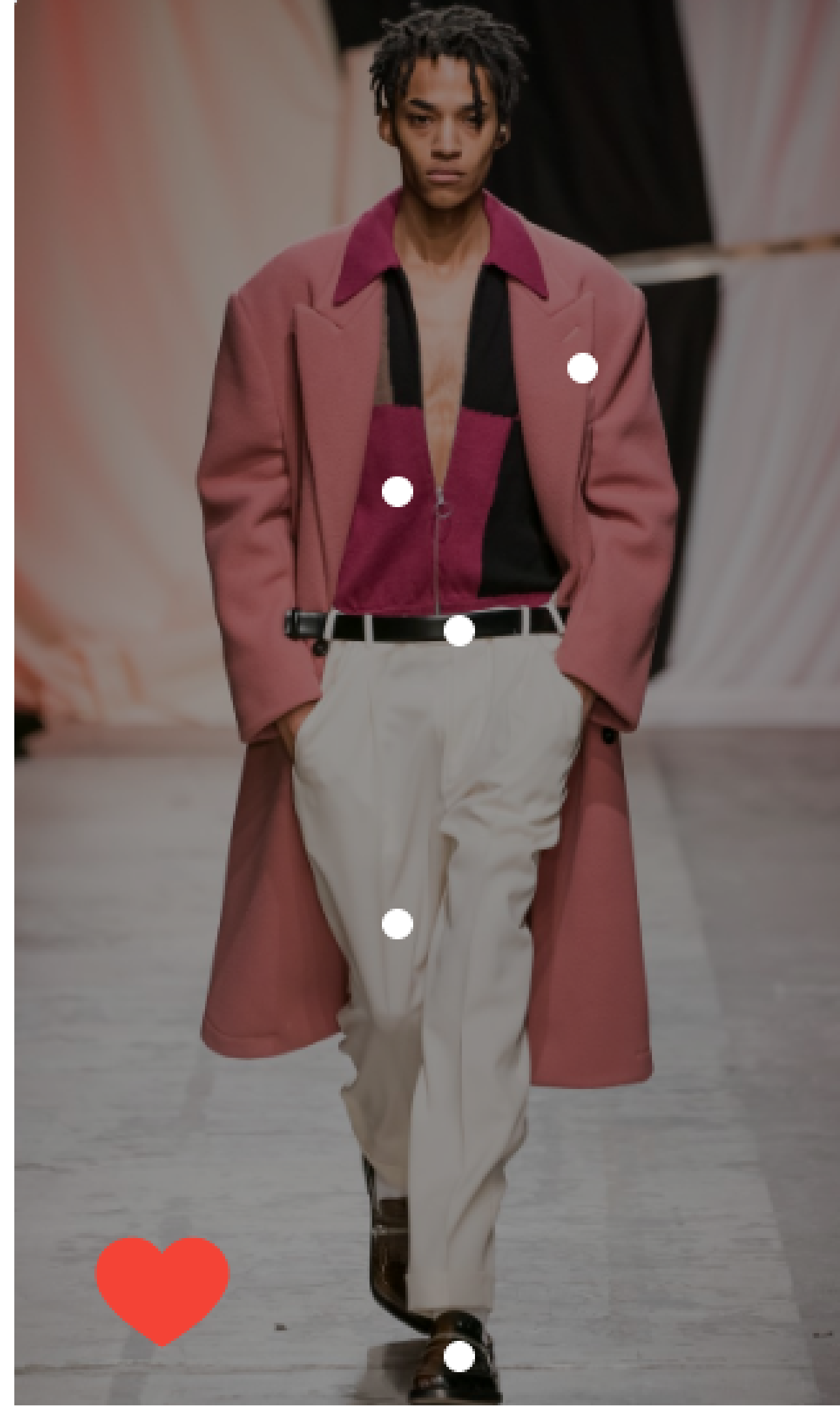




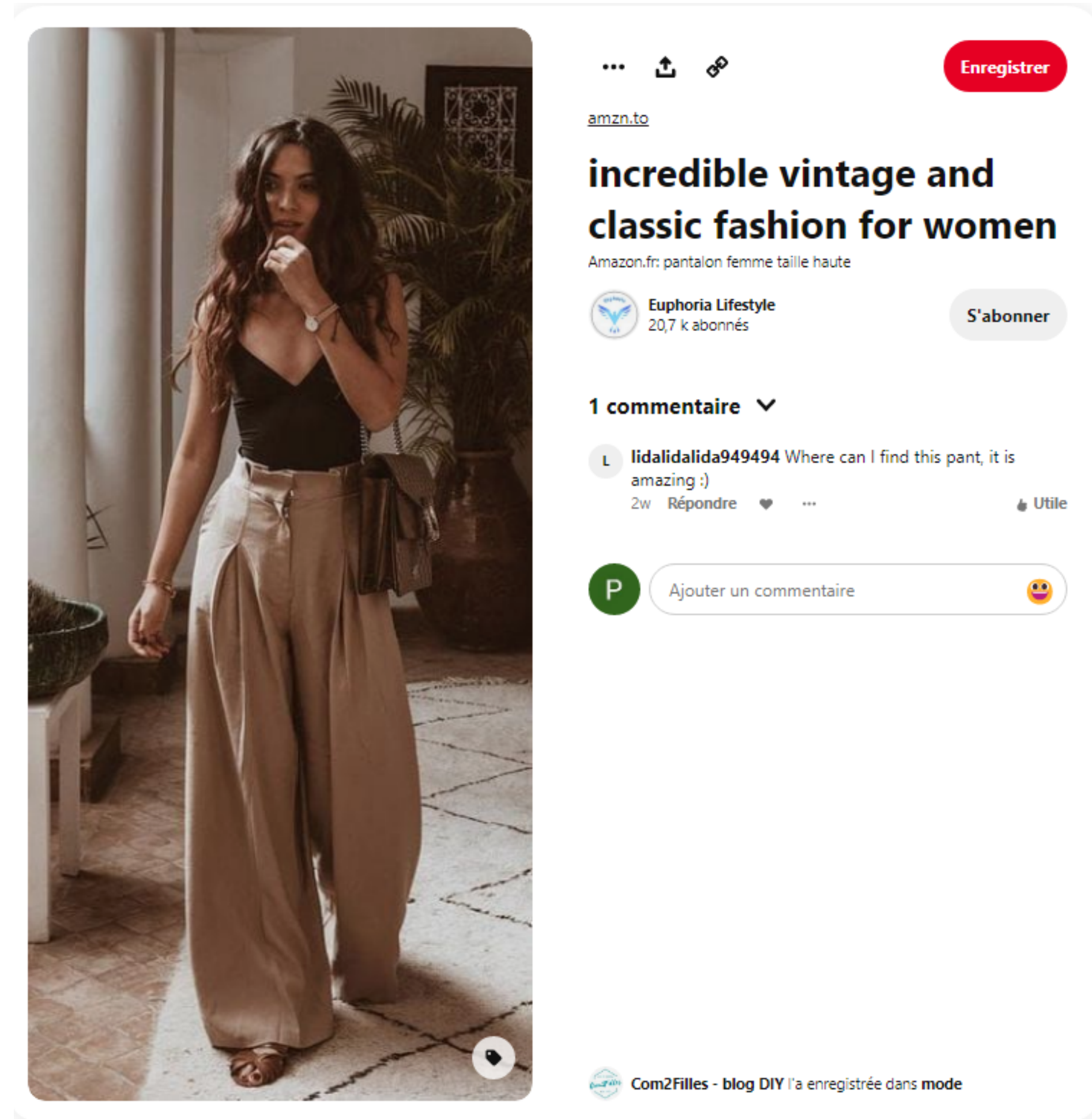
Bazar

Réduire le parcours d'achat client en ligne à un seul
clic



Use Case

Les produits exposés en ligne sont rarement identifiables



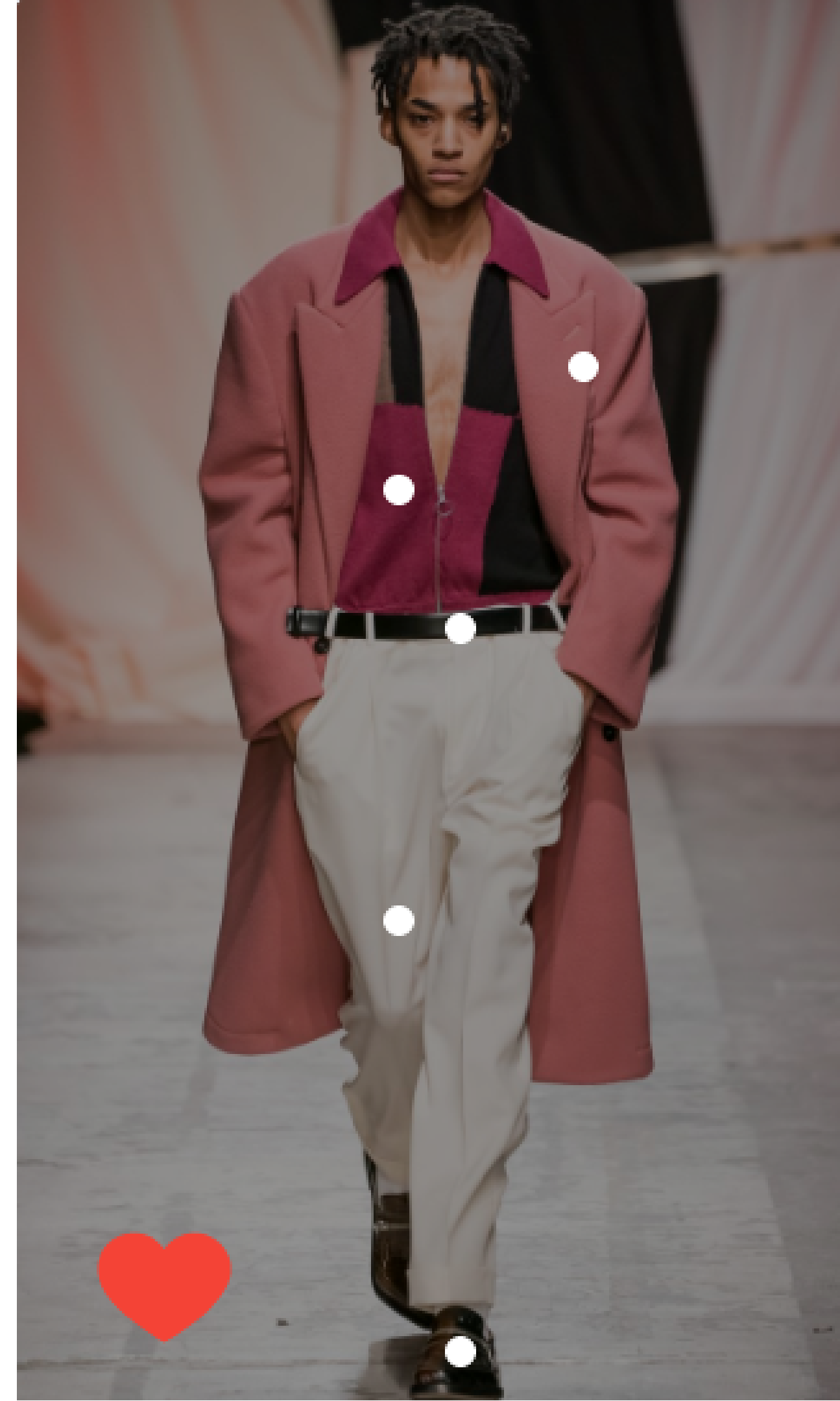
Solution

Bazar en action



Des images et vidéos interactives

Les produits sont référencés sur les images. Lorsque un produit exposé dans un post de réseaux