of Shares is (1,767,294) and Weighted-Average
Grant-Date Fair Value is $1,425.48.
```

```
*   Unvested as of December 31, 2021: Number of Shares is 16,894,713 and Wei
ghted-Average Grant-Date Fair Value is $1,626.13.
```

In [32]:

```python
# Text citations

print_text_to_text_citation(
    matching_results_chunks_data,
    print_top=False,
    chunk_text=True,
)
```

```
Citation 1: Matched text:

score:  0.77
file_name:  google-10k-sample-part1.pdf
page_number:  5
chunk_number:  1
chunk_text:  Executive Overview
The following table summarizes consolidated financial results for the years
ended
December 31, 2020 and 2021 unless otherwise specified (in millions, except f
or
per share information and percentages):
Revenues were $257.6 billion, an increase of 41%. The increase in
revenues was primarily driven by Google Services and Google Cloud. The
adverse effect of COVID-19 on 2020 advertising revenues also contributed
to the year-over-year growth.
Cost of revenues was $110.9 billion, an increase of 31%, primarily driven
by increases in TAC and content acquisition costs.
An overall increase in data centers and other operations costs was partially
offset by a reduction in depreciation expense due to the change in the
estimated useful life of our servers and certain network equipment.
Operating expenses were $68.0 billion, an increase of 20%, primarily
driven by headcount growth, increases in advertising and promotional
expenses and charges related to legal matters.

Citation 2: Matched text:

score:  0.72
file_name:  google-10k-sample-part2.pdf
page_number:  2
chunk_number:  2
chunk_text:  21 was due to an increase in TAC paid to
distribution partners and to Google Network partners, primarily driven by gr
owth
in revenues subject to TAC. The TAC rate decreased from 22.3% to 21.8% from
2020 to 2021 primarily due to a revenue mix shift from Google Network
properties to Google Search & other properties.
The TAC rate on Google Search & other properties revenues and the TAC rate
on Google Network revenues were both substantially consistent from 2020 to
2021. The increase in other cost of revenues from 2020 to 2021 was driven by

Citation 3: Matched text:

score:  0.71
file_name:  google-10k-sample-part2.pdf
page_number:  2
chunk_number:  1
chunk_text:  APAC revenue growth from 2020 to 2021 was favorably affected by
foreign
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Australian dollar, partially offset by the U.S. dollar strengthening rel
ative to
the Japanese yen.
Other Americas growth change from 2020 to 2021 was favorably affected by
changes in foreign currency exchange rates, primarily due to the U.S. dollar
```

```
weakening relative to the Canadian dollar, partially offset by the U.S. doll
ar
strengthening relative to the Argentine peso and the Brazilian real.
Costs and Expenses
Cost of Revenues
The following tables present cost of revenues, including TAC (in millions, e
xcept
percentages):
Cost of revenues increased $26.2 billion from 2020 to 2021. The increase was
due to an increase in other cost of revenues and TAC of $13.4 billion and $1
2.8
billion, respectively.
The increase in TAC from 2020 to 2021 was due to an increase in TAC paid to
distribution partners and to Google Network partners, prima
Citation 4: Matched text:

score:  0.71
file_name:  google-10k-sample-part2.pdf
page_number:  3
chunk_number:  1
chunk_text:  increases in content acquisition costs primarily for YouTube, d
ata center and
other operations costs, and hardware costs. The increase in data center and
Table of Contents Alphabet Inc. 36 other operations costs was partially offs
et by
a reduction in depreciation expense due to the change in the estimated usefu
l life
of our servers and certain network equipment beginning in the first quarter
of
2021.
Net Income Per Share
We compute net income per share of Class A, Class B, and Class C stock using
the two-class method. Basic net income per share is computed using the
weighted-average number of shares outstanding during the period. Diluted net
income per share is computed using the weighted-average number of shares and
the effect of potentially dilutive securities outstanding during the period.
Potentially dilutive securities consist of restricted stock units and other
contingently issuable shares. The dilutive effect of outstanding restricted
stock
units and other contingently issuable
Citation 5: Matched text:

score:  0.69
file_name:  google-10k-sample-part1.pdf
page_number:  6
chunk_number:  1
chunk_text:  Other information:
Operating cash flow was $91.7 billion, primarily driven by revenues
generated from our advertising products.
Share repurchases were $50.3 billion, an increase of 62%. See Note 11 of
the Notes to Consolidated Financial Statements included in Item 8 of this
Annual Report on Form 10-K for further information.
Capital expenditures, which primarily reflected investments in technical
infrastructure, were $24.6 billion.
In January 2021, we updated the useful lives of certain of our servers and
network equipment, resulting in a reduction in depreciation expense of
```

```
$2.6 billion recorded primarily in cost of revenues and R&D. See Note 1 of
the Notes to Consolidated Financial Statements included in Item 8 of this
Annual Report on Form 10-K for further information.
Our acquisition of Fitbit closed in early January 2021, and the related
revenues are included in Google other. See Note 8 of the Notes to
Consolidated Financial Statements included in Item 8 of this Annual Report
on F
Citation 6: Matched text:

score:  0.68
file_name:  google-10k-sample-part1.pdf
page_number:  6
chunk_number:  2
chunk_text:  te 8 of the Notes to
Consolidated Financial Statements included in Item 8 of this Annual Report
on Form 10-K for further information.

