Charles Ollion
Mar 6, 2018 · 9 min read

Why computer vision APIs won't do the trick for verticalized applications. Heuritech's take in Fashion

Beyond the AI hype, significant new possibilities in the world of computer vision have arisen in the last few years. However, deploying computer vision solutions still requires expert vision knowledge, business understanding, solid engineering and smart processes. I'll expose the challenges of computer vision applied to a vertical such as fashion, and how we solved them at Heuritech.

The Fashion World: social media is a game-changing

The Fashion Industry has been undergoing a major transformation:

> A few years ago, fashion trends would come from the top of the pyramid, while now, millions of people can potentially influence a brand's reputation.

It has resulted in trends continuously popping up from everywhere, millions of new products being launched everyday and even more contents posted online… in the form of images.

This has put an extra pressure on Fashion teams. Their eye & intuition is their superpower but with millions of new insights everyday, we feel **they need an extra help to spot only relevant waves & catch them on time.**

That is where the need for solid and tailored technology comes in to make sense of all that content posted everyday in a relevant and actionable way. This is why we decided to fuel our energy into building a powerful Computer Vision technology that perfectly matches this precise need.

Diving into the world of computer vision

There are generally 3 ways to go to tackle any computer vision problem at a business scale:

> 1. Use a generic API such as Google Cloud Vision, Amazon rekognition, Clarifai, etc.
>
> 2. Build your own system, starting from Open Source algorithms and your own dataset;
>
> 3. Use a domain-specific service trained specifically for your problem.

Qualifying inputs (images/videos) and outputs (elements we want to detect):

1. Class Granularity: In computer vision, we define a "Class" as an element or attribute to detect in an image. Here, we want to detect precise elements of Fashion: Identify each cloth, accessory, shape, attribute, color, pattern, style and even the exact product when it is identifiable.
2. Image diversity: Images/Videos may come from any user, and in that way are not normalized. There are plenty of contexts, zooms & resolutions, lighting conditions, etc. This makes the problem much, much more difficult than if it was on standardized pictures.
3. Class (think object) variability: Many of the classes we want to detect, "handbag", "floral texture", etc. may take several forms: a handbag might be a tote bag or a backpack, which are very different visually.
4. Class (think object) deformations: All the objects we are detecting may be seen under many different angles, deformations, occlusions (i.e. the handbag is worn and partially hidden).

5. Class Evolvability: We have a very large set of classes, organised hierarchically, and constantly evolving, as we want to detect new products or new attributes.

Charles Ollion
317 Followers
CTO Heuritech #machinelearning #deeplearning #nlp

Follow

Related

What are our constraints in tech terms

6. Precision / recall: The output of our product and attributes detection should always be relevant (we need a near perfect precision), but it is ok if a few items are missed (we need a good recall). To be able to track trends and to compare the performance of one product compared to another, we need to control precisely the recall (= have the same level of recall, for instance 90% for each product).

Keep in mind that this could be very different for another problem. In image moderation for instance it shouldn't miss any transgressive image (near perfect recall), but it's ok if we reject a bit too many images (good precision).

7. Dataset availability and quality: There is no available good quality dataset corresponding to the different classes we want to detect. Unfortunately, this is almost always how it goes with Machine Learning.

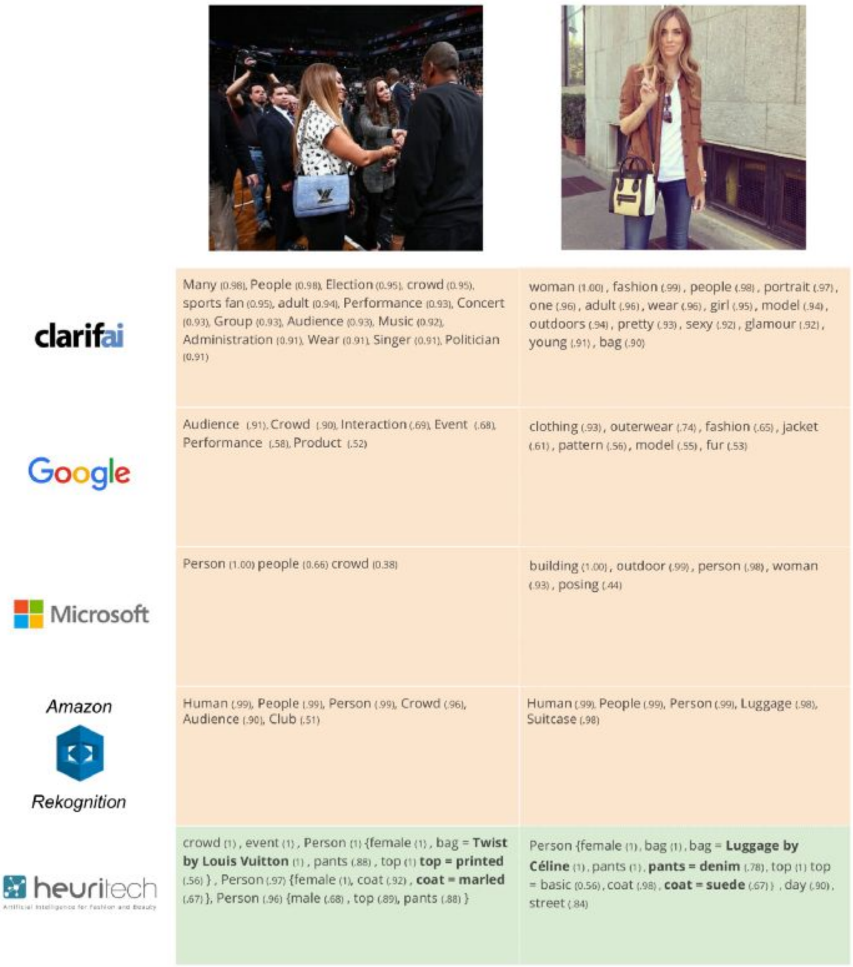
8. Scaling, speed and deployment:

