Vision-Based Website Characterization

Jieruei Chang
AIXON Backend Engineering
Mentors: **Wei-Chun Wang, Yung-Hsiu Chen**

Appier

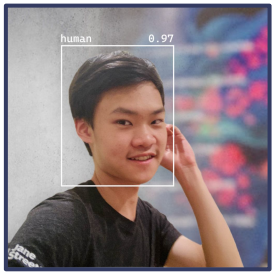
Outline

0. Introduction
1. Problem
2. First Approaches
3. A Simpler Solution
4. Sidequest
5. Putting Everything Together
6. Live Demo
7. Summary
8. Takeaways

Appier

$whoami

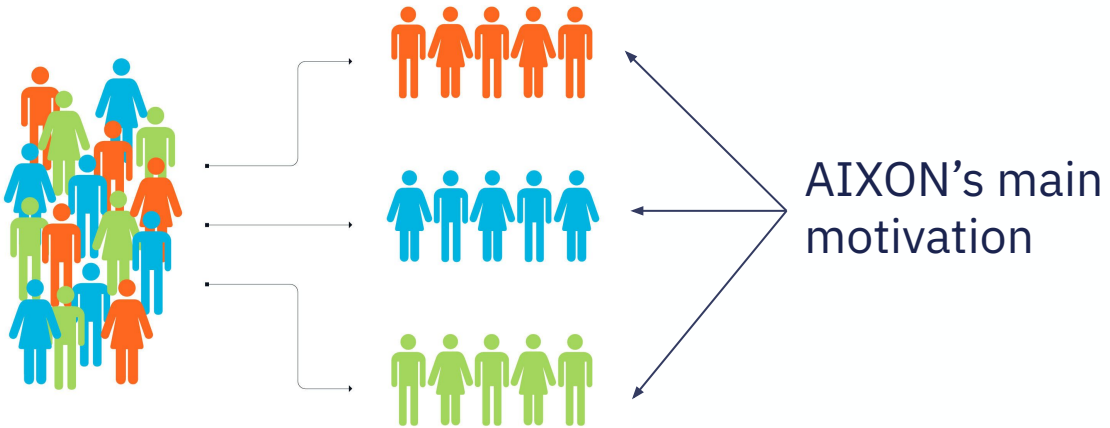
%YAML 1.2
---
name: **Jieruei Chang**
role: Summer Intern
team: AIXON Backend Engineering
academic_status: Freshman @ MIT
caffeine_intake: 126 mg/day
internship_time: 31 days
mentors:
- **yunghsiu.chen**
- **weichun.wang**



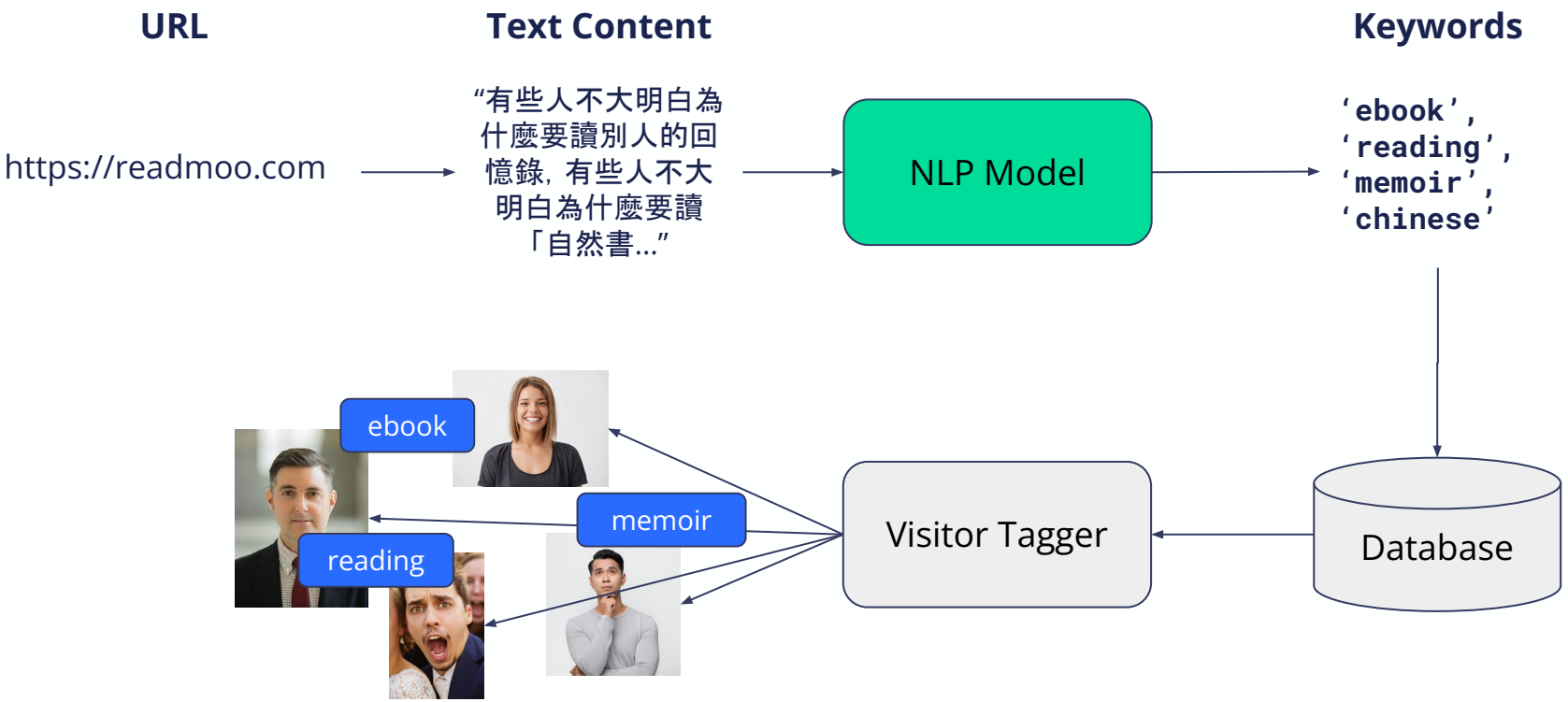
a nice simple question:
how do we make money?
or, how do we run a successful marketing campaign?

Understand your audience

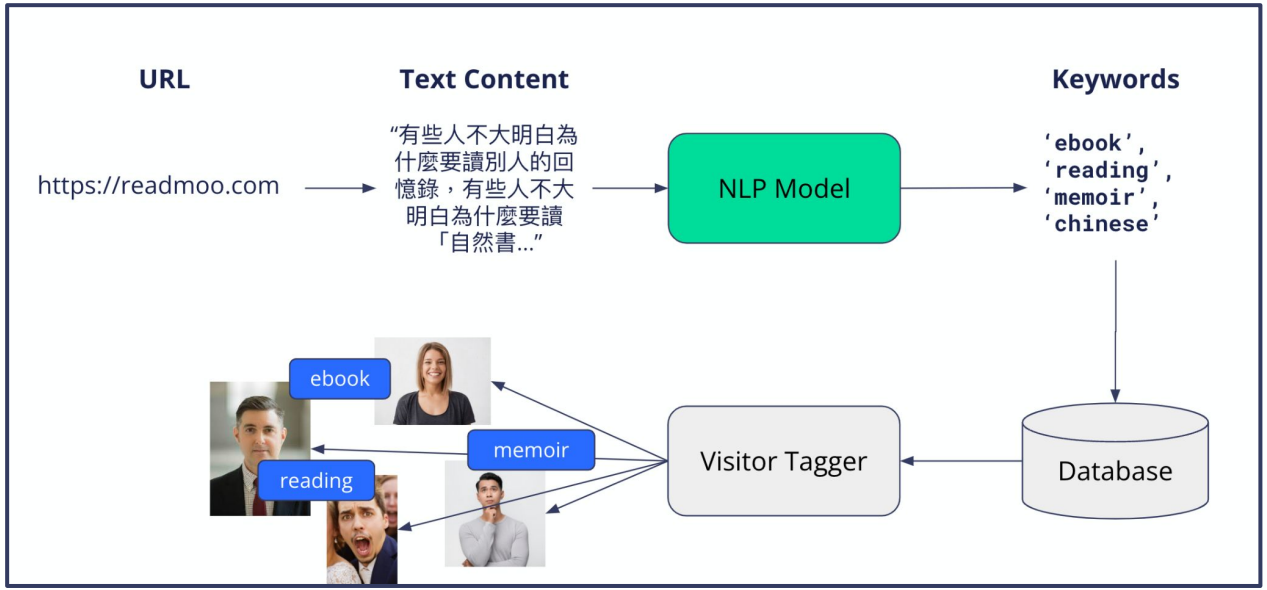
- What kind of user is likely to visit this site?
  - What **attributes** describe them?
  - What are their **interests**?
- How can we ***segment*** this **audience into groups** that we can target individually?



Keyword Segmentation: Auto Tagging Workflow

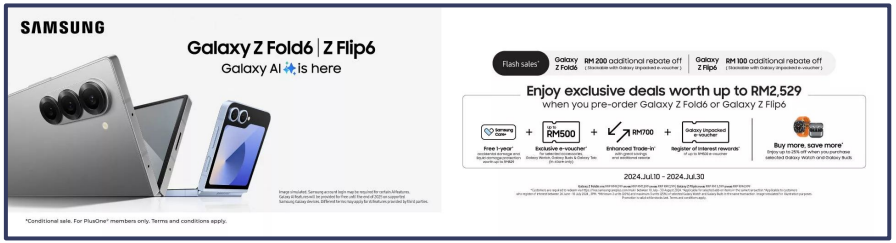


But what are we missing?



Visuals are important too!

- Websites often have content that is presented visually, not through text



Visuals are important too!

Screenshot to Keyword

Our Goal:

- Accurate **User Onsite Behavior Tagging**
- with Intelligent **Website Screenshot Analysis**

URL → **Screenshot** → **Vision Model** → **Keywords**

https://readmoo.com

Vision Model

```
{‘ebook’,
‘reading’,
‘memoir’,
‘chinese’  ...}
```

Screenshot to Keyword: First Thoughts

Find me some keywords!

