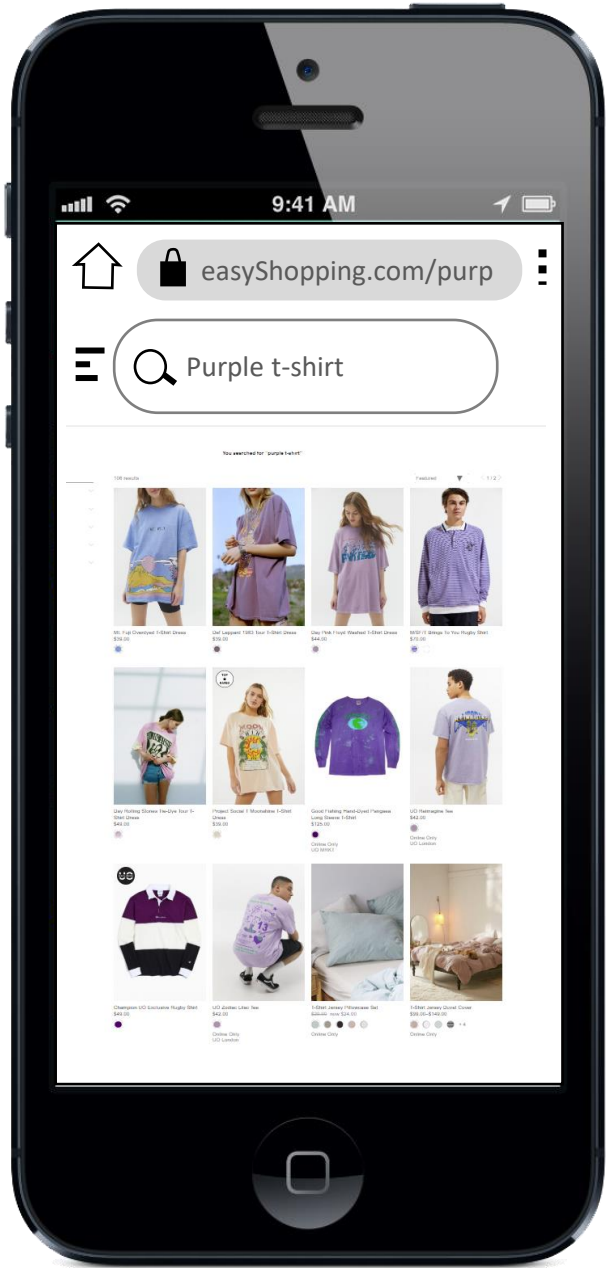


PROJECT 2:

Labelling

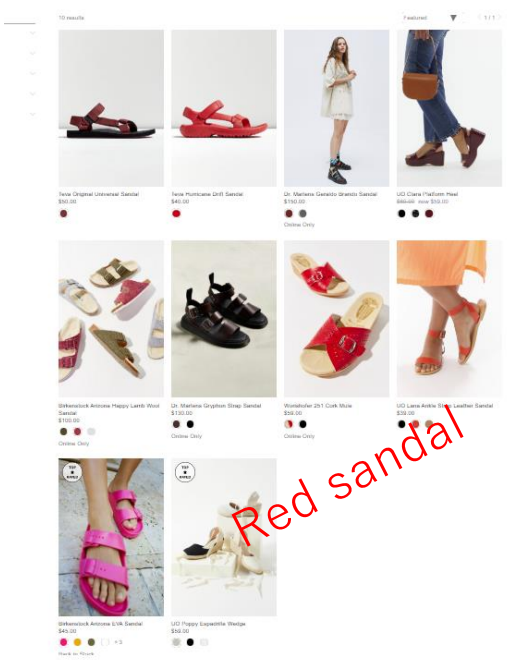
Artificial Intelligence

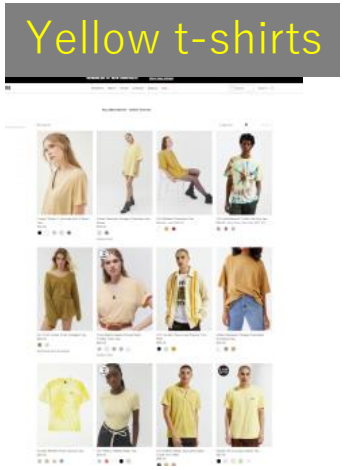
Universitat Autònoma de Barcelona

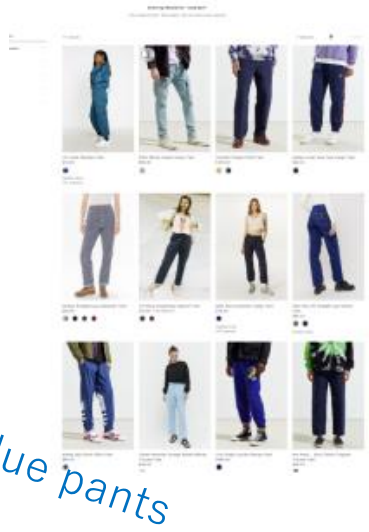
Project 2

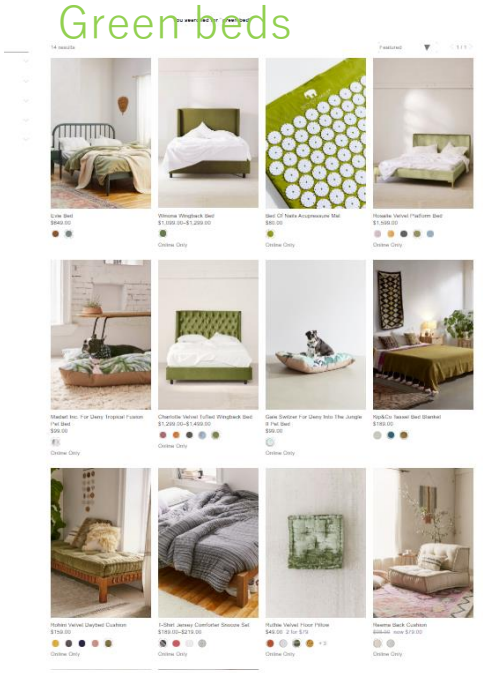
Goal: Building an agent able to automatically label images to provide the ability to make smart searches in natural Language for on a online shop that requires a constant update of the catalogue.

The system should be able to assign two kind of labels to the new products: **Colour and Shape**. Users should be able to search for: *"Red Shirt" or "Black Sandals"*









Project 2

It can be very complex!!! → We will simplify it

Simplifications:

- Labels are going to be in English
- We will only label 8 cloth classes:

  - ✓ *Dresses*
  - ✓ *Flip Flops*
  - ✓ *Jeans*
  - ✓ *Sandals*
  - ✓ *Shirts*
  - ✓ *Shorts*
  - ✓ *Socks*
  - ✓ *Handbags*

- We will label predominant colours for each cloth type, only the 11 universal colour terms:
  - ✓ *Red*
  - ✓ *Orange*
  - ✓ *Brown*
  - ✓ *Yellow*
  - ✓ *Green*
  - ✓ *Blue*
  - ✓ *Purple*
  - ✓ *Pink*
  - ✓ *Black*
  - ✓ *Grey*
  - ✓ *White*

