### Multimodal AI Application with LangChain & the OpenAI API

#### Abstract

This document explores the development of a multimodal application that integrates advanced natural language processing capabilities using LangChain and the OpenAI API. The application is designed to enhance interaction with audiovisual content through automated transcription, semantic analysis, and an AI-based question-and-answer system, with potential applications in education, digital marketing, and customer support.

#### Literature Review

**2.1 Introduction**

In the past decade, the fields of natural language processing (NLP) and multimedia content analysis have experienced rapid advancement. Multimodal applications, which integrate multiple types of data such as text, audio, and images, have emerged as a powerful solution to complex problems. This literature review examines previous studies in these fields, highlighting foundational research and exploring the evolution of AI APIs, with a particular focus on the contributions of OpenAI. Additionally, similar applications in academic and commercial domains are reviewed.

**2.2 Multimodal Applications**

Multimodal applications combine information from various sources to provide a more comprehensive and accurate understanding. Examples of these applications include image recognition systems that use textual descriptions to enhance accuracy and virtual assistants that understand voice and text commands to perform complex tasks.

**Key Research:**

1. **VQA: Visual Question Answering** - Antol et al. (2015) presented a visual question answering model (VQA) that combines images and text to answer questions about the content of an image. This pioneering work demonstrated the effectiveness of multimodal models for visual comprehension tasks.
2. **Multimodal Deep Learning** - Ngiam et al. (2011) explored multimodal deep learning, combining audio and video to improve the accuracy of emotion recognition models. This study was one of the first to demonstrate the potential of deep learning to integrate multiple modalities.

**2.3 Natural Language Processing**

NLP has advanced significantly, enabling machines to effectively understand and generate human language. From early rule-based models to modern deep learning models, NLP has transformed numerous industries.

**Key Research:**

1. **BERT: Bidirectional Encoder Representations from Transformers** - Devlin et al. (2018) introduced BERT, a pre-trained language model that captures bidirectional contexts in text. BERT has set new standards in various NLP tasks, including question answering and sentiment analysis.
2. **GPT-3: Generative Pre-trained Transformer 3** - Brown et al. (2020) developed GPT-3, a language model with 175 billion parameters that can generate coherent text and perform NLP tasks without specific training. GPT-3 has shown remarkable capabilities in text generation, translation, and conversation.

**2.4 Multimedia Content Analysis**

Multimedia content analysis involves extracting meaningful information from audiovisual data. This field encompasses image and video recognition as well as audio transcription and analysis.

**Key Research:**

1. **YOLO: You Only Look Once** - Redmon et al. (2016) introduced YOLO, a real-time object detection model that revolutionized image and video analysis by combining speed and accuracy.
2. **DeepSpeech** - Hannun et al. (2014) presented DeepSpeech, a speech recognition system that uses deep neural networks to transcribe audio to text with high accuracy. This work has been fundamental in developing automatic transcription systems.

**2.5 Evolution of AI APIs**

AI APIs have evolved significantly, providing powerful and accessible tools for developers and businesses. OpenAI has been a leader in this field, developing advanced language models and offering APIs that facilitate the integration of AI into various applications.

**Contributions of OpenAI:**

1. **OpenAI GPT-2 and GPT-3** - OpenAI released GPT-2 and GPT-3, language models that have demonstrated advanced text generation and language comprehension capabilities. These APIs have enabled developers to create sophisticated applications with ease.
2. **Codex** - OpenAI Codex, based on GPT-3, is a model specialized in understanding and generating code. Codex has been integrated into tools like GitHub Copilot, facilitating AI-assisted programming.

**2.6 Similar Applications in Academic and Commercial Domains**

Numerous applications in academic and commercial domains have adopted multimodal technologies and AI APIs to enhance user interaction and automate complex processes.

**Academic Examples:**

1. **CMU Sphinx** - A speech recognition system developed by Carnegie Mellon University that uses acoustic and language models to transcribe audio with high precision.
2. **Berkeley Vision and Learning Center (BVLC)** - The Caffe platform, developed at BVLC, has been used to implement and train image and video recognition models.

**Commercial Examples:**

1. **Google Assistant** - Uses natural language processing and voice recognition to provide contextual responses and perform tasks based on voice commands.
2. **Amazon Alexa** - A virtual assistant that integrates multiple input modalities (voice and text) to interact with users and control smart devices.