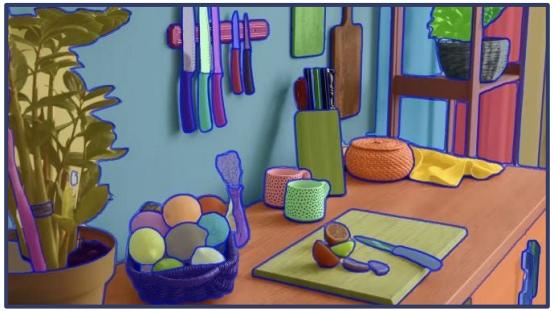
But this can be **10,000 px** long!

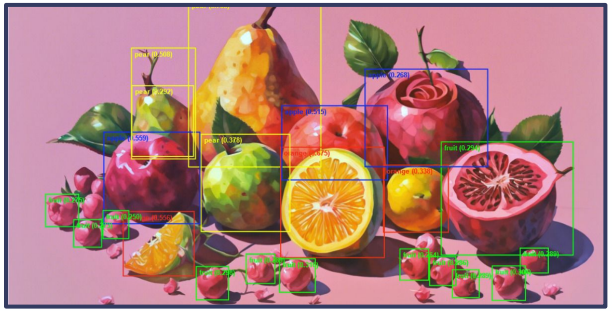
Screenshot to Keyword: Image Segmentation

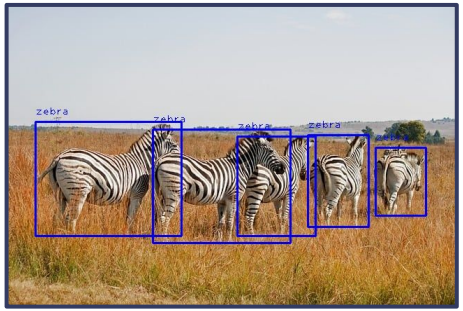
- How do we make GPT see important details, without giving it a huge input?
- We need to break up the screenshot and find only the most relevant pieces

Image Segmentation: First approaches

- Leverage existing zero-shot models
  - Facebook / **SAM**
  - Google / **OWL-ViT**
  - IDEA-Research / **GroundingDINO**

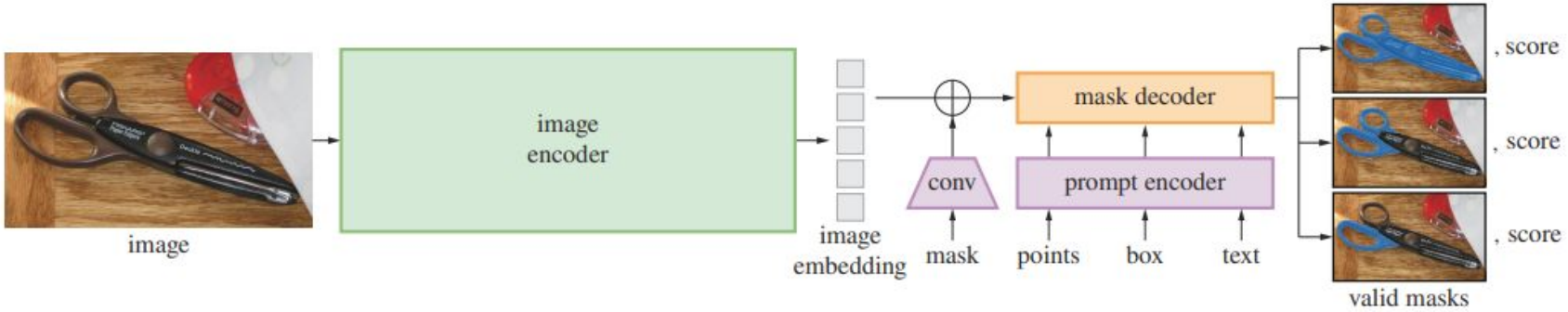






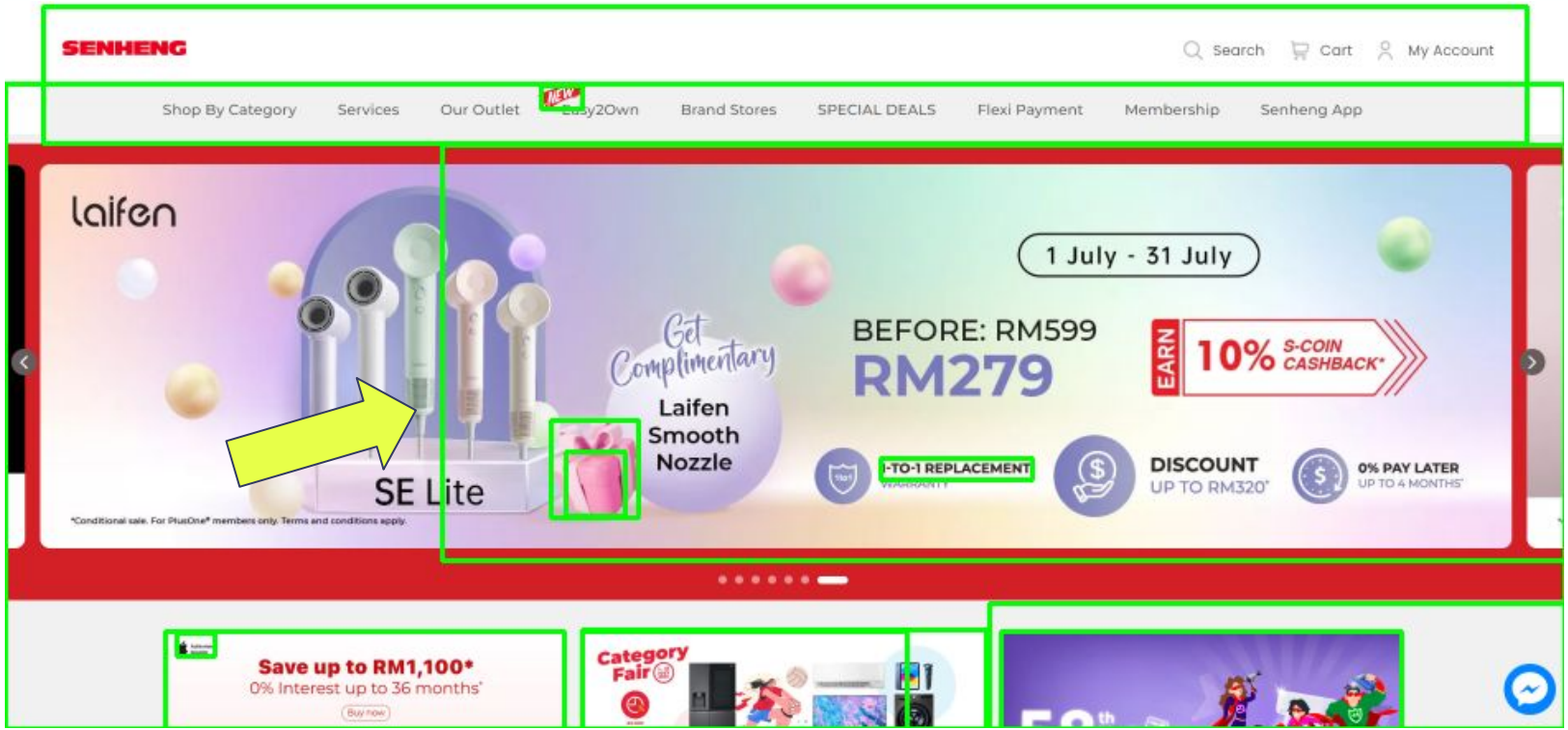
Zero-Shot Segmentation: Segment Anything

- Transformer-based segmentation model
- **~21.01s/image**
- Segmentation masks can be noisy → improve via post processing



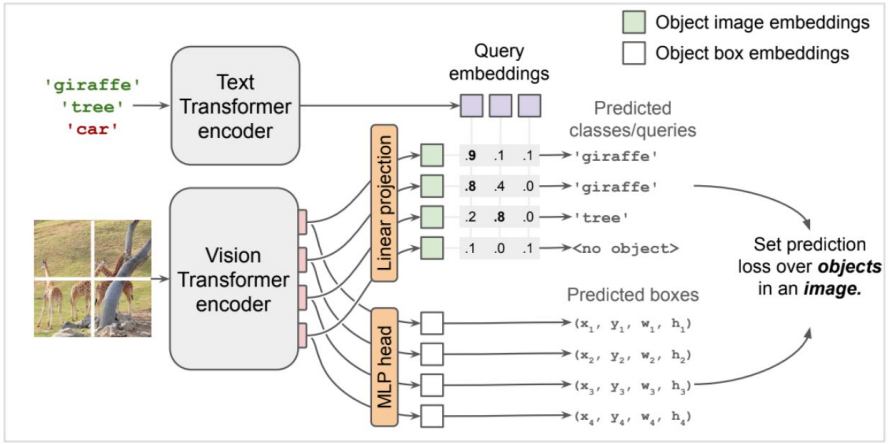
(Kirillov et al., 2023)

