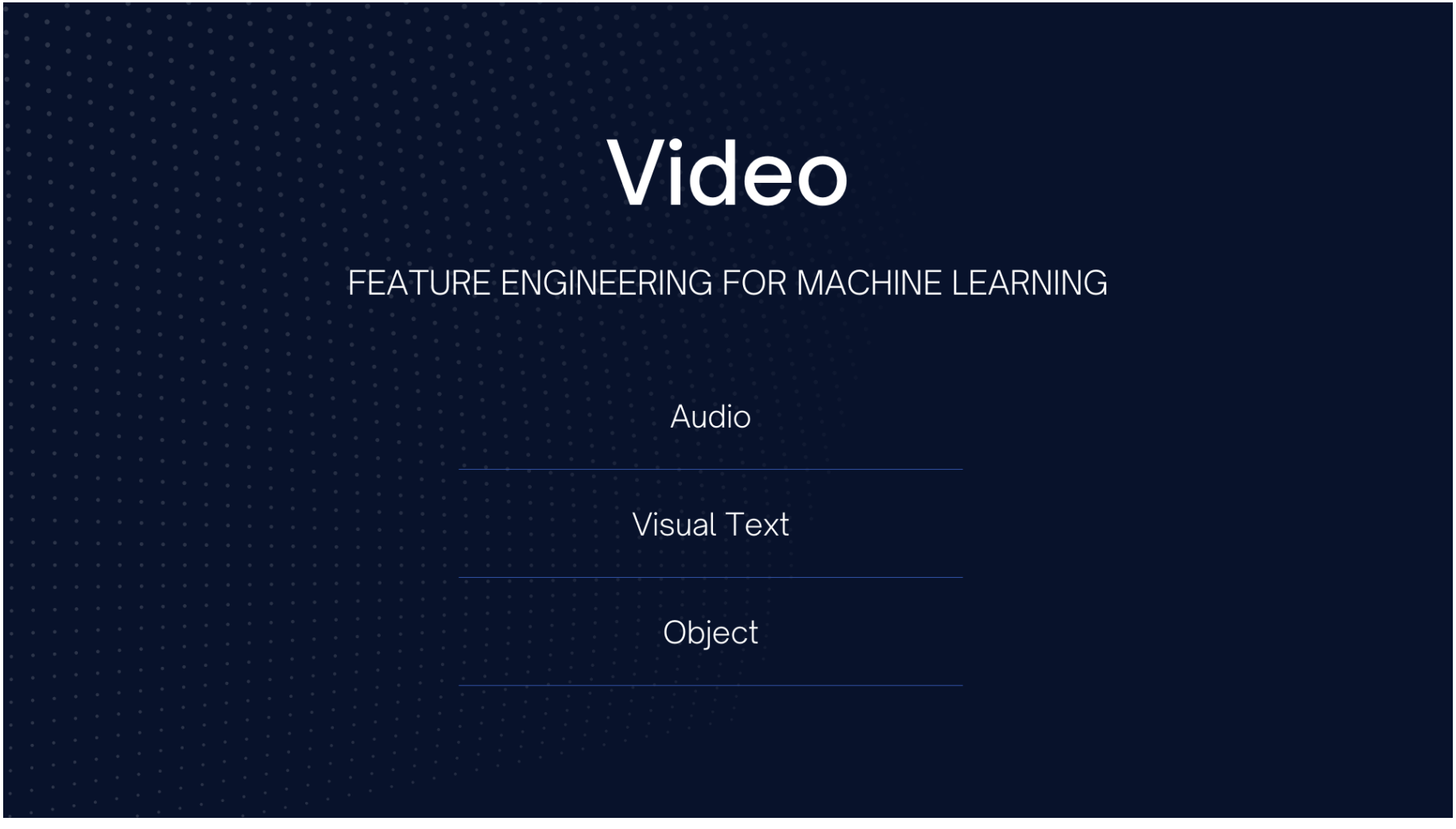
I am excited to share with you all what I've developed–I created a video library powered by machine learning.



First let's talk about the data type video. What features can we extract for analysis and machine learning?
A video has both audio and visual elements. So we can extract: (1) audio, (2) visual text, and (3) objects in video frames.
These are features that we should be able to use for searching and identifying videos, but that is not quite the case, yet.
For our use case, take the most popular video sharing platform right now.