sociaux ou sur une image intéresse un utilisateur, il lui suffit de cliquer dessus pour l'acheter.



Business model

Trois façons d'exploiter le concept

En B2C : Un réseau social

Les utilisateurs précisent les références des vêtements lorsqu'ils éditent un nouveau post.

chictopia.com

En B2B : Des publicités in feed intelligentes.

Concevoir des publicités en ligne interactive.

 invibes

En B2B : Concevoir des sites web.

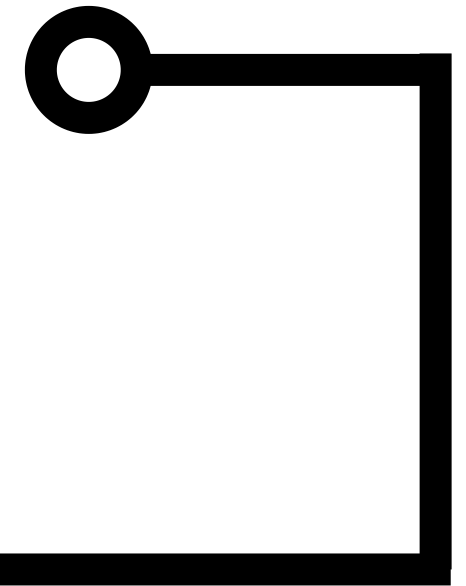
Concevoir des sites web sous forme de galerie photos et vidéos. Lorsque le client souhaite acheter un produit il lui suffit de cliquer sur le produit dans l'image.

 T-Fashion

Social media



post a new
picture/vidéo



the user adds
the reference of the items



AI detection of elements



when the item is clicked on, the
product is added to the user's
"shopping cart".