- We need to analyse 1M-10M images / videos a day.
- We don't need to be realtime (we can process images each day and have the results the day after).

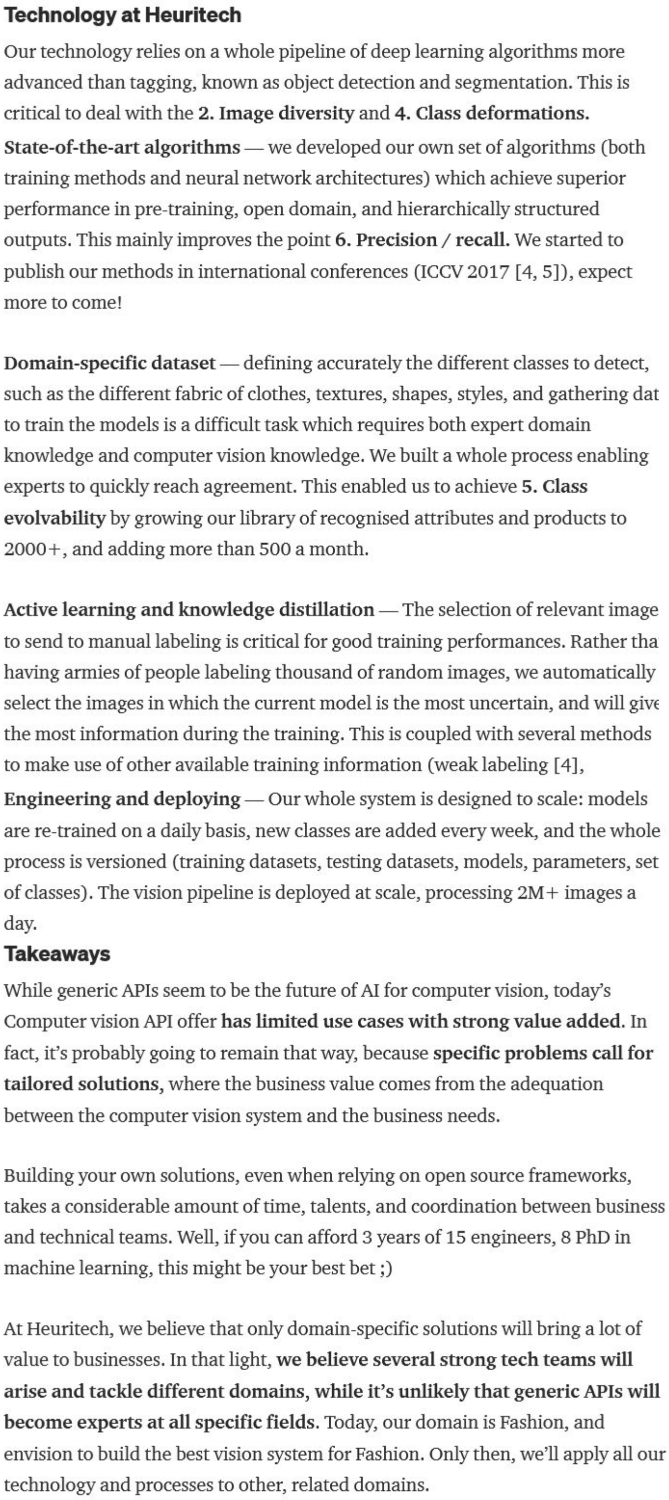
Is our problem solved by current offers?