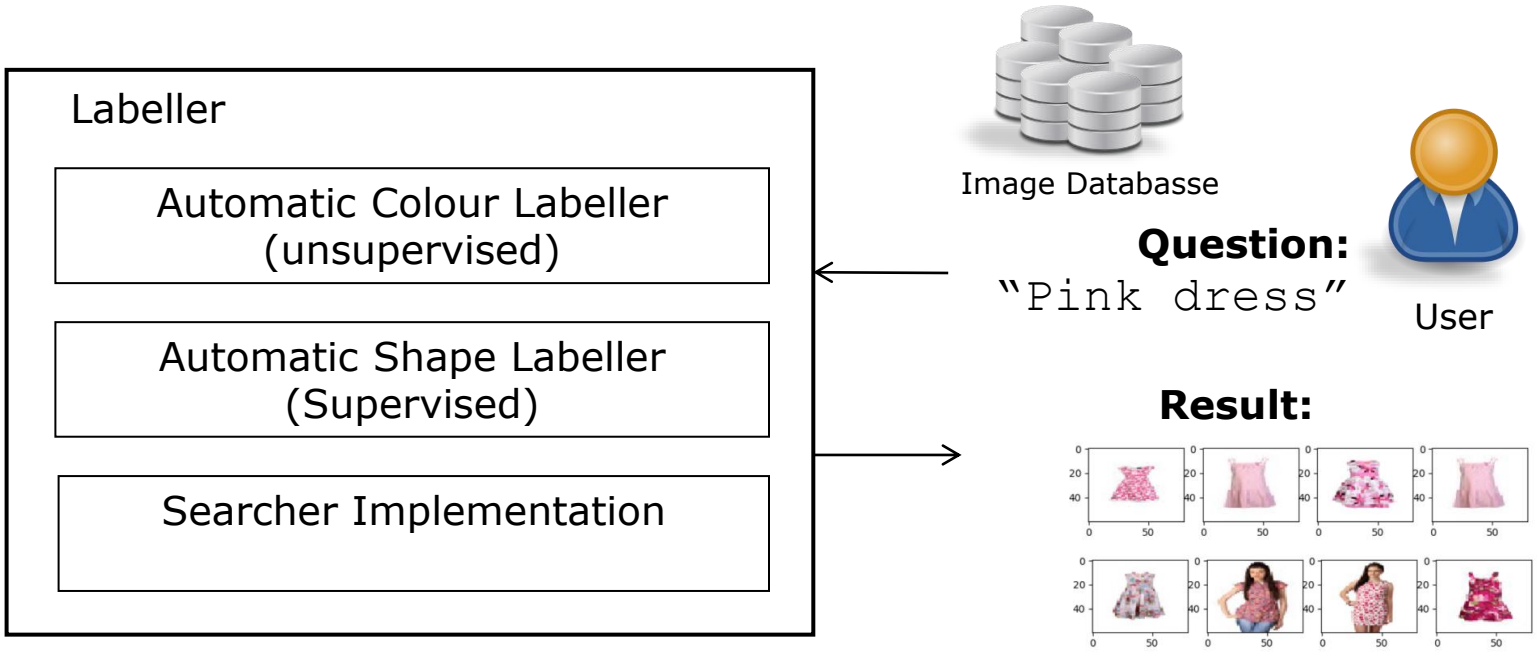
- To reduce running time we will work with low-resolution images (60x80 pixels). We will use the image dataset: Fashion Product Images Dataset del Kaggle

https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset

Kaggle *is a shared folder with datasets for research on Data Science*

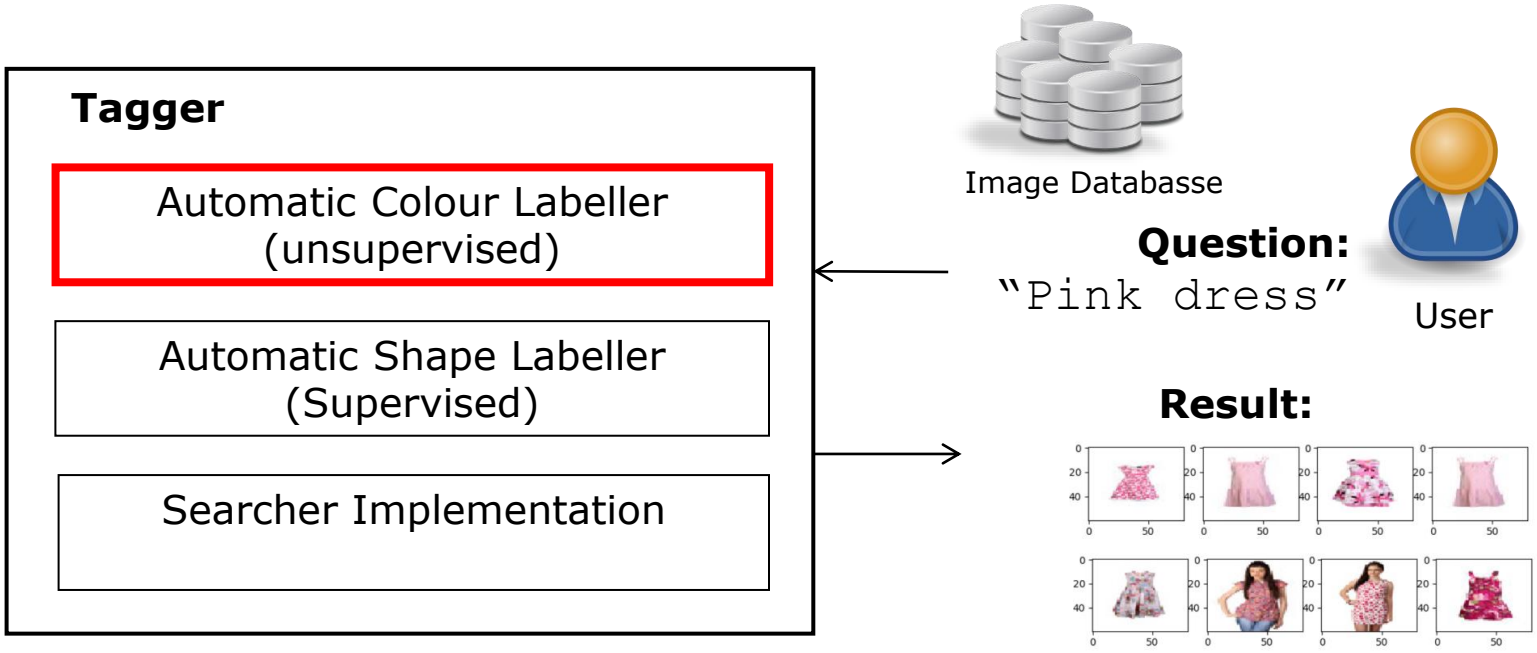
Project 2

Problems to solve to build this tagger:

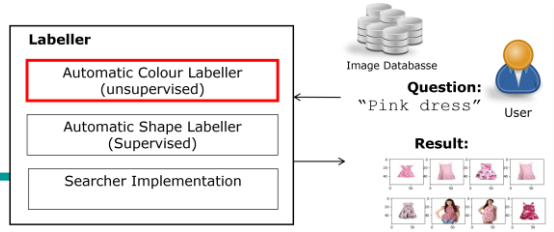


Project 2

Problems to solve to build this tagger:



Project 2: Colour Labelling



How can we label the predominant colours?

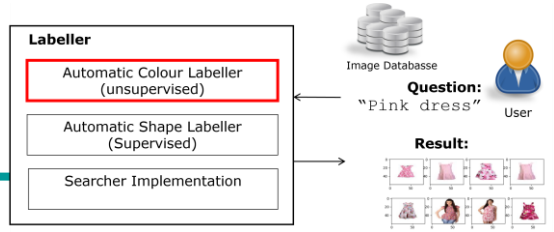
Predominant colour labels:

Yellow, Orange, Blue, Black, Green, White

3 Questions:

- How do we represent colour?
- How can we find the predominant colours of an image?
- How we do assign names to the predominant colours?

Project 2: Colour Labelling

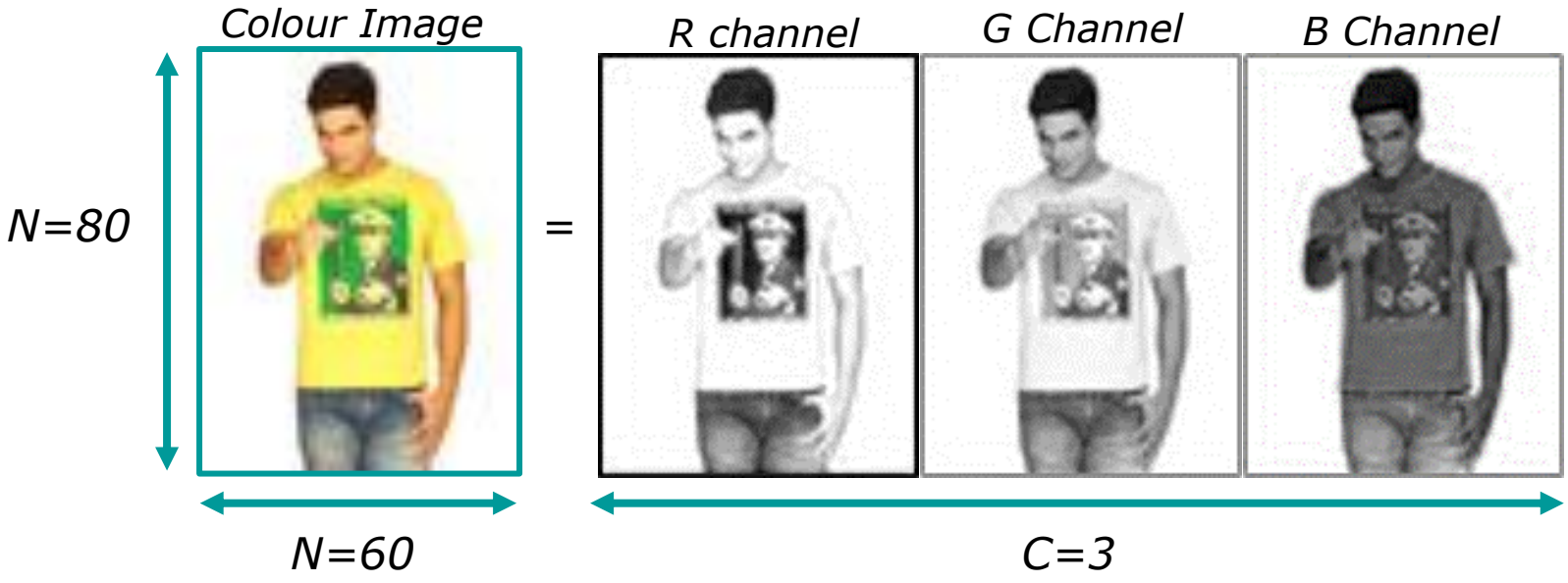


How do we represent colour?