Stack technique

Front

- React JS
- Redux
- Axios
- Tailwind
- Netlify

Back

- Node Js
- Express
- bcryptjs
- jsonwebtoken
- Heroku

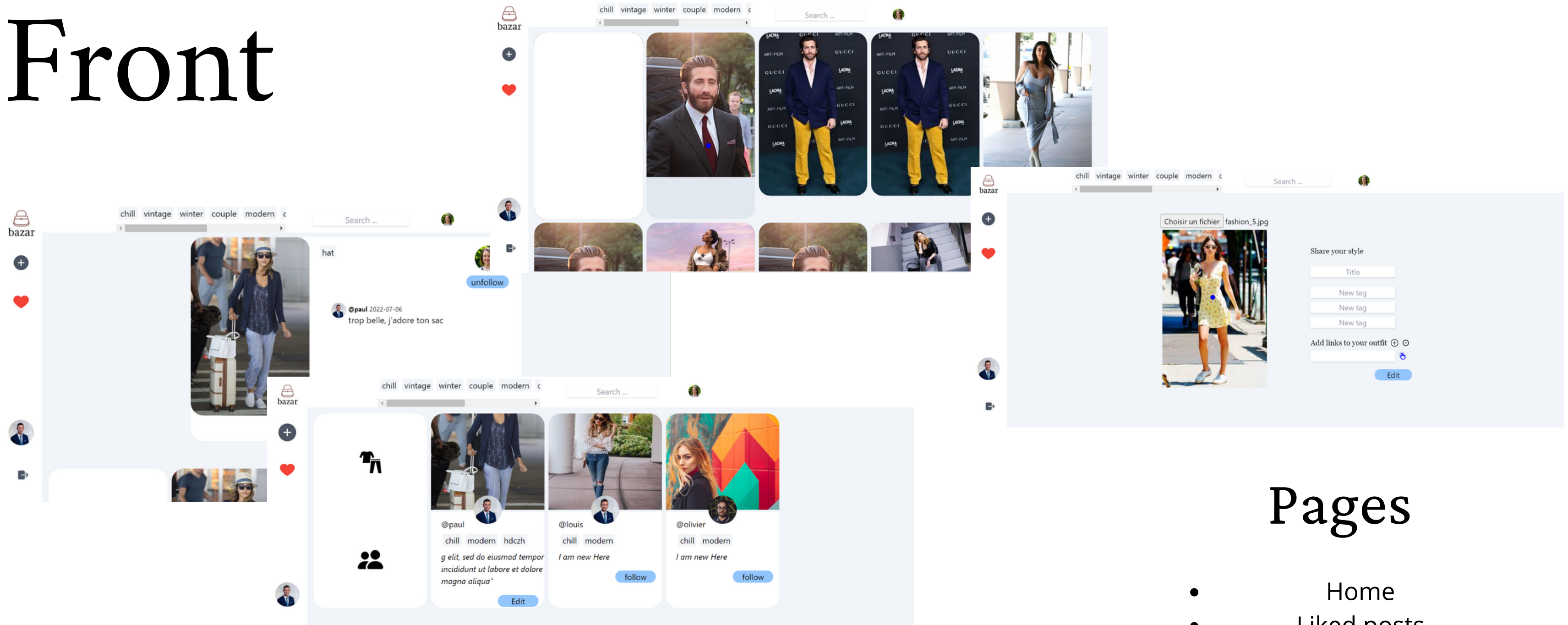
Storage

- Cloud MongoDB
- Amazon S3

IA

- Python