TikTok is a mobile app where users can scroll through hours of video content and save videos that they like to their profile.
However, TikTok lacks the functionality to search for any of the videos saved in the users archive of “liked” videos.
If you are one of its users, you've most likely experienced going through every single video you’ve ever liked to find a clip. Until today!

Summary

DEVELOP A VIDEO LIBRARY BUILT WITH A SEARCH ENGINE THAT PROCESSES AUDIO VISUAL TEXT AND OBJECT IN THE VIDEO TO RETURN ACCURATE RESULTS

To summarize, I've developed a video library built with an NLP-based search engine that processes (1) audio, (2) visual text, and (3) objects in the videos, using multiple machine learning models to return accurate results.

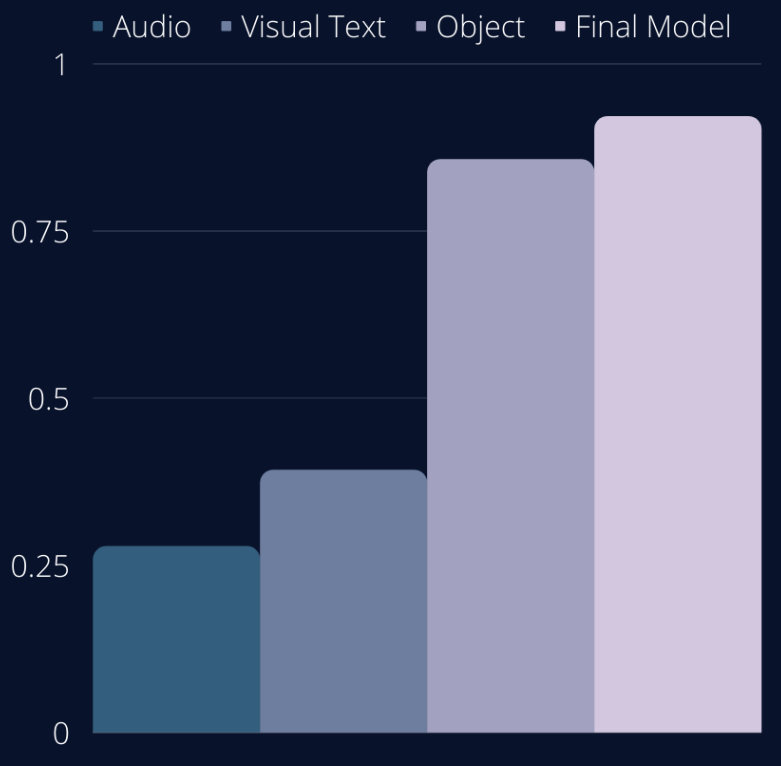
Data and Methodology

I used multimedia data from TikTok, analyzing over 2000 videos and labeling almost 150 of them myself.
Then I performed methods 1, 2, and 3–extracting the audio, text, and objects using these tools to be discussed shortly.
At the end, I used Natural Language Processing to build the search engine.