Zero-Shot Segmentation: Segment Anything



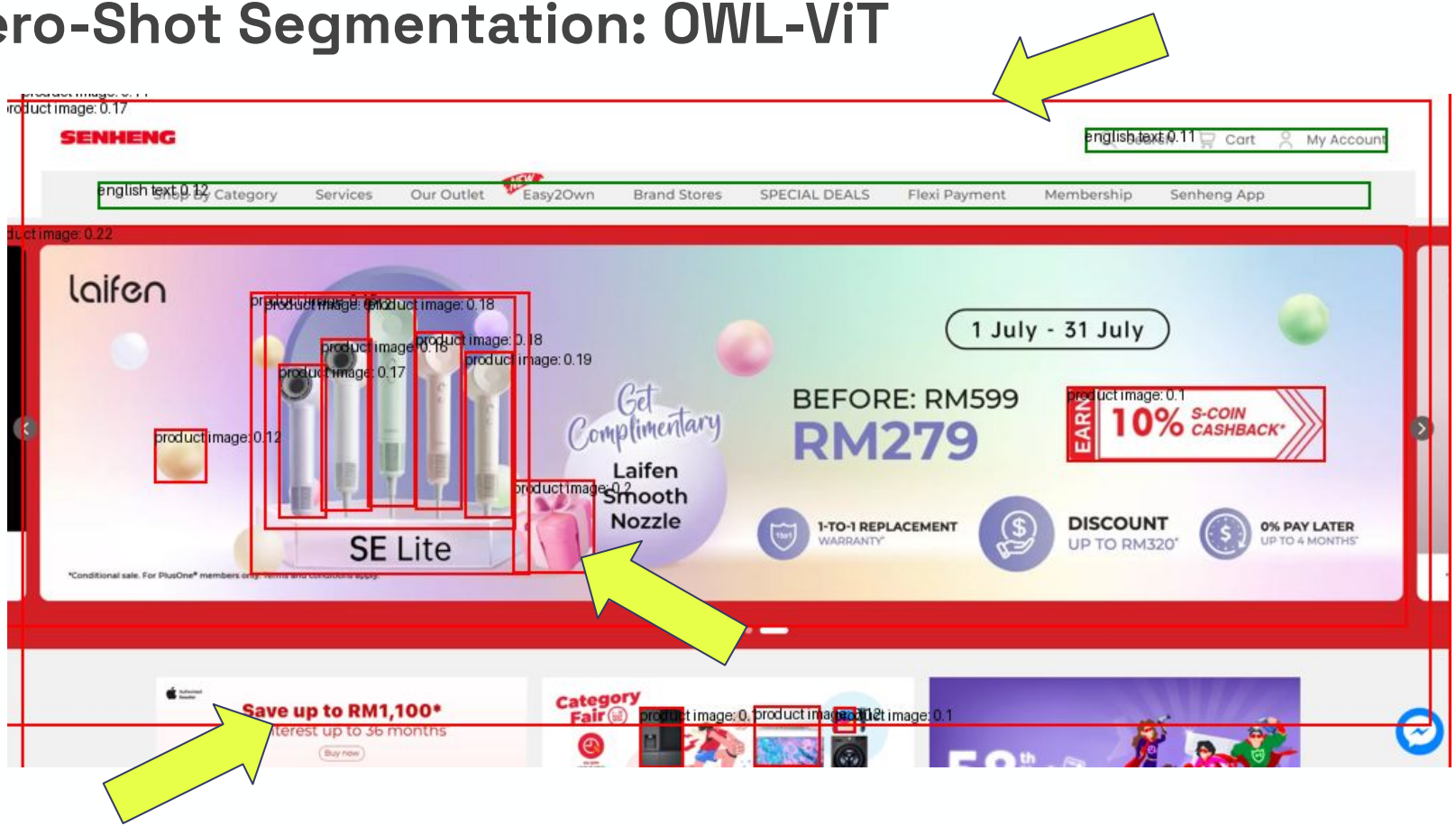
Zero-Shot Segmentation: OWL-ViT

- Zero-shot object detection (bounding boxes)
- Text-promptable model (CLIP embeddings); transformer-based
- **~71.9s/image**



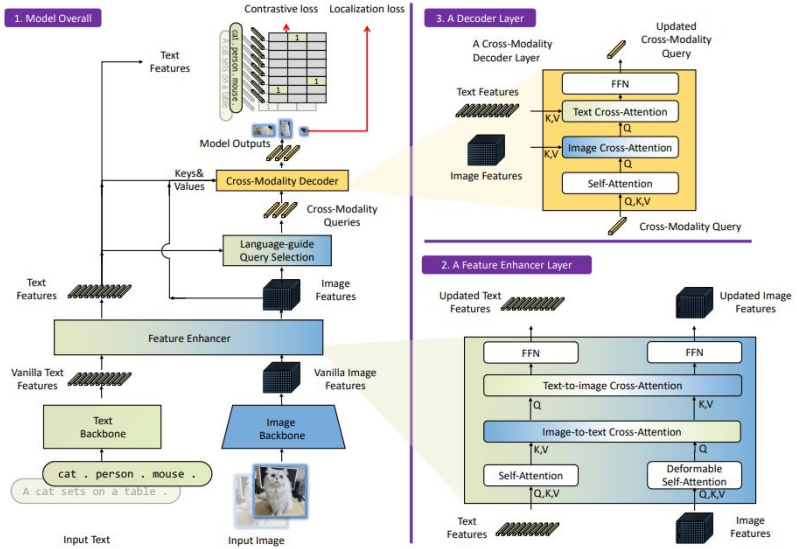
(Minderer et al., 2022)

Zero-Shot Segmentation: OWL-ViT



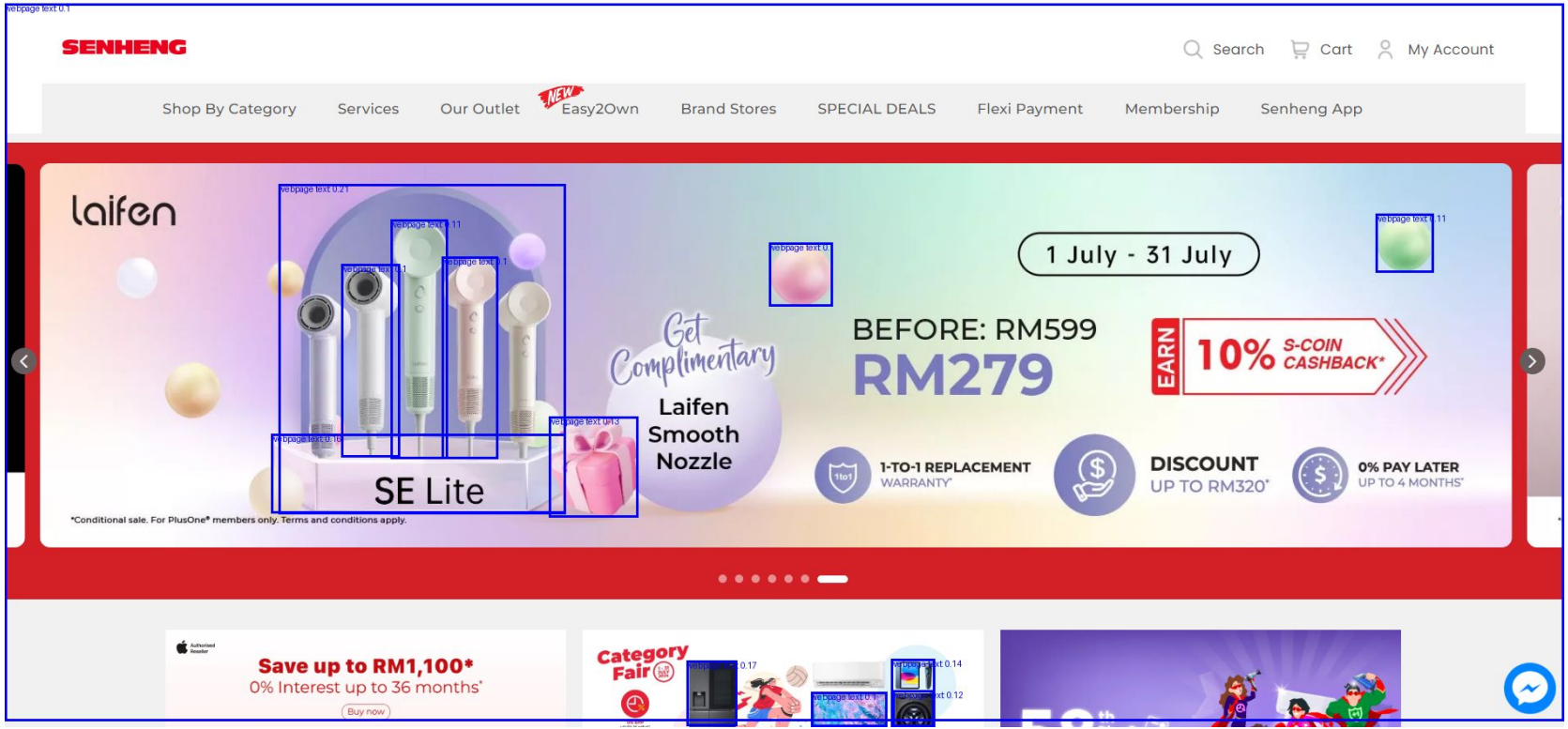
Zero-Shot Segmentation: GroundingDINO

- Zero-shot object detection
- Text-promptable model (CLIP embeddings); transformer-based
- **~15.2s/image**



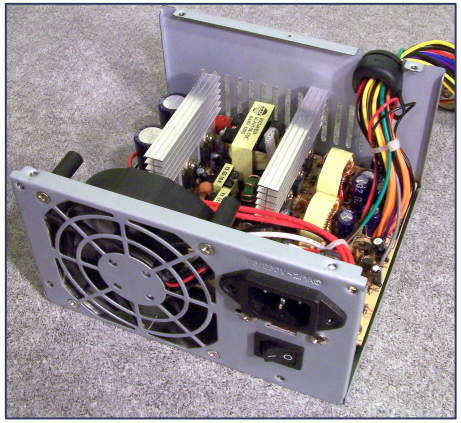
(Liu et al., 2023)

Zero-Shot Segmentation: GroundingDINO



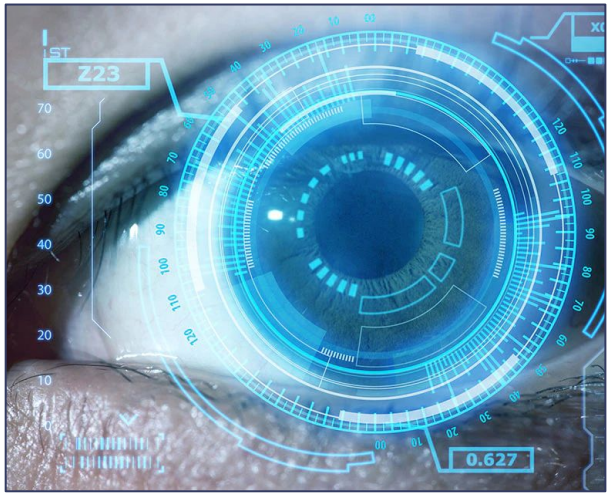
Zero-Shot Segmentation: Problems

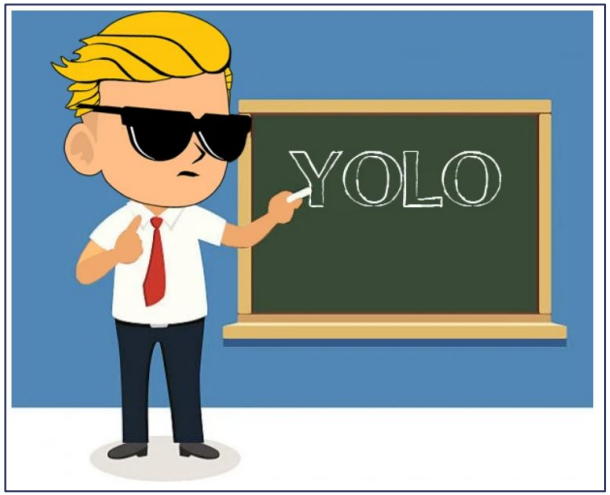
- Computation Time
- Processing Power
- Segmentation Failures


What about a simpler solution?