- Torch
- Yolov5
- Flask
- OpenCV

Front

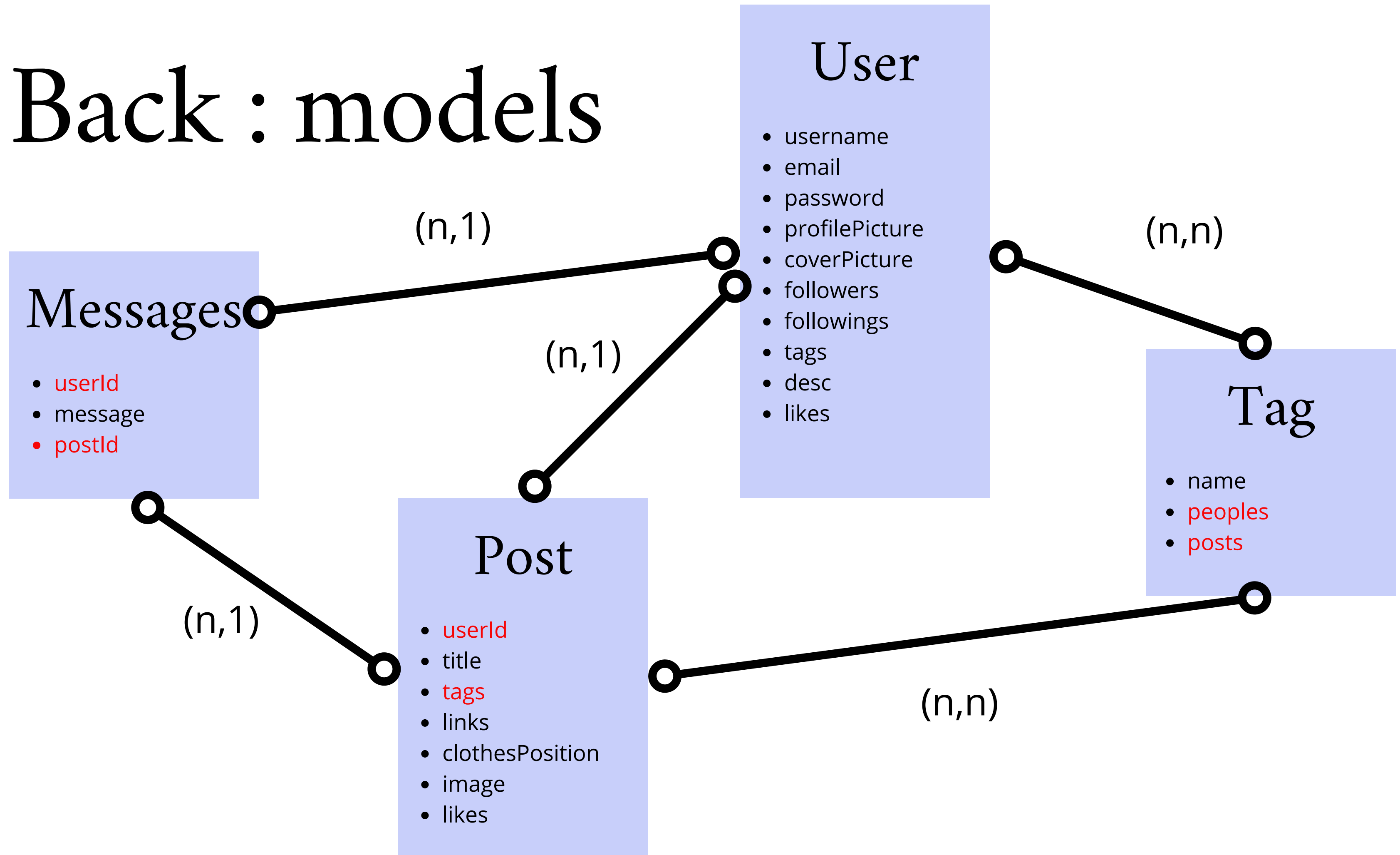


Pages

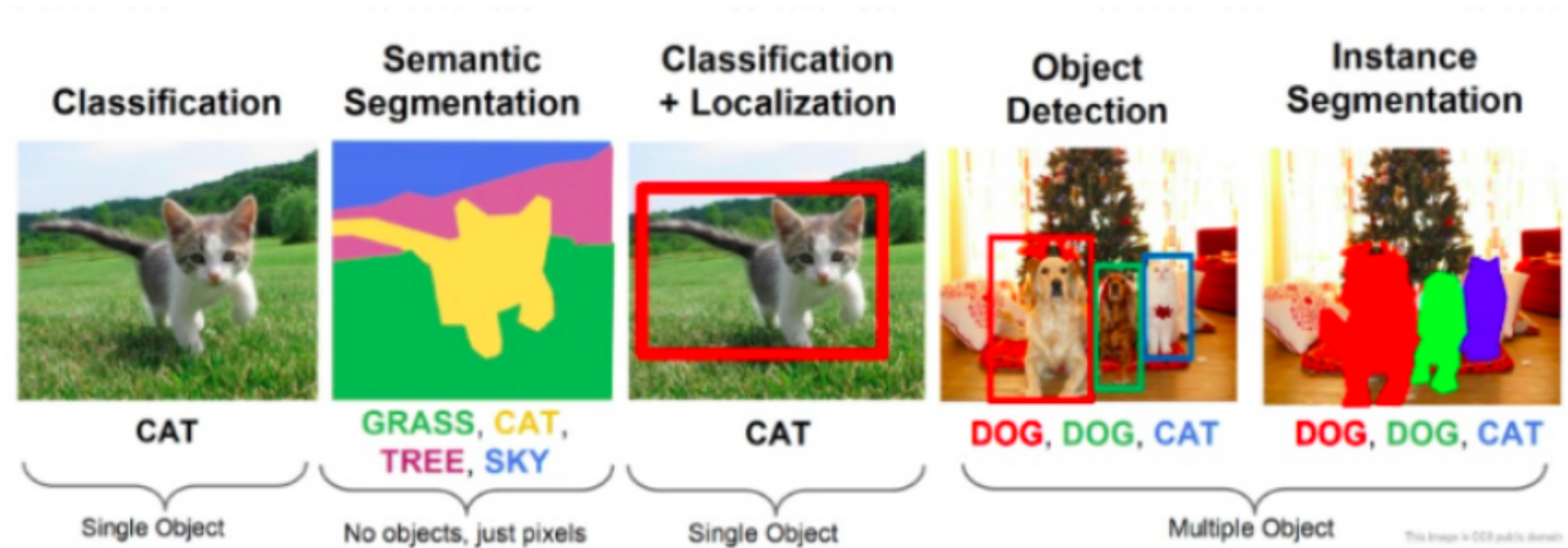
- Home
- Liked posts
- Login
- NewPost
- Pin
- Profile
- Register
- Search
- TrendsWall