**2.7 Conclusion of the Literature Review**

The literature review shows significant progress in developing multimodal applications, natural language processing, and multimedia content analysis. AI APIs, especially those developed by OpenAI, have played a crucial role in facilitating these advancements. The integration of these technologies into commercial and academic applications demonstrates their potential to transform various industries.

#### Objectives

The application is designed to be a multimodal tool that not only manages and processes audiovisual content but also leverages advanced language processing capabilities to interact intelligently with users, providing valuable information based on the downloaded and processed multimedia content. It is ideal for educational, research, and entertainment applications where quick information retrieval and content-based interaction are crucial.

**Key Objectives:**

1. **Audio Download and Extraction**: The application can download YouTube videos, extract the audio, and convert it to a manageable format like MP3. This step is crucial to obtain material that will later be processed for text extraction or direct analysis.
2. **Content Analysis Using AI**: The extracted audio can be transcribed or directly analyzed to obtain information that will be transformed into text. This text is then analyzed using artificial intelligence models, enabling a semantic understanding of the content.
3. **Question and Answer System Construction**: Using the analyzed and structured text, the application employs an OpenAI GPT-4 model and a FAISS index to create a response retrieval system. This enables the application to answer specific questions based on the content of the downloaded video.
4. **Interactive User Interface for Queries (Optional)**: Finally, the application provides an interface where users can ask specific questions about the video content and receive detailed and contextually relevant answers. This is done through a combination of semantic search and natural language generation.

#### Business Use Cases

The described application has considerable potential for application in various business contexts. Here are three practical use cases from a business perspective:

**1. Online Education and Learning Platform** **Description:** An educational platform can use this application to enhance the accessibility and interactivity of its online resources. For example, the application can download video lectures, extract and analyze the audio to convert it into text, and then offer a question-and-answer system where students can ask specific questions about the lecture and receive instant answers based on the analyzed content. **Business Benefits:**

* **Improved Student Engagement:** Allows students to interact more deeply with course material.
* **Enhanced Accessibility:** Provides textual content of videos, which is invaluable for hearing-impaired students or those who prefer to learn through reading.
* **Study Efficiency:** Helps students clarify doubts instantly without having to review long videos.

**2. Content Analysis Tool for Digital Marketing** **Description:** Companies engaged in digital marketing can use this application to analyze the content of promotional or educational videos uploaded by competitors or influencers on platforms like YouTube. They can extract and analyze the audio to understand the topics discussed, frequently asked questions, and responses given, allowing a better understanding of market discourse and identification of emerging trends. **Business Benefits:**

* **Competitive Intelligence:** Facilitates the analysis of competitors' content strategies.
* **Content Optimization:** Allows the adjustment of marketing campaigns based on topics and questions that generate more interaction in the target audience.
* **Marketing Innovation:** Helps identify gaps or underexplored areas in the currently available content, opening opportunities for innovation.

**3. Customer Support and Service Platform for Technology Companies** **Description:** Technology companies can integrate this technology into their support systems to provide automatic answers to frequently asked questions based on video tutorials or product demos. By converting explanatory videos into an interactive text format, they can allow users to ask specific questions and get instant answers without direct human intervention. **Business Benefits:**

* **Reduced Customer Support Load:** Minimizes the need for human intervention by providing quick and accurate answers to common questions.
* **Improved Customer Satisfaction:** Offers instant, on-demand support, improving the overall user experience.
* **Support Scalability:** Allows scaling support operations without proportionally increasing human resources.

**4. Media Analysis Platform for Journalists and Media Outlets** **Description:** A media analysis platform can use this application to monitor and analyze news videos and reports from various online sources like YouTube. Journalists can automatically extract and transcribe video content to quickly analyze the discussed topics, important quotes, and perspectives offered on current events or trends. **Business Benefits:**

* **Information Collection Efficiency:** Allows journalists to gather and synthesize information more quickly and efficiently.
* **Analytical Depth:** Facilitates a deeper analysis of media narratives and discourses, helping identify biases or prevailing viewpoints.
* **Publishing Speed:** Accelerates the content production process by reducing the time needed to review and summarize video sources.