Data Modeling

EVALUATING ACCURACY

Predicting Video Category

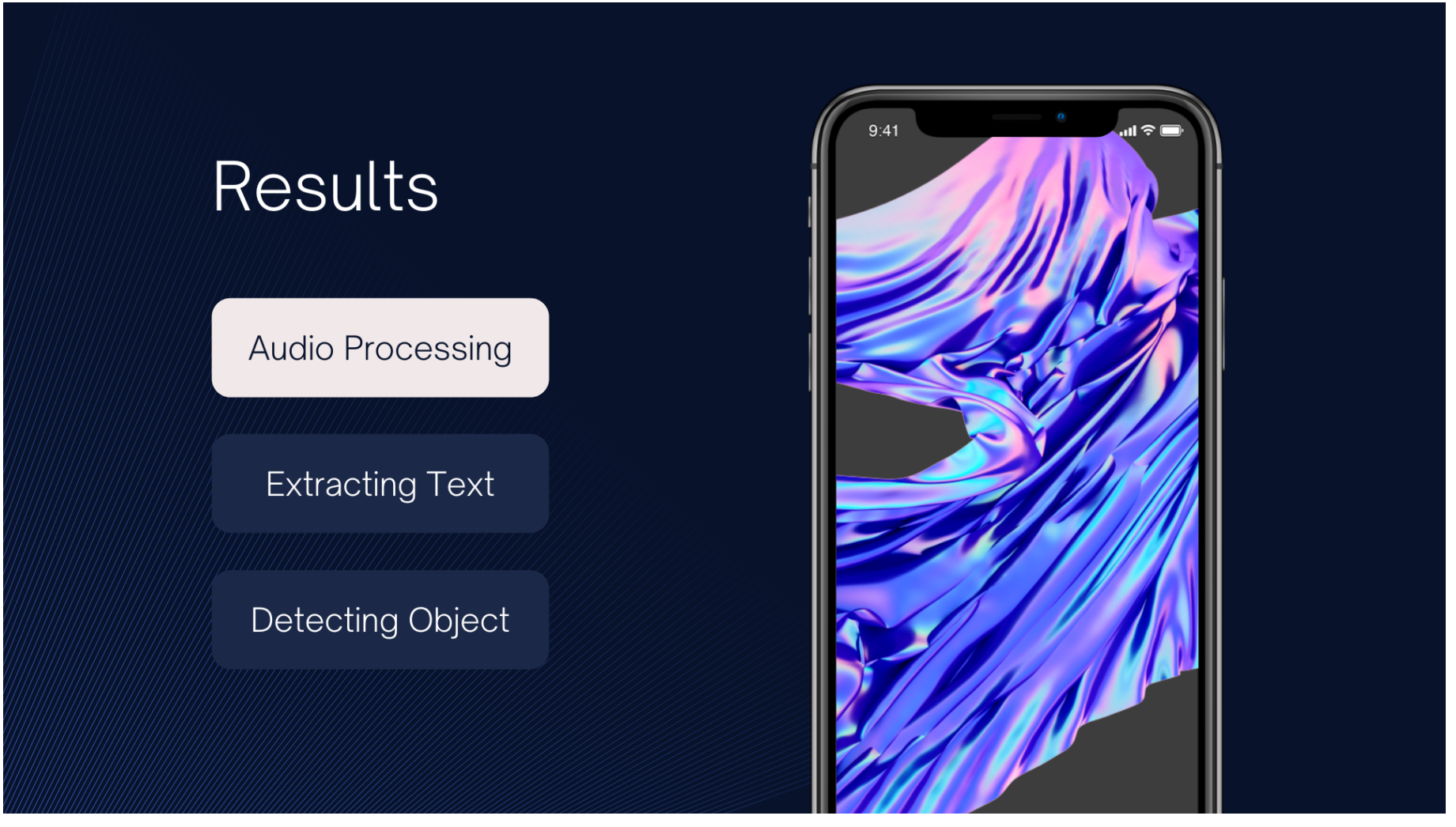


I used pre-trained models to classify videos and predict their categories.
The first model extracted audio with an accuracy of over 25%. Second extracted visual text with a higher score of 40%.
Third detected objects with an accuracy score of 85%. Compared to the two features, the object classes matched the video categories best.
And the final model used all of the features and increased the baseline by almost 3 quarters with 92% accuracy.

Results

- Audio Processing
- Extracting Text
- Detecting Object

Let’s walk through the results of each model one by one.



First, audio processing.

