Classifying images of clothes for online shopping

Louie Daans

Project Pitch

Problem based on market research

Online second-hand clothing marketplaces (e.g., Vinted) saw their sales skyrocketing during the COVID-19 pandemic, which is good for them, but also evokes a new problem. As those marketplaces are becoming bigger, they face challenges managing large amounts of clothing items. Currently, sellers manually add information like clothing type, brand, colour and size when listing an item on the marketplace. This is time-consuming, but more importantly, prone to errors, impacting search and user experience.

Proposed solution

Image classification could offer a solution to this by analysing the images of the listings being made and automatically generating tags for the corresponding clothing type (t-shirt, pants, shoes, …), brand (Nike, Lacoste, …) and colour (blue, red, blue, …). For this project I focused on classifying clothing types. Implementing this system would make users experience more fluid, as well for the seller (easier) as for the visitors (more accurate search-results).

Further potential implementations of Image classification

- Visual search (search for desired clothing items using a picture)
- Style categorisation (propose clothing items that match a certain style / outfit)
- Quality check (search for damage on clothes)

(1 minute)

Project Pitch

Main stakeholder and needs

Because of its high power and influence over the project combined with interest in the platform success, our main stakeholder is **the organisation itself (marketplace platform).** Key needs:

- Increased user engagement and satisfaction

(we offer improved search functionality and accurate product information)