The answer is related to **how an image is represented?**

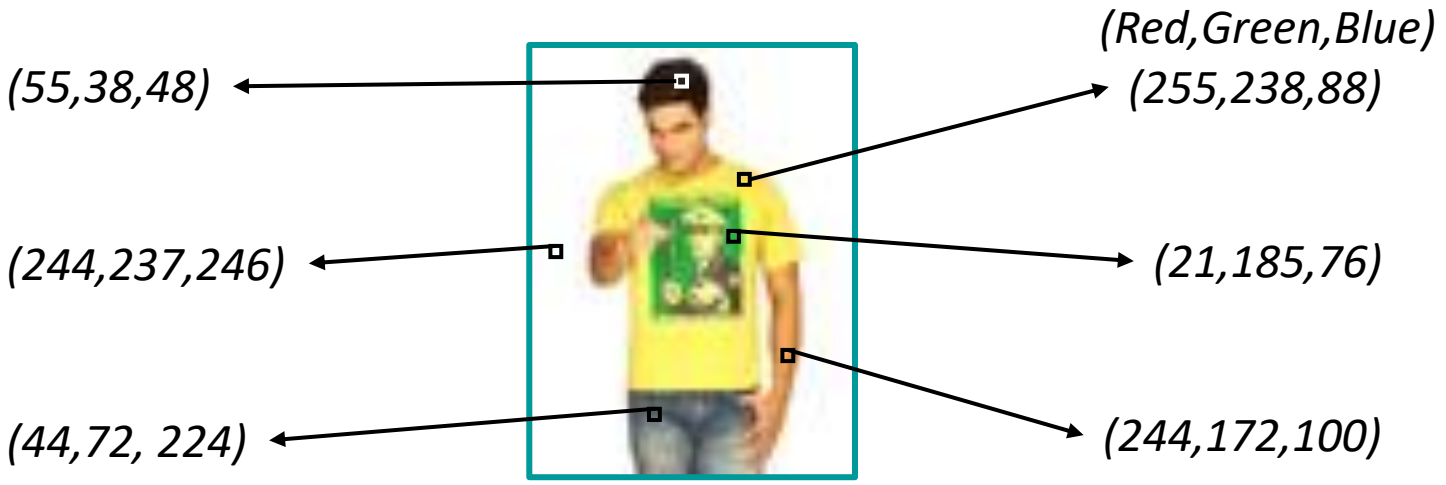
A colour image is a matrix of dimensions: *N x M x C*

Example: *Colour Image 80x60x3 (rows x columns x channels)*
Grey-level Image 80x60x1



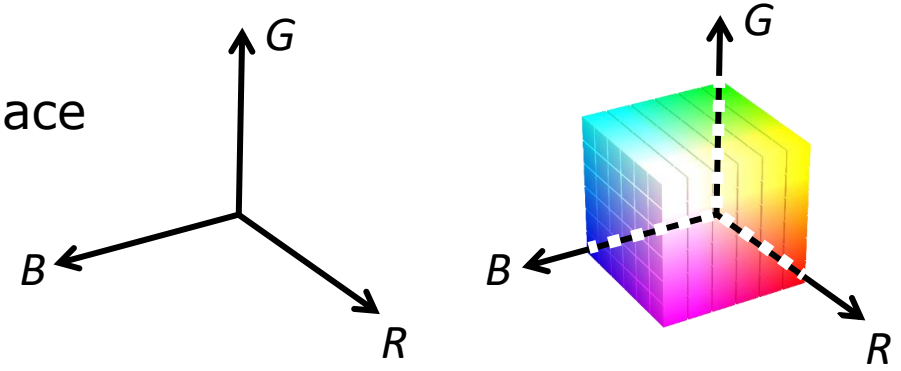
Project 2: Colour Labelling

Let's look at the pixel level:

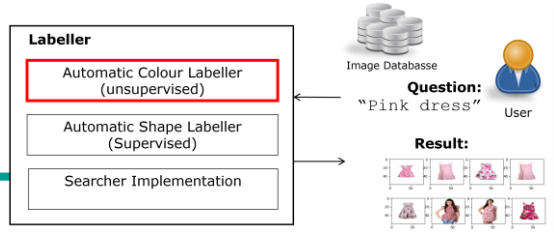


Go back to the initial question: **How is color represented?**

This is a 3-dimensional feature space



Project 2: Colour Labelling



How can we solve the problem of automatic colour labelling?

Labels of the predominant colours:

Yellow, Orange, Blue, Black, Green, White

3 Questions:

- How is colour represented? √
- How can we find the predominant colours of an image?
- How can we assign a name to the predominant colors?

Project 2: Colour Labelling

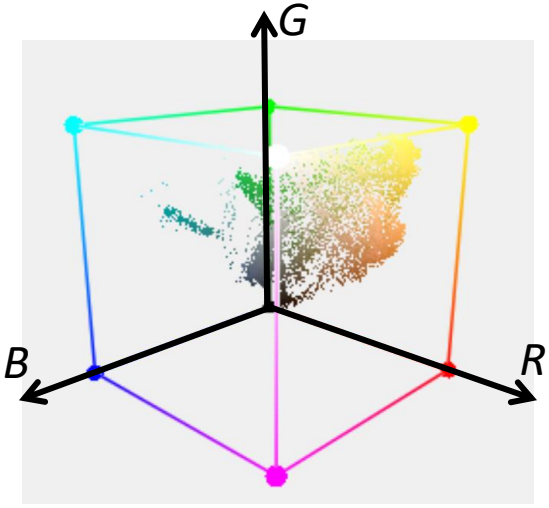
How can we find the predominant colours of an image?

Colour image (N x M x 3)
Nun. of pixels = rows x columns

$$X = \begin{pmatrix} x_R^1 & x_G^1 & x_B^1 \\ \vdots & \ddots & \vdots \\ x_R^n & x_G^n & x_B^n \end{pmatrix}$$

N x M pixels

3 channels



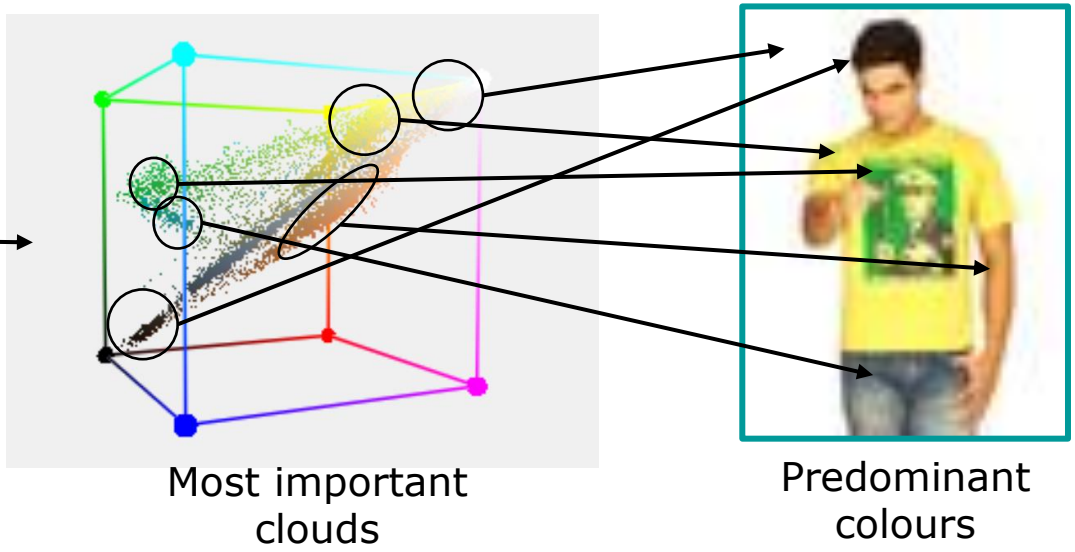
Dot colours are in the RGB colours of each pixel

How can we extract predominant colours?