link : <https://cool-otter-b6e1a8.netlify.app>
Reload the home page if needed

Back : models



Computer vision tasks



Yolov model

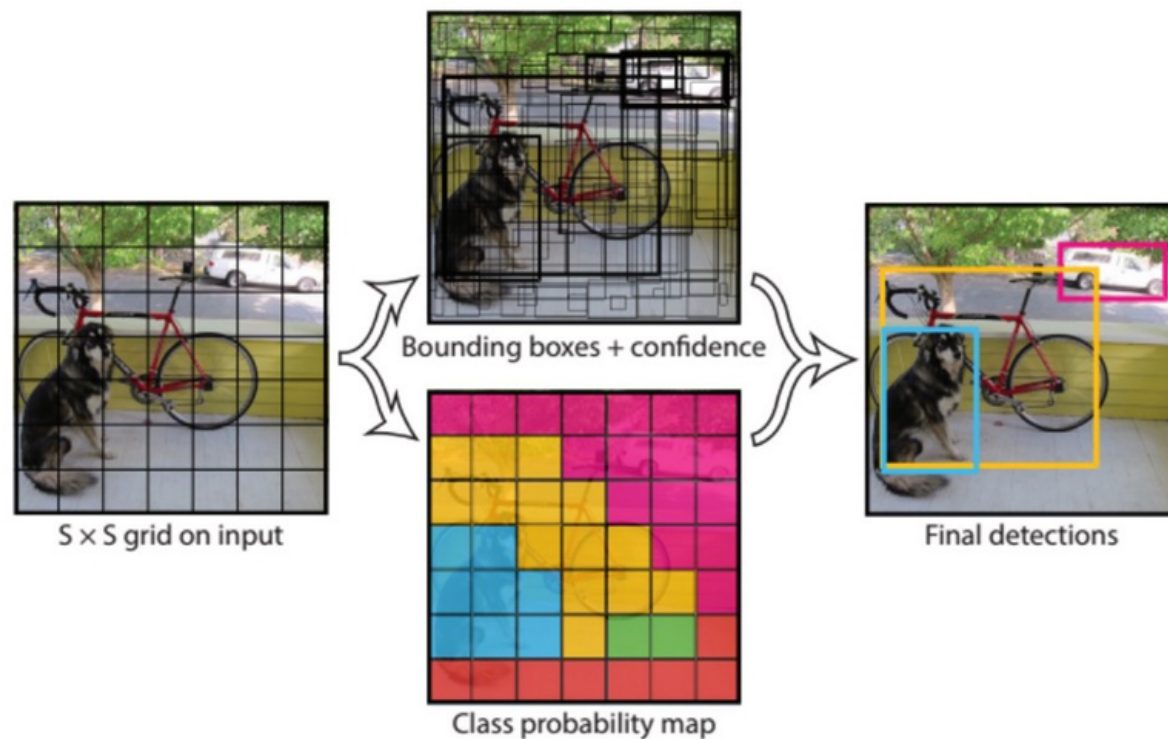


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

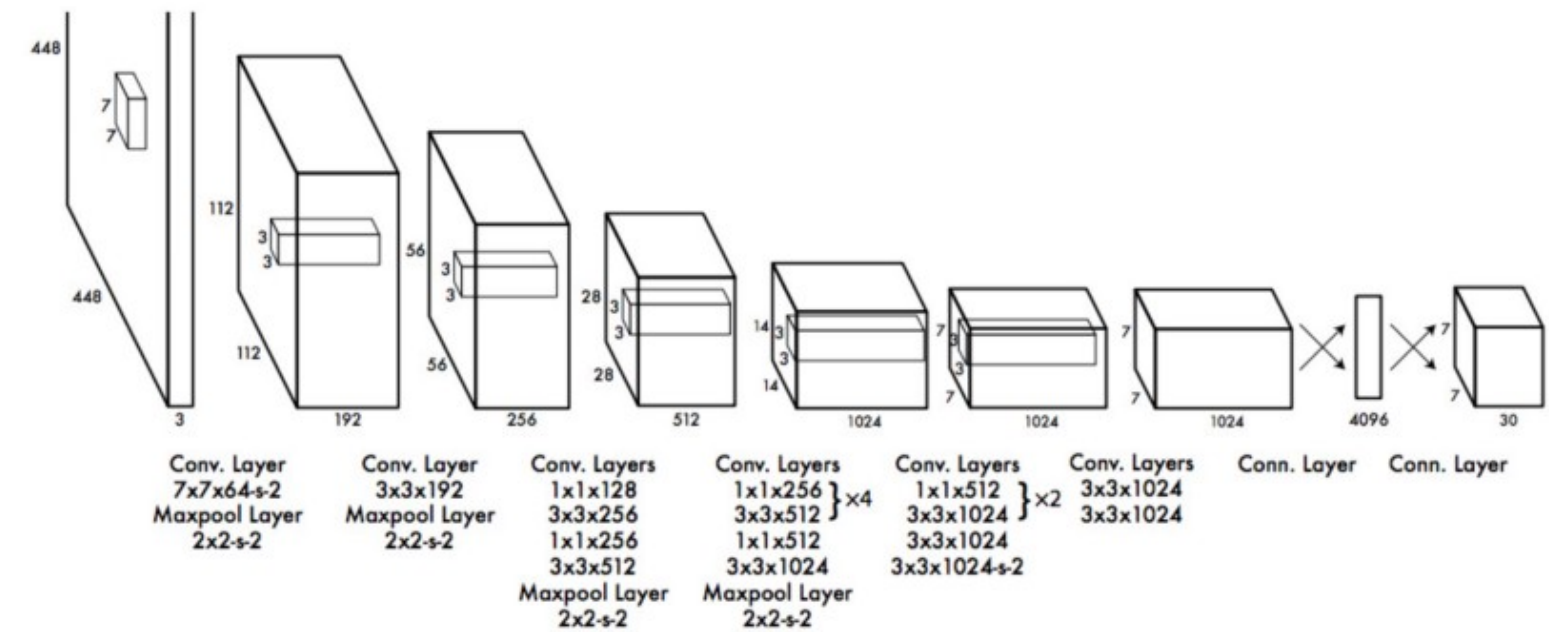
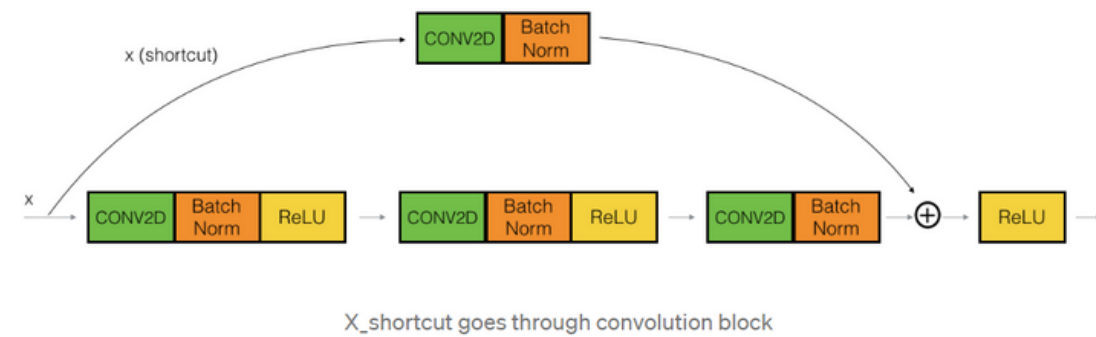


