TA’AM

Graduation Project Presentation

Team

Saleh Adel Saleh

Shehab Mostafa Fahmy

Abdelaziz Hossam Abdelaziz

Abdulrhman Hosny Muhammed

Muhammed Mahros Muhammed

Supervisors

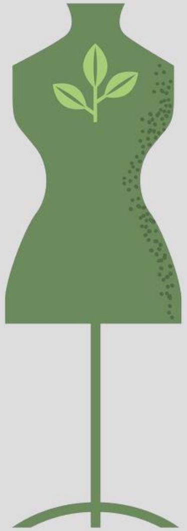
- Dr. Salsabil Amin
- TA. Ahmed Hatem

Agenda

- FlashBack
- System Architecture
- Diagrams
- Finished Phases
- Demo
- What’s Next?
- Time Plan
- References

FlashBack From first seminar

What is TA’AM?

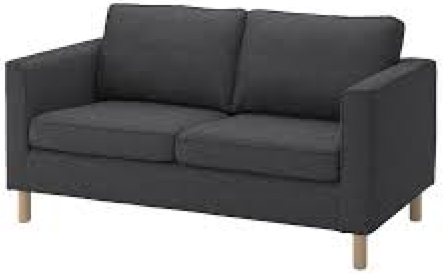


Objectives

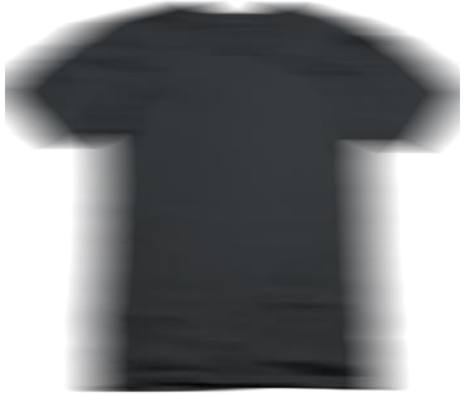
- Offer a specialized platform for users to allow them to showcase their surplus clothing, and buy suitable ones at a relatively low cost.
- Enhance the user experience by implementing multiple machine learning models to assess image suitability, extract various attributes, and offer diverse search capabilities.

Phases Description

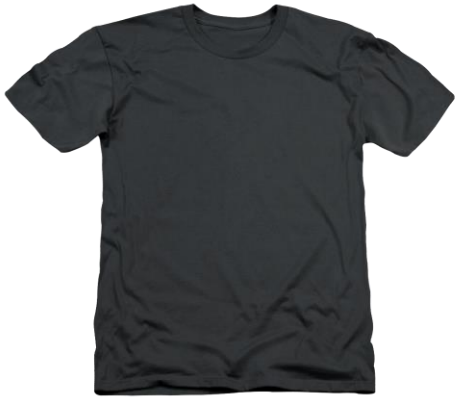
Image Assessment: This phase ensures that the quality of the image is good enough. for our processing efficiency and the user experience.



✗



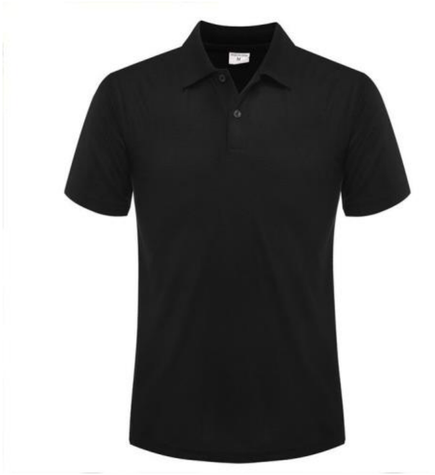
✗



✓

Phases Description

Attributes Extraction: In this phase, we extract different attributes from the image such as category, size, color, (season, gender, description, …) according to the data.



T-shirt

Black

Male

Summer

56 * 77

Phases Description

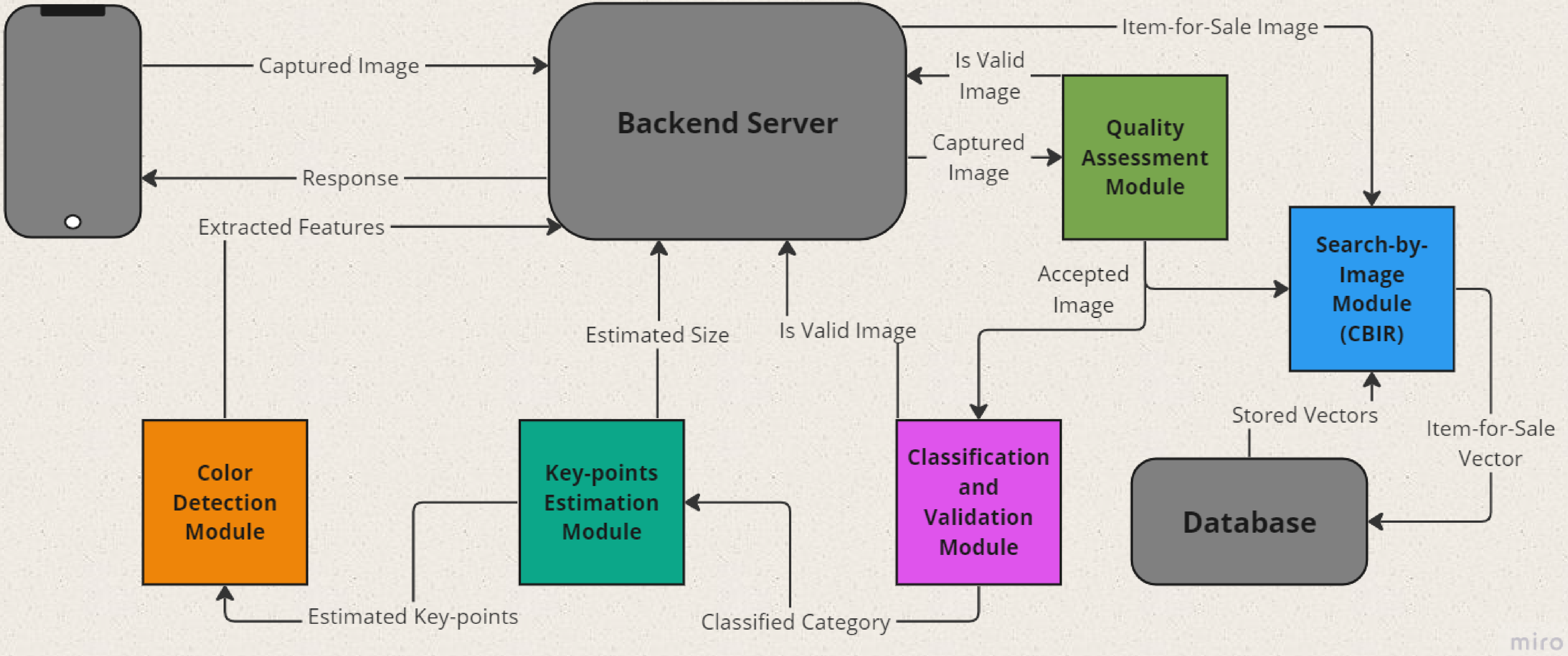
Search Capabilities: The phase we provide the user to search for a specific item using an image or a brief description to find the closest items.

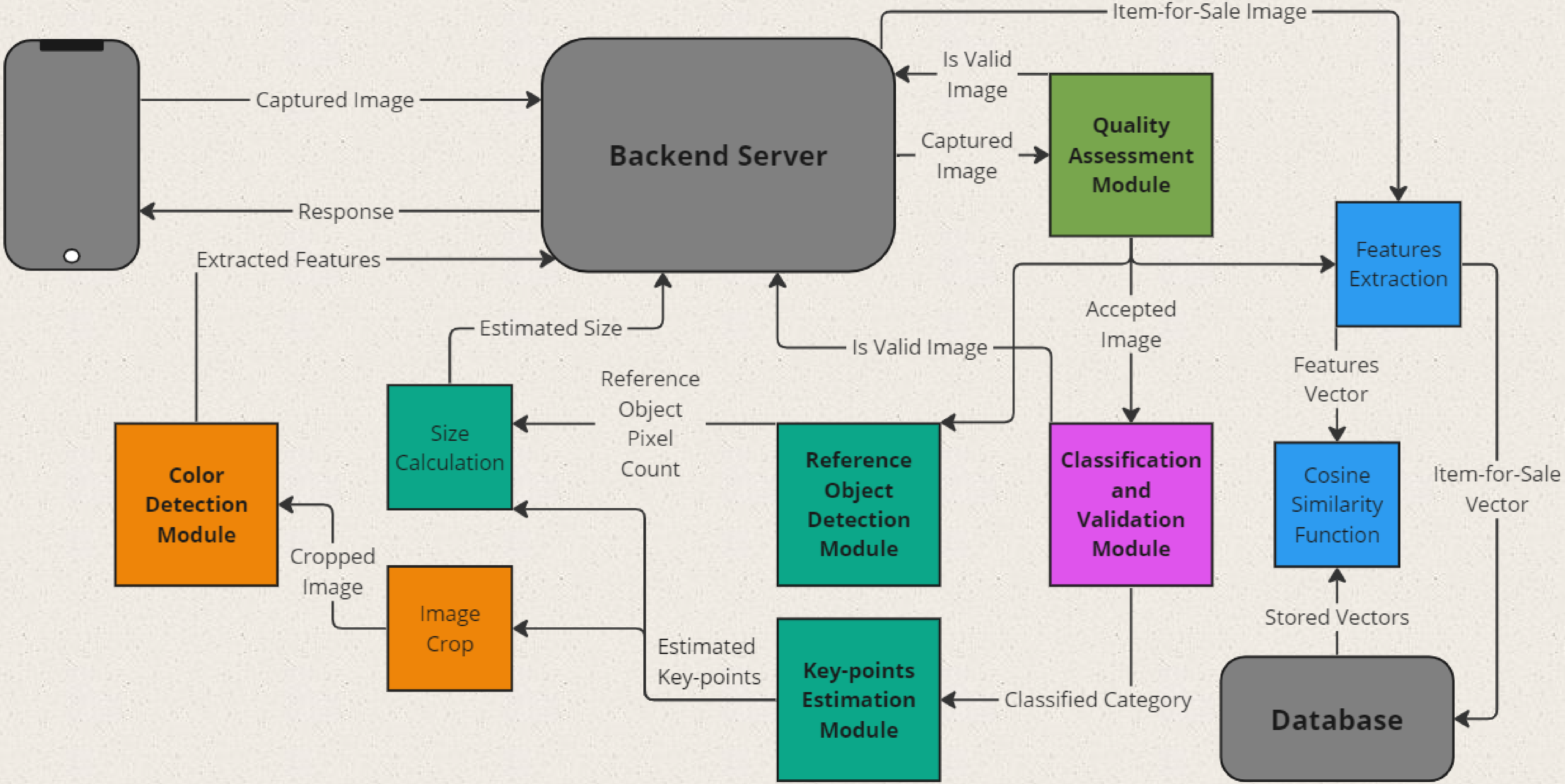
Search by images:

