COACTIVE

Unleashing **Visual Content**

A guide to multimodal AI infrastructure

Whitepaper

Table of Contents

Executive Summary

As video, image, and audio content explodes in volume, companies are finding themselves with massive and rapidly growing libraries of multimodal content, just waiting to be tapped. However, discovering the perfect content in the archive or making use of specific content in applications – for things like ad targeting, sponsorship validation, highlight reels, or content moderation– is uniquely challenging.

Traditional visual content discovery tools are very limited, since they often rely on manual tagging efforts and restrictive keyword search technology. This high-cost approach doesn't scale to cover the breadth and depth of content that organizations want to leverage. There are often huge labeling gaps, or shallow, sparse labeling that doesn't cover the detail and range needed to optimize content use. As a result, traditional keyword search tools are ineffective for multimodal content discovery, and human-powered tagging efforts fall far short of meeting metadata needs for readying content for application workflows and analytics.

At Coactive, we've built a multimodal application platform. Offered as a managed service, the platform makes it easy to do video and image content discovery at scale, while readying multimodal content for use in business applications through automated metadata tagging and analytics. It fits easily into your existing workflows and systems, while allowing you to leverage the best foundation models and leading cloud platforms.

By partnering with Coactive, developers can focus on building applications that address high-value business use cases, without worrying about configuring AI infrastructure.

We realize that many organizations have talented technical teams who are often tempted to build things on their own. In recognition of that, this guide walks technical leaders and contributors through the complex challenges these teams would need to crack at each level of the multimodal AI infrastructure stack. The goal is to help teams understand what developers face when they choose to go it alone. This guide also explains how Coactive can help your teams leapfrog these challenges, so that your developers are focused on quickly capturing business value, not the complexities of AI infrastructure.

Content moderation

A massive fan experience and content site used multimodal AI to automate the assessment and removal of harmful visual content in seconds, instead of the 24 hours previously required.

Cataloguing user generated content

A leading customer experience platform used multimodal AI to accelerate tagging and metadata creation for user generated content (UGC), benefiting retailers with large product catalogs.

Unleashing existing content

A news provider with over 125 years of visual content used multimodal AI to make its content more easily discoverable, opening up a significant new revenue stream from existing assets.

Build vs. Buy

The Multimodal Application Platform

Modern enterprises need systems that understand complex content in the intuitive way that humans do. While this is easy to state, achieving it requires cutting edge technology and serious technical skill. It requires the simultaneous processing of multiple information types and the ability to efficiently adapt to the rapid pace of new requirements and technologies. Coactive's Multimodal Application Platform (MAP) helps content and development teams leverage multimodal AI today, while ensuring that systems and workflows will continue to be effective in the future.

Should you build it yourself?

Building multimodal AI infrastructure and integrating it into your applications and solutions requires managing complex infrastructure layers and understanding their interdependencies. You need the right tools, skills, and architectural expertise to handle preprocessing workflows for different content types, foundation model integration, storage optimization, and sophisticated API architectures. Each component presents its own technical challenges and tradeoffs, from optimizing compute resources to ensuring efficient data flow.

> "With Coactive, Fandom automated content moderation and visual search across millions of assets — saving time and improving accuracy."
>
> **Florent Blachot**
> VP Data & Engineering at Fandom

Or should you leverage a pre-built platform like Coactive?

The Coactive multimodal application platform offers a strategically smart approach that minimizes infrastructure complexities. You can easily select your preferred foundation models for your use case and leverage your preferred cloud providers, all while using Coactive for the heavy lifting of configuring and maintaining multimodal AI infrastructure. The platform addresses critical components that would otherwise require significant investments in developer time, specialized expertise, and ongoing system optimization.

Coactive provides ready-to-use infrastructure that integrates seamlessly with existing workflows and technologies, so that developers can focus on creating applications that solve high-value use cases. This approach:

- Accelerates development cycles
- Reduces technical debt
- Offers the fastest path to realizing business value
- Maintains flexibility for future growth

What does it mean to build this on your own?

The following sections will go into detail about what the elements of a multimodal application stack are. Read on to find out what it means to take on these challenges yourself.