- Traditional Computer Vision
- YOLOv8





Computer Vision Approach

- Idea: apply image processing algorithms to manipulate and simplify the screenshot → handcrafted, not trained

Computer Vision Approach

- Idea: apply image processing algorithms to manipulate and simplify the screenshot
  → handcrafted, not trained

Computer Vision Approach

Step 1: Greyscale

- Simplify processing for subsequent steps

Computer Vision Approach

Step 2: Canny Edge Detection

- Find all the edges in the screenshot

Computer Vision Approach

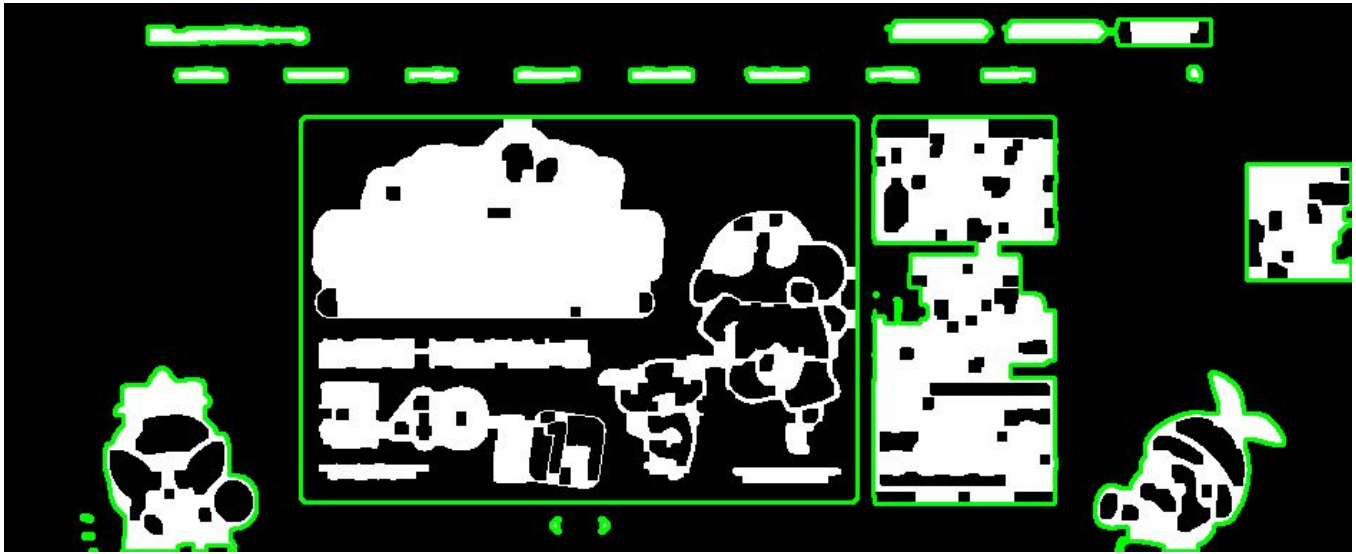
Step 3: Morphological Operations

- Dilation followed by erosion
- Clean up image, merge small regions (e.g. text) together

Computer Vision Approach

Step 4: Contour Detection

- Extract connected regions from binary image
- Hierarchical Filtering → remove contours entirely inside others



Computer Vision Approach

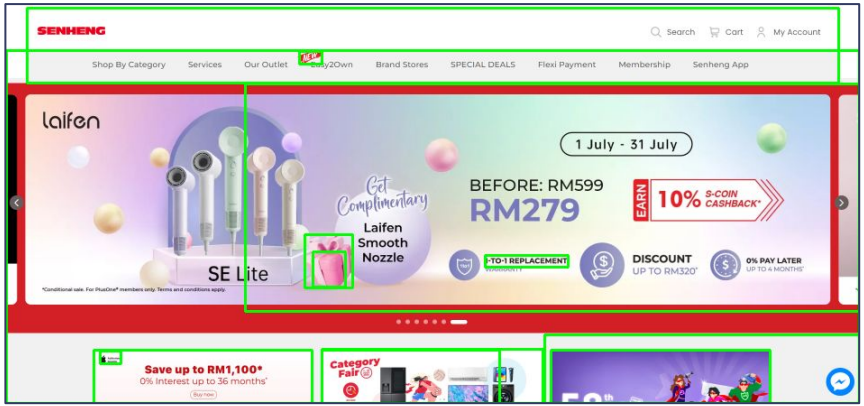
Step 4: Final processing

- Bounding box determination and cropping

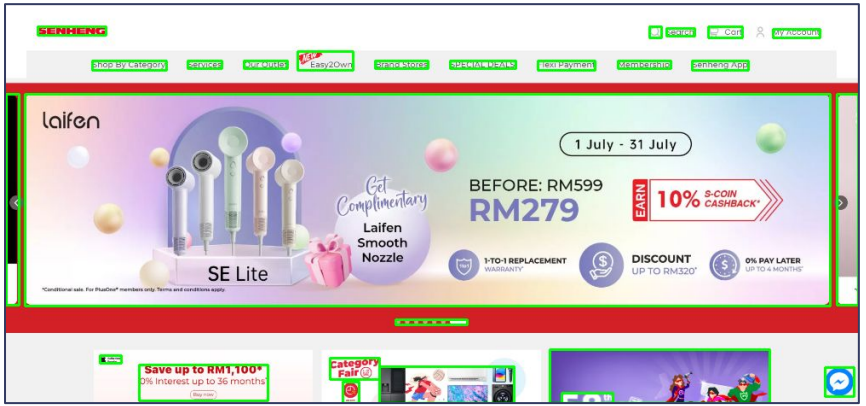
Computer Vision Approach

- Canny, Dilation + Erosion, Contour Detection, Hierarchical Cleaning
- **~0.06s/img → 250x speedup!**
- Segmentation results are generally very clean; can effectively extract isolated text and rectangular sections
  - Fails with more complex geometry

Computer Vision Approach: Results



SAM



CV

Computer Vision Approach: Results

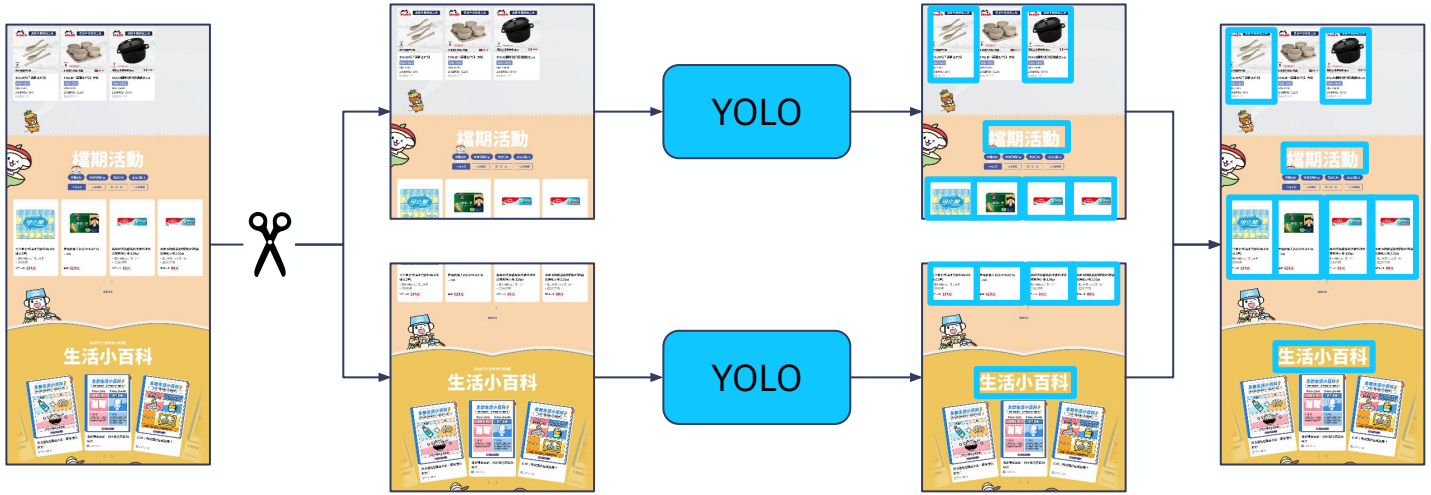
Computer Vision Approach: Results

YOLOv8 Approach

- CNN-based real-time object detection model: You Only Look Once
- Roboflow Website Screenshots Dataset
  - Automatically generated dataset
  - 1206 website screenshots
  - Annotations of images, text, headings, etc.