Video to Speech

Pydub

Audio file conversion

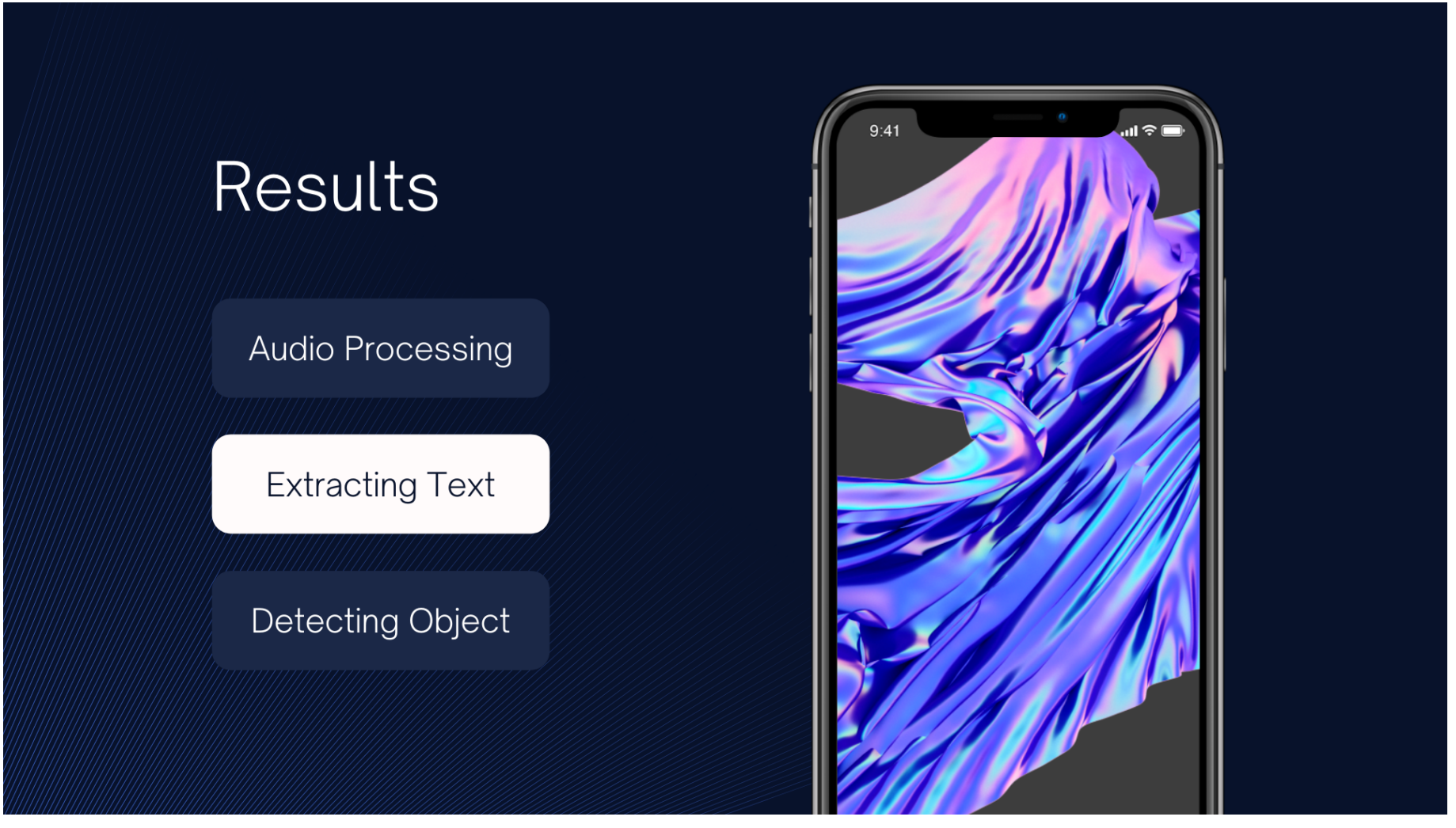
500 ms

Audio segmentation

Transcription

Google Speech Recognition

I used the Python library Pydub to convert the video format to wav files,
which I then segmented or split into chunks where there is silence for 500 milliseconds or more.
I passed these segmented chunks of audio on to Google Speech Recognition API that returned the transcription in a text format.



Second, extracting text.

I used the Python wrapper Pytesseract to perform optical character recognition OCR that transformed the video frames to printed text.

Video to Text

```
XeThis NY restaurant will “make you
    feel <like you’re inItaly! {

\\:@ Unique and diverse Italian |menu!
    %Private & romantic dininga

Open 7 days a week withbrunch options
    on {Saturdays & Sundays!— “
Make your reservationasap! =ro mm tT
```

This NY restaurant will make you feel like you're in Italy

Unique and diverse Italian menu
Private & romantic dining

Open 7 days a week with brunch options on Saturdays & Sundays
Make your reservations asap

The visual text extracted still needed text processing so I removed characters captured that were not part of any word and I segmented consecutive words that did not have spaces between them.
This is the result of an example video showing informative details.