Project 2: Colour Labelling

Goal: We have a set of points in a three-dimensional space and we need to find the most important clouds in this set.

$$X = \begin{pmatrix} x_R^1 & x_G^1 & x_B^1 \\ \vdots & \ddots & \vdots \\ x_R^n & x_G^n & x_B^n \end{pmatrix}$$



Most important clouds

Predominant colours

Solution: Unsupervised clustering of points

How do we do it? **K-means** algorithm

Project 2: Colour Labelling

In this project the K-means algorithm will be worked on

File: `Kmeans.py`
Class: `Kmeans`

Class parameters `Kmeans`

- `X`: Image we want to analyse.
- `K`: Number of clusters we will use
- `options`: Additional options (centroid initialization method, maximum number of iterations,...)
-

First all necessary variables will be initialized when called:

```
Kmeans(X,K=3,options=None)
```

Finally, the algorithm will be applied until it converges:

```
Kmeans.fit()
```

Obtained centroids will be stored at the variable `centroids`

Project 2: Colour Labelling

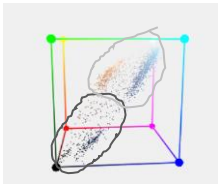
Example: K-means application for different K values

Original Image

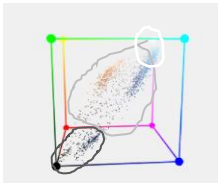
Result

Pixels assigned in each centroid

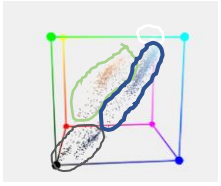
K=2



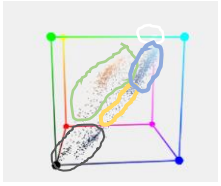
K=3



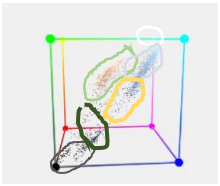
K=4



K=5



K=6



Note: This view is given by the function **`visualize_k_means()`**

Project 2: Colour Labelling

K-means problem: Which k is the best?

In theory lectures, we saw some ideas to choose the best k:

You can estimate a **Quality measurement of a given classification**, and study how it varies for different numbers of classes *(k=2, 3, 4, ...)*

(Usually this study is based on an analysis of class variance)

Some interesting statistics:

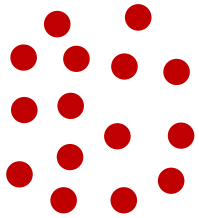
- Intra-class distance → **we will use this one!!**
- Inter-class distance
- Fisher's discriminant

Project 2: Colour Labelling

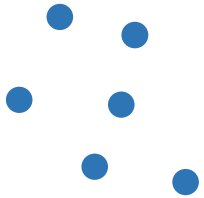
REMINDER from THEORY

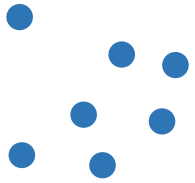
Intra-class distance

Example of 2 classes resulting from k-means:



Compact class
Good classification





Non-compact class
Bad classification

Estimation: Sum for all classes, the average distances between all the pairs of points of a class

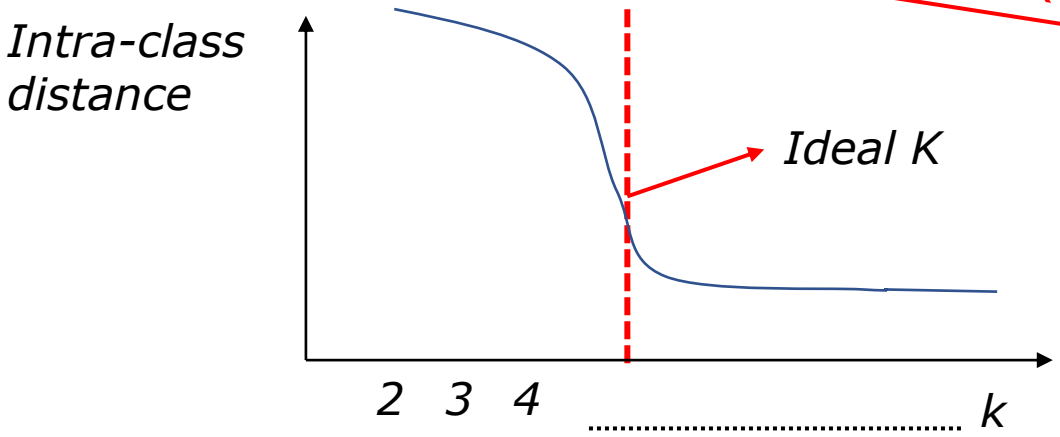
$$D(C) = \frac{2}{m(m-1)} \sum_{j=1}^{m} \sum_{i=j+1}^{m} d\left(\vec{x}^i, \vec{x}^j\right) : \vec{x}^i, \vec{x}^j \in C, i, \text{j}: 1 \dots m$$

$$\sum_{i=1}^{k} D(C_i) \longrightarrow$$ **it's good that it's small !!**

Project 2: Colour Labelling

Study of intra-class distance:

REMINDER from THEORY



In our case:

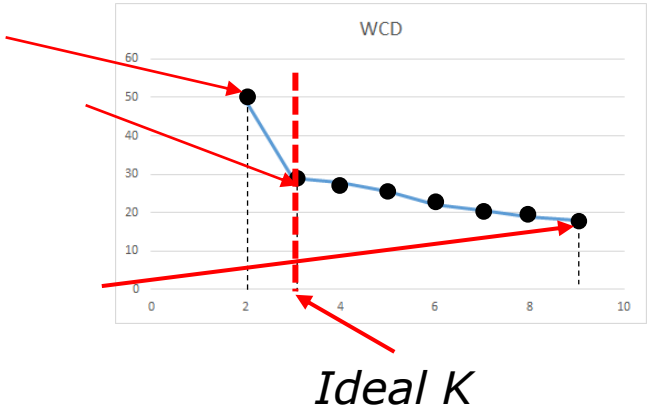
Given an image

`withinClassDistance()` for K=2

`withinClassDistance()` for K=3

⋮

`withinClassDistance()` for K=9



Project 2: Colour Labelling

Study of intra-class distance:

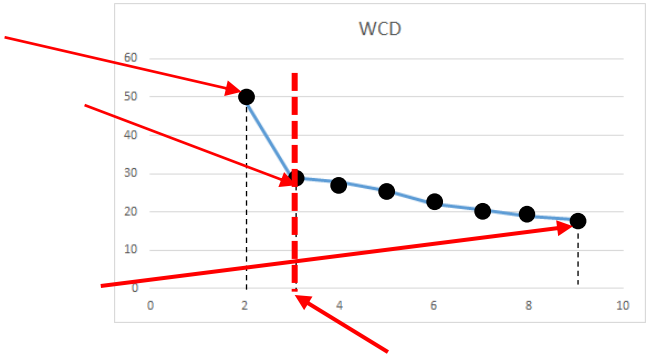
Given an image

`withinClassDistance()` for K=2

`withinClassDistance()` for K=3

⋮

`withinClassDistance()` for K=9



How we can find it ? ← *k ideal*

We can calculate the % of Decrement:

$$\%DEC = 100\frac{WCD_k}{WCD_{k-1}}$$

A possible threshold is to take the k from which

$$100 - \%DEC < 20\% \ (exemple)$$

K	WCD	%DEC	100-%DEC
2	49.09		
3	29.11	59.29	40.71
4	27.95	96.03	3.97
5	25.68	91.86	8.14
6	22.00	85.70	14.30
7	20.61	93.65	6.35
8	18.82	91.31	8.69
9	18.09	96.15	3.85

→ *Ideal K*

→ $< 20\%$

Project 2: Colour Labelling

Problem: Which k is the best?

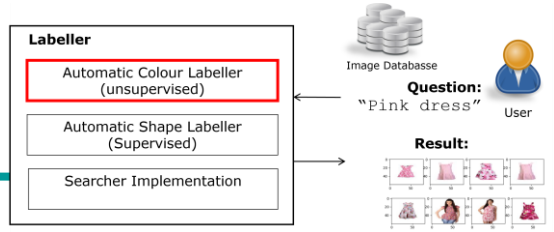
To compute intra-class distance you will program the function:

`whitinClassDistance()`
Input: self
Output: valor `WCD`

Per a seleccionar la millor k programareu la funció:

`find_bestK()`
Input: self, `max_K`
Output: `K`

Project 2: Colour Labelling



How can we solve the problem of automatic colour labelling?