YOLOv8 Approach

- Need to handle long and thin website screenshots
- Split screenshot into sections and run YOLO individually on each
- Merge bounding boxes on the edge between two slices



YOLOv8 Approach: Results

YOLOv8 Approach: Results

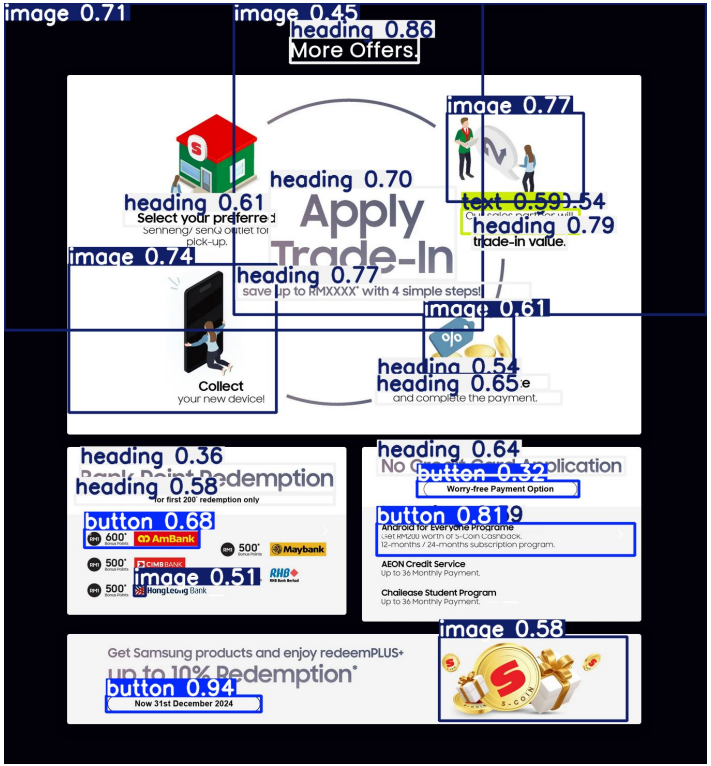
YOLOv8 Approach: Comparison

CV

YOLO

YOLOv8 Approach: Comparison

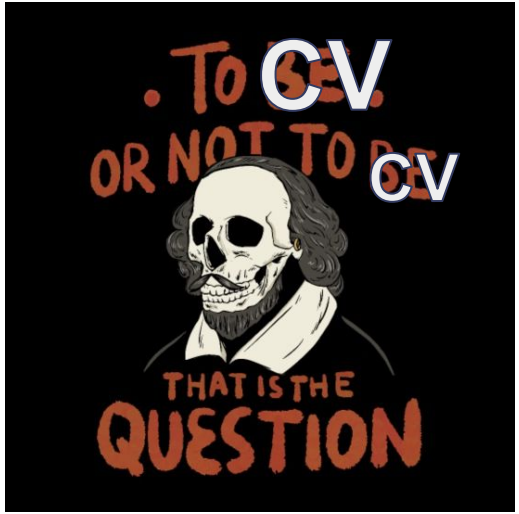
CV



YOLO

Which system do we choose?

- I don’t know
  - CV method is fast and precise, but fails with more complex geometry
  - YOLO can handle harder cases (and can filter out some irrelevant elements), but is slower and its bounding boxes are less clean

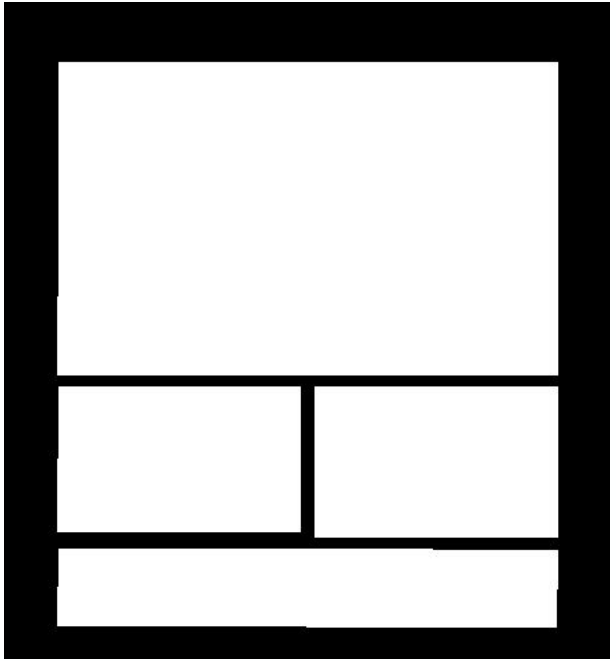


Dynamic Model Selection

- How suitable is this particular image for CV segmentation?
  - Independent determination for each webpage
- CV-based method
  - Canny Edge Detection
  - Contour Detection
  - Convex Hull + Polygonal Approximation
  - Axis-aligned Rectangularity
- Calculate percentage of webpage area that is covered in CV-friendly rectangular contours

Dynamic Model Selection

Rectangle-covered area: 65.7% → CV method selected



Dynamic Model Selection

Rectangle-covered area: 9.7% → YOLO method selected

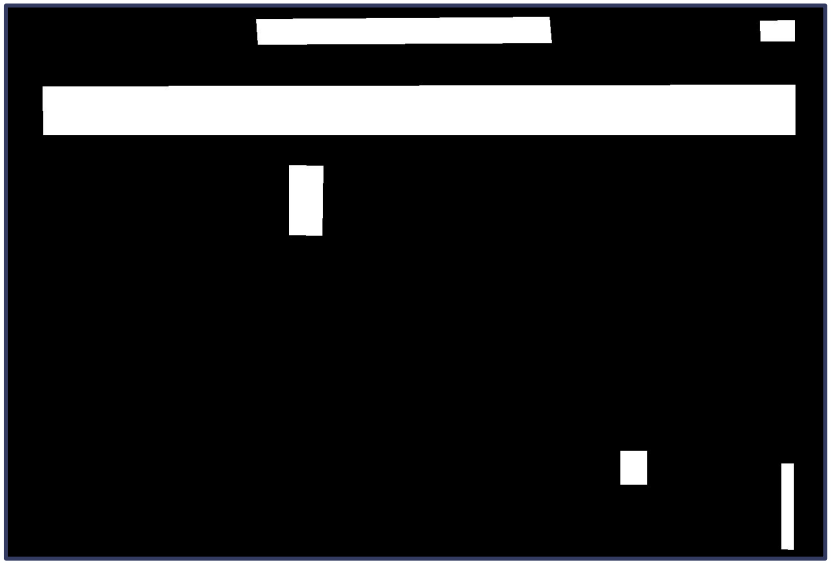
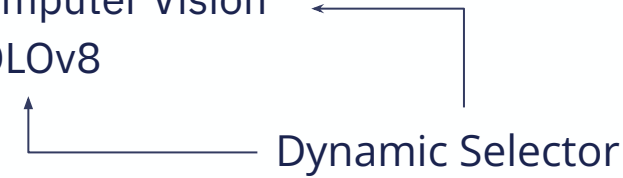


Image Segmentation: Recap

- Zero-Shot Models
  - SAM
  - OWL-ViT
  - GroundingDINO
- Computer Vision
- YOLOv8



Real World Problems

- In deployment, found that many segmentations looked incomplete

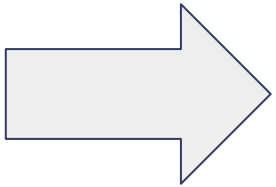
Sidequest: Web Crawler

- Without a good web crawler, an informative segmentation is impossible
- The current crawler (Hercules, in-house) fails to fully load many pages

Scroll Simulation

- Scroll through the website once to trigger JS lazy-loading
- Add event handlers to all images to check for loading
- Wait for all event handlers to fire while continuing to scroll

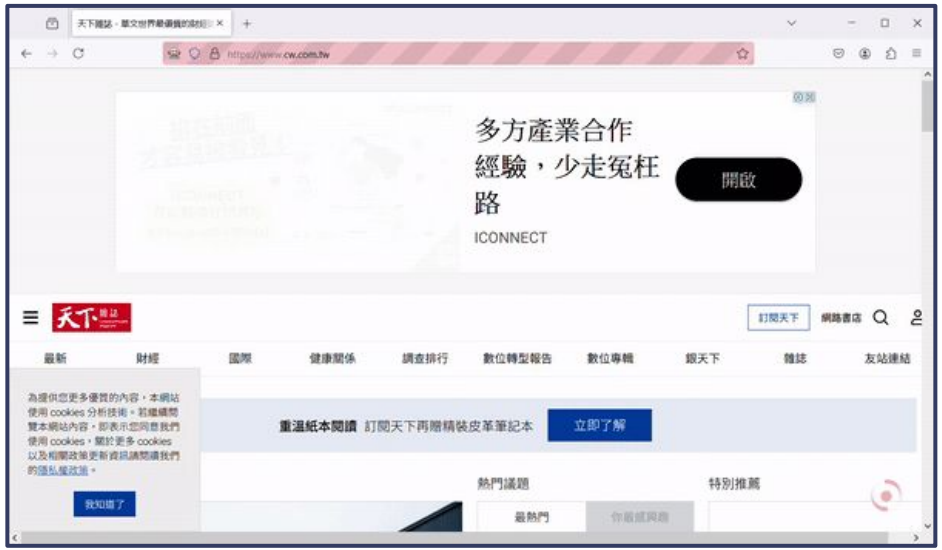
Before



After

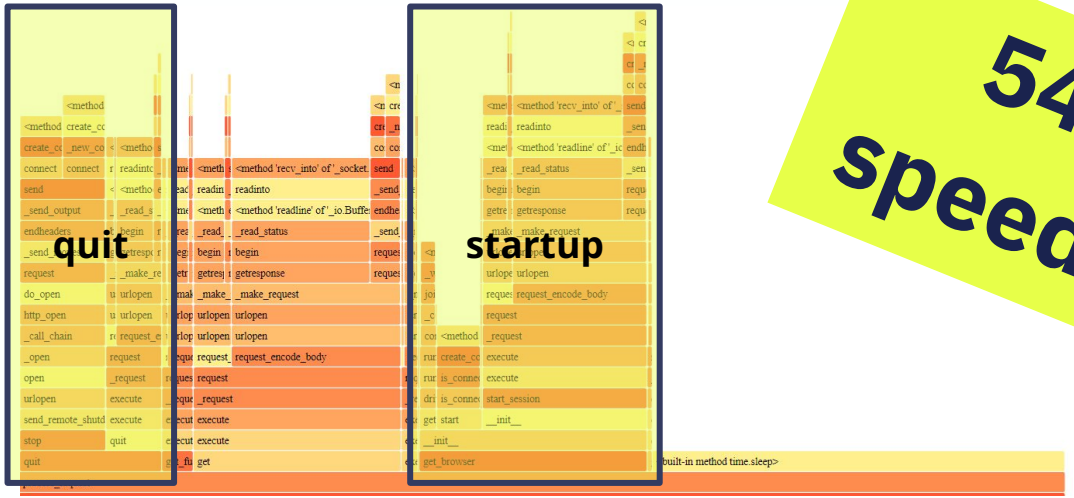
Ad and Popup Blocking

- Inject JS to remove:
  - All iframes from different origins (typically how ads are served)
  - All fixed elements that cover the entire screen (popups/modals)