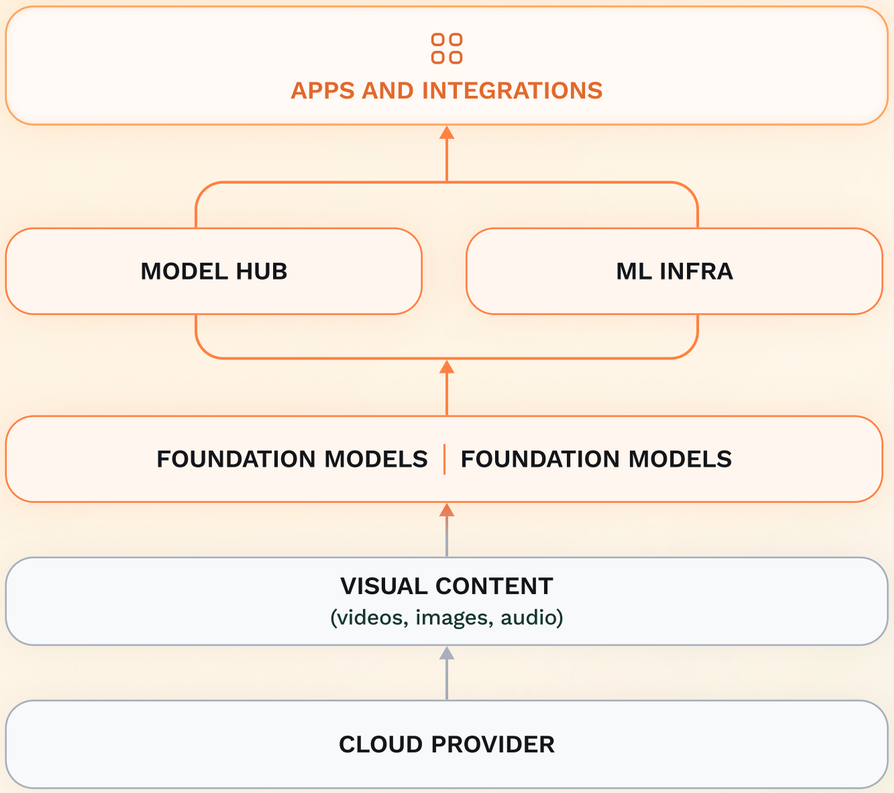
See the diagram on page 5

Build vs. Buy

The Multimodal Application Platform

Build It Yourself

Developers have to build and maintain complex infrastructure



COACTIVE

Developers focus on building effective apps

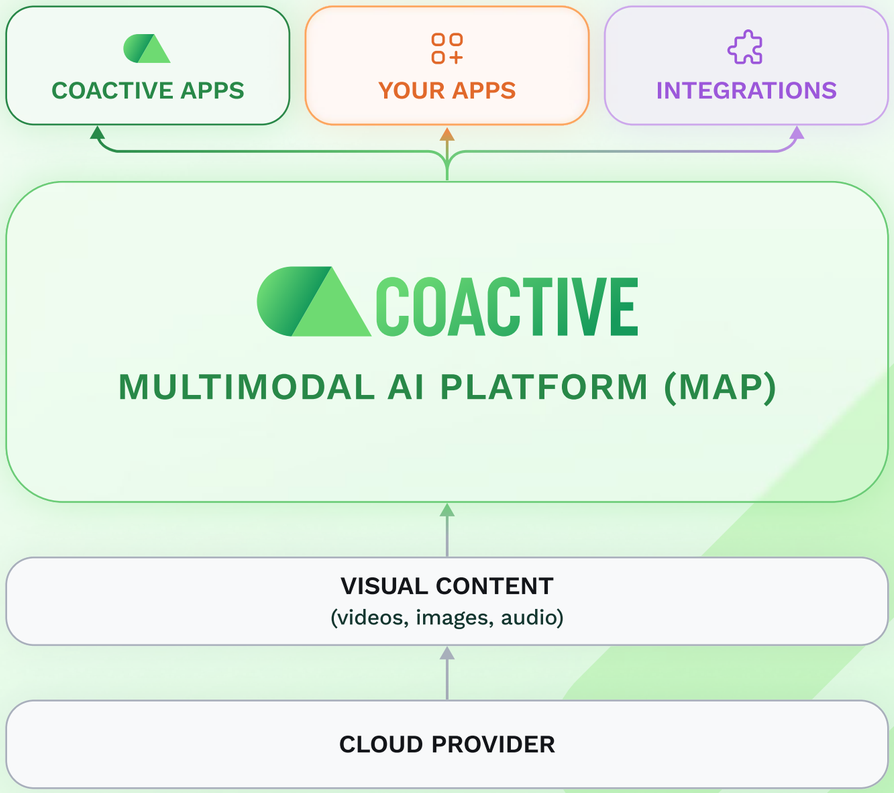


Figure 1: Simplified view of the core capabilities that developers would have to build and maintain if they built a multimodal application platform on their own, versus selecting a provider like Coactive.