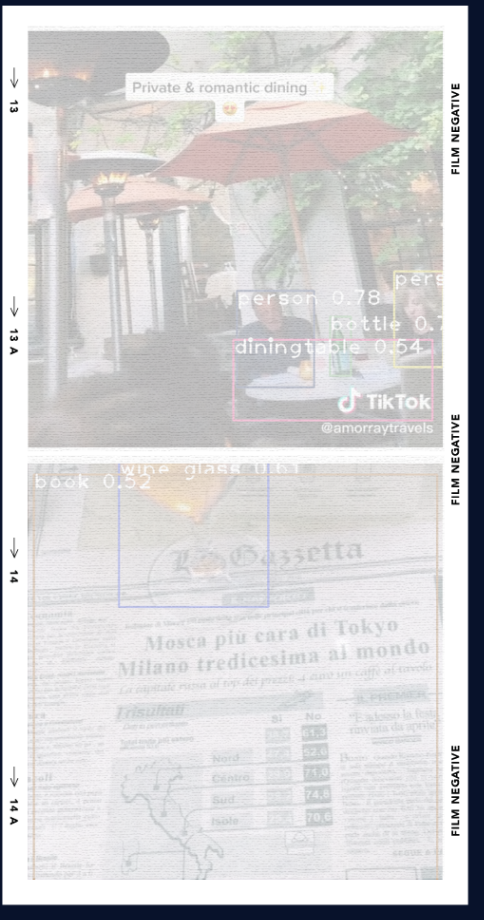
Results

Audio Processing

Extracting Text

Detecting Object

Finally, detecting objects.
I used the object detection algorithm YOLO via the Open Computer Vision library to detect objects in every nth frame of the video.



book

bottle

chair

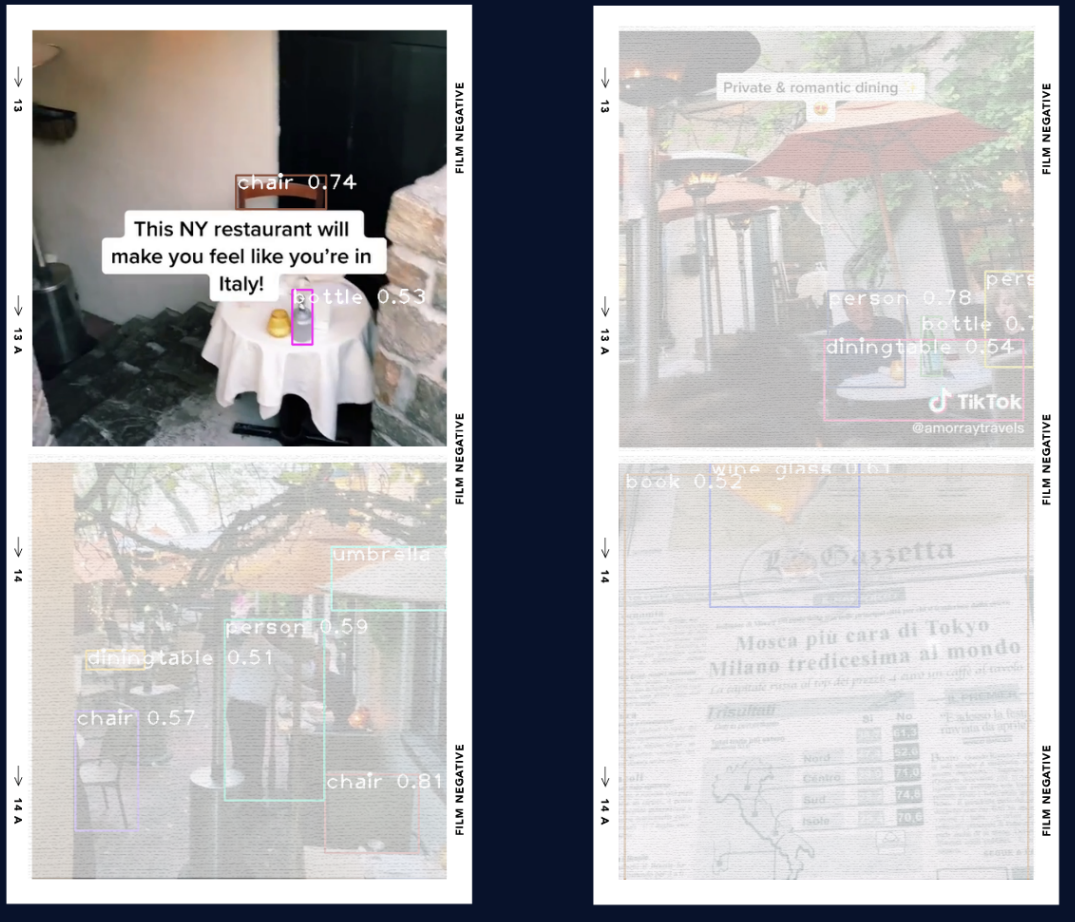
dining table

person

umbrella

wine glass

To demonstrate, these are four sample frames from the video.