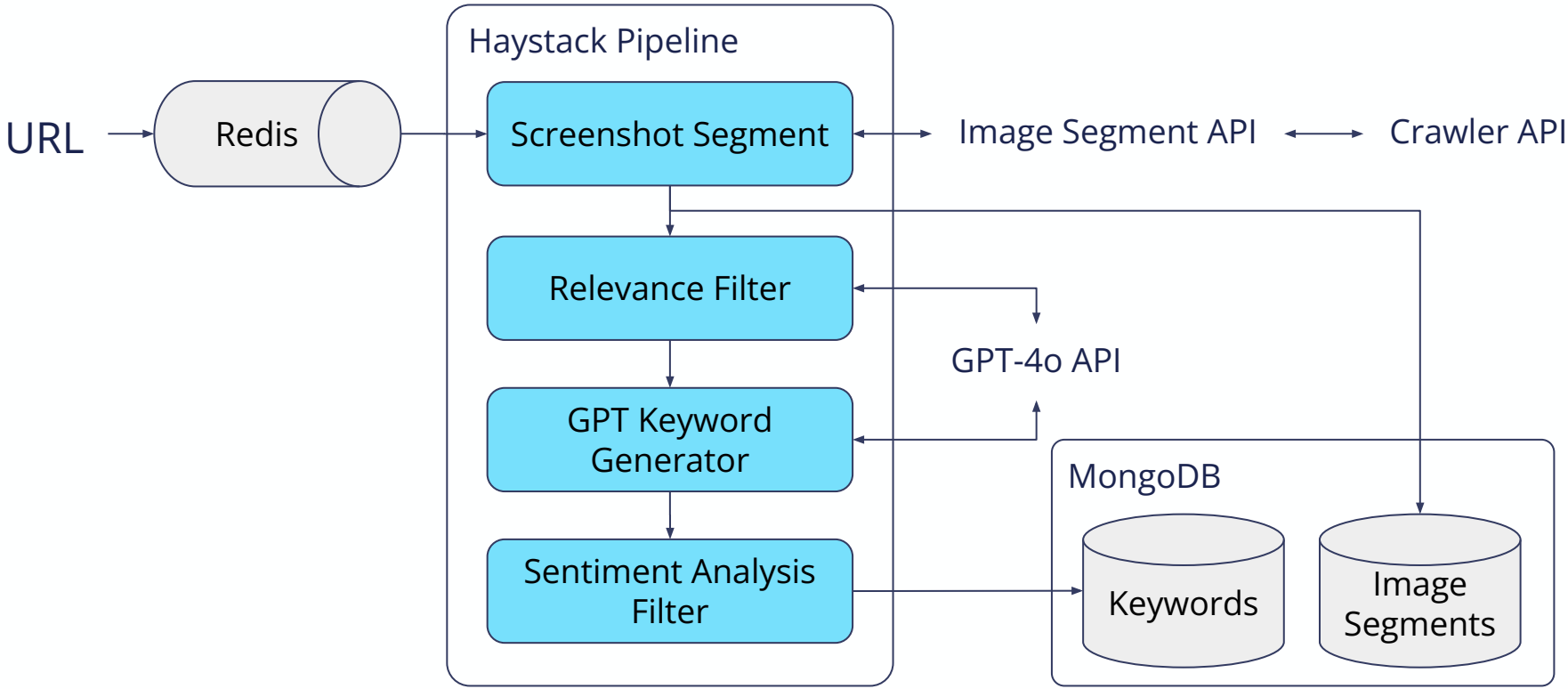
Performance Speedups

- A large percentage of the crawl time is **due to starting up and shutting down** the Selenium-controlled Firefox browser
- We can **keep the browser running**, and reset browser state between each crawl (cookies/cache/localStorage)

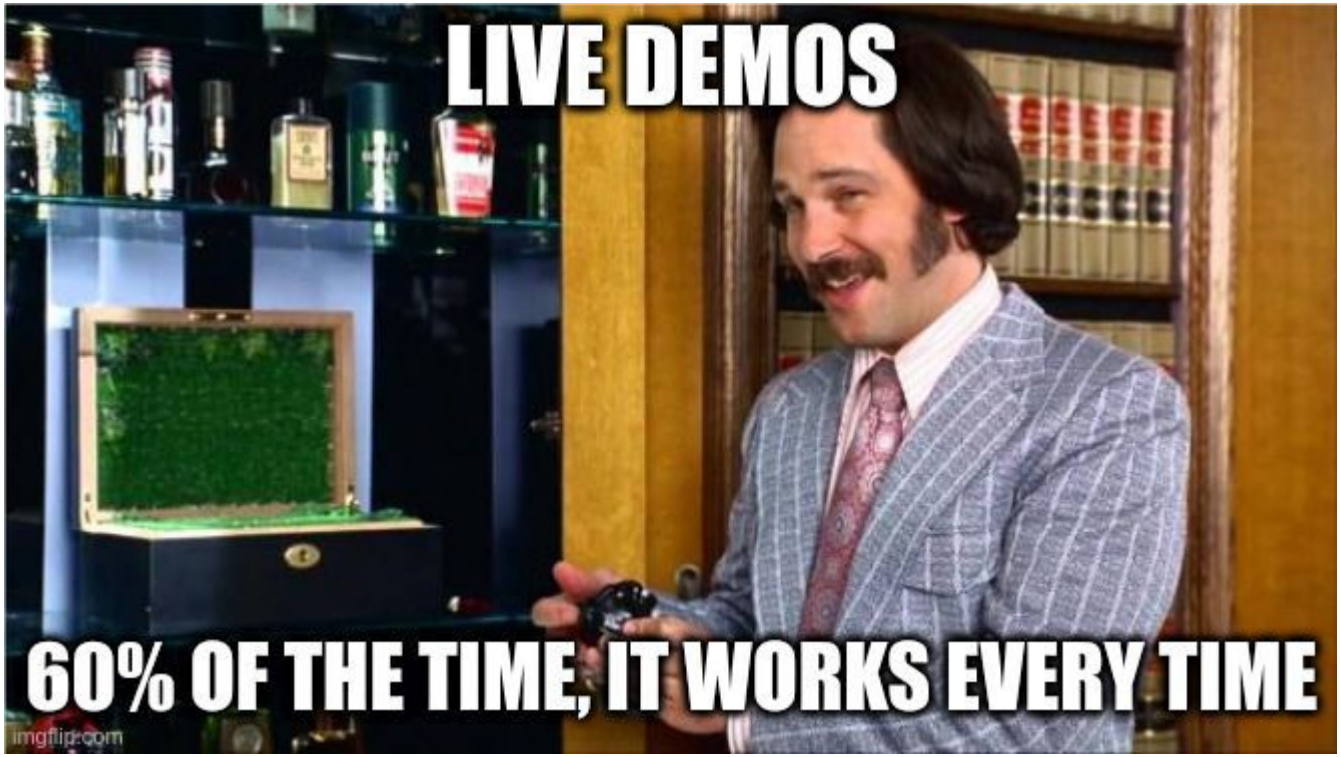


Putting everything together

Keyword Generation Workflow



Live Demo



Summary

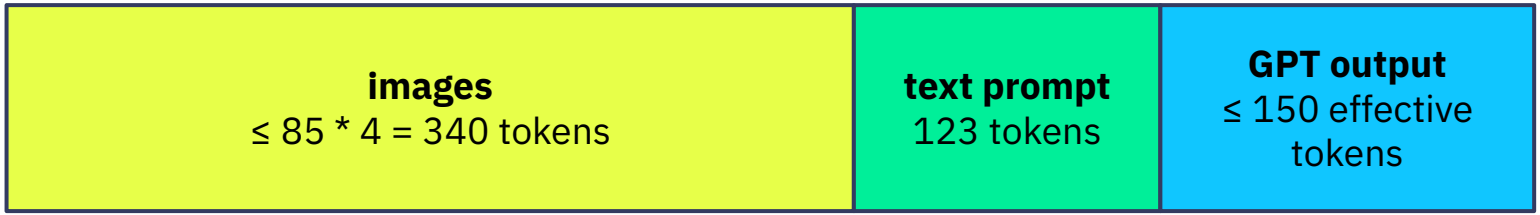
- Use CV/ML to visually extract relevant regions from a website
- Use LLMs to automatically generate keywords to describe the customer base
- Better user tagging → better audience segmentation → better targeted marketing campaigns → more profit $$$
- **AI → ROI**