The challenges of building multimodal AI infrastructure

At a high level, building a multimodal AI platform for visual content discovery and use involves three things:

- managing infrastructure
- implementing preprocessing workflows for different content types, and
- deploying the right foundation for each major use case.

5 key challenges of building a multimodal AI platform

Given the many moving parts and numerous choices available for building your platform, you need a firm understanding of the key components and how they work, both together and independently.

The considerations associated with these components are nontrivial. No decision should be taken lightly; each one is critical to success. For example, storing full-resolution images and high-dimensional embeddings without optimization can cause the size of a database to balloon, affecting search response times. This problem can cascade into others like application performance and API calls. Getting everything right is crucial, since corrections can become increasingly complex and costly over time.

In the following chapters, we explore the challenges and opportunities of each component to help you understand key decisions you'll need to make, if you decide to build your own platform.

COACTIVE

Preprocessing

Converts images and videos into a format that can be ingested, embedded in a vector database, and optimized for GPU and hardware.

Foundation model

A large machine learning model trained on a massive, diverse dataset, making it adaptable to a wide range of tasks through fine-tuning.

Storage

Where visual assets, embeddings, metadata, and models are stored. Storage decisions impact the speed, efficiency, and cost-effectiveness of the platform.

API layer

Enables users to access different AI-powered services, such as: search, metadata tagging, or analytics.

App integration and UI

Delivers a user experience that makes it easy to interact with results, such as enabling semantic search for video archives.

CHALLENGE 1:

Preprocessing video and image assets

Visual content comes in all shapes and sizes as raw image and video assets. These assets need to be transformed into AI-ready data. Preprocessing encodes them into a format that models can ingest.

Preprocessing is where files are converted into standardized digital formats. It determines how basic compression and decompression are handled, and how color spaces and bit depths are kept consistent. Applying basic technical tags and extracting existing metadata tags also takes place during preprocessing, along with simple, rules-based content flagging. Preprocessing requires that you manage technical format compatibility, create tensor representations for model input, and determine the right frame rates and resolution.

In general, preprocessing is the most straightforward component of building multimodal AI infrastructure. However, preprocessing becomes more complex if it needs to be done quickly to meet service level agreements (SLAs) and if the input types are not restricted. While you could use standardized packages to decode videos, this can become very challenging and costly in highly parallel ecosystems.

Therefore, there are important decisions to make during preprocessing related to ingestion, storage, compute, and optimization.

Ingestion

There are multiple methods and tools for ingesting images, videos, and other data that can be used in preprocessing. A popular practice is building standardized pipelines or frameworks for ingestion. However, it can take several months of development time, which is followed by:

- extensive testing
- edge case discovery (often multiple instances)
- performance evaluation, and
- documentation.

Ingestion also requires deciding how to run your machine learning models—in-memory or through model serving. A key part of this involves weighing the trade-offs between speed and cost.

Challenge 1:

Preprocessing video and image assets

Determining the storage strategy

Another part of preprocessing is deciding how to store the images and videos that you will be ingesting for your in-house project. Storage architecture choices profoundly affect workflow. Your decisions must take into account:

- accessibility
- processing efficiency, and
- storage costs.

For example, you need to consider whether you should store videos as individual files or collect them into serialized output stored in a large binary file with multiple videos.

Compute architecture and transformation

Preprocessing requires significant computational resources for converting images and videos into tensors, which are numerical matrices that AI models can process. Transforming visual content into these multi-dimensional arrays of data that organize and represent information requires thoughtful, measured decisions about compute architecture.

One example involves deciding whether scaling should be horizontal, using multiple smaller machines, or vertical, utilizing fewer, more powerful machines. For processing, you need to weigh the pros and cons of CPU versus GPU processing based on workload characteristics and cost. It's also important to consider the memory and processing requirements for all of the different content types.

Content-specific optimization

Different content types require different approaches to preprocessing. Your solution needs to be able to handle all of them. For example:

1. Video preprocessing involves separating audio into an independent signal stream. Plus, you must make crucial video sampling decisions, such as one frame per second or using full frame rates.
2. Video conference preprocessing requires specific sampling strategies in order to capture meaningful frames.
3. Image processing may require resolution adjustment, particularly for high- resolution content.