book

bottle

chair

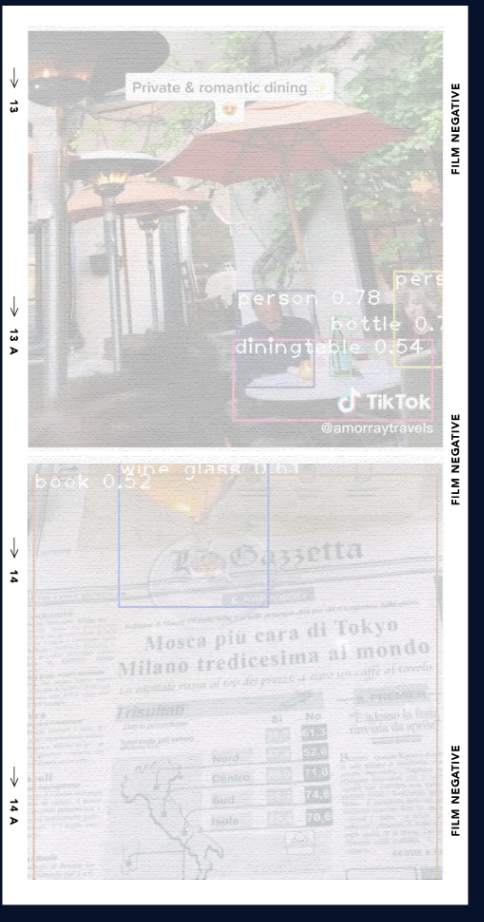
dining table

person

umbrella

wine glass

In the first frame, a chair and a bottle were detected.



- book
- bottle
- chair
- dining table
- person
- umbrella
- wine glass

In the second frame, two chairs, a dining table, a person, and an umbrella were detected.

- book
- bottle
- chair
- dining table
- person
- umbrella
- wine glass

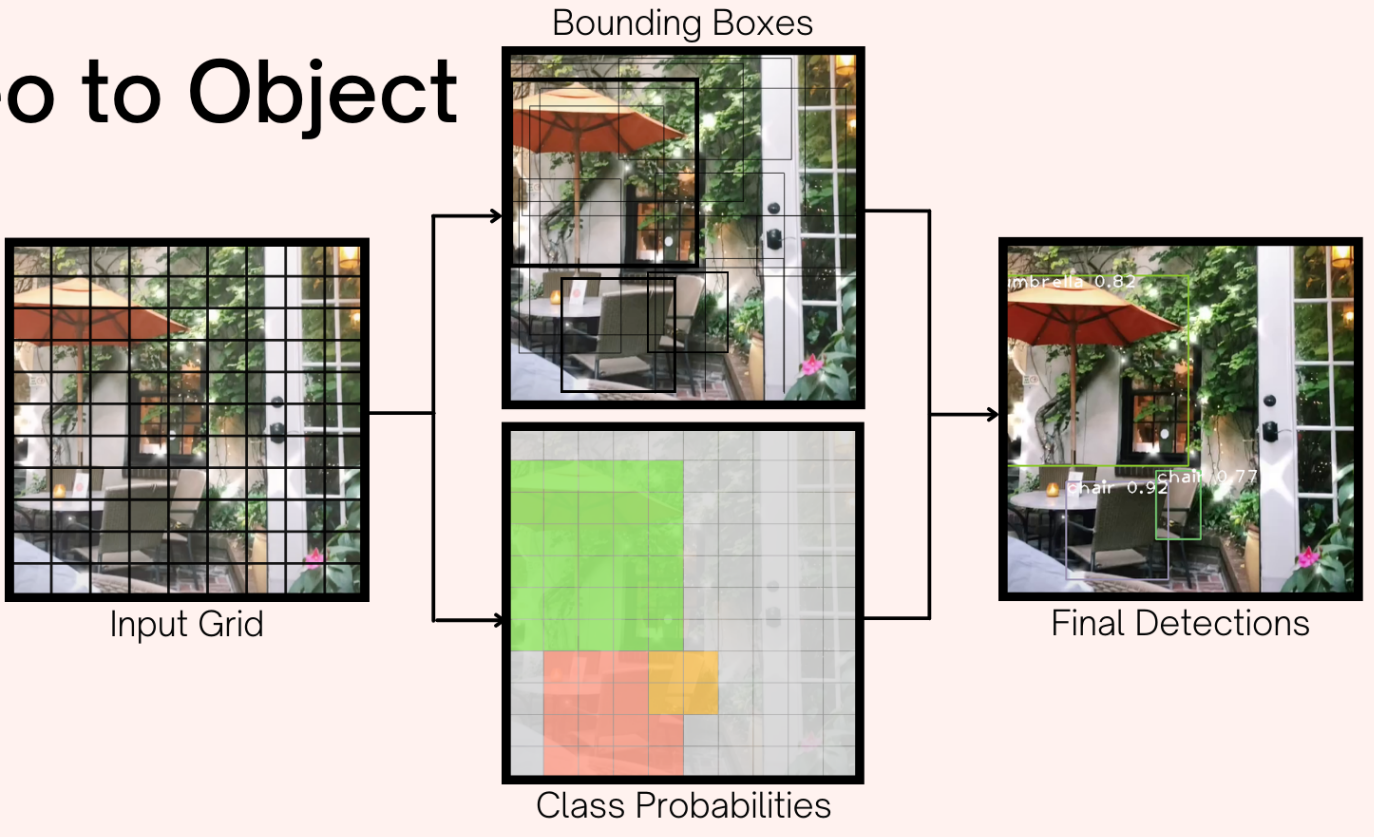
In the third frame, a bottle, a dining table, and two persons were detected.