Search by text:

I need a basic navy T-shirt with a round nick

System Architecture

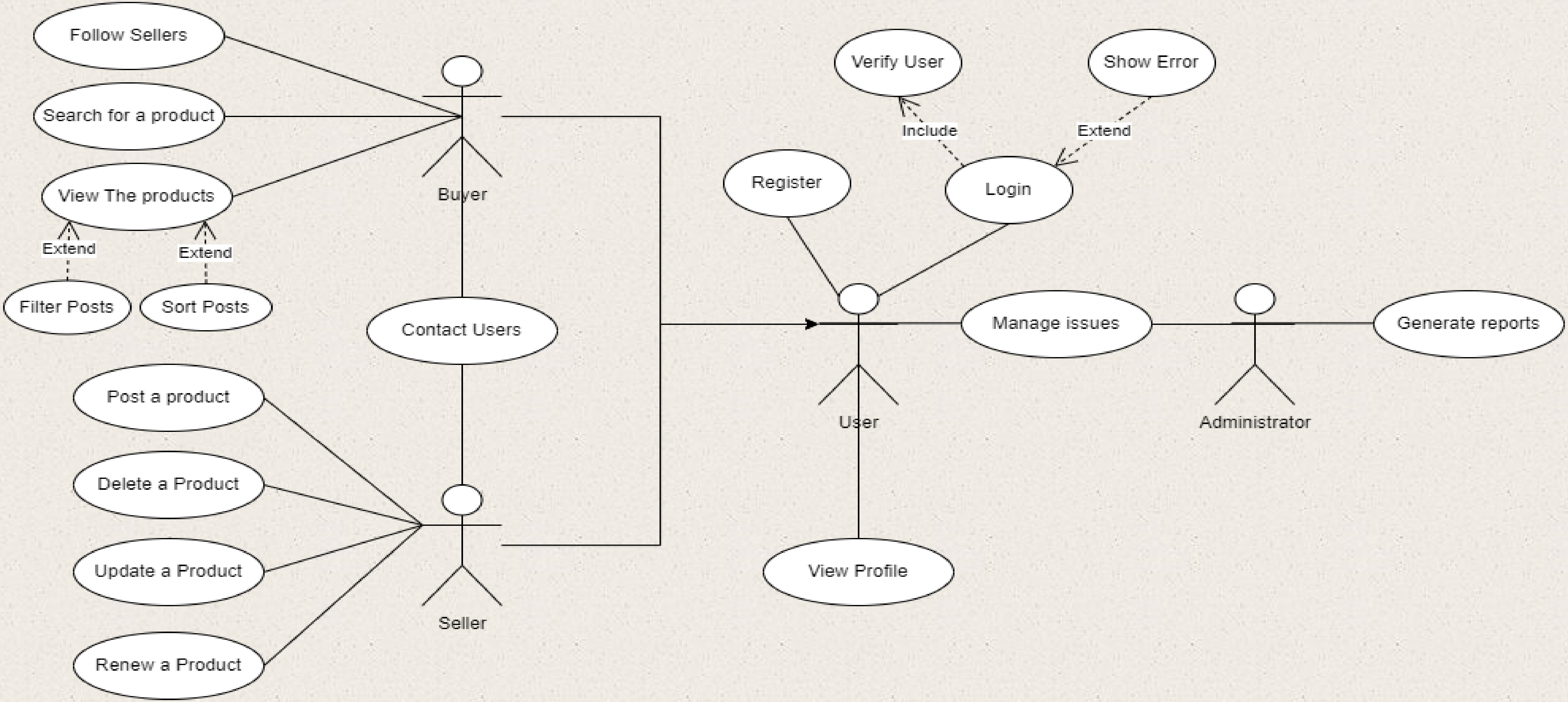

Backend Server
Captured Image
Response
Extracted Features
Item-for-Sale Image
Is Valid Image
Captured Image
Quality Assessment Module
Accepted Image
Search-by-Image Module (CBIR)
Estimated Size
Is Valid Image
Stored Vectors
Item-for-Sale Vector
Color Detection Module
Key-points Estimation Module
Classification and Validation Module
Database
Estimated Key-points
Classified Category
miro



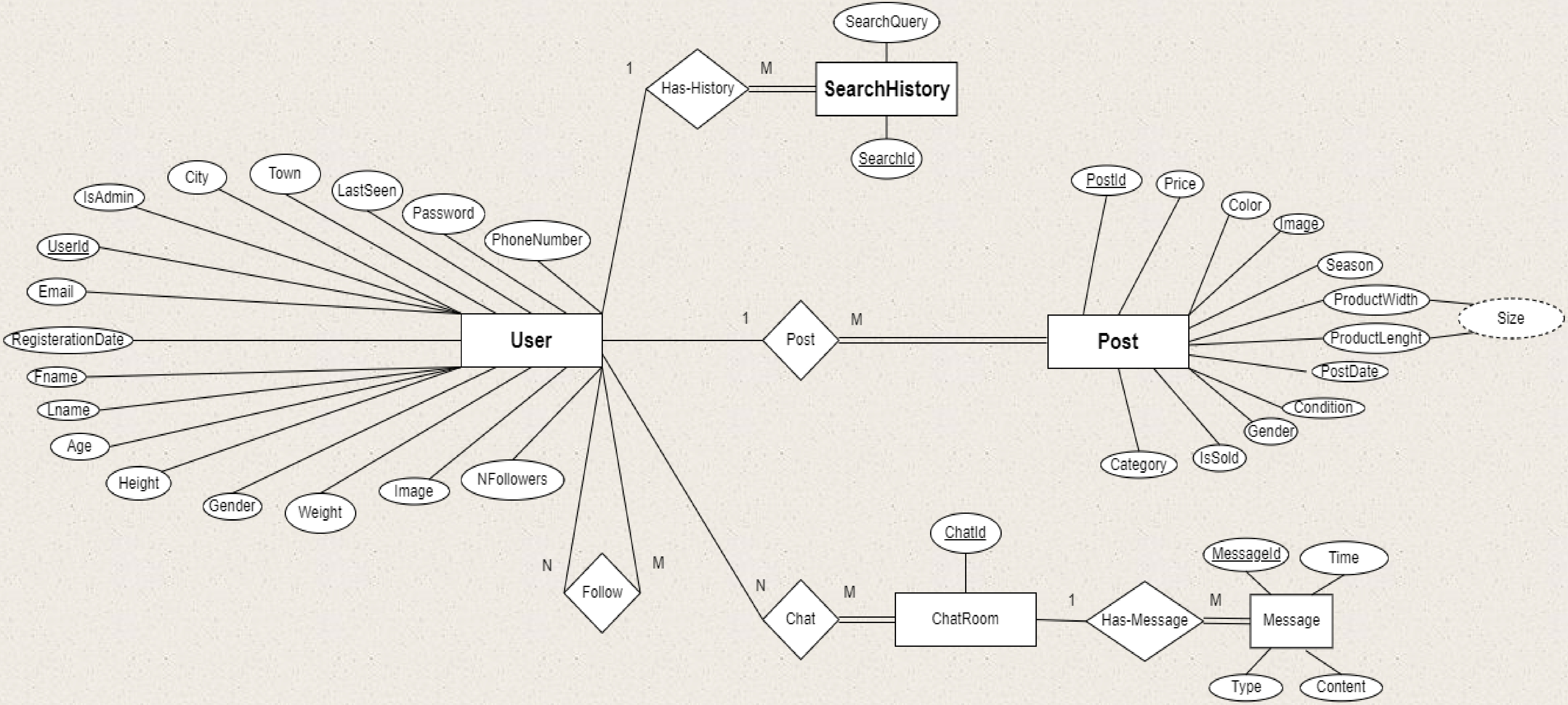
Backend Server
Captured Image
Response
Extracted Features
Item-for-Sale Image
Is Valid Image
Captured Image
Quality Assessment Module
Features Extraction
Accepted Image
Features Vector
Estimated Size
Is Valid Image
Reference Object Pixel Count
Size Calculation
Color Detection Module
Reference Object Detection Module
Classification and Validation Module
Cosine Similarity Function
Item-for-Sale Vector
Cropped Image
Image Crop
Estimated Key-points
Key-points Estimation Module
Classified Category
Stored Vectors
Database

