

ECE 445  
SENIOR DESIGN LABORATORY  
DESIGN DOCUMENT

---

**InWhat: An Intelligent System for  
Wardrobe Design Document for ECE 445**

---

**Team #19**

CHUQIN ZHANG  
(chuqinz2@illinois.edu)

BO PAN  
(bopan3@illinois.edu)

SHUHONG XIAO  
(shuhong3@illinois.edu)

TA: Xinyi Xu

March 24, 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Solution . . . . .	1
1.3	Visual Aid . . . . .	1
1.4	High-level Requirements List . . . . .	2
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	Block Diagram . . . . .	3
2.2	Subsystem Overview . . . . .	4
2.2.1	Coat Hanger Subsystem . . . . .	4
2.2.2	Clothes Photo Info Extraction Subsystem . . . . .	7
2.2.3	Outfit Recommendation Subsystem . . . . .	10
2.2.4	Interact Subsystem . . . . .	12
<b>3</b>	<b>Cost and Schedule</b>	<b>15</b>
3.1	Cost Analysis . . . . .	15
3.2	Schedule . . . . .	16
<b>4</b>	<b>Ethics and Safety</b>	<b>17</b>
4.1	Ethics . . . . .	17
4.2	Safety . . . . .	17
	<b>References</b>	<b>18</b>

# 1 Introduction

## 1.1 Background

Many people worry about what to wear tomorrow and waste a lot of time trying to find the clothes they want. Moreover, with the improvement of living standards, people pay more attention to how to dress better and how to choose a satisfying outfit according to different needs. To solve these problems, our group come up with an amazing solution, an Intelligent System for Wardrobes.

## 1.2 Solution

To address the problem stated above, we decided to build an intelligent system for the wardrobe. Whenever a user gets new clothes, he/she can register the new clothes by inputting the photo of clothes to our system by phone or pad and it will be labeled automatically (labels could be categories like “jeans” and attributes like “collared”, “woolen”, “red”, etc.). The user can also modify labels if they are not satisfied with them. The system will also allocate a free coat hanger for it and send a message by a Wi-Fi microchip to light up the LED on that coat hanger. After putting the clothes on the hanger, the user can push the button on the hanger to turn off the LED and finish registering. Every day when the user wakes up, he/she can ask our system to recommend what he/she should wear (the decision will be based on criteria like the weather, the place user want to go to, the color matching degree, etc.) After the user become satisfied with a recommended outfit, the LED on the hanger will be lit to help the user find the clothes.

## 1.3 Visual Aid

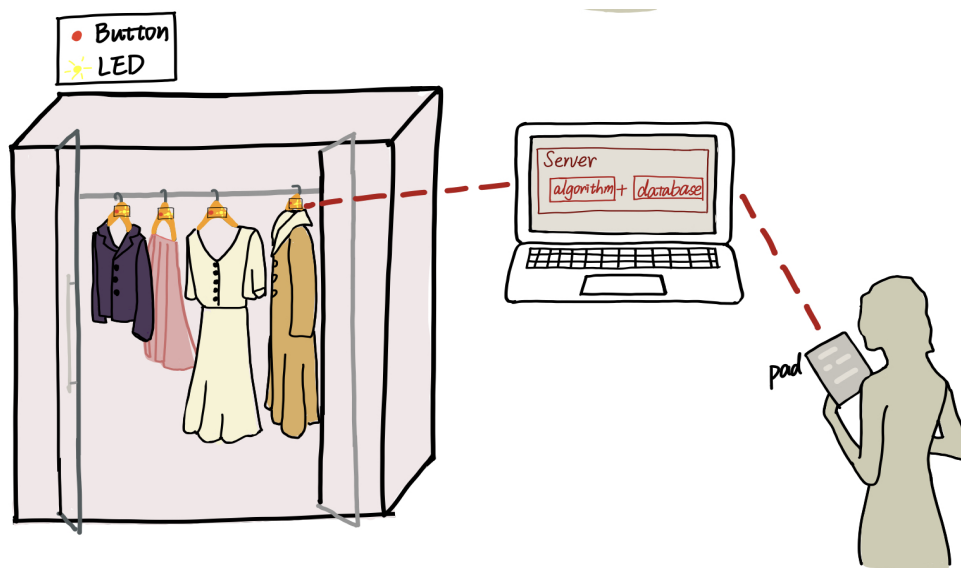


Figure 1: Physical Design

## 1.4 High-level Requirements List

- Must recognize the features of the clothes put in with an acceptable accuracy (accuracy 50% for category recognition and recall 30% for attributes recognition).
- Must have a usable recommendation function that give some valuable suggestion (at least 50% of the recommendations should make sense for at least 50% test users). Should include some randomness to make the recommendations flexible and allow the unsatisfied users a second chance.
- Must have a user-friendly interface (at least 80% test users should find it easy to use). Must allow the users easily and quickly find where the chosen clothes are (the users can find the recommended clothes within 5 seconds).