Again, these optimizations need to balance quality requirements against desired processing speed and resource utilization.

The choices made during preprocessing cascade throughout the entire system. Incorrect decisions can create bottlenecks, increase costs, or reduce the accuracy of AI analysis. For example, JPEG compression can produce a massive shift in generated embeddings, even though it looks the same as the original. However, careful planning and implementation of preprocessing can have a positive effect downstream—increasing and maximizing AI performance.

Optimizing for efficiency

To fine tune your build process, remember that smart preprocessing decisions significantly impact system efficiency. For example, if you are processing a 4K video, it's better to pre- process it into a 480p proxy first

CHALLENGE 2:

Choosing and using foundation models

Despite their power and promise, foundation models have limitations. Each has its own nuances, and determining the best foundation model to use for particular use cases, such as content discovery or automated cutsheets, is hard.

Foundation models generate low-dimensional, semantically rich representations of multimodal data. These representations can be used for downstream tasks such as semantic search, similarity search, zero-shot learning, cross-modal retrieval, and more. Foundation models also:

- Output vector embeddings from visual content or tensors.
- Apply deep learning to understand content semantics.
- Assist with generating metadata to make sense of visual content.
- Enable advanced search and comparison capabilities.

The practical limitations of foundation models for multimodal data can often be obscured by the creators and researchers of those models. Although foundation models can perform well on academic tests, these tests are markedly different from real-world, industry use cases. Many of these foundation models lack flexibility and accuracy—at best, they can deliver search and discovery accuracy of 75 percent. However, that accuracy can also be as low as 40 percent.[1] For many organizations, 40-75% accuracy is inadequate. This means that foundation models alone cannot meet industry requirements.

How to address the use of foundation models in your multimodal application platform requires a well-thought-out, comprehensive strategy. Here are some key things to consider.

Rapidly changing models

Multiple foundation models can generate embeddings, and model providers are rapidly developing and announcing new capabilities. If you build functionality with the model that appears to be the best today, it could be out of date in the next three months. **Therefore, your team needs to understand how to add knowledge to the model without retraining – or you need a strategy for swapping out models.**

[1] Coactive, Introducing the Multimodal Application Platform (MAP), 2024.

Challenge 2:

Choosing and using foundation models

Adding knowledge without retraining

When it comes to adding knowledge, there are several options that enable you to keep models up to date without requiring a full retraining process on a massive dataset. Do you use prompt engineering, model adaptation through embedding adapters, or model fine-tuning?

Swapping foundation models

To offer the ability to swap models, you will need to implement adapters for each model to conform to the interface and build a registry system that manages configurations and versioning. You must also develop a dynamic loading system and flexible index management to handle model transitions without disrupting search capabilities.

Maximizing flexibility, now and in the future

To get the most out of foundation models, consider a solution that can support many of them. The ability to change between models gives you access to the latest versions, to models that specialize in identifying people or objects, and to those that can relate things to one another. As a result, you can stay connected to the best-performing systems and the latest research.

Adapting to your specific domain

Foundation models are one-size-fits-all. This means that they are not necessarily tailored to the domain-specific concepts and terminology of any particular industry. Further, it's likely that they have also not been exposed to nor used your specific organization's terminology and visuals, because they are trained on the public internet. Therefore, if you are developing an in-house solution, you will have to bridge the performance gap between your selected foundation model's native accuracy and the margin of error you find acceptable.

To increase accuracy, you can configure your data systems and machine learning workflows. For example, implementing confidence scoring with automatic approval of high-confidence results and using validated results can move the needle closer to 100 percent.1 However, you'll need to be sure to budget enough time and resources. It can take months to deploy these infrastructure and machine learning workflows into production to improve accuracy.

Closing the distance from 75% accuracy to 100% is the most challenging and critical in an enterprise context.

Once you have chosen a foundation model and adjusted it for accuracy, you're ready to output embeddings. However, you need a place to store them.

[1] Coactive, Introducing the Multimodal Application Platform (MAP), 2024.

CHALLENGE 3:

Planning storage for optimal retrieval

Selecting the right storage and databases for the components of your solution is a pivotal step in ensuring a seamless integration between asset ingestion and extraction. The decisions you make impact your ability to quickly, accurately, and cost effectively find and retrieve visual content through your application while continuously improving the overall system.