Appier

Keyword Generation Cost Analysis

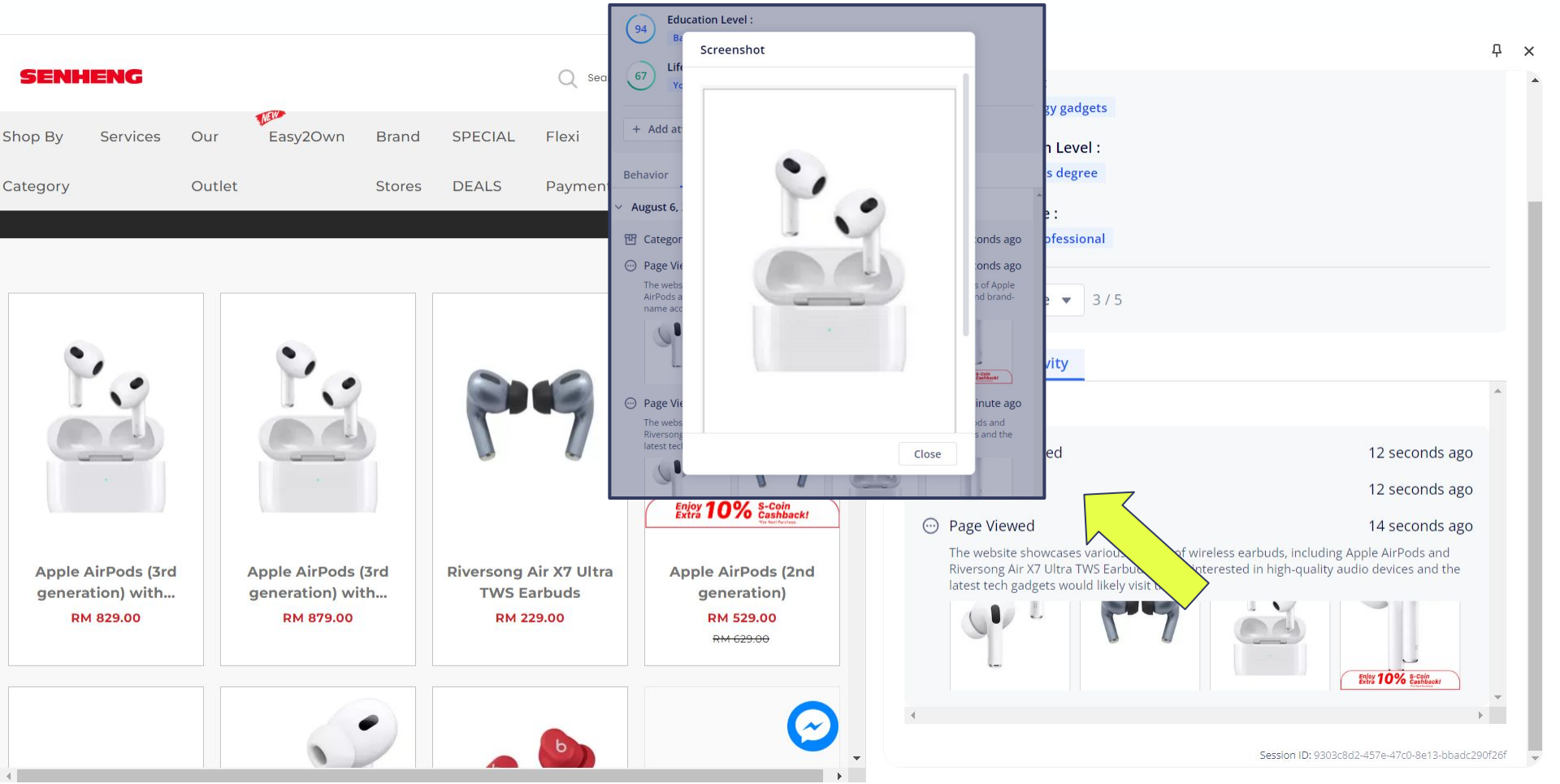
(4 * 85 + 123) * ($5.00 / 1M input tokens) + 50 * ($15.00 / 1M output tokens)
→ **$0.003065** / URL on GPT-4o

Cost Breakdown



Next steps and current usage

- Future work could be to integrate this system into AIXON's onsite behavior NLPaaS system for user tagging
  - Compare tags generated by text and image-based methods
- For now:
  - Webcrawler improvements are in production
    - Performance boost in crawl time
    - More complete content for text-based website characterization
  - Image segmentation API is online
    - Image previews in LLM Auto-Tagging extension
    - Input for website summaries



Takeaways

- Cutting-edge is not always better
  - Practical considerations (e.g. runtime cost); pick the right tool for the job
  - SAM is very powerful in general, but a narrow solution (even CV) may fit better
- Enterprise software is complex because it needs to be scalable and maintainable
  - Webcrawler is 300 lines of code
  - Infrastructure to manage webcrawler is 10,000 lines
- Don’t use a Windows computer for software development
  - “Just Use Linux” – commonly heard phrase

What did I gain?

- A taste of what it's like to work in an enterprise environment
  - Being part of a larger whole
  - Docker, Kubernetes, Jenkins
  - "If a company doesn't use CI/CD, don't work for them"
- A taste of all the good restaurants in Xinyi

Appier

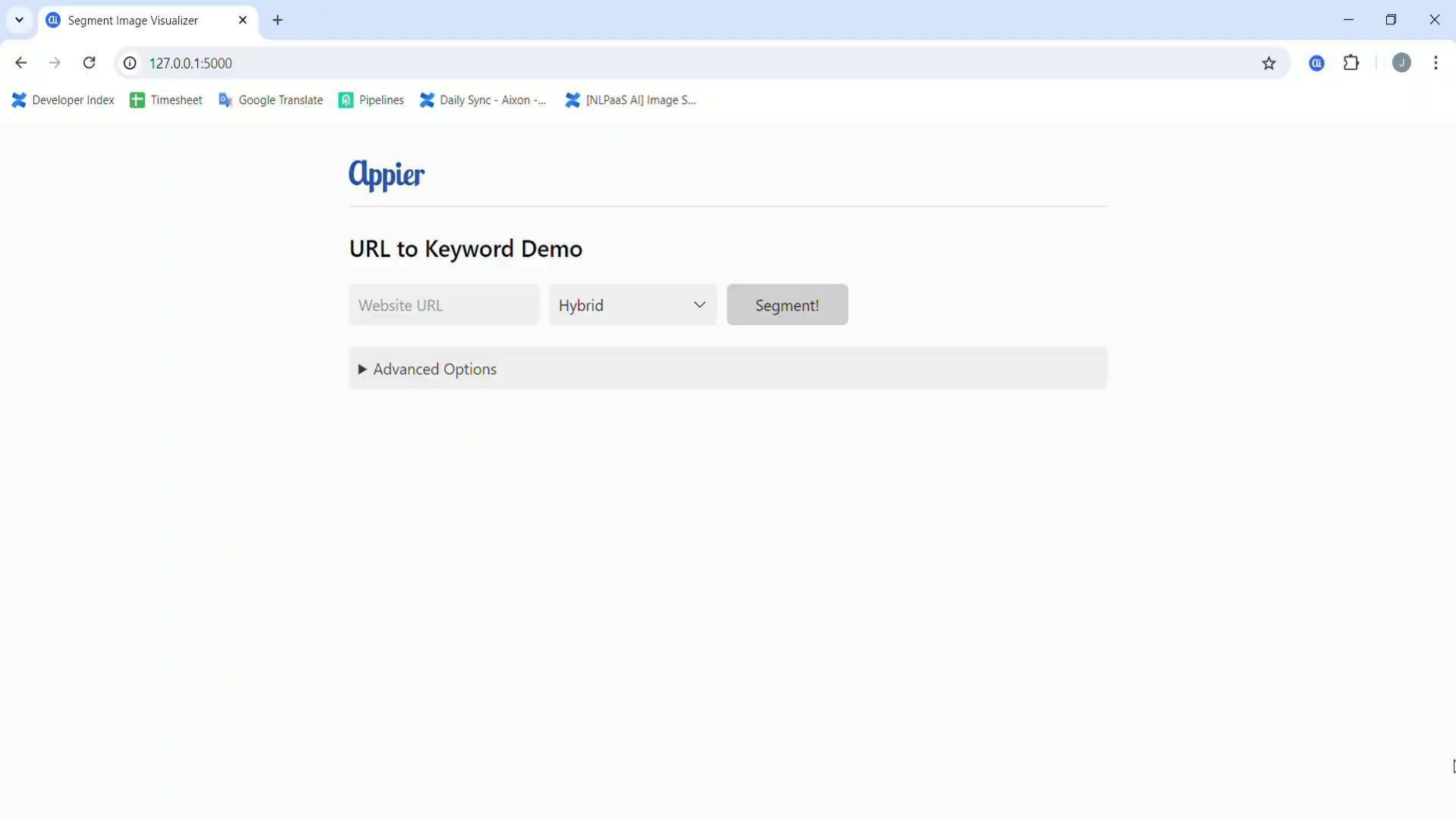
Thank you!

Jieruei Chang / jieruei@mit.edu / github.com/knosmos

AIXON Backend / Vision-Based Website Characterization

https://bank.sinopac.com/sinopacBT/webevents/fraudprevention/

Appendix

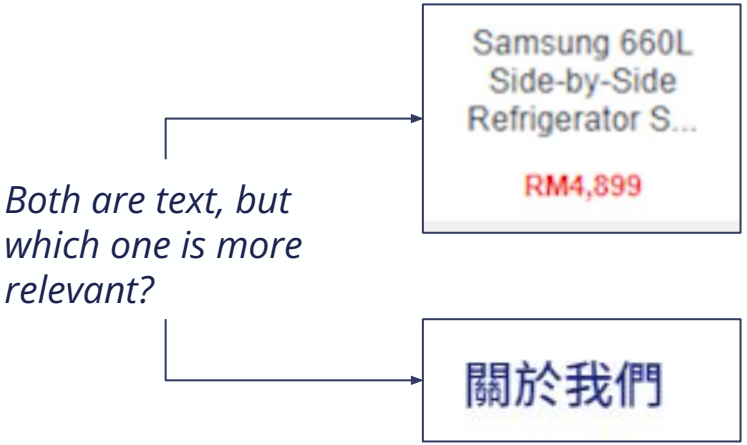

Segment Image Visualizer
127.0.0.1:5000
Developer Index
Timesheet
Google Translate
Pipelines
Daily Sync - Aixon -...
[NLPaaS AI] Image S...
Appier
URL to Keyword Demo
Website URL
Hybrid
Segment!
▶ Advanced Options

Relevance Determination

- Segmentation system can output 500+ segments
  - How do we choose the 4 that are most relevant?
- Simple filtering:
  - Dimensions, aspect ratio
    - Larger images are probably more relevant
    - Strangely sized images are probably less relevant
  - Model prediction confidence
- Image entropy
  - “Disorder” of image segment

Relevance Determination

- “What elements are important” is a hard problem
- Get GPT to do it



Prompt Engineering

```
### TASK DESCRIPTION
The following image is a cropped section of a website screenshot. Your response must
consist of properly formatted JSON with the following attributes:
- class: ONE word, classifying the image into ONE of the following categories: [...]
- relevant: "yes" or "no" depending on whether the image is relevant to understanding the type
of user that would be interested in this website.
    - DO NOT answer "yes" if the image is PURELY decorative and contains no text.
    - DO NOT answer "yes" if it is a navigational element.

### ANSWER FORMAT
{"class":<class>, "relevant":<relevant>}
```