Labels of the predominant colours:

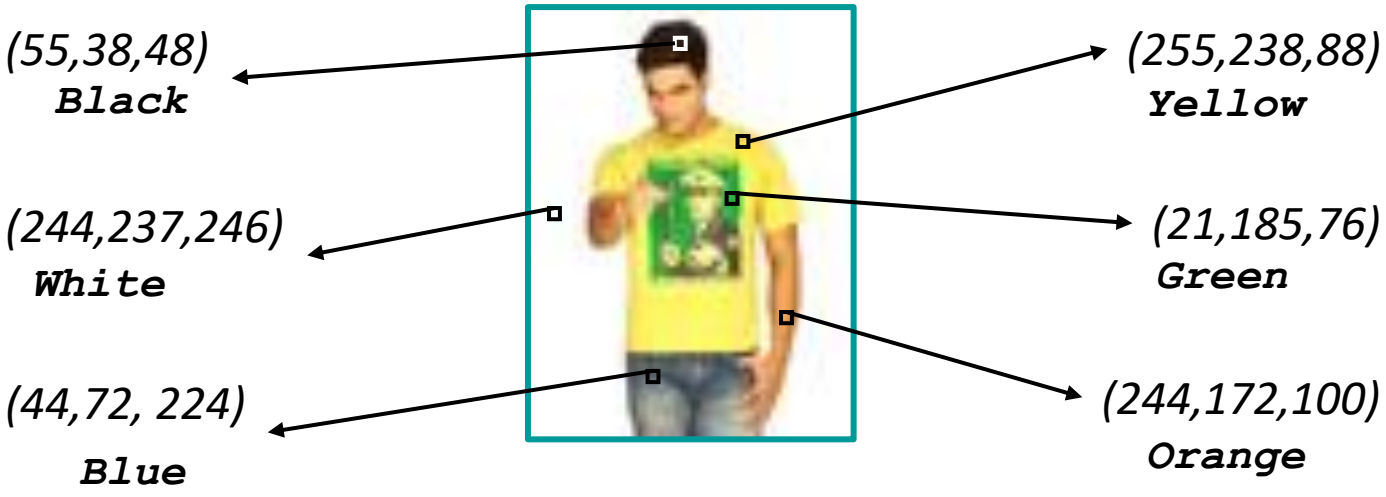
Yellow, Orange, Blue, Black, Green, White

3 Questions:

- How is colour represented? ✓
- How can we find the predominant colours of an image? ✓
- How can we assign a name to the predominant colours? ↙

Project 2: Colour Labelling

How can we assign a name to the predominant colors?



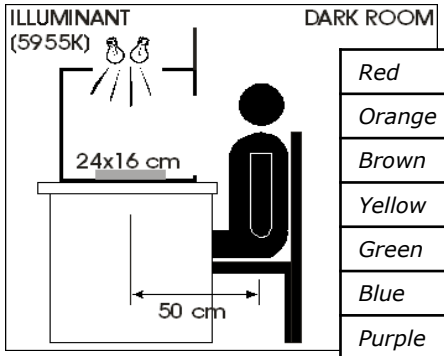
This problem requires simulating how humans perceive color !!!

Project 2: Colour Labelling

This problem has already been solved in a multidisciplinary way:

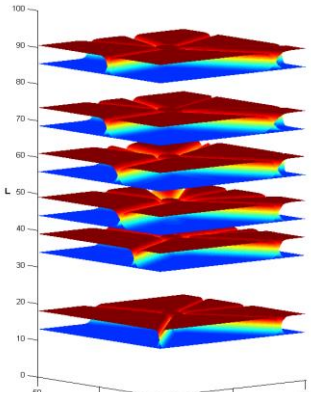
Experiments in Anthropology + Experiments in Experimental Psychology + Mathematical Models in Computer Vision

Studies on 78 languages have shown that <u>there are 11 universal basic color names</u> shared by the most evolved languages



Red	0
Orange	0
Brown	0
Yellow	0
Green	5
Blue	5
Purple	0
Pink	0
Black	0
Grey	0
White	0

Distribute 10 point to 11 basic colors



$$(R, G, B) \rightarrow (P_{Red}, \ldots, P_{White})$$

Code available at:
http://www.cvc.uab.cat/colour_naming

Berlin, B., & Kay, P. (1991) Basic color terms: Their universality and evolution. Univ of California Press.

R. Benavente, M. Vanrell, R. Baldrich (2006) A dataset for fuzzy color naming, Color Research and Applications

R. Benavente, M. Vanrell, R. Baldrich (2008) Parametric fuzzy sets for automatic color naming, Journal of the OSA.

We will use this code!!!

Project 2: Colour Labelling

How we can assign a name to the predominant colors?

Using the results of the previous works, we will move from the RGB space to the space of the 11 color names:

$$(R, G, B) \rightarrow (P_{Red}, P_{Orange}, P_{Brown}, P_{Yellow}, P_{Green}, P_{Blue}, P_{Purple}, P_{Pink}, P_{Black}, P_{Grey}, P_{White})$$

for each RGB returns a vector of 11 probabilities of a human assigning each of the color names.

The code of this conversion is given to you:

Function: `get_color_prob()`

File: `utils.py`

To assign labels to all the predominant colors, you will program:

Function: `get_color()`

File: `kmeans.py`

Project 2: Colour Labelling

Example:

centroid

[241,184,94]

(R , G , B)

Get_color_prob([241,184,94])

[0 ,0.0562, 0 , 0.943, 0 , 0 , 0 , 0 , 0 , 0 , 0]

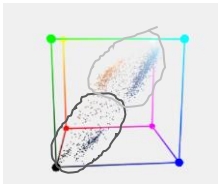
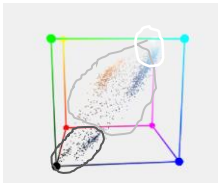
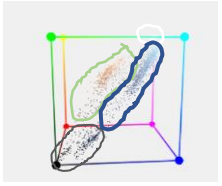
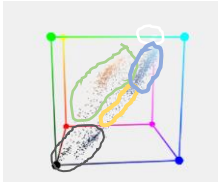
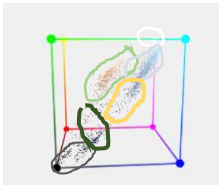
(Red, Orange, Brown, Yellow , Green, Blue, Purple, Pink , Black, Grey, White)

Probability((241, 184, 94)=Yellow) = 0.943

Maximum

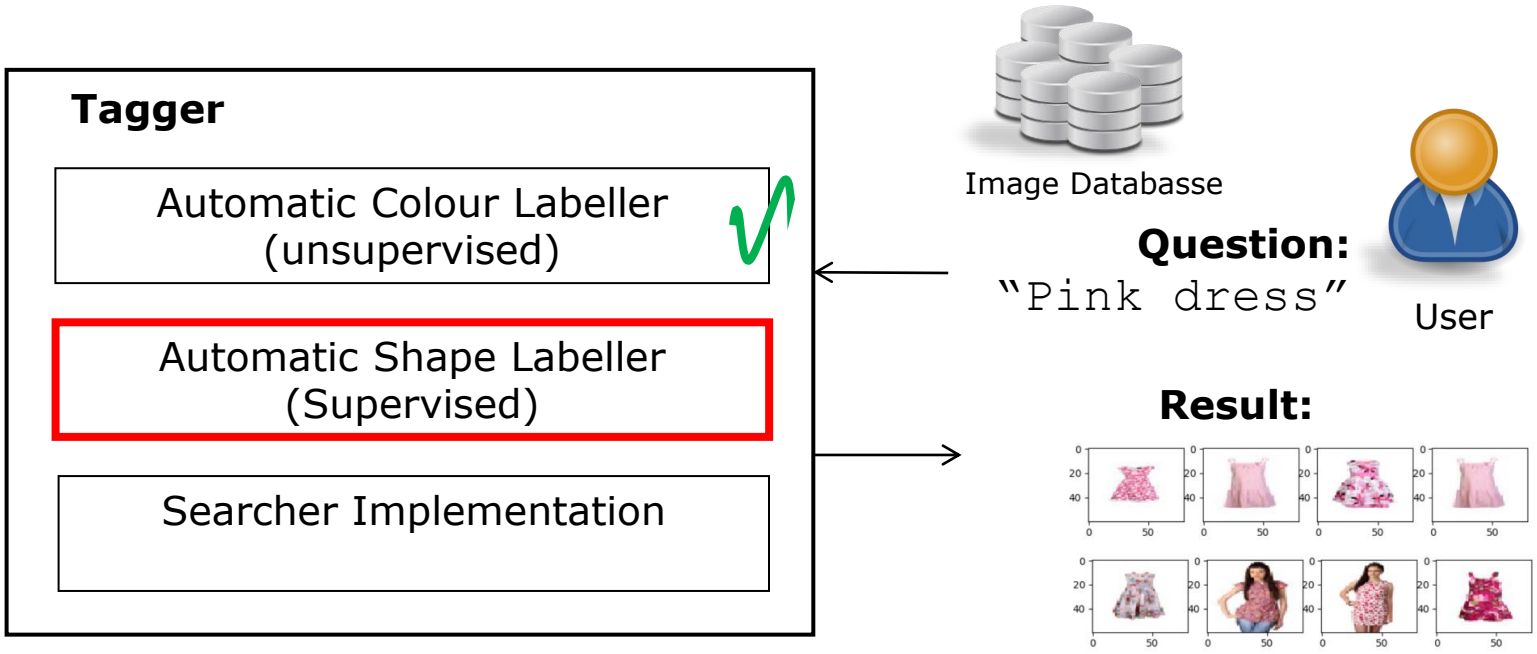
Project 2: Colour Labelling

Example: application of labels for different results

Input image		Result	Assigned Pixels to centroids	Centroid labels
	K=2			*[Grey, Grey]*
	K=3			*[White, Grey, Black]*
	K=4			*[White, Blue, Orange, Black]*
	K=5			*[White, Blue, Purple Orange, Black]*
	K=6			*[White, Blue, Purple, Brown, Orange, Black]*

Project 2

Problems to solve to build this tagger:



Project 2: Shape Labelling

How can we solve the problem of automatically labelling clothes?