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

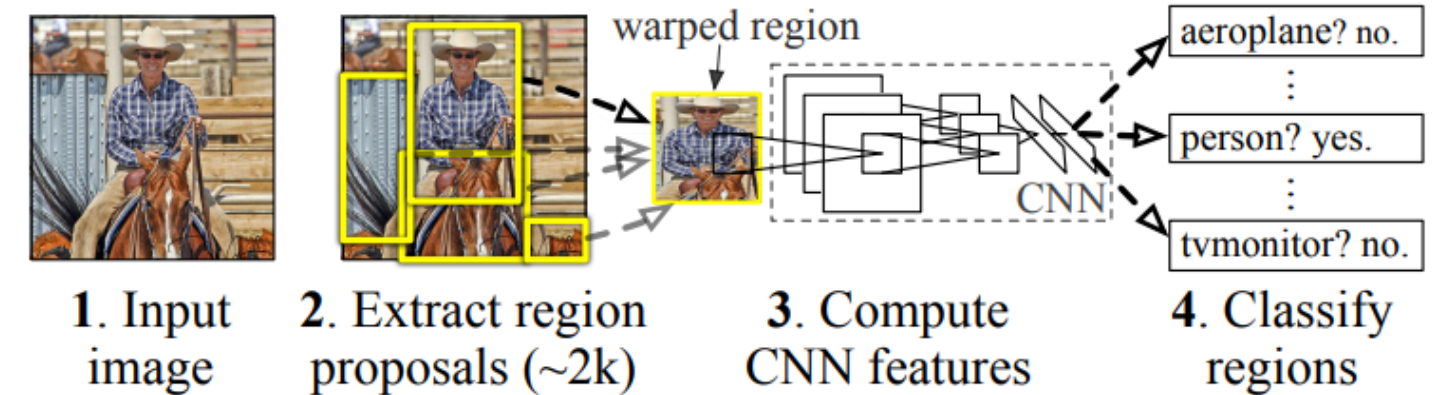
Competing models

R-CNN / Fast R-CNN / Faster R-CNN

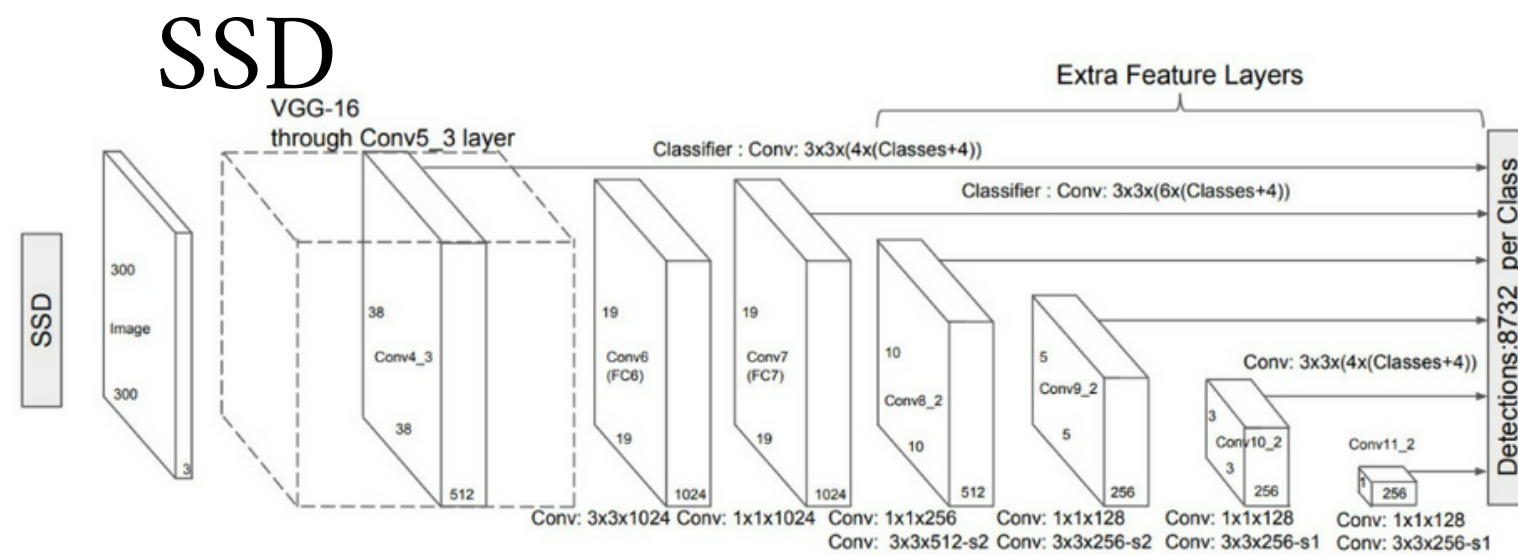


ResNet

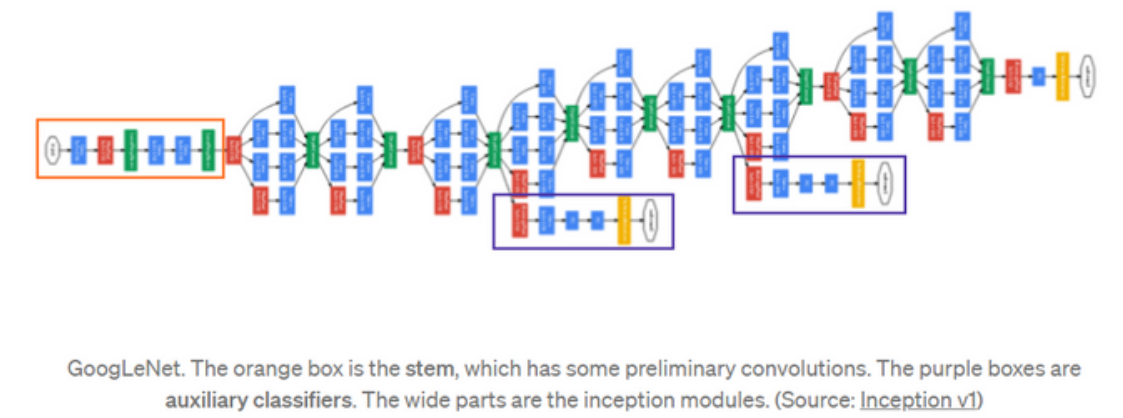
R-CNN: *Regions with CNN features*



Inception



Detectron



Database

DeepFashion

<https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html>

- train : 1400 pictures
- validation : 2000 pictures
- test : 4000 pictures
- Categories : 50 or 4

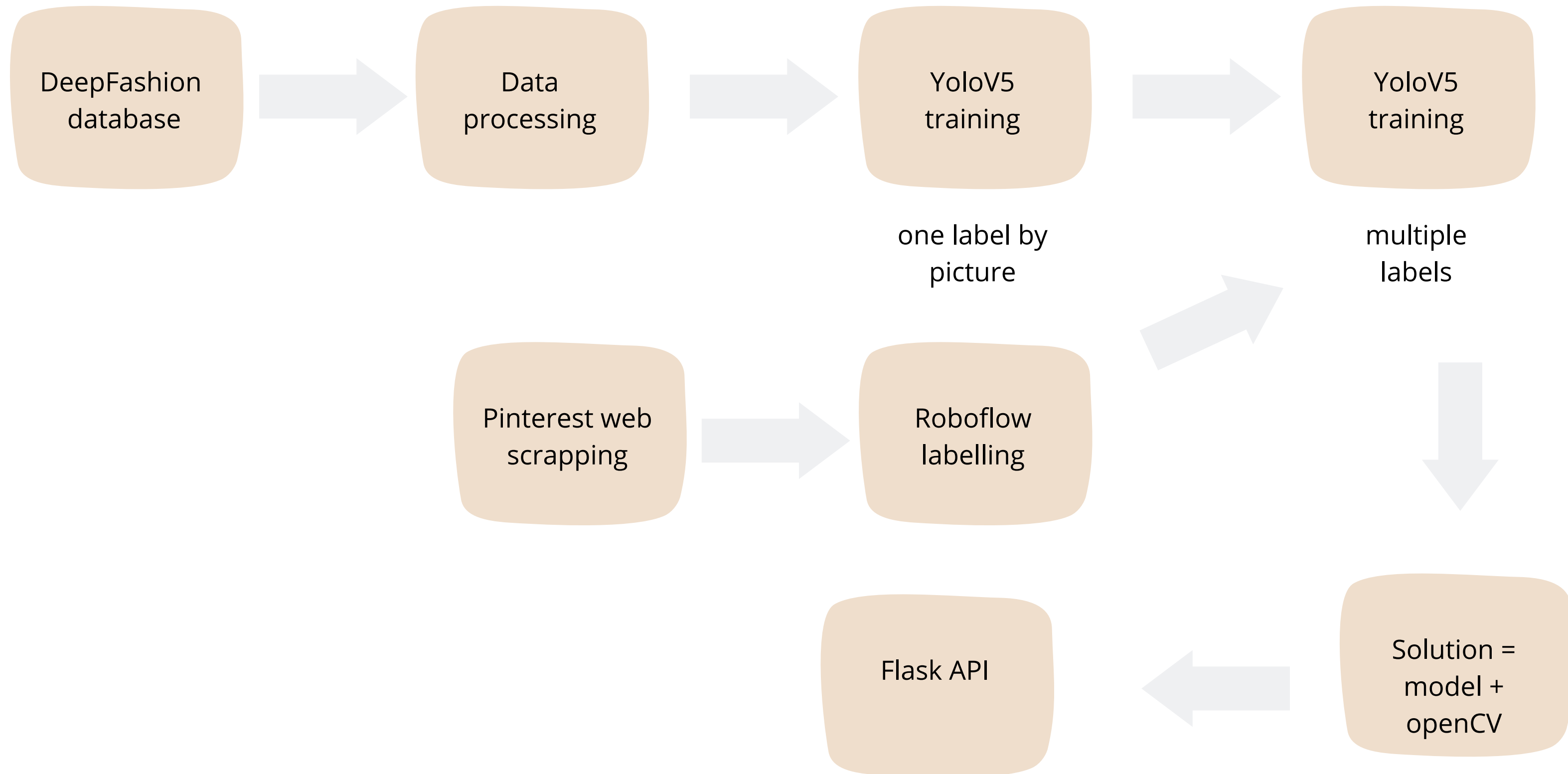
But

- only one label by picture

Homemade database

- ~150 pictures
- Pinterest webscraping + roboflow
- up to 7 labels
- multiple labels

Data pipeline



Other ideas

- Image/vidéo interactive : lorsque l'on clique sur le produit dans l'image, il est ajouté à notre panier
 - Moteur de recherche par photo : trouver dans un catalogue de marques, les produits qui ressemblent à un vêtement que l'on cherche.
 - Proposer des vêtements en fonction des images de l'utilisateur et du contexte (entretien, date ...)
 - Virtual try on : model numérique pour tester les vêtements
-