Appier

Prompt Engineering

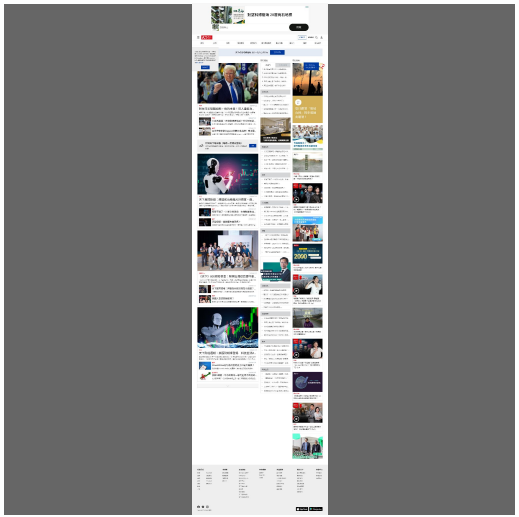
```
### TASK DESCRIPTION
The following are cropped sections of a website. Please generate 10 mutually independent
marketing keywords that may describe some attributes or interests of a user of this website.
This keyword list should include a summary of the website's content, an inference about the
preferences, behavior, personality, and interests of the user browsing the website.
Specifically, avoid any keywords or phrases associated with common web page buttons, actions,
pricing, or deals. Return keywords in comma separated format.

### ANSWER FORMAT
keyword1,keyword2,keyword3...
```

Sentiment Analysis

- Sometimes, GPT gives results that portray the audience negatively
  - A website about banking security → “interested in identity fraud”
    - Probably more relevant are phrases like “fraud protection”
  - We can directly ask GPT to not include such terms
  - Filter with sentiment analysis → SiEBERT (fine-tuned version of RoBERTa transformer model)

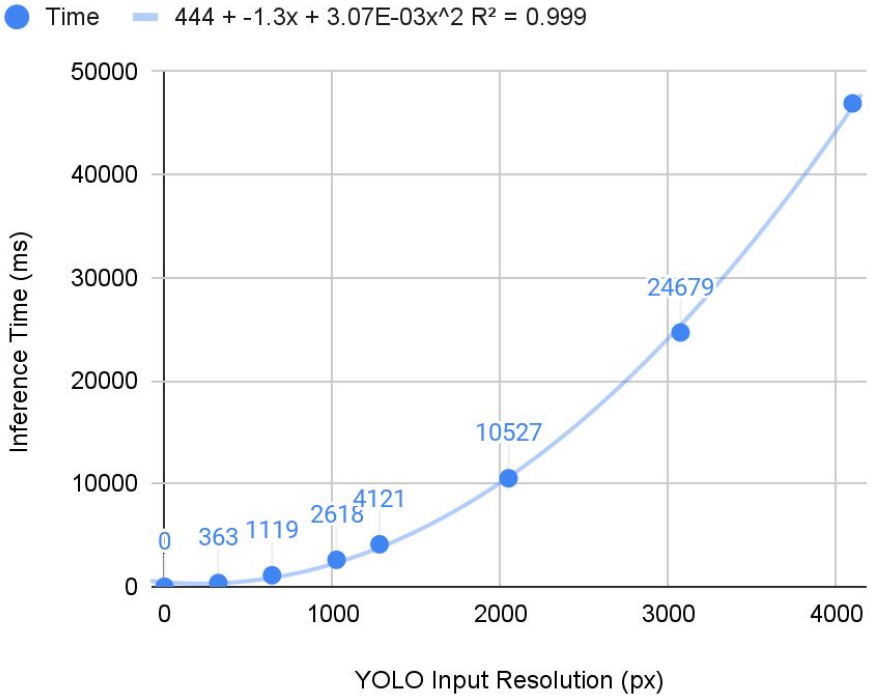
YOLO Implementation Challenges

- YOLO pads inputs to a square (default 640px)
  - But typical website screenshots are very long and skinny
  - We lose significant resolution, leading to poor inferences



Solution 1: Brute Force

- Increase YOLO's input resolution (1024px, 2048px)
  - Runtime scales quadratically
  - Screenshots can have a height of 10,000 px
  - Memory concerns

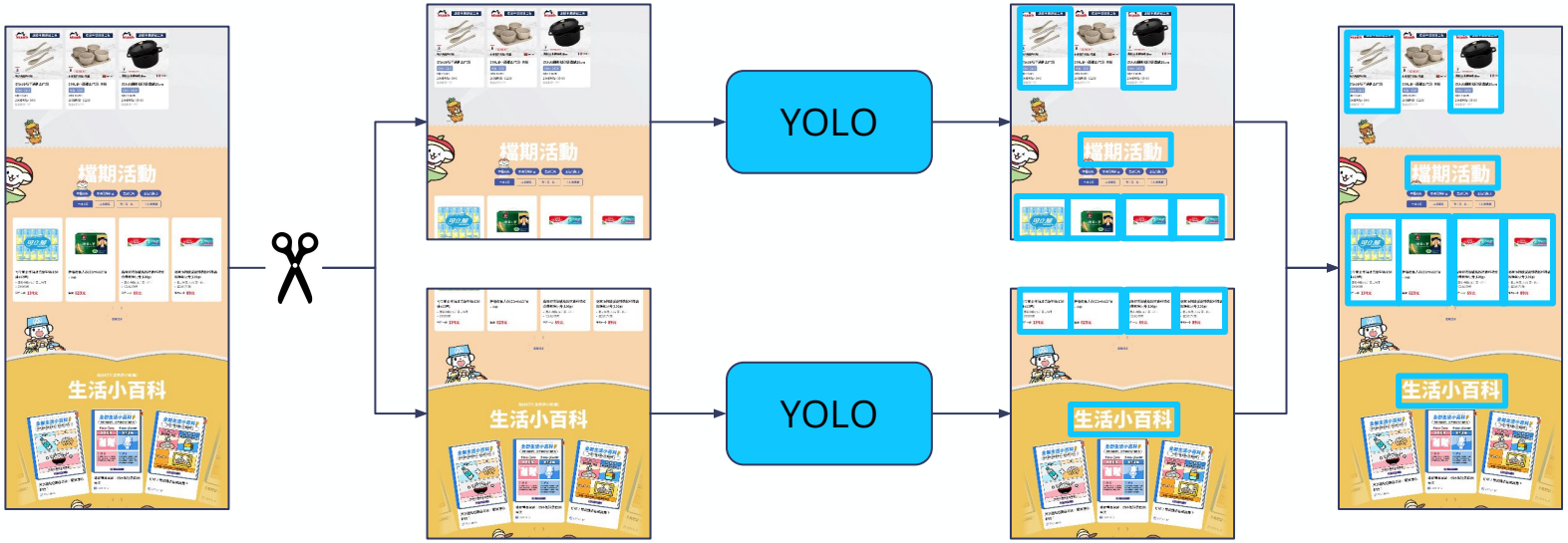


Solution 2: Shifting Window

- Inspiration: FFT frames
- Algorithm
  - Split image into overlapping frames; run YOLO individually on each frame
  - Combine results (merge on sufficiently high IOU); ignore boxes on edge
- Repeated computations

Solution 3: Edge Merge

- Directly slice image into sections
- Run YOLO individually on each section
- Merge bounding boxes on the edge between two slices



Solution 3: Edge Merge

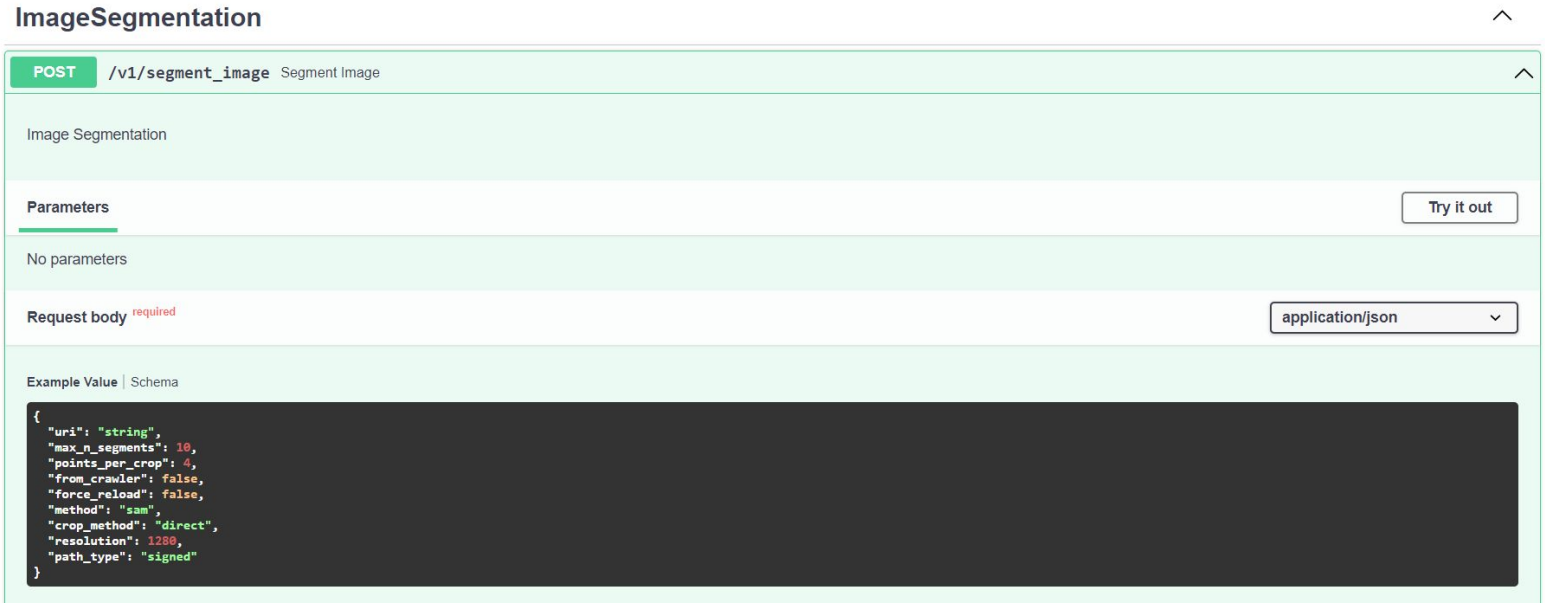
Input

Original YOLO

Slice + Edge Merge

Deployment

- Integrate into FastAPI
- Wrap inside Docker container and deploy to Kubernetes cluster



YOLOv8 Approach: Comparison

CV

YOLO