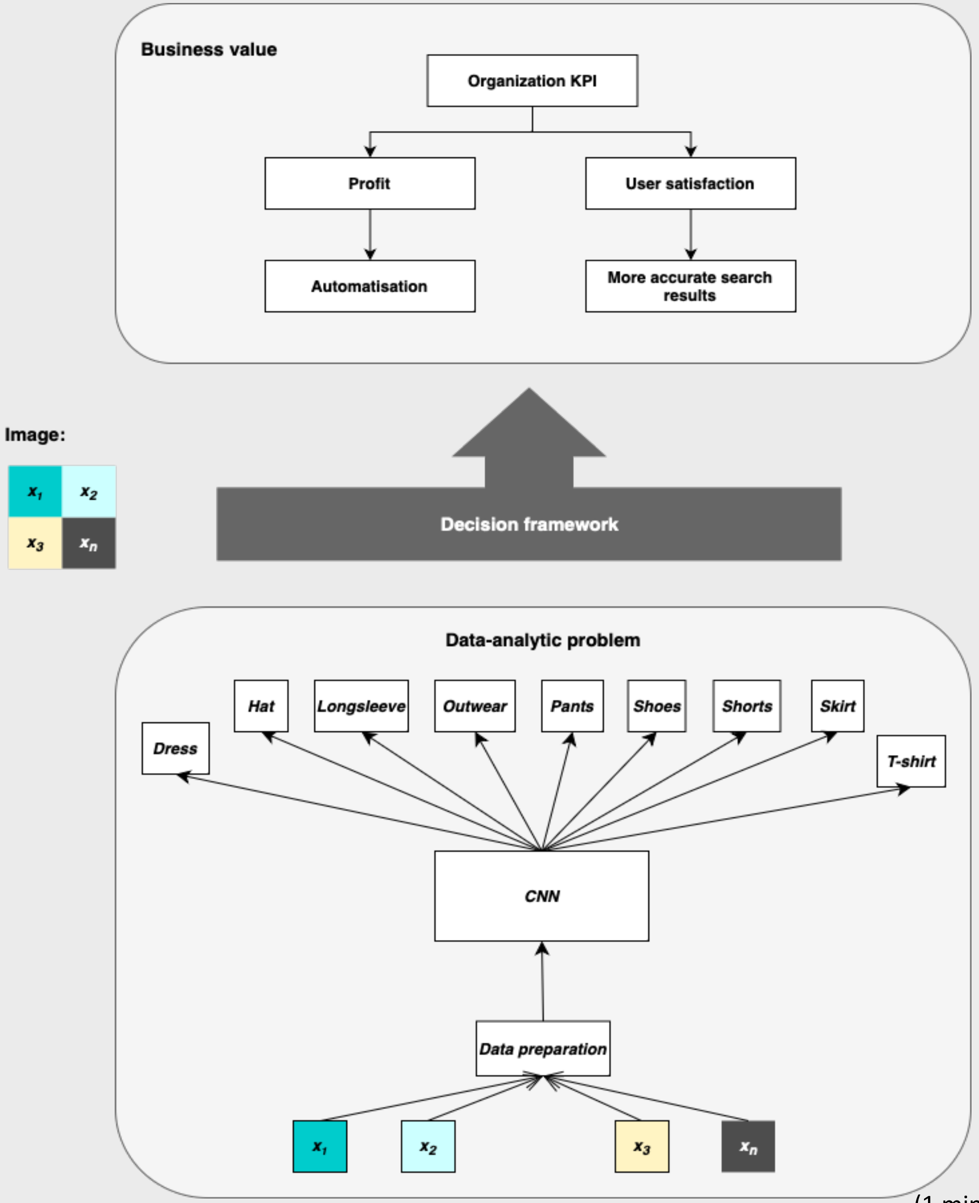
- Improved operational efficiency

(we offer automatisation, which can free up resources for other areas)

- Cost savings

(our solution can potentially reduce manual workload and associated costs.)

DAPS diagram



Business value

Organization KPI

Profit

User satisfaction

Automatisation

More accurate search results

Image:

x_1 x_2 x_3 x_n

Decision framework

Data-analytic problem

Dress

Hat

Longsleeve

Outwear

Pants

Shoes

Shorts

Skirt

T-shirt

CNN

Data preparation

x_1 x_2 x_3 x_n

(1 minute)

Problem Overview

Dataset information

The model classifies images of clothes from marketplaces such as H&M, Zalando and Vinted.

Although more categories of clothes exist, we will use only 9 classes for this project.

class:	Dress	Hat	Longsleeve	Outwear	Pants	Shoes	Shorts	Skirt	T-shirt
image example:									
n images in dataset:	190	188	153	141	176	194	177	153	171

Range ->]140:195[

Accuracy baselines

Random guess accuracy: 0.111

Human-level performance accuracy: 0.954

Basic Multilayer Perceptron accuracy: 0.467

(1 minute)

Iterations Overview

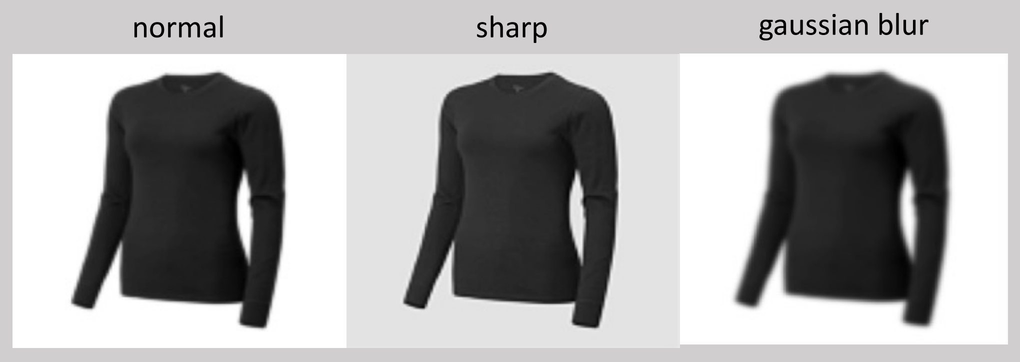
Iteration 1

For the first iteration I compared filters (normal / sharp / gaussian blur).
I imported my dataset as grayscale with a shape of 128x128 pixels.

Model size: 2.75 MB

Accuracy(normal): **0.656**

Problem: Only very small differences in performance between the 3 different types of images used, which is not enough to conclude on what's best to use.



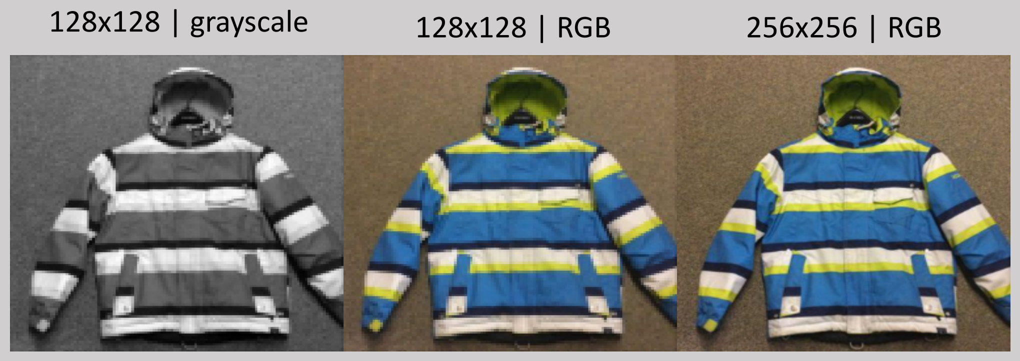
Iteration 2

Compared colour and size in the second iteration.