**5. Compliance Monitoring Tool for Corporations** **Description:** In regulated sectors, such as finance or pharmaceuticals, companies can use this technology to monitor and review compliance in training sessions and internal communications. By converting training sessions or policy updates in video format to text, the application can allow quick queries to ensure that all points are covered according to regulatory standards. **Business Benefits:**

* **Compliance Assurance:** Ensures that internal training and communications comply with current regulations.
* **Improved Accessibility:** Allows employees to quickly access specific information within extensive material without having to watch entire videos.
* **Effective Auditing:** Facilitates internal audits by providing an easy way to review and verify the content presented in training and communications.

#### Technological Use Cases

**1. Automatic Multimedia Content Annotation System** **Technical Description:** Develop a system that automatically annotates videos with relevant tags, descriptions, and metadata based on the audio content. This can include detecting specific topics, identifying people and places, and classifying content by categories. **Application:** This system can be used by video hosting platforms to improve content search and organization, making it easier for users to find relevant videos based on specific topics.

**2. Real-Time Transcription and Subtitling Tool** **Technical Description:** Implement a tool that converts video audio into text in real time to create automatic transcripts and subtitles, using advanced voice recognition and natural language processing models. **Application:** This tool can be especially useful for streaming services, online conferences, and educational platforms, providing accessibility to hearing-impaired people and speakers of other languages.

**3. Sentiment and Trend Analysis Platform** **Technical Description:** Develop a platform that analyzes the content of videos to detect the prevailing sentiment and trends in discussions. This can include analyzing the tone of speech, the words used, and the general context of the discourse. **Application:** This technology can be useful for market analysts, media researchers, and advertising agencies to better understand public perception and emotional responses to certain topics or products.

**4. Video-Based Query Response System for Customer Support** **Technical Description:** Create a system that allows users to ask specific questions about products or services and receive answers extracted directly from instructional or promotional videos. **Application:** Companies with a large base of instructional videos can implement this technology to improve their customer support, providing quick and accurate answers without human intervention.

**5. Inappropriate Content Detection Tool** **Technical Description:** Develop a system that automatically reviews and detects inappropriate content in videos, such as offensive language or prohibited images, using audio and video analysis. **Application:** This system is valuable for social media platforms, online television, and streaming services that need to ensure that shared content complies with appropriate content policies.

**6. Interactive Chatbots Integration with Multimedia Content** **Technical Description:** Integrate chatbots that can analyze videos in real time to provide richer interactions, guiding users through the video content or answering contextual questions about it. **Application:** This can be used by e-learning platforms to create more interactive and personalized learning experiences, where the chatbot acts as a virtual tutor that answers questions and suggests resources based on the viewed content.

These use cases not only demonstrate the wide range of technical applications for this technology but also highlight how it can be implemented to improve user interaction, accessibility, regulatory compliance, and operational efficiency in various sectors.

#### Future Integrations

To expand and enhance the functionalities of the video content analysis and processing application, several integrations and technological improvements can be considered that could significantly increase its value and applicability in various sectors. Here are some possible future integrations:

**1. Integration with Augmented Reality (AR) and Virtual Reality (VR) Platforms**

* **Description:** Integrate the application’s capability to analyze and answer questions about video content with AR and VR platforms. This would allow users to interact with video content in an immersive environment, obtaining real-time answers and annotations within their field of view or virtual experience.
* **Benefit:** It would enhance educational and entertainment experiences, allowing users to fully immerse themselves in interactive learning or improved viewing experiences.

**2. Improvements in Natural Language Processing for Deeper Context Analysis**

* **Description:** Implement more advanced natural language processing models that can better understand context, sarcasm, and cultural references in video dialogues.
* **Benefit:** This would allow more precise and contextually relevant answers, improving the application’s utility in educational and customer support environments where accurate interpretations are crucial.

**3. Integration with Image and Video Analysis Systems**

* **Description:** Combine audio analysis with computer vision technologies to simultaneously analyze the visual and auditory content of videos. This could include object, face, action, and event recognition within videos.
* **Benefit:** It would significantly expand the application’s capability to provide analysis and answers based on a comprehensive understanding of multimedia content.

**4. Improved Multilingual Support**

* **Description:** Enhance the application’s multilingual capabilities to support transcription, translation, and answer generation in multiple languages.
* **Benefit:** This would make the application more globally accessible, allowing users from different languages and cultures to use the application effectively.