COACTIVE

Key decisions for faster vector computations

The vector embeddings generated by the foundation models need to be stored in a vector database. Where they are stored in proximity to one another is determined by the similarity between the assets. Therefore, you need to decide how you will compute the distance between vectors to identify a group of related assets. Measuring the exact distance is the most accurate method; however, this brute force method takes an exorbitant amount of time when dealing with millions of assets.

An index applies a mathematical model to approximate distances between vectors in a fraction of the time compared to exact distance. Different indexes produce results with different tradeoffs for precision, recall, and accuracy. By selecting and properly configuring the right index for your vector database, you can significantly reduce the time required to deliver results. Some popular indexes include Hierarchical Navigable Small Worlds (HNSW), Inverted File (IVF), and Locality Sensitive Hashing (LSH).

Optimizing downstream access

Planning a decoupled and flexible system for storage enables you to optimize downstream access patterns and usage. Consider the elasticity of each type of storage to ensure it will scale the way you need it to in the future.

Challenge 3:

Planning storage for optimal retrieval

Refining results with metadata

Search relies on the critical assumption that a query vector is close to candidate vectors with similar attributes. However, when the definition of an attribute is more subjective, the act of tagging (also known as classification) can be used to create metadata that emphasizes similarities between vector embeddings.

This metadata needs to be associated with specific embeddings so that the related assets can be identified. While some vector databases will store tags, using a vector database for this purpose isn't always the right decision, because it can slow the performance of the vector database and lead to a backlog of queries, especially in cases when you need to store other artifacts and link assets to your embeddings.

Establishing databases for asset and artifact mapping

right video, image, or audio asset. Many developers use a tabular format in a PostgreSQL or NoSQL database to store embeddings alongside links to the actual asset and other artifacts like metadata. Additionally, if you plan to provide a user interface (UI) for review, you'll also need to create and store preview images that are linked to the embeddings.

Setting up a separate database for embeddings and artifacts enables faster metadata propagation (or tagging and classification) than relying solely on the vector database.

Setting up storage analytics

After you select the optimal storage solutions for different components, you can run analytics on your data. This helps you keep track of what visual assets you have and provides you with the opportunity to enrich metadata and classification tags for improved content curation and discovery.

CHALLENGE 4:

Architecting the API layer for classification, search, and analytics

Once you make your preprocessing, foundation model, and storage decisions, you need an API layer that you can use for classification, search, and analytics.

This layer enables your backend to communicate with your applications and users. It also coordinates interactions between the storage components you've selected, such as vector embeddings, relational data, NoSQL documents, raw assets, and model artifacts. Therefore, it requires careful attention to interface design, error handling, and system reliability.

Production requirements of the API layer

To handle production workloads, the API layer should support request queuing for compute- intensive operations, strategic caching for frequent queries, and streaming responses for large result sets. For security, it should include API key authentication, request signing, and granular access controls to manage component and operation access.

Powering SQL analytics

With a SQL interface, users can submit SQL queries to access analytical data like probability, precision, and recall metrics on the assets. Using this interface, users can write very complex queries to perform analytics.

Challenge 4:

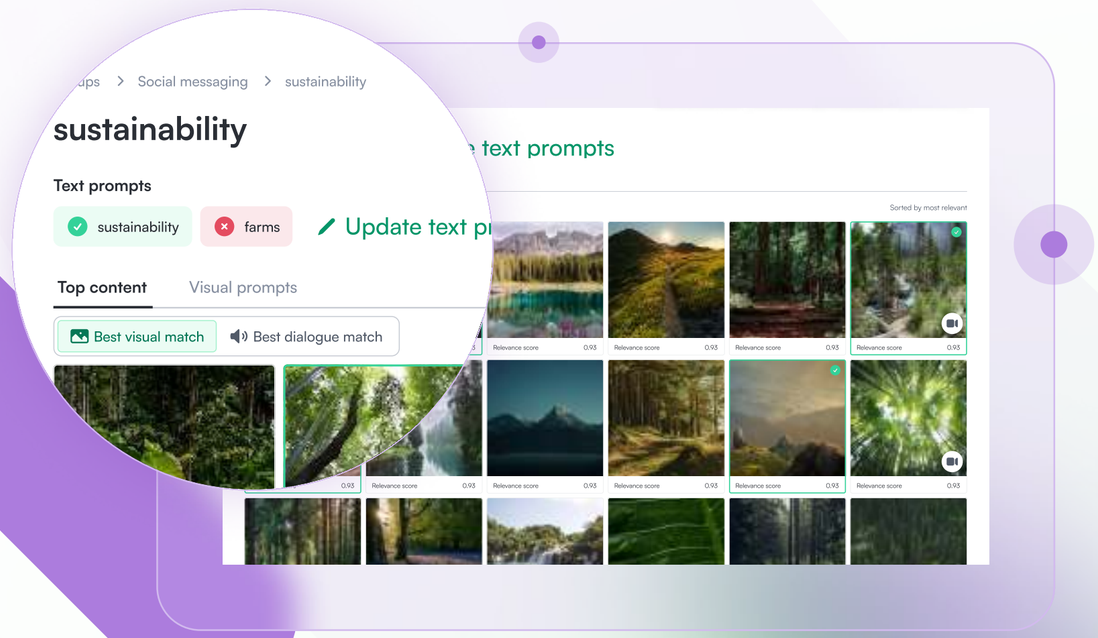
Architecting the API layer for classification, search, and analytics