Shape Label:

Shirt

3 Questions:

- How can we represent the shape of clothes?
- How can we learn to classify clothes?
- How we assign the clothing type label to a new image?

Project 2: Shape Labelling

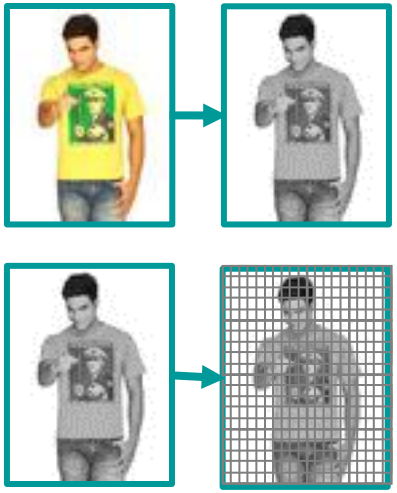
How can we represent the shape of clothes?

What **feature space** we could use to represent the shape of clothes?

This is **a computer vision problem** that since we do not know enough, we will solve it in a very simple way as follows:

1) We will remove color, since we do not need it to represent the shape

2) We will take the pixels of the image directly as the feature of each position of the image.



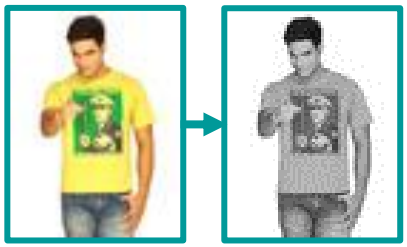
Project 2: Shape Labelling

How can we represent the shape of clothes?

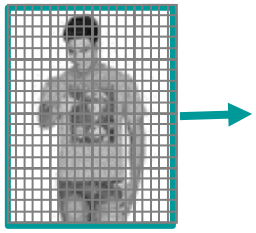
Extracting **image shape features**:

1) Removing colour, since we do not need it to represent the shape

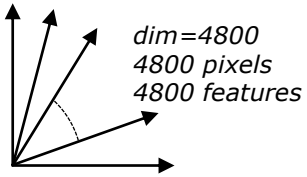
$$(R, G, B) \rightarrow \left(\frac{R+G+B}{3}, \frac{R+G+B}{3}, \frac{R+G+B}{3}\right)$$



2) We will take the pixels of the image directly as the feature of each position of the image.



$(1,0,0.5,\ldots,1) \rightarrow$ *80x60 pixel vector*
dimension=4800



Project 2: Shape Labelling

How can we solve the problem of automatically labelling clothes?

Shape Label:

Shirt

3 Questions:

- How can we represent the shape of clothes? ✓
- How can we learn to classify clothes?
- How we assign the clothing type label to a new image?

Project 2: Shape Labelling

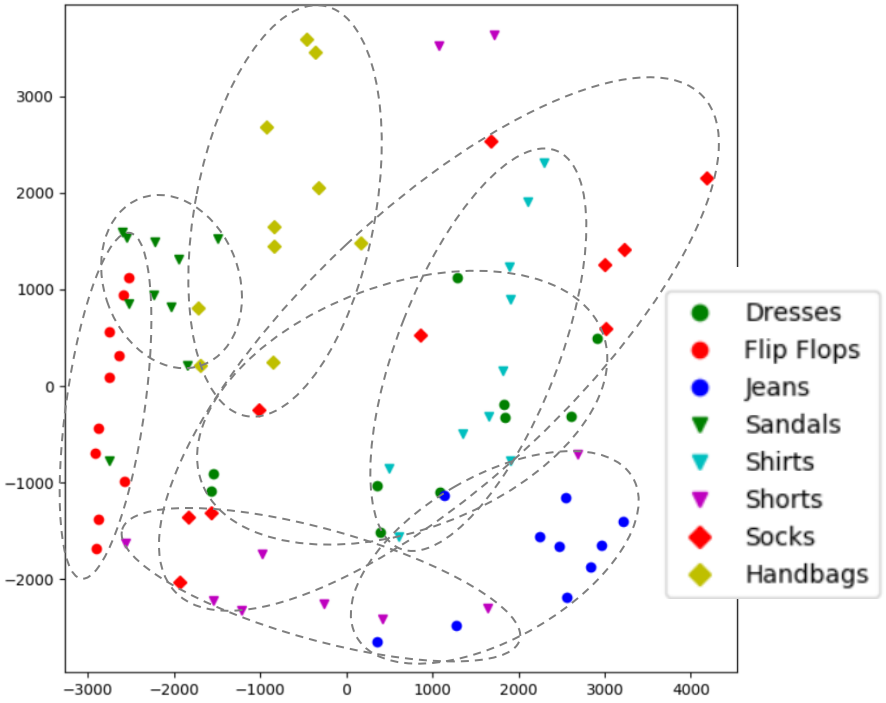
How can we learn how to classify clothes?

Given the sample that we will use as a learning set, we can visualize this space of 4800 dimensions to an observable space of 2 dimensions:

$$X = \begin{pmatrix} x_1^1 & \cdots & x_{4800}^1 C_1 \\ \vdots & \ddots & \vdots \\ x_1^{n_1} & \cdots & x_{4800}^{n_1} C_1 \\ x_1^1 & \cdots & x_{4800}^1 C_2 \\ \vdots & \ddots & \vdots \\ x_1^{n_2} & \cdots & x_{4800}^{n_2} C_2 \\ & \vdots \quad \vdots & \vdots \\ x_1^1 & \cdots & x_{4800}^1 C_k \\ \vdots & \ddots & \vdots \\ x_1^{n_k} & \cdots & x_{4800}^{n_k} C_k \end{pmatrix}$$

Principal Component Analysis (PCA)

Change to the basis of the eigen vectors and take the first 2 dimensions



Observation of clusters despite the high loss of information (4800 to 2)

Project 2: Shape Labelling

How can we learn to classify clothes?

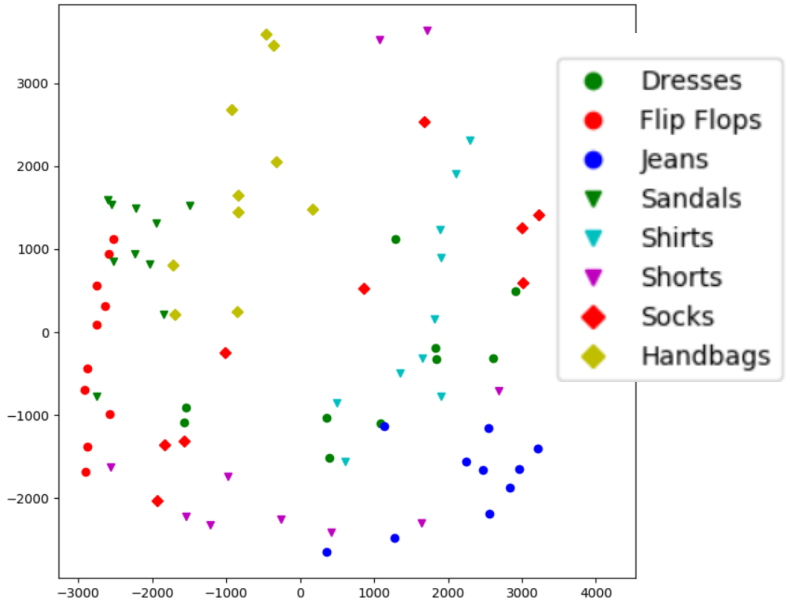
In theory lectures we have seen different families of classifiers:

- Linear classifier
- Nonlinear classifier
- Probabilistic classifier

When the data present a clear model (linear, non-linear, probabilistic, …)

- Nearest k-neighbor classifier (KNN)

When there is no clear model



Project 2: Shape Labelling

How can we solve the problem of automatically labelling clothes?