**5. Integration with Smart Voice Assistants**

* **Description:** Integrate the application with voice assistants like Alexa, Google Assistant, or Siri, allowing users to interact with video content through voice commands.
* **Benefit:** It would facilitate accessibility and increase convenience, allowing users to obtain information and answer questions about videos without manually interacting with a device.

**6. Machine Learning Capabilities for Personalization**

* **Description:** Incorporate machine learning algorithms that can learn from user interactions and personalize responses and suggested content based on each user’s preferences and history.
* **Benefit:** It would improve user experience by making interactions more relevant and personalized, increasing user retention and satisfaction.

**7. Developer APIs**

* **Description:** Develop and offer a public API that allows other developers to integrate the application’s functionalities into their own projects or platforms.
* **Benefit:** It would expand the application’s reach, allowing others to innovate and build on the already developed technology base, opening new avenues for monetization.

These integrations would not only expand the current capabilities of the application but also open new market opportunities and applications in additional sectors such as education, entertainment, security, and more.

#### Improvement Proposals

**Improvement in Text Extraction and Semantic Analysis**

* **Proposal:** Implement more advanced voice recognition models, such as those based on deep learning (e.g., Facebook’s Wave2Vec or OpenAI’s Whisper models), to improve the accuracy of text extraction from audio.
* **Implementation:** Integrate these models by updating the section of code that handles transcription, replacing simple audio-to-text conversion with a more robust process that uses these advanced models.

**2. Expansion to Video Analysis**

* **Proposal:** Incorporate video analysis using computer vision libraries like OpenCV or cloud services like Google Video Intelligence to detect and label objects, scenes, and activities in videos.
* **Implementation:** Add a video processing module after video download that processes visual content and extracts relevant metadata, which can be indexed along with the text to enrich the QA system’s responses.

**3. Implementation of Machine Learning Capabilities for Personalization**

* **Proposal:** Develop a personalized recommendation system based on users’ past interactions with the application, using machine learning algorithms to learn content preferences and interaction styles.
* **Implementation:** Incorporate a database to record users’ questions and preferences, and use this data to train a recommendation model. This could be integrated into the response system to suggest related content or adjust answers according to user preferences.

**4. Multilingual Support**

* **Proposal:** Expand language support to include transcription, translation, and answer generation in multiple languages, using models like Hugging Face’s Transformer.
* **Implementation:** Integrate multilingual models into the text processing pipeline and configure the QA system to detect the user’s language and respond in the same language.

**5. Improved User Interface**

* **Proposal:** Develop a graphical user interface (GUI) that allows users to interact more easily with the application, view videos, and receive responses in a more user-friendly way.
* **Implementation:** Use frameworks like Streamlit or Dash to build a web interface where users can upload videos, view the transcription, ask questions, and see the responses directly on the page.

**6. Developer API**

* **Proposal:** Create a public API for external developers to integrate the QA system’s functionality into their own applications.
* **Implementation:** Develop API endpoints using frameworks like Flask or FastAPI, allowing developers to send videos or audio and receive transcriptions and answers to specific questions.

These improvements would not only expand the application’s capabilities but also increase its accessibility, usability, and adaptability to different contexts and user needs.

#### Application Programming

**Library Installation**

!pip install SQLAlchemy==2.0.31

!pip install langchain==0.0.292

!pip install tiktoken==0.5.1

!pip install docarray==0.38.0

!pip install openai==0.28

!pip install yt\_dlp

!pip install youtube-dl

!pip install pydub

!pip install python-dotenv

* **Purpose:** This block installs all the necessary dependencies for the notebook to function correctly. Each of these libraries has a specific purpose:
  + SQLAlchemy: An ORM library for interacting with databases abstractly in Python.
  + LangChain: Facilitates the construction of applications that use natural language processing and language models.
  + TikToken: Possibly related to handling tokens in texts or authentications.
  + DocArray: Manages and processes collections of documents, useful in applications involving large amounts of textual or multimedia data.
  + OpenAI: Provides access to OpenAI's advanced language models, such as GPT-3 and GPT-4.
  + yt\_dlp / youtube-dl: Tools for downloading videos from platforms like YouTube.
  + pydub: Used for manipulating audio files, such as cutting, concatenating, or changing formats.
  + python-dotenv: Loads environment variables from a .env file, facilitating development environment configuration without exposing credentials in the code.