API layer architecture

The API layer's architecture consists of three primary endpoint groups: classification, search, and analytics. Each endpoint group requires its own complex integration logic while depending on shared infrastructure for authentication, rate limiting, and request validation.

Classification APIs

Classification APIs combine raw assets with model inference to generate predictions and metadata tags. These APIs handle the full pipeline from media upload through preprocessing and model inference to result formatting. Users can submit an asset and get the output in real- time. Scalability, flexibility, and performance are critical here, because the layer must support the submission of thousands of images for classification in real time.



Search APIs

Search APIs merge vector similarity with traditional database queries. Search endpoints need to execute parallel operations, combining vector similarity lookups with metadata filtering. They should also support query parameters that can target specific asset types or metadata fields.

Analytics APIs

Analytics APIs aggregate data across storage systems and translate high-level requests into optimized, cross-database queries that can span multiple storage systems. Native integrations with analytical systems enable users to access the data and perform advanced analytics.

For example, you can understand critical business questions like:

1. Which sports content gets the highest clicks or likes? Cross-reference engagement metadata with a semantic analysis of the image and video content can advance a company's data-driven content strategy.
2. Which customers are driving metadata creation vs search? And what types of search queries? You can find user access patterns that you will need to analyze to improve the platform as a whole.
3. What gets the highest clickthrough on ads - image, video, or audio?
4. Which user types are associated with posting high-risk content?

CHALLENGE 5:
Integrating multimodal AI into your application and UI

The final component you need to build for an effective multimodal AI application platform is the actual application and its UI. You need to design and build a UI that enables users to upload and enrich content, use search and filter to discover it, and create dashboards and reports to monitor and understand it.

The UI needs to be intuitive and easy to use, while delivering the right visual content quickly. Plus, it should seamlessly connect to what you've built—databases, foundation models, API layers, and more—to ensure the best experience. Therefore, each aspect of building the UI and integrating it with the backend takes careful consideration of user mental models and expectations.

Key interface components

Front-end implementation centers on several critical elements:

- **Media upload zones** that provide clear progress indicators and batch processing status.
- **Search interfaces** that balance power with simplicity, incorporating filters, facets, and visual previews that help users quickly find what they need.
- **Content preview and player** components that surface AI-generated metadata seamlessly.
- **Analytics dashboards** that translate raw data into meaningful visualizations of content insights and usage patterns.

Administrative interfaces round out the system, giving technical users the control they need over models and processing pipelines.

Select pre-built components to save time

Use pre-built, production-ready components that handle upload processing, content discovery, interaction, and analytics. This lets you focus on customizing the user experience for your specific needs rather than architecting AI integration from scratch.

Challenge 5:

Integrating multimodal AI into your application and UI

Technical considerations

When developing the UI layer, you'll need to plan how to integrate it with the backend calls by answering questions such as:

- How will storage state management efficiently handle large result sets, while responsive loading states keep users informed during API calls?
- How can you make sure that error handling and recovery gracefully manage failures?
- What is your caching strategy for faster rendering?
- How will you ensure that real-time updates smoothly communicate the status of long-running processes?

The goal is to abstract away the technical complexity built in the API layer and present users with clear, purposeful interfaces that help them achieve their goals without needing to understand the underlying ML infrastructure.