Diagrams

Use Case


Follow Sellers
Search for a product
View The products
Extend
Extend
Filter Posts
Sort Posts
Buyer
Contact Users
Post a product
Delete a Product
Update a Product
Renew a Product
Seller
Verify User
Show Error
Include
Extend
Register
Login
User
Manage issues
Administrator
Generate reports
View Profile

ERD


SearchQuery
1
Has-History
M
SearchHistory
SearchId
PostId
Price
Color
Image
Season
ProductWidth
Size
ProductLenght
PostDate
Condition
Gender
IsSold
Category
Post
City
Town
LastSeen
IsAdmin
Password
PhoneNumber
UserId
Email
RegisterationDate
User
1
Post
M
Fname
Lname
Age
Height
Gender
Weight
Image
NFollowers
N
Follow
M
N
Chat
M
ChatId
ChatRoom
1
Has-Message
M
MessageId
Time
Message
Type
Content

Finished Phases

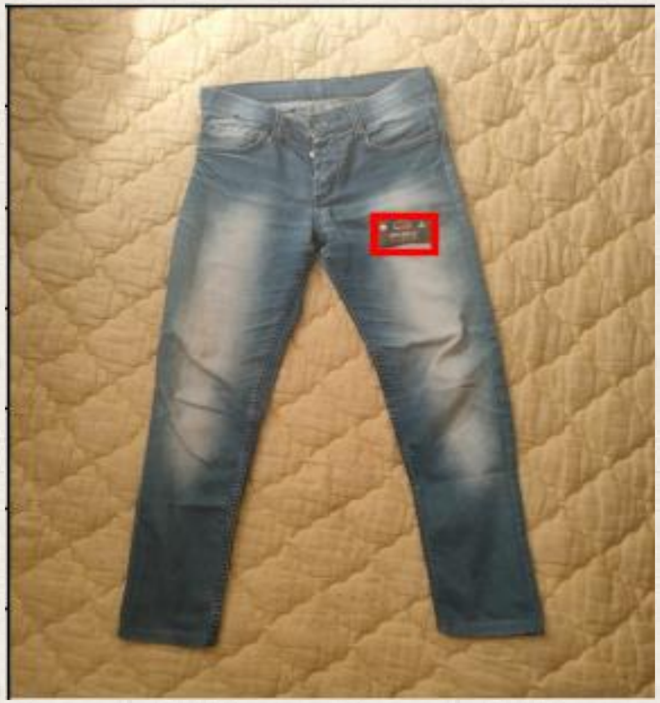
Quality Assessment

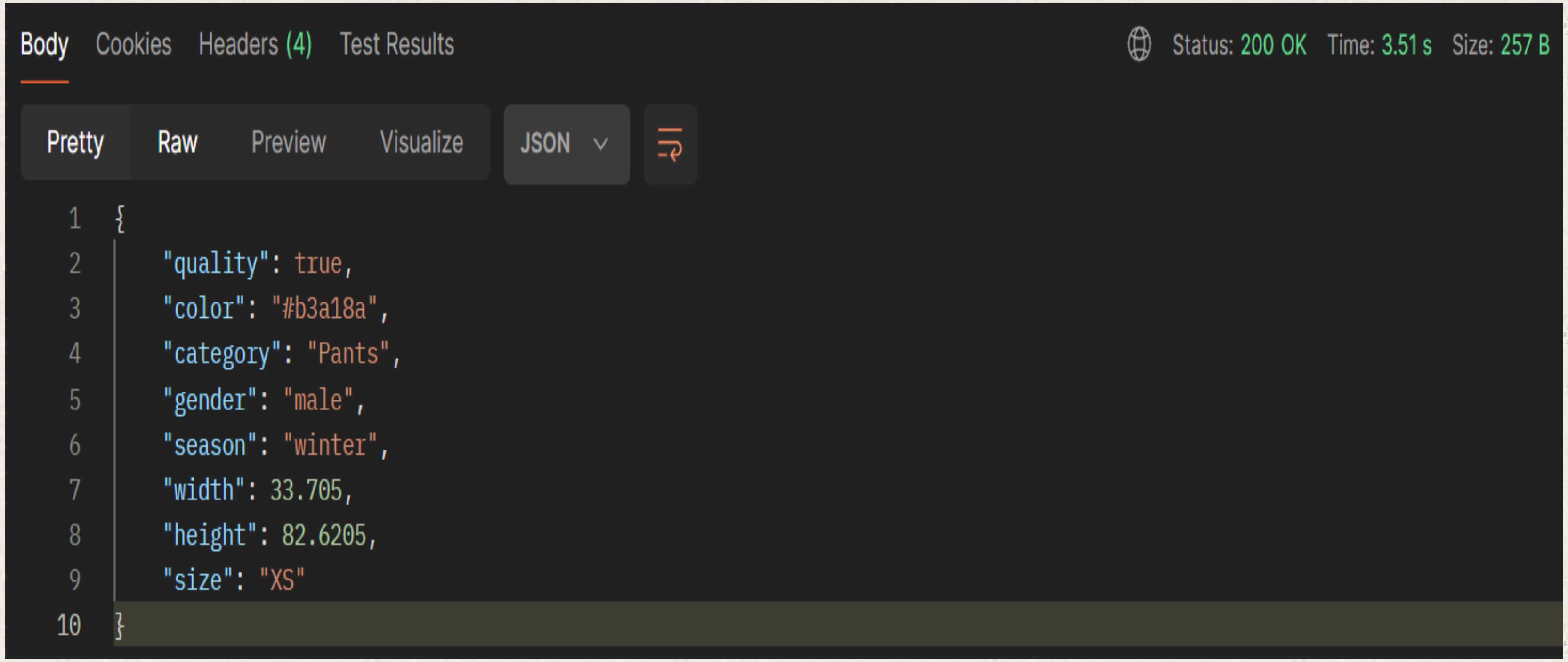
- Using the <u>BRISQUE</u> model, the image quality score was computed, and a specific threshold of **30%** was applied for image assessment.
- Brisque is a model that predicts how good an image looks without needing another image for comparison.
- Brisque extracts features like contrast, brightness, and texture from images, then uses Support Vector Regression (SVR), to link these features to quality scores based on a dataset of manually rated images.

✖

✓

Features Extraction





Status: 200 OK Time: 3.51 s Size: 257 B

```
{
    "quality": true,
    "color": "#b3a18a",
    "category": "Pants",
    "gender": "male",
    "season": "winter",
    "width": 33.705,
    "height": 82.6205,
    "size": "XS"
}
```

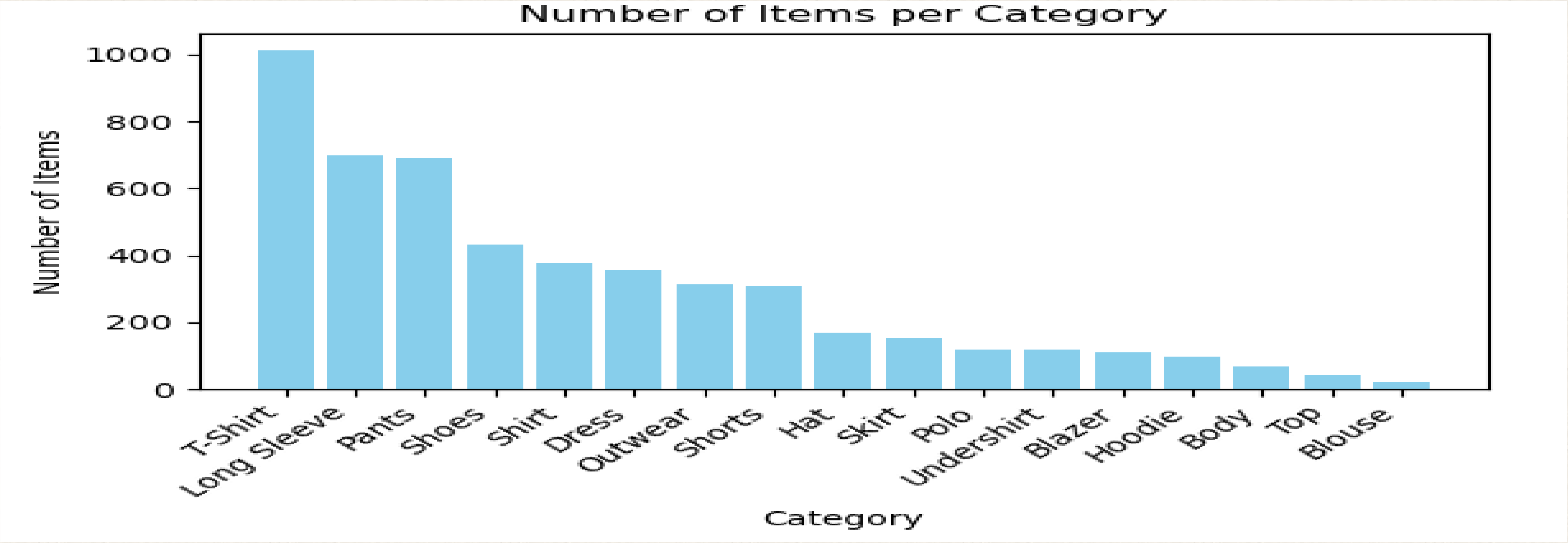
Category Dataset

Agrigorev's Custom Kaggle Dataset

- This dataset is a 7GB "**5k** product image" of **20 different classes** taken by users.
- **80%** of the dataset used for training and **20%** for testing.