Model size: 8.37 MB

Accuracy(128x128 | RGB): **0.675**

Problem: Again, only very small differences in performance between the 3 different types of images used, which is not enough to conclude on what's best to use.



Iteration 3

Using Data augmentation

Model size: 2.75 MB

Accuracy: **0.773**

Problem: Data augmentation helps a lot, but the model still makes the same type of wrong predictions (error analysis)

(1 minute)

Iterations Overview

Iteration 4 (FINAL MODEL)

Data augmentation + Transfer learning (VGG-16)

Model size: 64.26 MB

Accuracy: **0.864**

Problem: The predictions are getting better, but we haven’t reached our HLP-baseline of 0.954.

Architecture of final model

VGG16	(4, 4, 512)
(Flatten)	(8192)
Dense	(256)
Dense	(128)
Dropout	
Dense	9

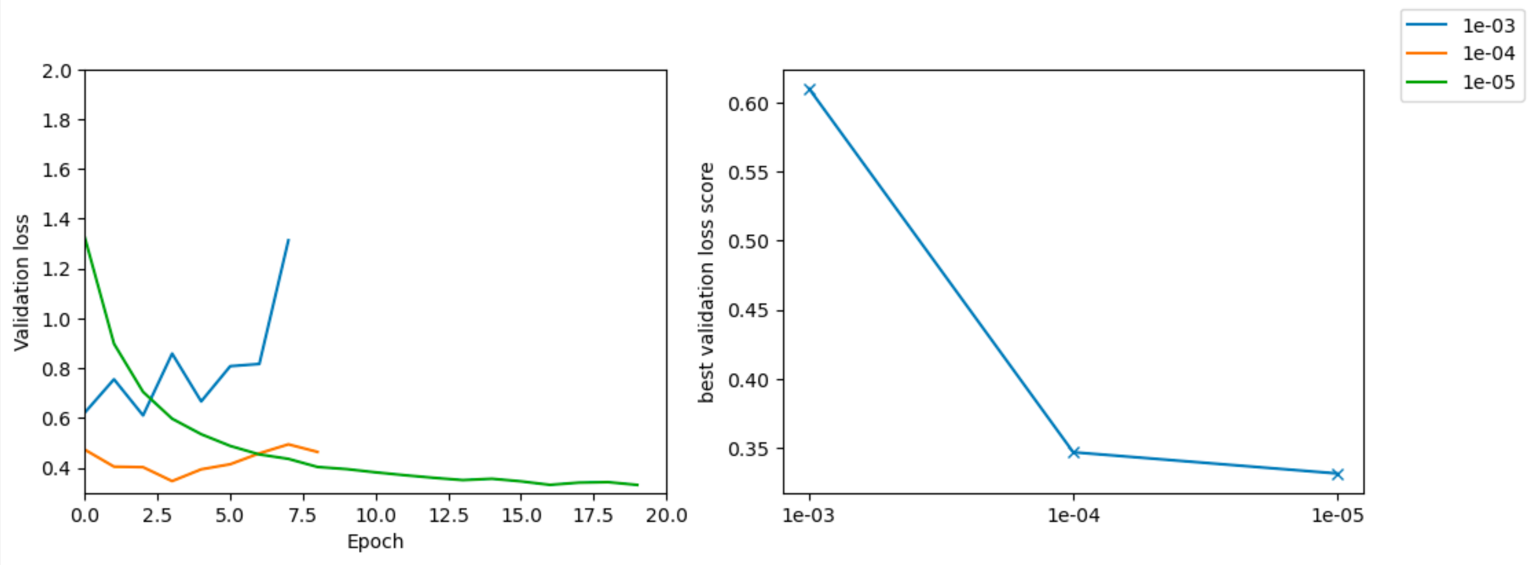
Total params: **16 846 153**

trainable params: 2 131 465
non-trainable params: 14 714 688

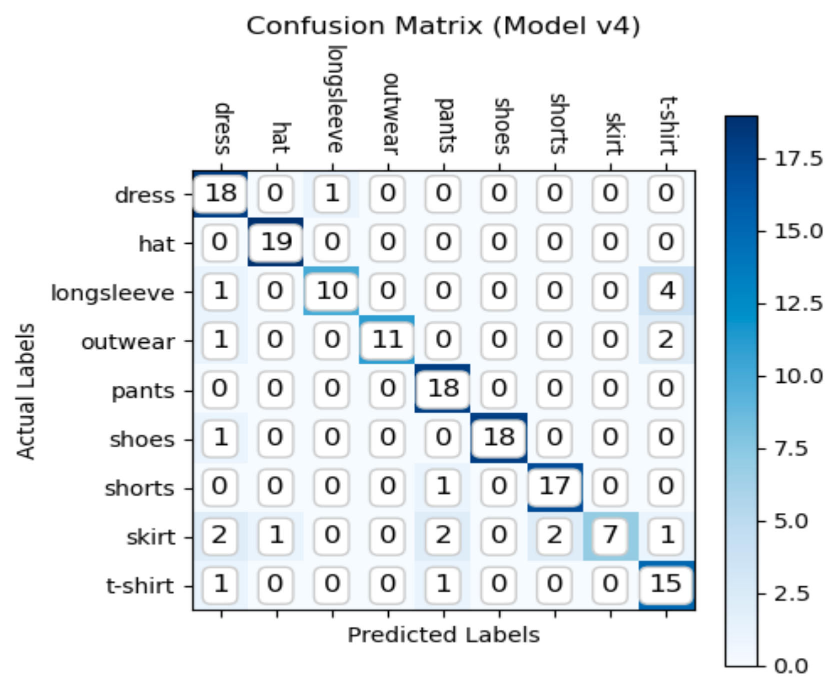
(1 minute)

Model Performance