Shape Label:

Shirt

3 Questions:

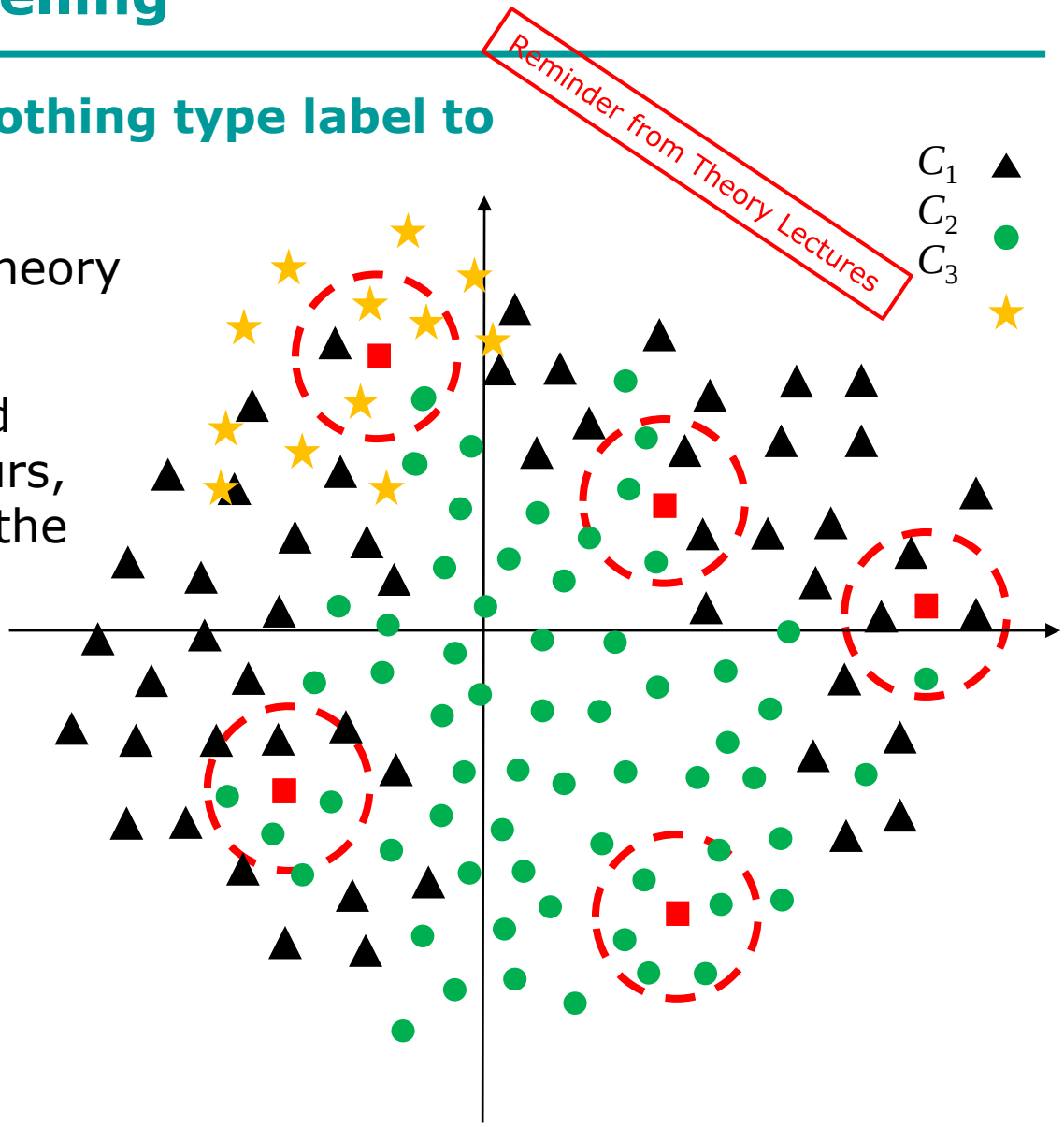
- How can we represent the shape of clothes? ✓
- How can we learn to classify clothes? ✓
- How we assign the clothing type label to a new image?

Project 2: Shape Labelling

How do we assign the clothing type label to a new image?

K-NN Algorithm seen in theory lectures,

Idea: The decision is based on the closest neighbours, considering what class the closest N-neighbours belongs to.

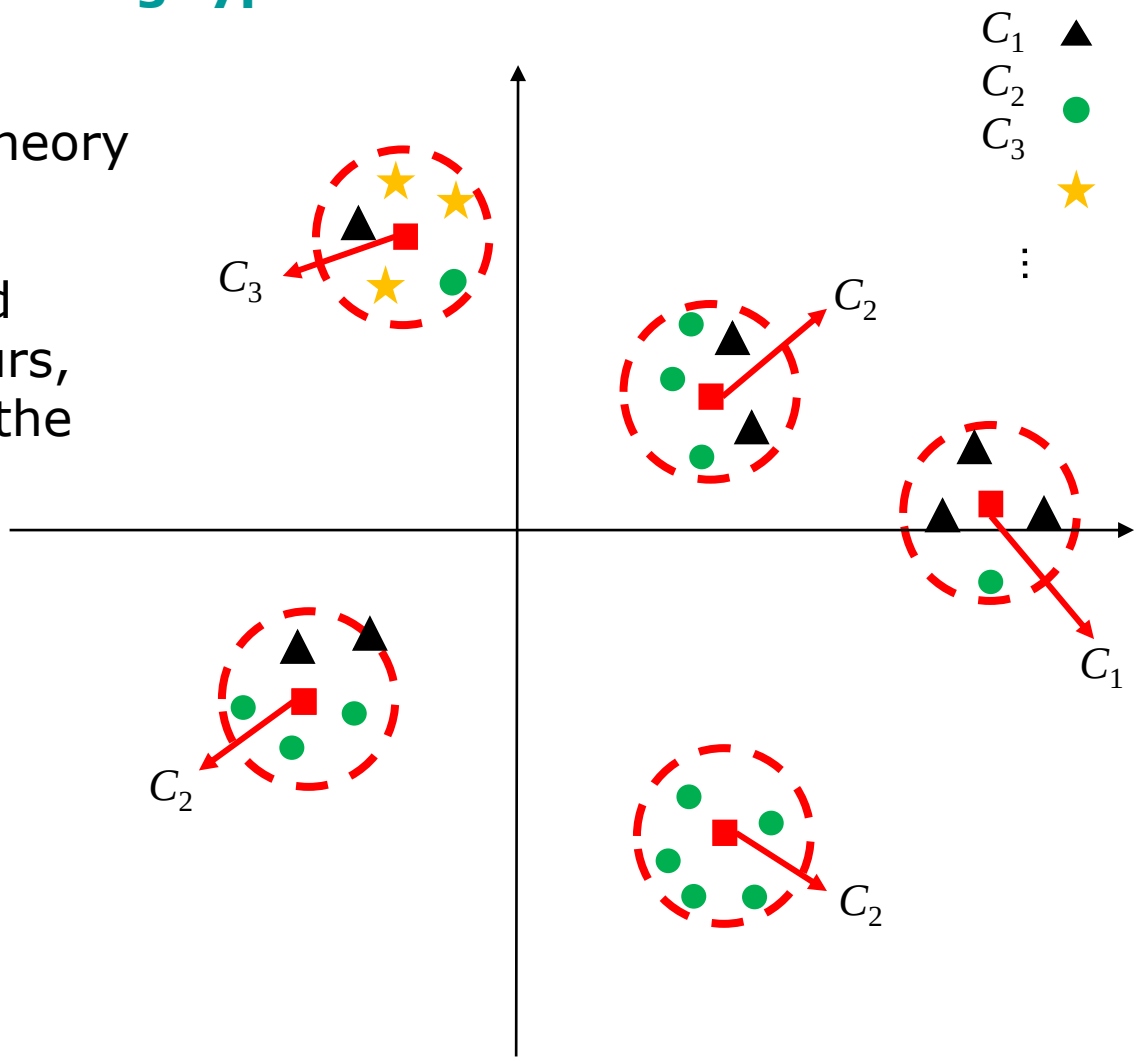


Project 2: Shape Labelling

How do we assign the clothing type label to a new image?

K-NN Algorithm seen in theory lectures,

Idea: The decision is based on the closest neighbours, considering what class the closest N-neighbours belongs to.



Project 2: Shape Labelling

How do we assign the clothing type label to a new image?

K-NN Algorithm

Function decision
(to classify $\vec{y}$)

$For\ (\vec{x}^j \in \boldsymbol{X})\ do$
$\quad List = insert([d(\vec{y}, \vec{x}^j), C_j], List)$
$fFor$
$Neighbours = First\_k(sorted\_d(List))$
$If\ (\#(Neighbours, C_1) > \#(Neighbours, C_2))$
$\quad \vec{y} \in C_1$
$Sinó$
$\quad \vec{y} \in C_2$
fSi

$$d(\vec{y}, \vec{x}^i) \rightarrow \begin{pmatrix} x_1^1 & \dots & x_{4800}^1 C_1 \\ \vdots & \ddots & \vdots \\ x_1^{n_1} & \dots & x_{4800}^{n_1} C_1 \\ x_1^1 & \dots & x_{4800}^1 C_2 \\ \vdots & \ddots & \vdots \\ x_1^{n_2} & \dots & x_{4800}^{n_2} C_2 \\ \vdots & \vdots & \vdots \\ x_1^1 & \dots & x_{4800}^1 C_k \\ \vdots & \ddots & \vdots \\ x_1^{n_k} & \dots & x_{4800}^{n_k} C_k \end{pmatrix}$$

Example:

$\vec{y}$ =[1,1,1,0.2,0.5,0,0,1,... ,,1]

d=23 [1,1,1,0.1,0.4,0.2,1,1,... ,1]

d=59 [0,0,0,0.7,0.5,1,0,1,... ,1]

d=103 [0,0,0,0.2,1,1,0,0.4,... ,0]

Project 2: Shape Labelling

How can we solve the problem of automatically labelling clothes?