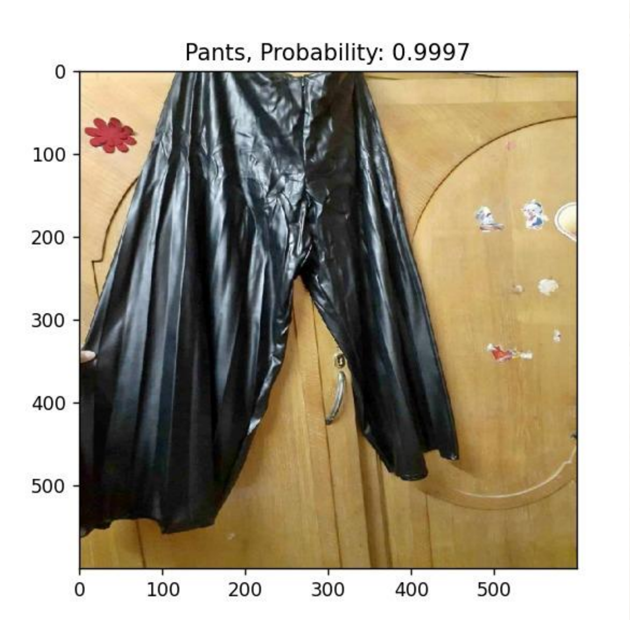
Dataset Description

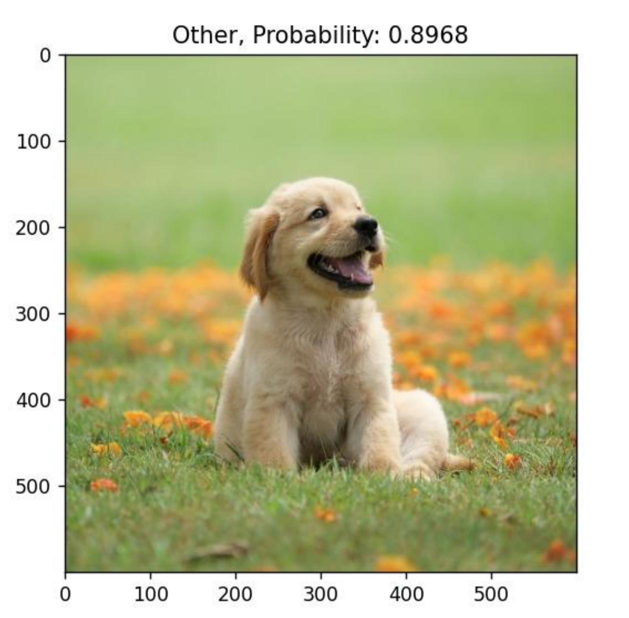
- To solve the problem of **data imbalance**:
  - **Data Augmentation** using FastAI library applied.
  - **Pre-trained** model was used.



Category Classification

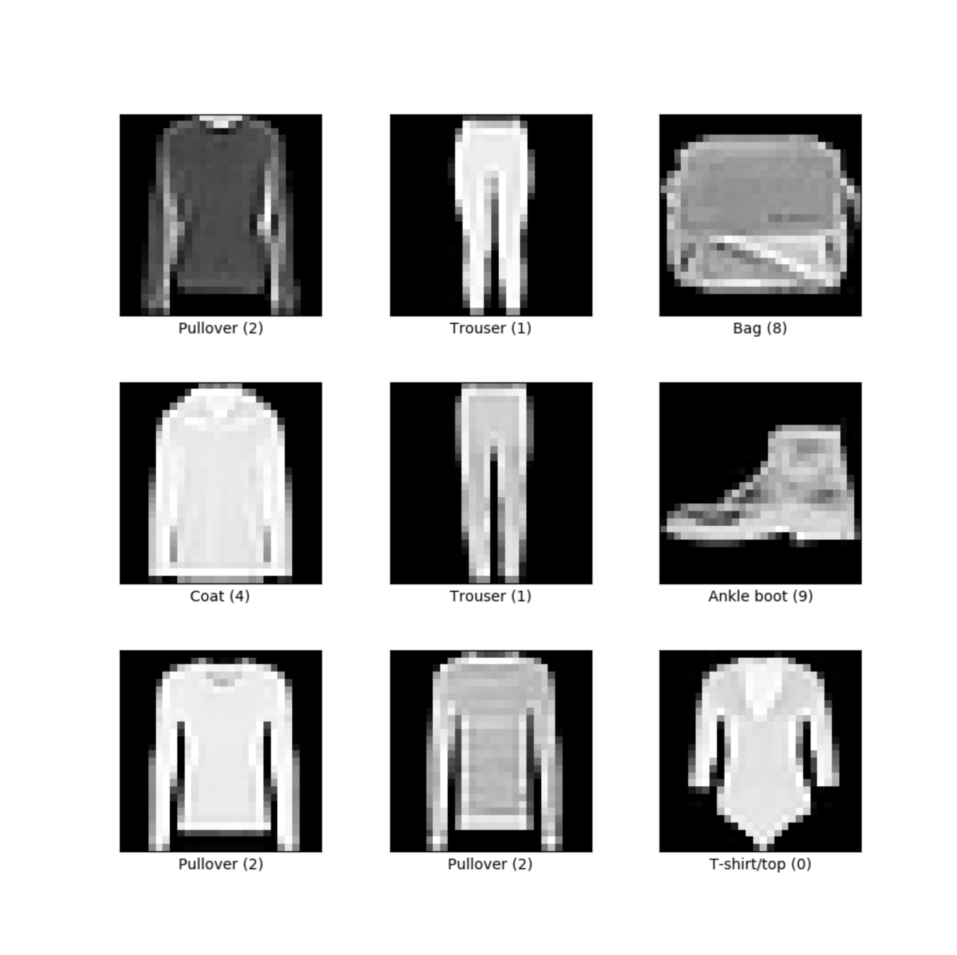
- This model validates whether the image **contains clothes or not**, based on that, it returns the classified category or “Other”.
- In the case of **“Other”**, the image is **rejected**.
- We utilized the Pre-trained ResNet34 architecture on Agrigorev’s custom Kaggle dataset for category classification.





Experiments and results

- **First Trial:**
  - In the initial attempt, a custom CNN architecture was employed on FashionMNIST dataset to classify categories. Despite achieving a notable accuracy of approximately **92.4%**, the approach was deemed inadequate due to its reliance on 28x28 grayscale images, which **did not meet our specified criteria**.



Experiments and results

- **Second Trial:**
  - Next, we utilized Agrigorev’s custom Kaggle dataset with a pre-trained ResNet34 architecture which achieved an accuracy of **88.5%** after **50 epochs**. This dataset, sourced directly from users, is closely aligned with our application's criteria, making it a **suitable choice for our purposes**.

Experiments and results

- **Third Trial:**
  - DeepFashion2 dataset was employed with a pre-trained ResNet50 architecture. However, it was **not accepted**, as **Agrigorev's** dataset proved to be more compatible with the images likely to be submitted, given that a significant portion of the DeepFashion2 dataset comprised images of clothing being worn.

Experiments and results

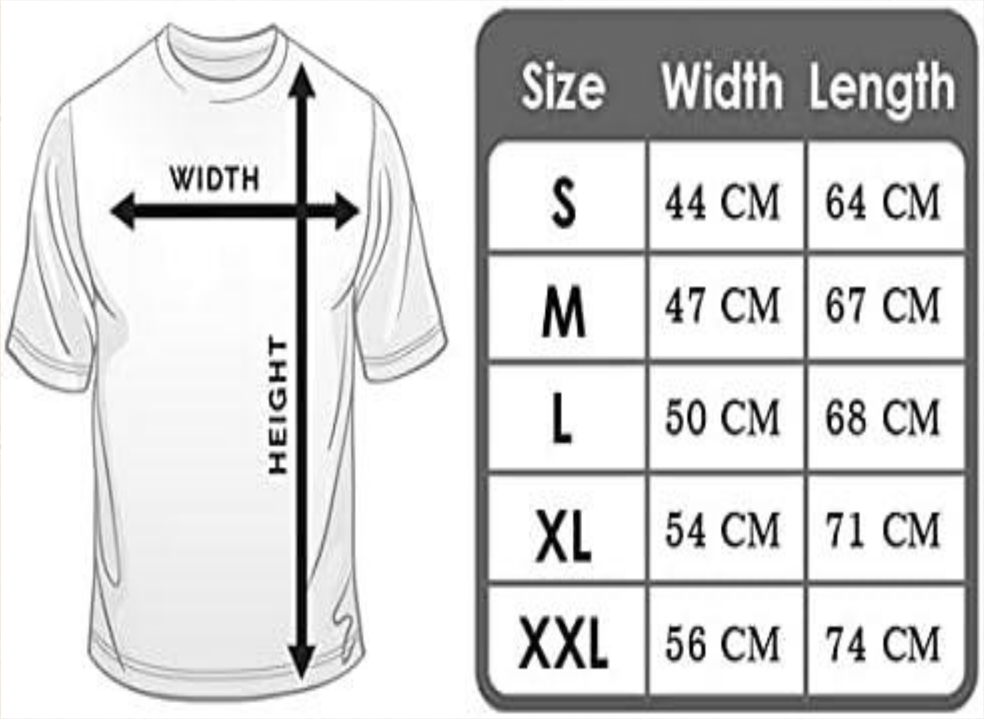
- Here's a summary table outlining the datasets and architectures employed for category classification:

Dataset	Architecture	Accuracy
FashionMNIST	Custom CNN architecture	e=10 -> 92.4%
Agrigorev’s Custom Kaggle Dataset	**ResNet34**	**e=50 -> 88.5%** e=100 -> 82.3%
DeepFashion2	ResNet50	e=40 -> 90.5%

- **Higher accuracy doesn’t mean it satisfies our criteria.**

Size Estimation

- This model is used to calculate the classified cloth size (width x height).
- Model Pipeline is:
  - Key-points detection model.
  - Card detection model.
  - The predicted key-points and the card measurements are inserted into an equation to calculate the estimated size of the cloth.