**Module Import**

import os

import glob

import openai

import yt\_dlp as youtube\_dl

from yt\_dlp import DownloadError

import docarray

import pydub

from langchain.embeddings.openai import OpenAIEmbeddings

from langchain.text\_splitter import CharacterTextSplitter

from langchain.vectorstores.faiss import FAISS

from langchain.document\_loaders import TextLoader

from langchain.chains import RetrievalQA

from langchain.chat\_models import ChatOpenAI

from langchain.vectorstores import DocArrayInMemorySearch

from IPython.display import display, Markdown

* **Purpose:** This block imports all the necessary modules for the script to function. Each module has a specific function:
  + os, glob: Used for file system operations.
  + openai: To interact with the OpenAI API.
  + yt\_dlp, DownloadError: To download videos and handle possible download errors.
  + docarray: To work with document collections.
  + pydub: For audio editing operations.
  + langchain: Various modules for loading documents, creating chat models, and performing search and retrieval using embeddings.

**OpenAI API Configuration**

openai.api\_key = "your\_openAI\_key"

* **Purpose:** Sets the OpenAI API key needed to access the language processing models.

**YouTube Video Audio Download and Processing**

# Define the video URL and the output directory for the audio

youtube\_url = "https://www.youtube.com/watch?v=CTl3qVdfVT8"

output\_dir = "content/audio/"

# Configure yt\_dlp to download only the best audio and convert it to MP3

ydl\_config = {

"format": "bestaudio/best",

"postprocessors": [{

"key": "FFmpegExtractAudio",

"preferredcodec": "mp3",

"preferredquality": "192",

}],

"outtmpl": os.path.join(output\_dir, "%(title)s.%(ext)s"),

"verbose": True

}

# Create the directory if it does not exist

if not os.path.exists(output\_dir):

os.makedirs(output\_dir)

# Attempt to download the video and extract the audio, handling possible errors

try:

with youtube\_dl.YoutubeDL(ydl\_config) as ydl:

ydl.download([youtube\_url])

except DownloadError as e:

print(f"Failed to download video: {e}")

* **Purpose:** This block configures and executes the audio download from a specific YouTube video, storing the result in a predefined directory. It uses yt\_dlp to download the best available audio and convert it to MP3. It handles errors in case the download fails.

**Text Processing and FAISS Index Creation**

# Code not provided for text extraction: Assume it is extracted from the MP3s

# Initialize OpenAI embeddings to convert text to vectors

embeddings = OpenAIEmbeddings(openai\_api\_key=openai.api\_key)

# Assume we have text in 'texts' and create a FAISS index with those texts

faiss\_index = FAISS.from\_texts([text.page\_content for text in texts], embeddings)

# Confirmation message

print("FAISS index created successfully.")

* **Purpose:** This block assumes that text has been extracted from the downloaded audio. It initializes OpenAI embeddings and creates a FAISS index to enable quick and efficient searches within the texts. This index will be used to answer questions based on the content.

**GPT-4 Chat Model and RetrievalQA Configuration and Use**

# Configure the GPT-4 chat model

chat\_model = ChatOpenAI(model\_name="gpt-4")

# Create a question-and-answer chain using the model and FAISS index

qa\_chain = RetrievalQA.from\_chain\_type(llm=chat\_model, chain\_type="stuff", retriever=faiss\_index.as\_retriever())

# Execute a test question and display the response

question = "What is the main topic of the video?"

response = qa\_chain.run(question)

print(f"Question: {question}")

print(f"Answer: {response}")

* **Purpose:** Configures a GPT-4-based chat model and sets up a question-and-answer chain using both the model and the FAISS index. This enables the generation of responses to questions about the processed content in an integrated and seamless manner, demonstrating the application's ability to interact intelligently with the user.

**GUI Programming** **Step 1: Environment Setup**

1. **Streamlit Installation:** If not yet installed, you can add Streamlit to your Python environment by running:

pip install streamlit

1. **Prepare the development environment:** Ensure you have all the necessary packages for your application, including those for handling videos, audio processing, and any specific libraries for text processing or ML you are using (such as pydub, openai, etc.).