Comparing **validation losses**:

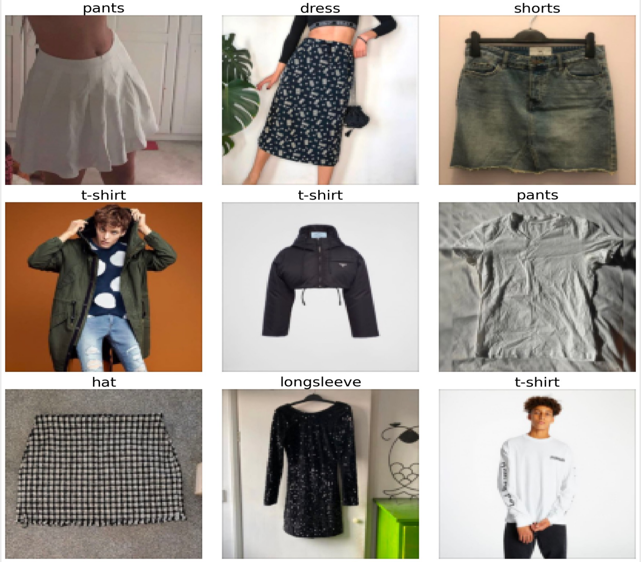


Confusion Matrix (21/154 misclassified)



	dress	hat	longsleeve	outwear	pants	shoes	shorts	skirt	t-shirt
dress	18	0	1	0	0	0	0	0	0
hat	0	19	0	0	0	0	0	0	0
longsleeve	1	0	10	0	0	0	0	0	4
outwear	1	0	0	11	0	0	0	0	2
pants	0	0	0	0	18	0	0	0	0
shoes	1	0	0	0	0	18	0	0	0
shorts	0	0	0	0	1	0	17	0	0
skirt	2	1	0	0	2	0	2	7	1
t-shirt	1	0	0	0	1	0	0	0	15

Example of 9 **False predictions** (error analysis)



Final scores

Best validation loss: 0.331

Precision: 0.89

Recall: 0.85

Accuracy: **0.864**

(1 minute)

Model Interpretability

Trade-off between accuracy and interpretability

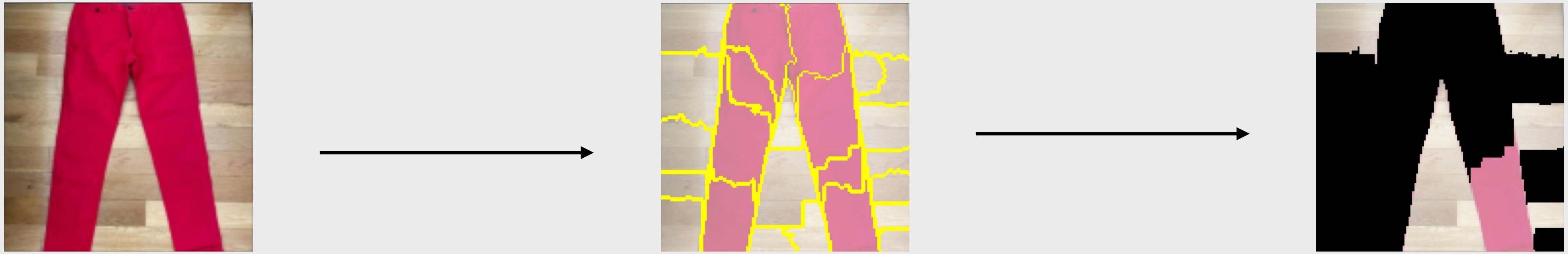
Discussing the trade-off between accuracy and interpretability within a ML-project is important. Do we want to focus on explaining ‘why?’ our model predicted a certain class, or do we just want to have the most accurate prediction possible without caring about the rest?