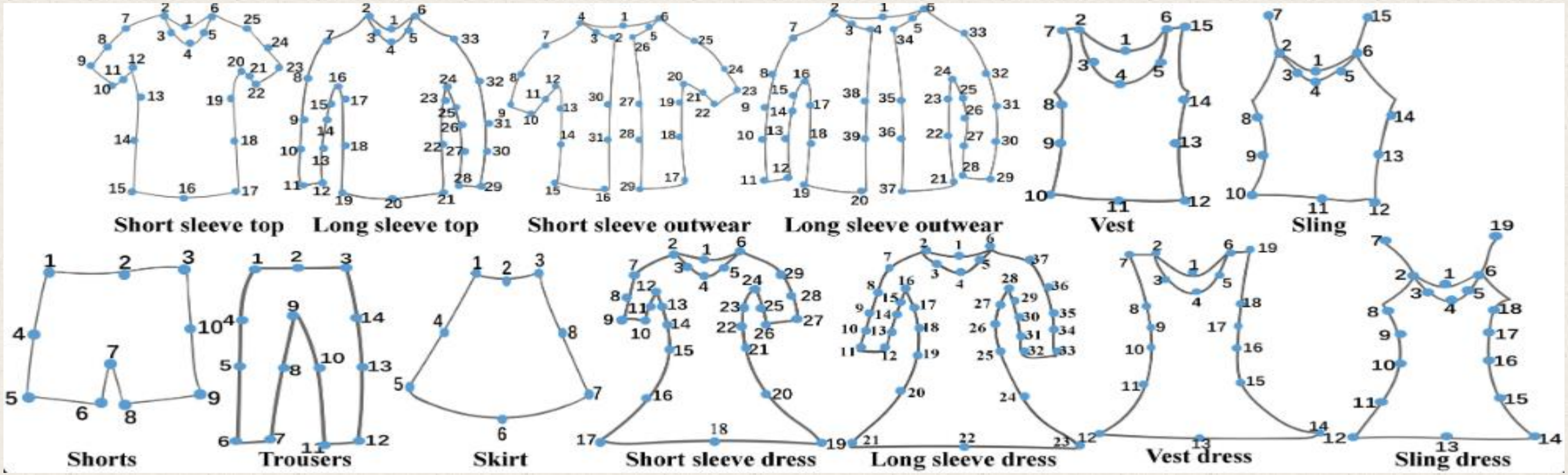


Size	Width	Length
S	44 CM	64 CM
M	47 CM	67 CM
L	50 CM	68 CM
XL	54 CM	71 CM
XXL	56 CM	74 CM

Key-points Dataset

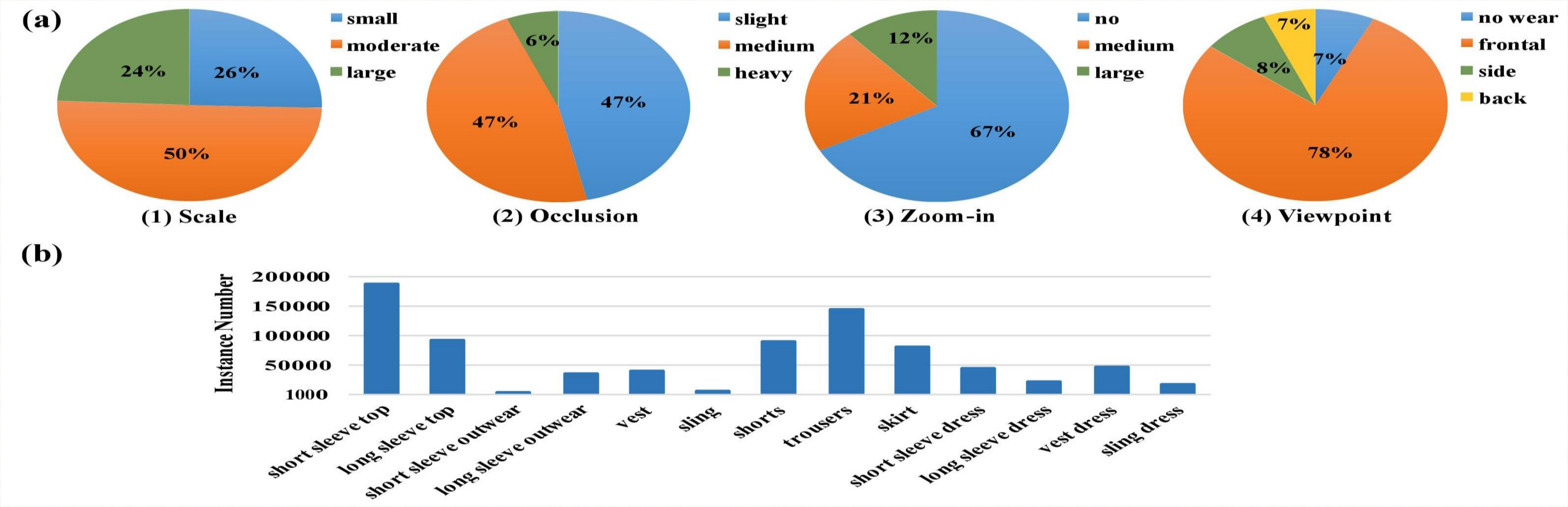
DeepFashion2

- DeepFashion2 is a comprehensive fashion dataset. It contains **286K** diverse images of **13 categories** from both commercial shopping stores and consumers.



Dataset Description

- To solve the problem of **data imbalance**:
  - Each category has a **separate** model



DeepFashion2 Challenges

- Variability in the number of items per image:

  Solutions:

  - Ignore images with more than one item.
  - Crop images to ensure each one contains only one category.

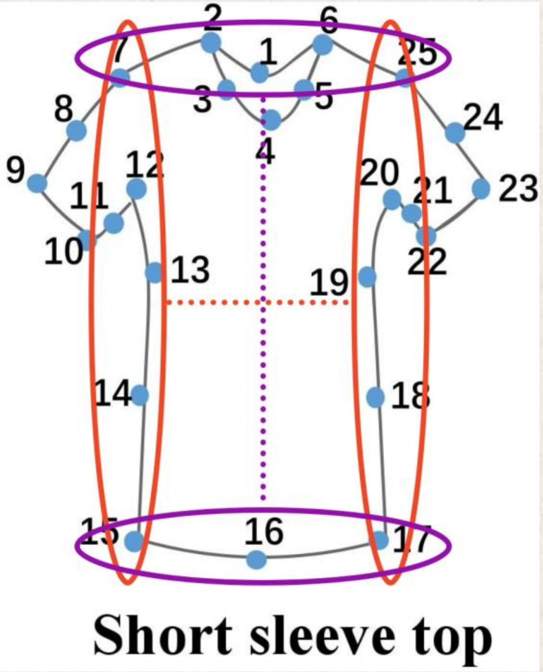
- Differences in the number of key-points per category:

  Solutions:

  - Train a separate model for each category.
  - Standardize the number of landmarks across all categories.
    - Choose some specific points.
    - Pad and Trim.

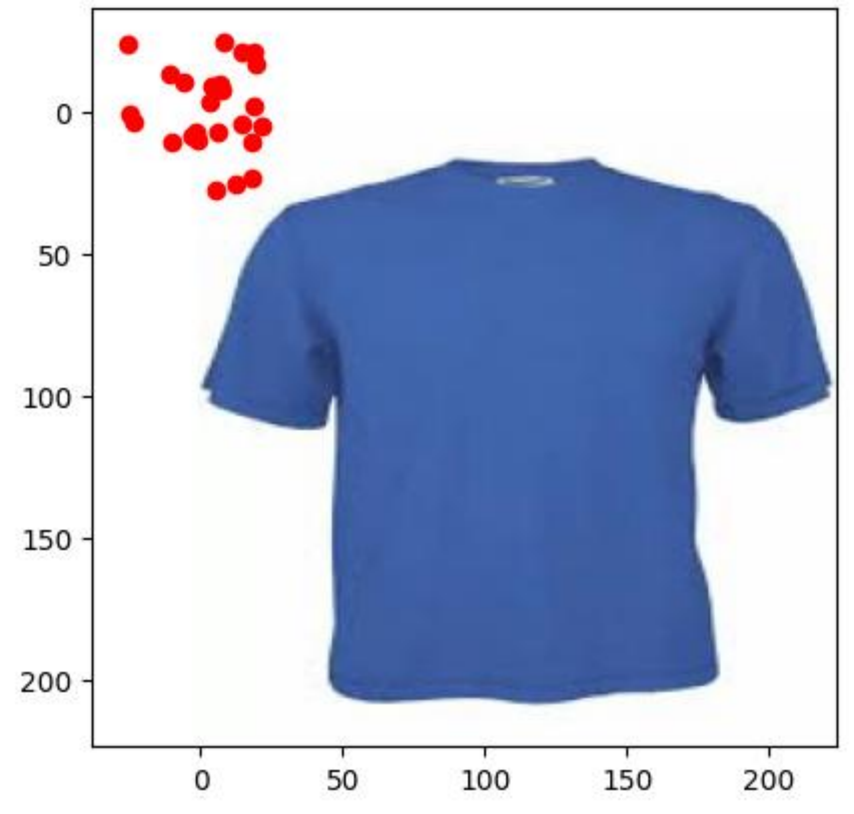
Key-points Detection

- 13 pre-trained ResNet50 models were employed to extract key-points **for each category**.
- The height and width of the product were determined by calculating the distances between specific key-points.