On February 1, 2022, the Company announced that the Board of Directors
had approved and declared a 20- for-one stock split in the form of a
one-time special stock dividend on each share of the Companys Class A,
Class B, and Class C stock. See Note 11 of the Notes to Consolidated
Financial Statements included in Item 8 of this Annual Report on Form
10-K for additional information.
The Effect of COVID-19 on our Financial Results
We began to observe the effect of COVID-19 on our financial results in March
2020 when, despite an increase in users' search activity, our advertising
revenues declined compared to the prior year. This was due to a shift of use
r
search activity to less commercial topics and reduced spending by our
advertisers. For the quarter ended June 30, 2020 our advertising revenues
declined due to the continued effects of COVID-19 and the rel
Citation 7: Matched text:

score:  0.68
file_name:  google-10k-sample-part2.pdf
page_number:  4
chunk_number:  1
chunk_text:  liquidation and dividend rights are identical, the undistribute
d earnings are
allocated on a proportionate basis.
In the years ended December 31, 2019, 2020 and 2021, the net income per
share amounts are the same for Class A, Class B, and Class C stock because
the holders of each class are entitled to equal per share dividends or distr
ibutions
in liquidation in accordance with the Amended and Restated Certificate of
Incorporation of Alphabet Inc.
The following tables set forth the computation of basic and diluted net inco
me per
share of Class A, Class B, and Class C stock (in millions, except share amou
nts
which are reflected in thousands and per share amounts):

Citation 8: Matched text:

score:  0.67
file_name:  google-10k-sample-part1.pdf
```

```
page_number:  3
chunk_number:  1
chunk_text:  Stock Performance Graphs
The graph below matches Alphabet Inc. Class A's cumulative 5-year total
stockholder return on common stock with the cumulative total returns of the
S&P
500 index, the NASDAQ Composite index, and the RDG Internet Composite
index. The graph tracks the performance of a $100 investment in our common
stock and in each index (with the reinvestment of all dividends) from Decemb
er
31, 2016 to December 31, 2021. The returns shown are based on historical
results and are not intended to suggest future performance.

Citation 9: Matched text:

score:  0.67
file_name:  google-10k-sample-part2.pdf
page_number:  1
chunk_number:  1
chunk_text:  source: https://abc.xyz/assets/investor/static/pdf/20220202_alp
habet_10K.pdf
source: https://abc.xyz/assets/9a/bd/838c917c4b4ab21f94e84c3c2c65/goog-10-k-
q4-2022.pdf
Note: Tables and figures are converted to images for demonstration purposes.
Revenues by Geography
The following table presents revenues by geography as a percentage of
revenues, determined based on the addresses of our customers:
The following table presents the foreign exchange effect on international
revenues and total revenues (in millions, except percentages):
EMEA revenue growth from 2020 to 2021 was favorably affected by foreign
currency exchange rates, primarily due to the U.S. dollar weakening relative
to
the Euro and British pound.

Citation 10: Matched text:

score:  0.64
file_name:  google-10k-sample-part1.pdf
page_number:  4
chunk_number:  1
chunk_text:  The graph below matches Alphabet Inc. Class A's cumulative 5-ye
ar total
stockholder return on common stock with the cumulative total returns of the
S&P
500 index, the NASDAQ Composite index, and the RDG Internet Composite
index. The graph tracks the performance of a $100 investment in our common
stock and in each index (with the reinvestment of all dividends) from Decemb
er
31, 2017 to December 31, 2022. The returns shown are based on historical
results and are not intended to suggest future performance.
```

Conclusions

Congratulations on making it through this multimodal RAG notebook!

While multimodal RAG can be quite powerful, note that it can face some limitations:

- **Data dependency:** Needs high-quality paired text and visuals.
- **Computationally demanding:** Processing multimodal data is resource-intensive.
- **Domain specific:** Models trained on general data may not shine in specialized fields like medicine.
- **Black box:** Understanding how these models work can be tricky, hindering trust and adoption.

Despite these challenges, multimodal RAG represents a significant step towards search and retrieval systems that can handle diverse, multimodal data.