For our project, the most important is to focus on high level of accuracy. While interpretability can be interesting for the project-team to improve their model, it isn’t useful for the users of our model. Moreover, implementing too much interpretability could affect our accuracy and can have large computational cost.

In more diagnostic tasks e.g., analysing medical images for disease detection, it is important to prioritise interpretability since doctors might gain insights into how the disease manifests visually, helping in treatment decisions.

Explainable AI method applied (used MODEL V3 -> Iteration 3)

As explainable AI method I used LIME to understand why my model predicts that an image belongs to a certain class.

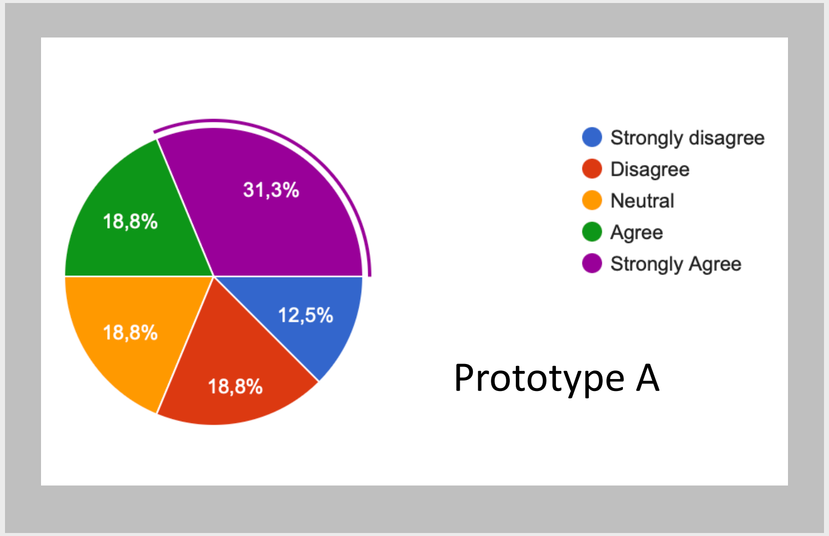


We can see on the last image that the most important features in the picture are the ones that form the shape of the pants. More interestingly, we can also see that the lower part of the pants is important, perhaps to distinguish between pants and shorts. It’s concerning that our model doesn’t prioritise key-features that differentiate pants from other clothes, like pockets or belt-holders.

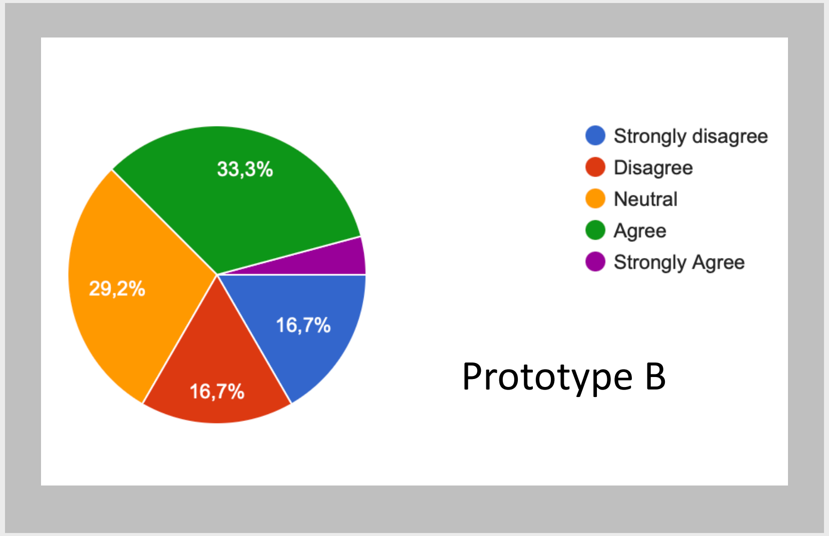
(2 minutes)

User Study

For the A/B test I left prototype (A) unchanged and changed some visuals for when the model predicts something in prototype (B). My hypothesis was that prototype B would have better responses for ‘The implementation of AI in the app is clear’. I had 16 participants for prototype A and 24 for prototype B.