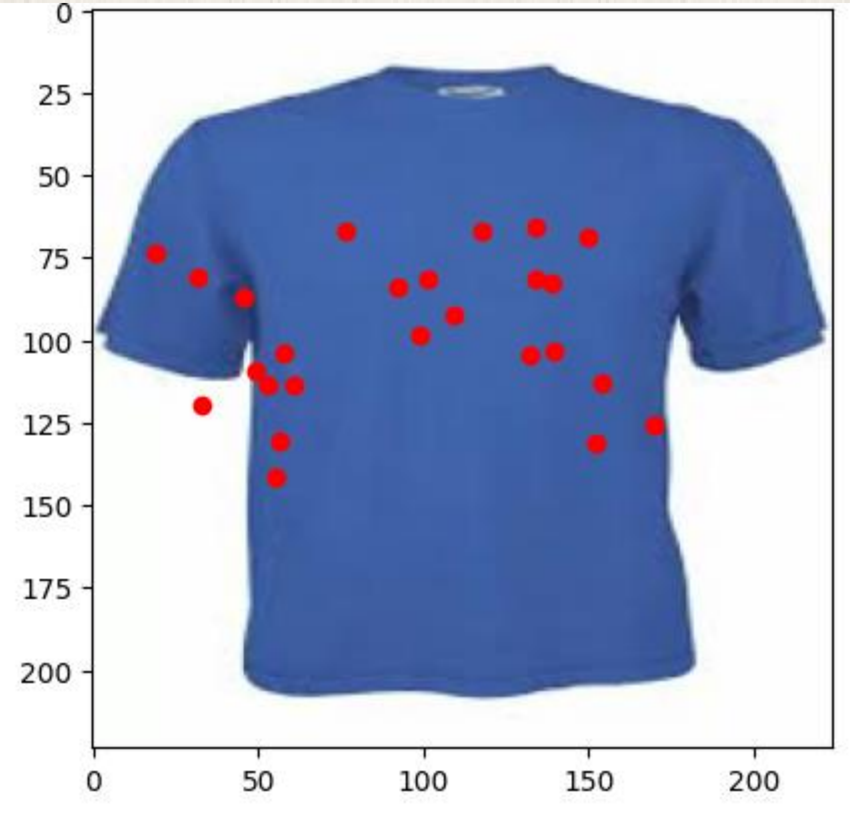
Experiments and results

- ## First Trial:
  - A <u>basic CNN model</u> was employed for key-point detection, but it yielded unsatisfactory results with a **loss of 7000** after 5 epochs, indicating its **poor performance**.
  - The model utilized here focused on **single-category** images.



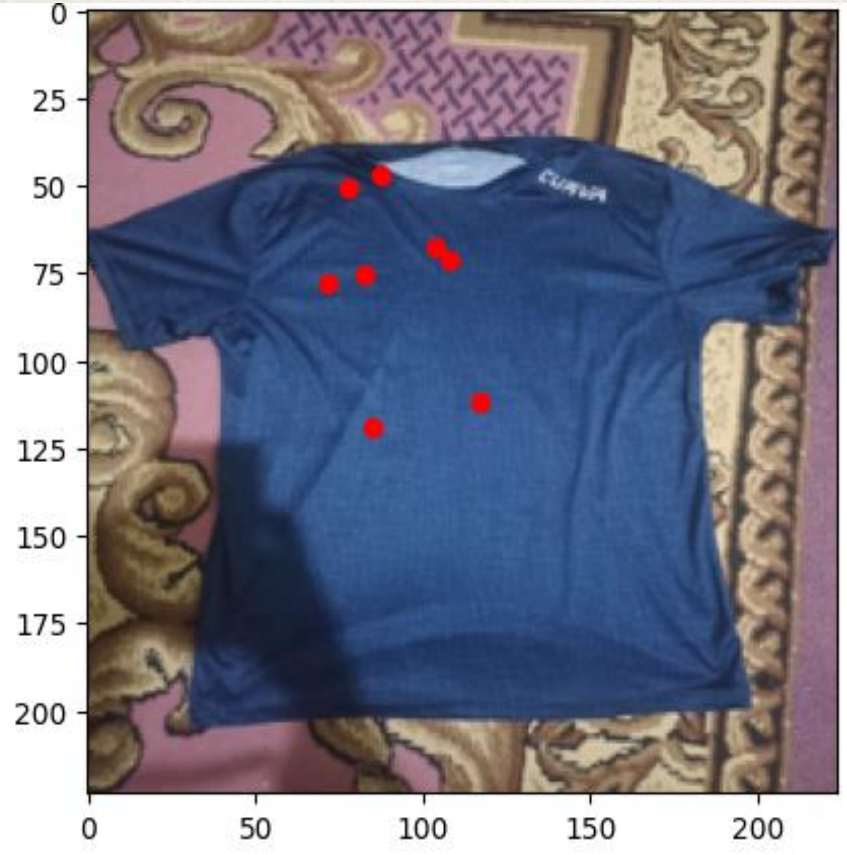
Experiments and results

- **Second Trial:**
  - MobileNetV2 was utilized for keypoint detection, achieving a **loss of 2000** over 10 epochs.
  - The model utilized here focused on **single-category** images.



Experiments and results

- ## Third Trial:
  - ResNet50 was employed for **some specific** key-points detection with the following results:
    - With a learning rate of **0.001** and **5 epochs**, the validation loss was **794.31640625**.
    - With a learning rate of **0.001** and **15 epochs**, the validation **loss decreased to 598.56103515625**.
    - Utilizing a learning rate of **0.003019** and training for **10 epochs**, the validation loss increased to **912.736328125**.



Experiments and results

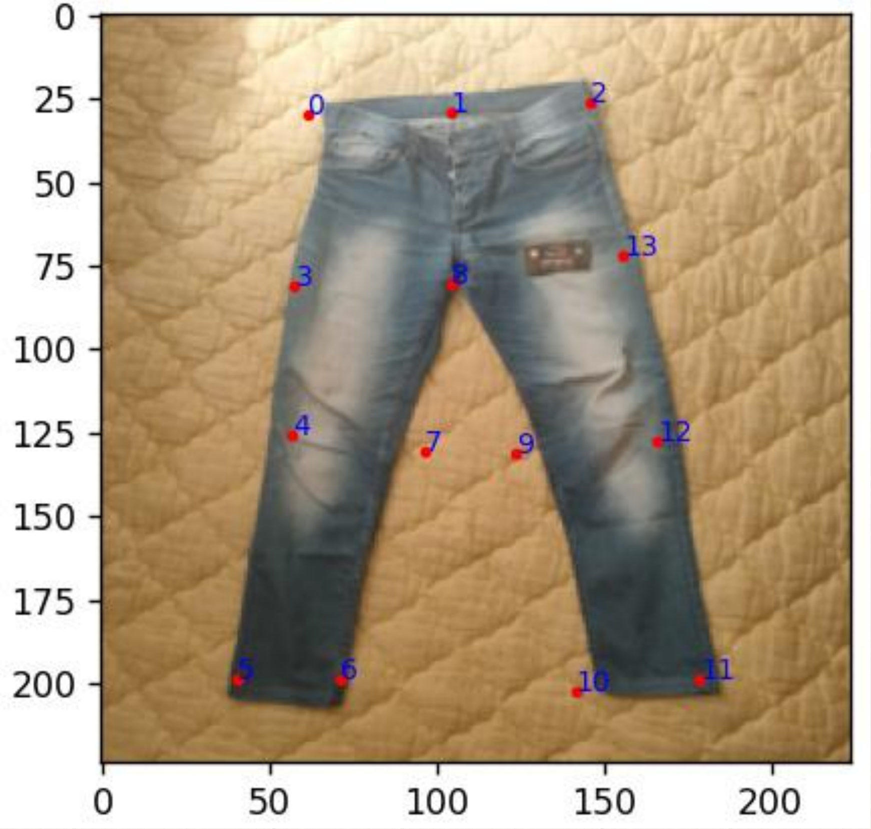
- **Fourth Trial:**
  - 13 pre-trained ResNet50 models were employed to extract key-points **for each category**.