And in the fourth frame, a book and a wine glass were detected, but the page of a newspaper was mislabeled as a book.

- book
- bottle
- chair
- dining table
- person
- umbrella
- wine glass

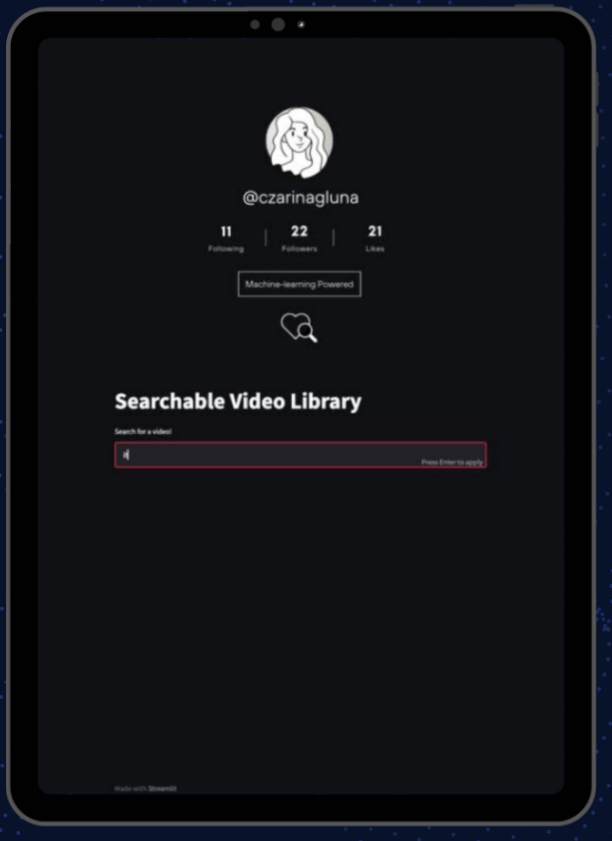
Let's look at how YOLO works.

Video to Object



YOLO looks at the image once, hence the name You Look Only Once. It divides the image into a grid of x by x number of cells. Then it predicts bounding boxes around each cell with output of confidence score for each of the bounding boxes, mapped to class probabilities. And depending on the threshold we set, we get the final detections with confidence scores of say above 0.5.

NLP-Based Search Engine



Putting it all together, I created a corpus of all the words from which I retrieve text information given a search query.
I measured the cosine similarity with TF-IDF weighting to find the top results. TF-IDF stands for term frequency-inverse document frequency that is a measure of how relevant the word is to a document, which in this case is the video.
Finally, I deployed a web app to create the searchable video library.

Further Developments

Music Recognition

Neural Network-Based Search

Other Applications

For further developments, I plan to add music recognition due to the limitations of speech recognition that could not always extract lyrics. It is also worth exploring a neural network based search engine. Lastly, I think this technology should be used for other applications. For instance, analyzing hours of footage from historical video archives, among others.

Thank you! Feel free to reach out to me for any questions.

LinkedIn.com/in/CzarinaGLuna