Shape Label:

Shirt

3 Questions:

- How can we represent the shape of clothes? ✓
- How can we learn to classify clothes? ✓
- How we assign the clothing type label to a new image? ✓

Project 2: Shape Labelling

The answer to the 3 questions in the code:

- How can we represent the shape of clothes?

  Function: `read_dataset()`
  Folder: `images/train`
  File: `utils_data.py`

- How can we learn to classify clothes?

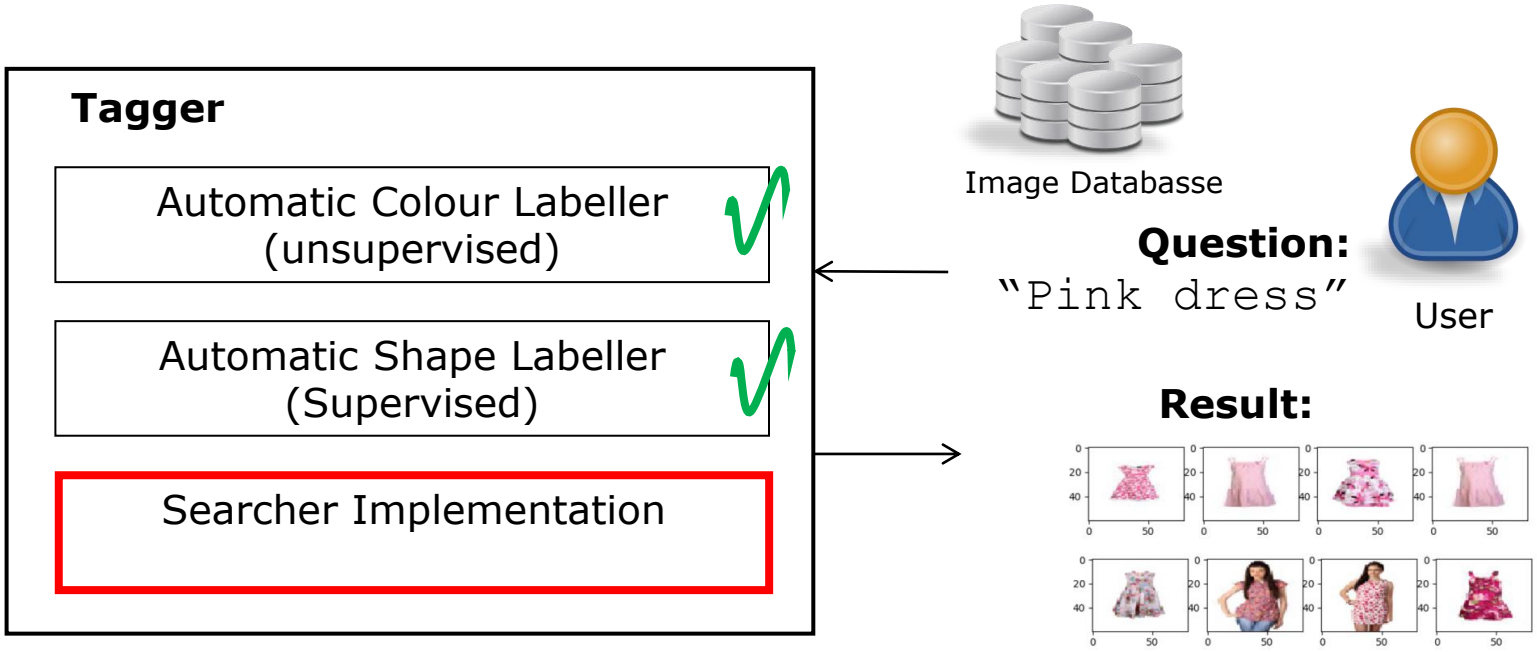
  Function: `KNN.__init__()`
  File: `KNN.py`

- How do we assign the clothing type label to a new image?

  Function: `KNN.predict()`
  File: `KNN.py`

Project 2

Problems to solve to build this tagger:



Project 2: Image Searcher

How do we implement a search engine based on color and shape labels?

We already have images labeled COLOR and SHAPE,

To search with labels, you need to code the functions:

```
Retrieve_img_by_color()

Retrieve_img_by_class()

Retrieve_combine()
```

File: `my_labeling.py`

Project 2

Planning

Part 1: **CODING** Kmeans and colour

Online Sessions: Week of **March 27th**

Delivery: What? Exercises indicated in Part 1 Guidelines (`GuiaP2_Part1.pdf`)

When? Before **Monday, April 10th** at 23:55h.

Part 2: **CODING** kNN and shape

Online Sessions: Week of **April 17th**

Delivery: What? Exercises indicated in Part 2 Guidelines (`GuiaP2_Part2.pdf`)

When? Before **Tuesday, May 2nd** at 23:55h.

Part 3: **Performance Evaluation and Analysis**

Online Sessions: Week of **May 8th**

Delivery: What? Report indicated in Part 3 Guidelines (`GuiaP2_Part3.pdf`) and whole code with the experiments

When? Before **Thursday, May 18th** at 23:55h.

ORAL Presentation, explanation of your whole Project

Online Sessions: Week of **May 22nd**

Delivery: Slides of the presentation before **Thursday, June 1st** at 23:55h

Project 2

Practical tips for the 1st Delivery:

- Exercises are in file `<Practica2_1.pdf>` you will find it at `cv.uab.cat` in `Practicum Section > Project 2`. This document will guide you in all the coding.

- Save all the functions in the file `<Kmeans.py>`

- Code the functions exactly how they are specified. Pay attention on the input parameters and output results.

- Delivery will be at cv.uab.cat, you will deliver the file `Kmeans.py` with all the functions you worked on this Part 1.

- We highly recommend you to **attend the online session with the exercises practically coded** in order you can use the session to solve the final details with your lecturer, and in the way to achieve the deadline.

Project 2

Practical tips for the 2nd Delivery:

- Exercises are in file `<Practica2_2.pdf>` you will find it at `cv.uab.cat` in `Practicum Section > Project 2.` This document will guide you in all the coding.

- Save all the functions in the file `<KNN.py>`

- Delivery will be at cv.uab.cat, you will deliver the file `KNN.py` with all the functions you worked on this Part 2

- We highly recommend you to **attend the online session with the exercises practically coded.** In this way, you can use the session to solve the final details with the help of your lecturer to achieve the delivery deadline.

Project 2

Practical tips for the 3rd Delivery:

- Exercises are in file `<Practica2_3.pdf>` you will find it at `cv.uab.cat` in `Practicum Section > Project 2`. This document will guide you in all the coding.

- Save all the functions in the file `<my_labeling.py>`

- Delivery will be at cv.uab.cat, you will deliver the files `Kmeans.py`, `KNN.py`, and `my_labeling.py` with all the functions you work on this project, and the **report** where you describe all the obtained results and analysis done.

- We highly recommend to **attend the online session with the exercises practically coded.** In this way, you can use the session to solve the final details with the help of your lecturer to achieve the delivery deadline.

Project 2

Tips to prepare Report and Oral Presentation:

Both should be organized as follows:

- **Introduction** (list of contents and summary about what you did diferent from the rest)
- **At least 3 analysis** from those we mentioned in the guideline 3. An analysis should contain the following:
  - ***Brief Introduction about which parameter****/method are you analysing*
  - ***Comparison between the original result and the new results.***
  - ***Explanation of the results*** *(Why it works better?, or worse?, if is it more efficient?, Could you find cases where one method works better than the other?*
- **Conclusion**

  Main problems you found out, what have you learnt?, what would you improve now?

Different parameters or methods to be analysed

- Centroid initialization method
- Using a colour space different from RGB
- Using different K values
- Using different methods to find the best K Interclass variance, Fisher,...)
- Using different methods to label color (multiple labels, new color terms, …)
- Using different features for KNN → Different image sizes
- Using different features for KNN → Features computed separately (*average value of all pixels, pixels on the right side versus left side, pixels at top versus bottom, …)*
- *…*

Project 2

Evaluation:

Mark Project 2 *= 0.75 * Group Mark + 0.25 * Individual Mark*

- ***Group Mark*** *= 0.6 * Code Mark + 0.3 * Report + 0.1 * Group Presentation*
- ***Individual Mark*** *= 0.5 * Individual Presentation + 0.5 * Group Participation*