Category	mAP
short sleeve top	0.81113302167
skirt	0.86460204081
vest	0.79334654842
vest dress	0.82715819403
short sleeve dress	0.79205821998
short sleeve outwear	0.65667514548
trousers	0.86246394137
shorts	0.89364662171
sling dress	0.72155837022
long sleeve dress	0.62783778636
long sleeve outwear	0.69307984813
sling	0.68970313825
long sleeve top	0.69068483233
short sleeve top	0.81113302167

Category	MSE
short sleeve top	0.058824
skirt	0.065241
vest	0.062255
vest dress	0.057008
short sleeve dress	0.063092
short sleeve outwear	0.098977
trousers	0.060573
shorts	0.057842
sling dress	0.081455
long sleeve dress	0.106946
long sleeve outwear	0.094994
sling	0.084006
long sleeve top	0.093227
short sleeve top	0.058824

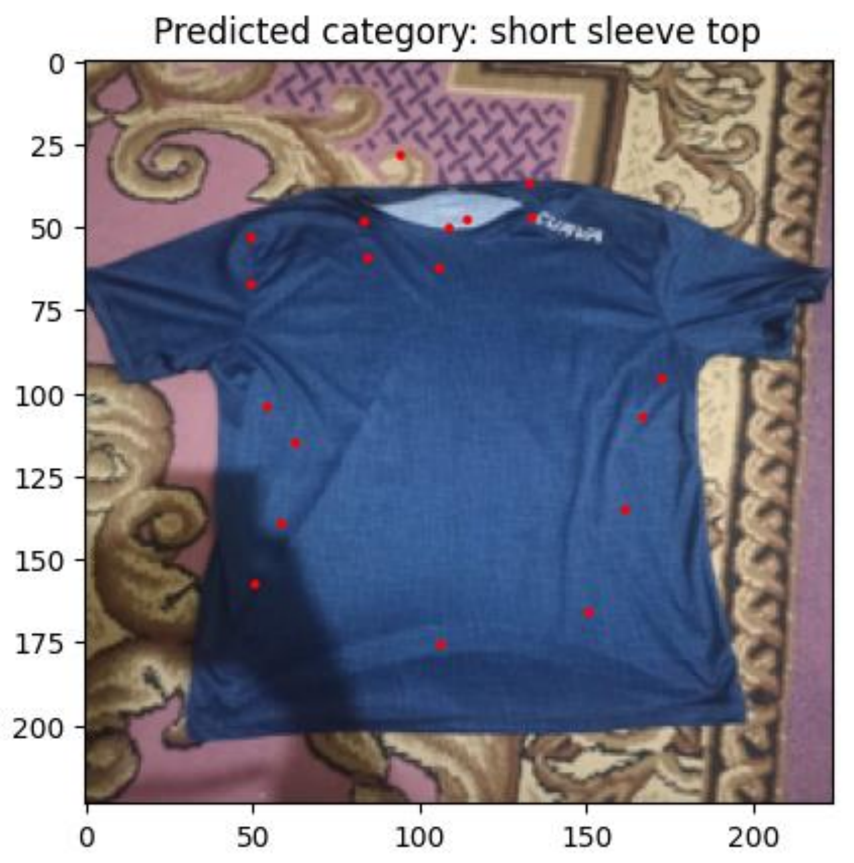
Experiments and results

- **Fourth Trial:**
  - 13 pre-trained ResNet50 models were employed to extract key-points for each category.



Experiments and results

- **Fifth Trial:**
  - Pre-trained ResNet50 model was employed to extract key-points and classify categories simultaneously. The best results were obtained with a **loss of 0.085964** and a **category accuracy of 88.3%**, trained for **40 epochs** with a learning rate of **0.0001**.



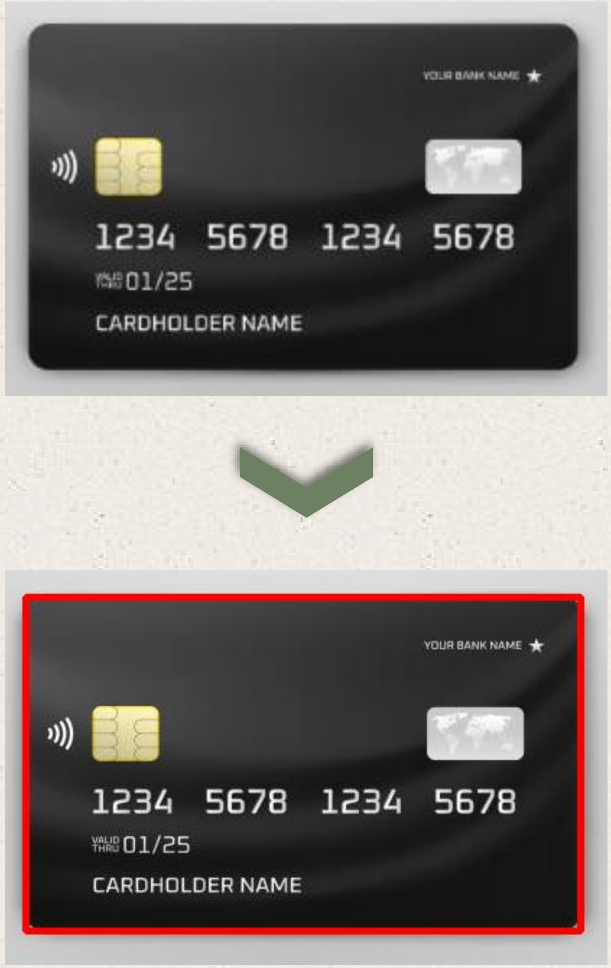
Experiments and results

Target	Architecture	Validation MSE Loss
Single Category	Basic CNN	e=5 -> 7000+
Single Category	MobileNetV2	e=10 -> 2000+
Single Category	ResNet50	e=15 -> 598+
Single Category	**13 pre-trained ResNet50**	-
All Categories	Multi-task using pretrained ResNet50	e=40 -> 0.0859

Card Detection

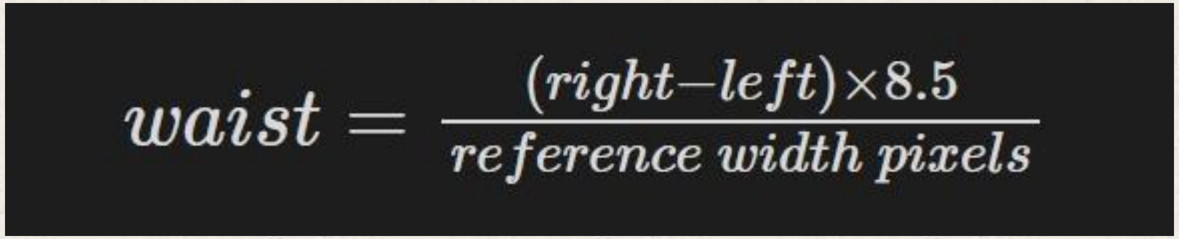
- This model **processes an image** containing a card, detects its boundaries, detects the largest contour, calculates the conversion factor, and returns it.
- This conversion factor allows for measurements in the image to be translated to real-world units (centimeters).
- Model Pipeline is :

  - Preprocessing
  - Contour Detection
  - Detect the largest contour
  - Get the width and height in pixels
  - Convert it from pixels to cm



Estimated Size Equation

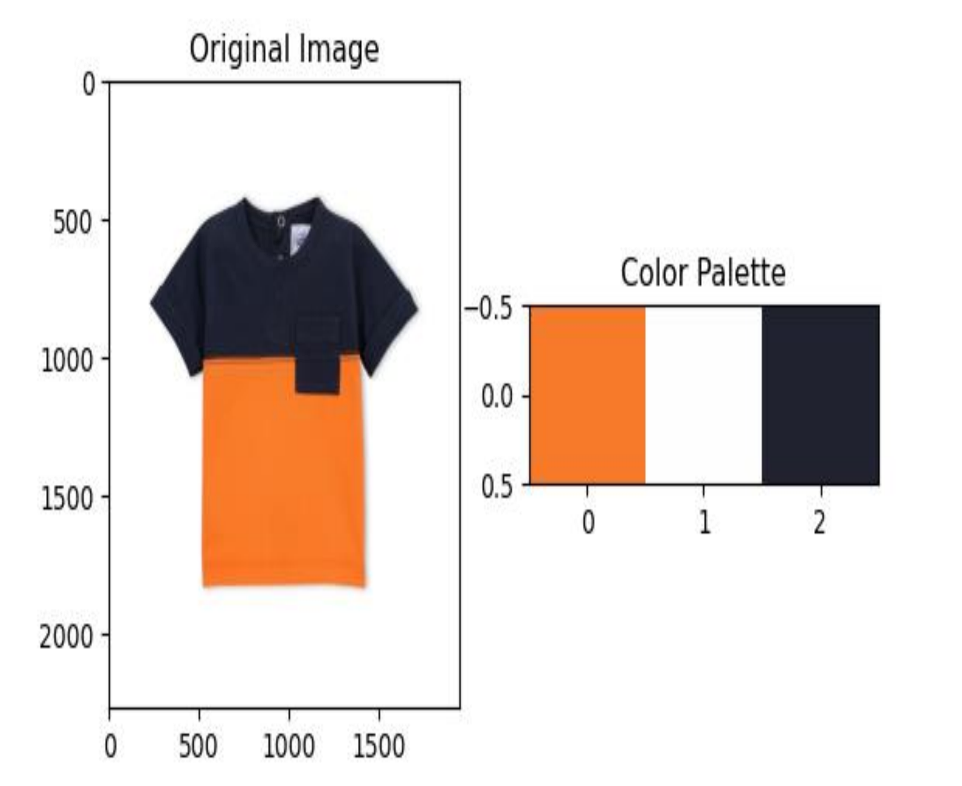
- An EN 13402 Standard for clothing size designation was used to map the measurements from centimeters to real-world measurements “S, M, L, XL”.
- This is an example of the equation applied to get the waist of trousers:



$$waist = \frac{(right - left) \times 8.5}{reference\ width\ pixels}$$

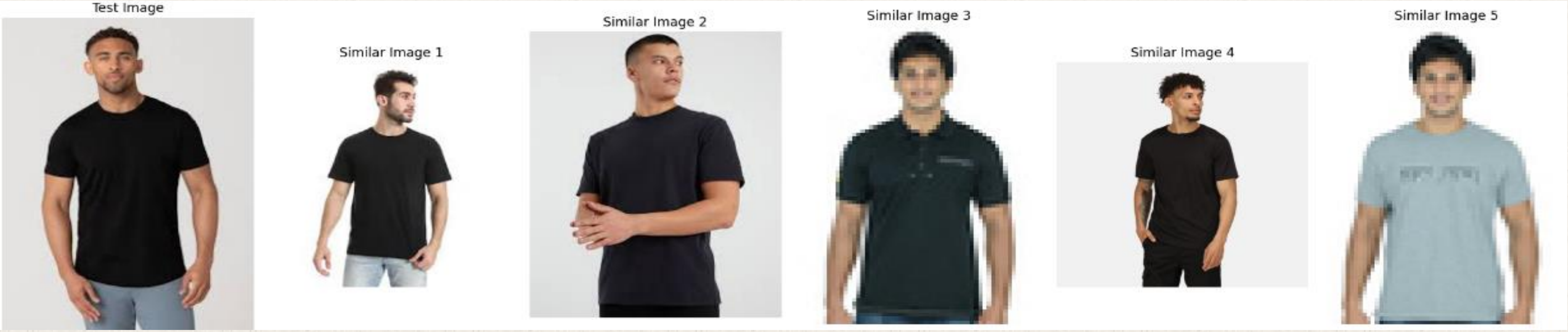
Color Detection

- The image was cropped using the detected key-points to **ignore background colors.**
- Kmeans is used to segment the image into regions of similar colors and identify representative colors for each region, forming the color palette.
- The execution time to extract the most three dominant colors is approximately **0.1 seconds**.