Prototype A

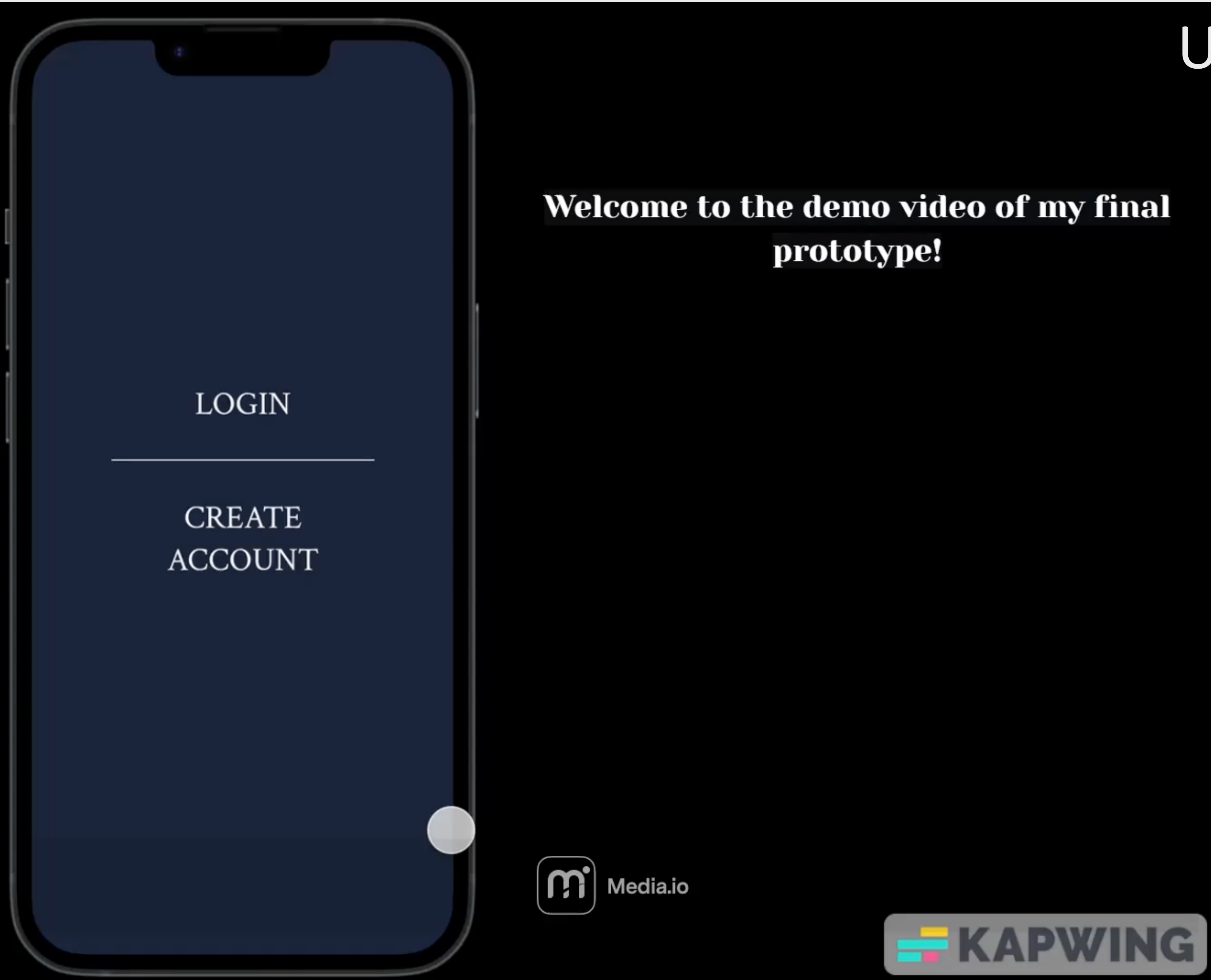


Prototype B

Prototype B has worse responses than A, which is concerning. After analysing the results in a t-test, I concluded that my hypothesis can’t be validated nor rejected because the number of participants is too small for each test, giving too much space to personal opinion.

(1 minute)

User Study



LOGIN

CREATE ACCOUNT

Welcome to the demo video of my final prototype!

Any questions?

Thank you!