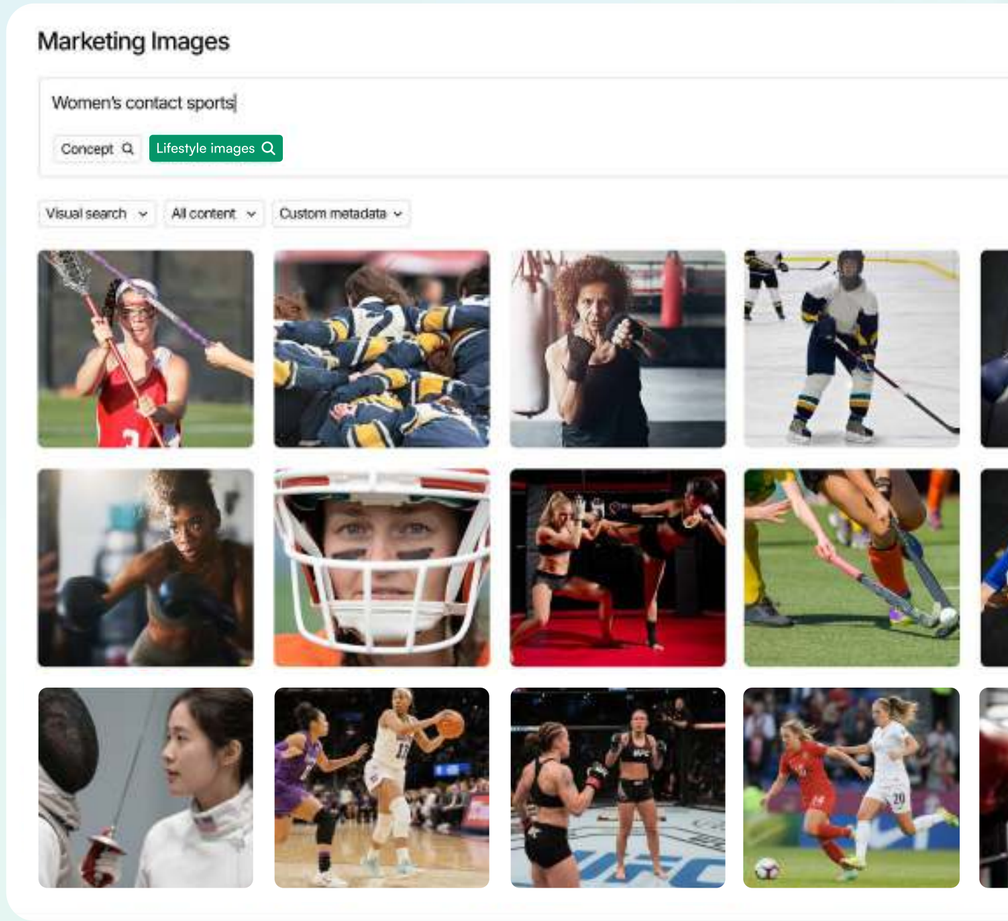


Figure 2: An example of an AI-powered image search that could be used in a targeted advertising workflow or application.

Do you really want to focus on core AI infrastructure?

The multimodal application that you are building will power your business. Visual content enrichment, discovery, and analytics functionality is just part of its development. Building this core functionality in-house might seem to be the better investment, but in many cases, it can pull developer focus away from delivering the final application and possibly slow your time to market.

Build with confidence and speed.
Build with Coactive.

As we've covered in this whitepaper, there are 5 major challenges to building your own multimodal application platform:

- Preprocessing video and image assets
- Choosing and using foundation models
- Planning storage for optimal retrieval
- Architecting the API layer for classification, search, and analytics
- Integrating multimodal AI into your application and UI

Each of these is a significant challenge to initially architect and then to maintain, especially at the rapid pace of change in AI.

COACTIVE

While building such a platform yourself may be tempting, many development teams prefer to stay focused on where they add the most business value: the application itself. If you're interested in spending more time building applications and minimal time configuring AI/ML infrastructure, consider using a platform that provides ready-to-use content enrichment, discovery, and analytics.

Coactive provides a robust, easy to use Multimodal Application Platform that minimizes the effort and risk of building applications that rely on visual content - such as solutions for ad targeting, brand safety/content moderation, content discovery, or automated highlights.

Coactive is used by major global media and retail brands, and has been named to the Forbes AI 50 twice.

If you'd like to learn more, contact us at coactive.ai.