Search By Image

- The MobileNet model extracts features from the query image, generating a **1 x 50176-dimensional vector** within a time frame of **20ms to 50ms**.
- Utilizing **cosine-similarity metrics**, the closest 5 products to the query image are calculated. Comparing the query image with **5000 images**, from the closest to the farthest, typically takes approximately **0.7s to 1s**.



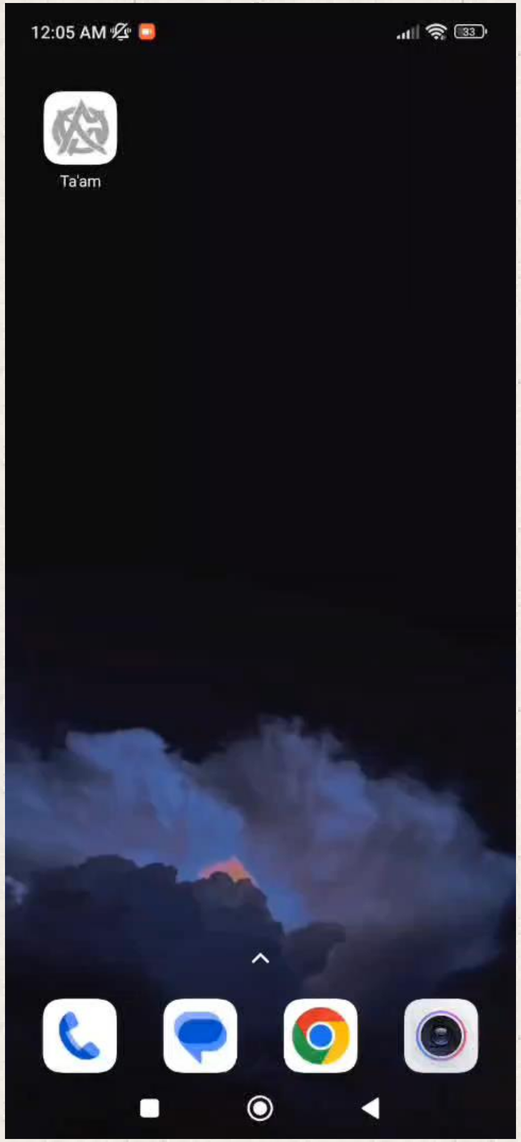
Experiments and results

- The **vector dimensionality is reduced to 5000** using an AvgPooling Layer followed by a Dense Layer, resulting in the model being **10 times faster**.
- Comparison is **limited to the same category** rather than all categories, making the model **13 times faster.**
- Overall, the model now takes around **30ms to 50ms** to execute.

	VGG16	ResNet50	MobileNet
Feature Extraction time per image	1s to 2s	20ms to 50ms	20ms to 50ms
Vector Dimension	25088	100352	50176
Cosine-similarity with 5000 images	0.4s to 0.7s	1.5s to 1.8s	0.7s to 1s

- VGG16 Results weren't satisfying compared to ResNet50 and MobileNet.
- Given that the results of ResNet50 and MobileNet were similar, MobileNet was selected due to its **faster processing speed**.

Demo

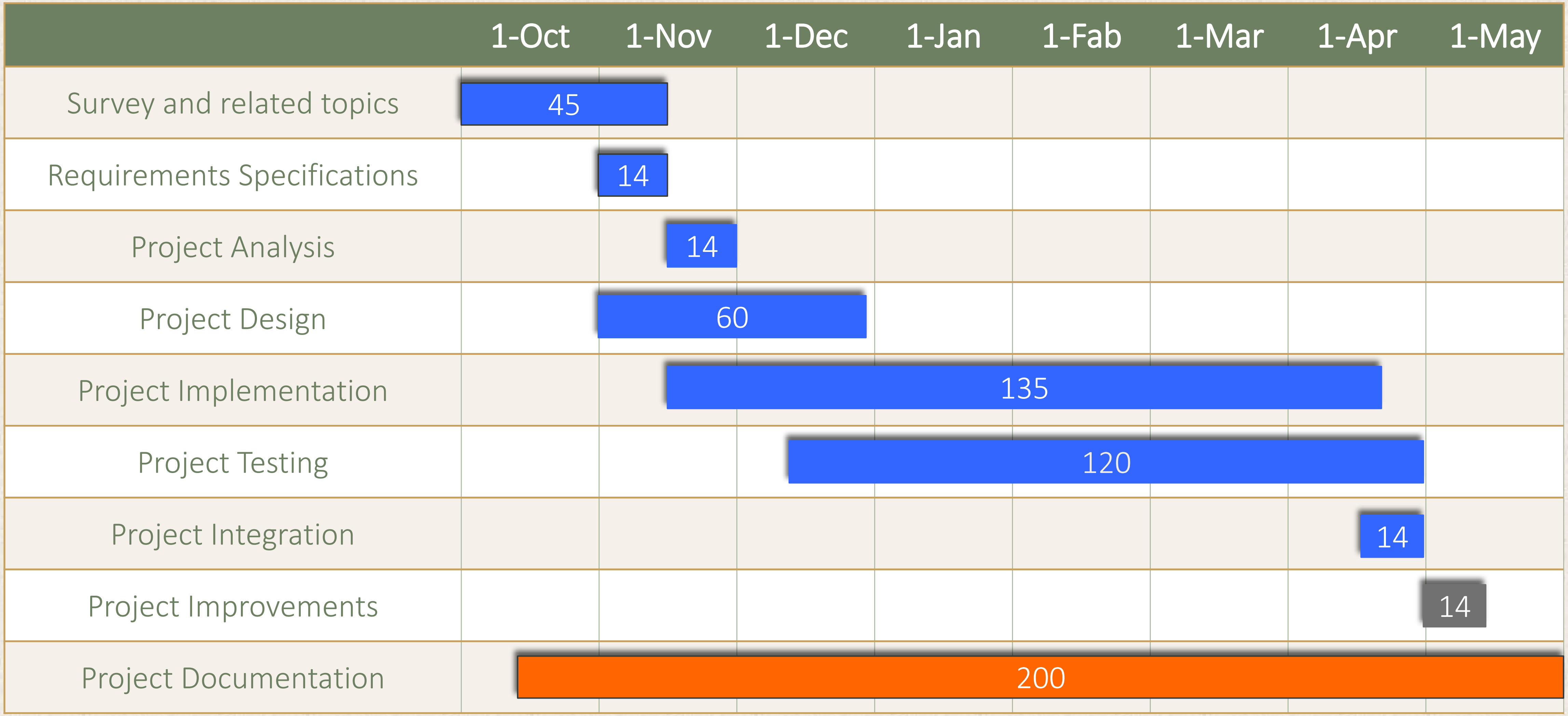

12:05 AM
33
Ta'am

What’s Next?

- Apply the text-based search model.
- Complete the documentation by providing comprehensive information on all aspects of the project.

Time Plan

Time Plan



🟦 Finished
🟧 In Progress
⬛ To Do
1-Oct
1-Nov
1-Dec
1-Jan
1-Fab
1-Mar
1-Apr
1-May
Survey and related topics
45
Requirements Specifications
14
Project Analysis
14
Project Design
60
Project Implementation
135
Project Testing
120
Project Integration
14
Project Improvements
14
Project Documentation
200

References

References

1. *A. Paulauskaite-Taraseviciene, et al., "An intelligent solution for automatic garment measurement using image recognition technologies," Applied Sciences, vol. 12, no. 9, pp. 4470, 2022.*

2. *B. Lao and K. A. Jagadeesh, "Convolutional Neural Networks for Fashion Classification and Object Detection," 2015.*

3. *S. S. Islam, E. K. Dey, M. N. A. Tawhid, and B. M. M. Hossain, "A CNN Based Approach for Garments Texture Design Classification," Adv. technol. innov., vol. 2, no. 4, pp. 119–125, May 2017*

4. *T. Sennikova, "Clothes Classification with the DeepFashion Dataset and Fastai," towardsdatascience, Available: https://towardsdatascience.com/clothes-classification-with-the-deepfashion-dataset-and-fast-ai-1e174cbf0cdc, Accessed: 19/11/2023.*

5. *M.Ibrahim Mohamed, “CalorieMe Image-based Calorie Estimator System”, 2023*

Thank you

Any Questions?