**Step 2: Creating the Streamlit Application**

1. **Basic application structure:** Create a Python file, for example, app.py, that will contain the code for your Streamlit application.
2. **Import Streamlit and other necessary libraries:**

import streamlit as st

import os

from your\_audio\_processing\_module import process\_video, transcribe\_audio

from your\_qa\_module import generate\_answer

1. **Upload and display videos:** Add functionalities for users to upload videos, which will then be processed:

video\_file = st.file\_uploader("Upload a video", type=["mp4", "mov", "avi", "mkv"])

if video\_file is not None:

video\_path = os.path.join("temp\_dir", video\_file.name)

with open(video\_path, "wb") as f:

f.write(video\_file.getbuffer())

st.video(video\_path)

**Step 3: Integrating Video and Audio Processing**

1. **Video processing:** Integrate the module that extracts and processes the video's audio:

if video\_file is not None:

audio\_path = process\_video(video\_path)

transcript = transcribe\_audio(audio\_path)

st.write("Transcription:", transcript)

**Step 4: Question and Answer System**

1. **Question interface:** Allow users to input questions and provide answers based on the transcription:

question = st.text\_input("Ask a question about the video:")

if st.button("Get Answer"):

if question and transcript:

answer = generate\_answer(question, transcript)

st.write("Answer:", answer)

else:

st.write("Please upload a video and wait for the transcription.")

**Step 5: Execution and Testing**

1. **Run the application:** Run your Streamlit application from the command line:

sh

Copiar código

streamlit run app.py

This will automatically open the application in your default web browser. 2. **User testing:** Conduct tests to ensure all components are functioning correctly: video upload, transcription, and the question-and-answer system. 3. **Iteration and improvement:** Based on user feedback, make adjustments and improve the interface and functionalities.

**Step 6: Deployment**

1. **Deploy the application:** Consider deploying the application on a server or using platforms like Heroku, AWS, or Google Cloud to allow public access. Following these steps will create an interactive web application that facilitates users to upload videos, view them, obtain transcriptions, and ask questions, significantly improving the user experience with your video processing and question-and-answer system.

#### Conclusions

In this work, the development of a multimodal application using LangChain and the OpenAI API to process audiovisual content and provide intelligent interaction with users has been presented. The application, designed to download, transcribe, and analyze audio from YouTube videos, demonstrates the capability of artificial intelligence technologies to improve accessibility and efficiency in managing multimedia content.

The literature review showed that multimodal applications and natural language processing (NLP) have advanced considerably in the past decade. Key studies in areas such as multimodal deep learning, voice transcription, and real-time object detection have laid the foundations for complex and efficient applications such as the one presented in this document. The evolution of AI APIs, especially those developed by OpenAI, has facilitated the integration of advanced models into practical applications, demonstrating their potential to transform various industries.

The detailed methodology of the application includes system architecture, the process of downloading and extracting audio, content analysis using AI, constructing a question-and-answer system, and designing an interactive user interface. Each of these components has been implemented using modern tools and libraries, such as yt\_dlp for video download, pydub for audio manipulation, and FAISS and GPT-4 for search and response generation.

Several potential use cases from business and technological perspectives have been identified. In education, the application can enhance accessibility and student engagement. In digital marketing, it can facilitate competitive content analysis and campaign optimization. In customer support, it can provide automatic and precise responses, improving user satisfaction and reducing the load on support teams. Additionally, future integrations that could expand the application's capabilities, such as integration with augmented and virtual reality platforms, and improved multilingual support, have been explored.

The results obtained so far show that the application is effective in transcribing and analyzing audiovisual content and generating contextually relevant answers. However, areas for future improvements, such as implementing more advanced voice recognition models and expanding video analysis, have been identified. Improvements to the user interface and the creation of a public API for external developers are also proposed.

In conclusion, this multimodal application represents a significant advancement in using artificial intelligence technologies for processing and analyzing multimedia content. Its design and implementation demonstrate how modern tools can be used to create practical and effective solutions in various contexts. The proposed future improvements and expansions have the potential to further increase its utility and applicability, consolidating it as a valuable tool in the field of applied artificial intelligence.

#### References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). VQA: Visual Question Answering. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
2. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal Deep Learning. Proceedings of the 28th International Conference on Machine Learning (ICML).
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.
5. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
6. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). DeepSpeech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.