As Images may come from any source, we absolutely need to use solutions which can handle large **2. Image diversity**. We may then turn to APIs, but we encounter the following problems:

- Not a good **1. Class Granularity** and **5. Class evolvability** — this makes these solutions irrelevant for fashion trends spotting: knowing there's a "dress" and a "tree" in a picture doesn't bring any value. The business value only when we get to domain-specific (here Fashion, including specific styles, brands, patterns…) tagging.
- The **6. precision and recall** for coarse grained classes is good, but recall is lacking for finer grained classes. By the way, if you want to know more about the actual performance of these APIs, here is an excellent article about them.
- APIs operate at the full image level — we want to operate at the level of each clothes and qualify them precisely. If an API gives 'denim' 'jacket' and 'pants', we wouldn't know if it's the jacket or the jeans that have the 'denim' fabric.
- Super expensive — when we get to large scale analysis of images, such as a few millions per day, it's easy to climb up to a few 100K$ monthly bills.



clarifai	Many (0.98), People (0.98), Election (0.95), crowd (0.95), sports fan (0.95), adult (0.94), Performance (0.93), Concert (0.93), Group (0.93), Audience (0.93), Music (0.92), Administration (0.91), Wear (0.91), Singer (0.91), Politician (0.91)	woman (1.00) , fashion (.99) , people (.98) , portrait (.97) , one (.96) , adult (.96) , wear (.96) , girl (.95) , model (.94) , outdoors (.94) , pretty (.93) , sexy (.92) , glamour (.92) , young (.91) , bag (.90)
Google	Audience (.91), Crowd (.90), Interaction (.69), Event (.68), Performance (.58), Product (.52)	clothing (.93) , outerwear (.74) , fashion (.65) , jacket (.61) , pattern (.56) , model (.55) , fur (.53)
Microsoft	Person (1.00) people (0.66) crowd (0.38)	building (1.00) , outdoor (.99) , person (.98) , woman (.93) , posing (.44)
Amazon Rekognition	Human (.99), People (.99), Person (.99), Crowd (.96), Audience (.90), Club (.51)	Human (.99), People (.99), Person (.99), Luggage (.98), Suitcase (.98)
heuritech Artificial Intelligence for Fashion and Beauty	crowd (1) , event (1) , Person (1) {female (1) , bag = **Twist by Louis Vuitton** (1) , pants (.88) , top (1) **top = printed** (.56) } , Person (.97) {female (1), coat (.92) , **coat = marled** (.67) }, Person (.96) {male (.68) , top (.89), pants (.88) }	Person {female (1) , bag (1) , bag = **Luggage by Céline** (1) , pants (1) , **pants = denim** (.78), top (1) top = basic (0.56) , coat (.98) , **coat = suede** (.67) } , day (.90) , street (.84)



Technology at Heuritech

Our technology relies on a whole pipeline of deep learning algorithms more advanced than tagging, known as object detection and segmentation. This is critical to deal with the **2. Image diversity** and **4. Class deformations.**

State-of-the-art algorithms — we developed our own set of algorithms (both training methods and neural network architectures) which achieve superior performance in pre-training, open domain, and hierarchically structured outputs. This mainly improves the point **6. Precision / recall.** We started to publish our methods in international conferences (ICCV 2017 [4, 5]), expect more to come!

Domain-specific dataset — defining accurately the different classes to detect, such as the different fabric of clothes, textures, shapes, styles, and gathering dat to train the models is a difficult task which requires both expert domain knowledge and computer vision knowledge. We built a whole process enabling experts to quickly reach agreement. This enabled us to achieve **5. Class evolvability** by growing our library of recognised attributes and products to 2000+, and adding more than 500 a month.

Active learning and knowledge distillation — The selection of relevant image to send to manual labeling is critical for good training performances. Rather tha having armies of people labeling thousand of random images, we automatically select the images in which the current model is the most uncertain, and will give the most information during the training. This is coupled with several methods to make use of other available training information (weak labeling [4],

Engineering and deploying — Our whole system is designed to scale: models are re-trained on a daily basis, new classes are added every week, and the whole process is versioned (training datasets, testing datasets, models, parameters, set of classes). The vision pipeline is deployed at scale, processing 2M+ images a day.

Takeaways

While generic APIs seem to be the future of AI for computer vision, today's Computer vision API offer **has limited use cases with strong value added**. In fact, it's probably going to remain that way, because **specific problems call for tailored solutions,** where the business value comes from the adequation between the computer vision system and the business needs.

Building your own solutions, even when relying on open source frameworks, takes a considerable amount of time, talents, and coordination between business and technical teams. Well, if you can afford 3 years of 15 engineers, 8 PhD in machine learning, this might be your best bet ;)

At Heuritech, we believe that only domain-specific solutions will bring a lot of value to businesses. In that light, **we believe several strong tech teams will arise and tackle different domains, while it's unlikely that generic APIs will become experts at all specific fields**. Today, our domain is Fashion, and envision to build the best vision system for Fashion. Only then, we'll apply all our technology and processes to other, related domains.

https://medium.com/serena-capital/why-computer-vision-apis-wont-do-the-trick-for-verticalized-applications-93abe59fb021