## 2 Design

### 2.1 Block Diagram

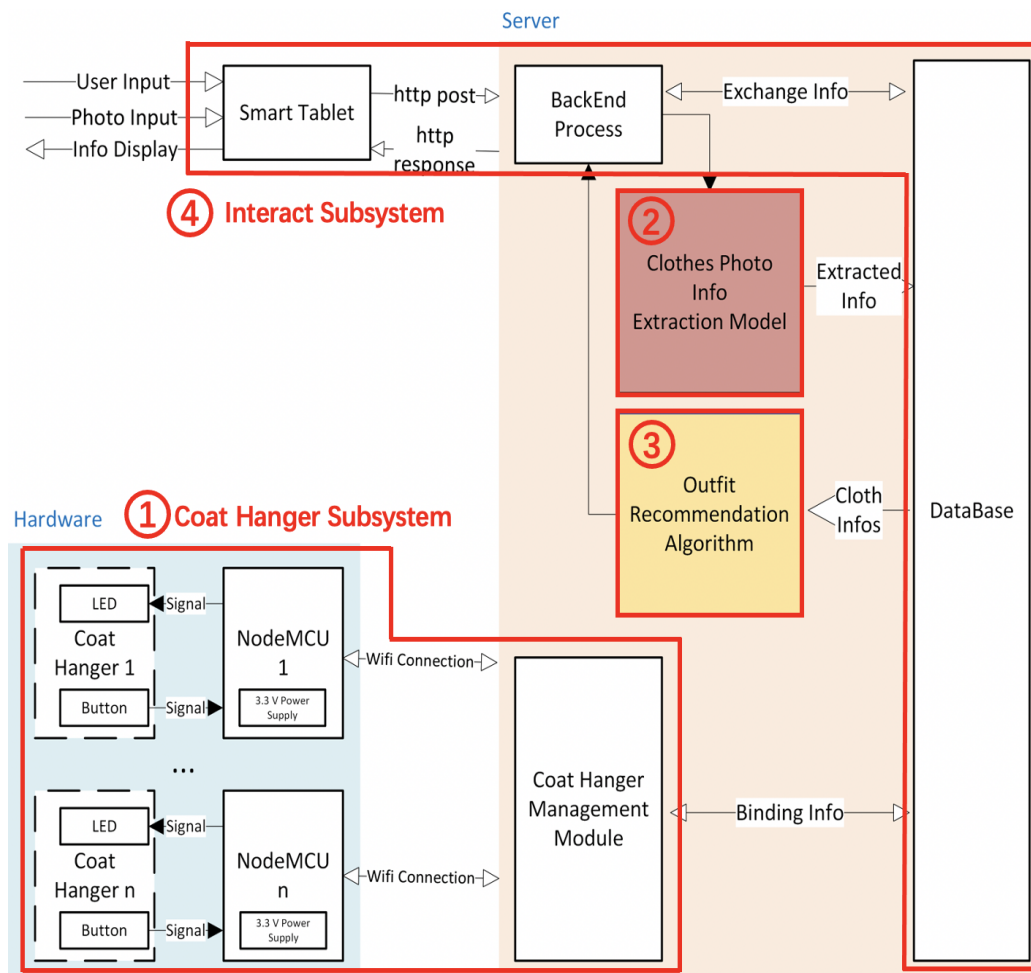


Figure 2: block diagram

The block diagram gives an overview for our whole system. The blue part are hardware and the orange part are software running on the server. The system is divided into four subsystems: The interact subsystem contains the end user device (a pad), and a back end process dealing with the user's need. The back end process and the end device communicate through the http protocol and Wifi connection. And the back end process will use the Clothes Photo Info Extraction Subsystem to extract clothes info or use the Outfit Recommendation Subsystem to give user recommendations when needed. The Coat Hanger Subsystem contains the hardware part (LED, button, and NodeMCU chips which are attached on the hanger) and the Coat Hanger Management Module which is running on the server. The the Coat Hanger Management Module and the NodeMCU chip communicate through the http protocol and Wifi connection.

## 2.2 Subsystem Overview

### 2.2.1 Coat Hanger Subsystem

The Coat Hanger Subsystem consists of a physical part (including an ordinary coat hanger and a set of circuit to control the LED light and button, and a Wi-Fi microchip to communicate with our Interact Subsystem ) and a coat hanger management algorithm running on server, as shown in the Fig. 3 below.

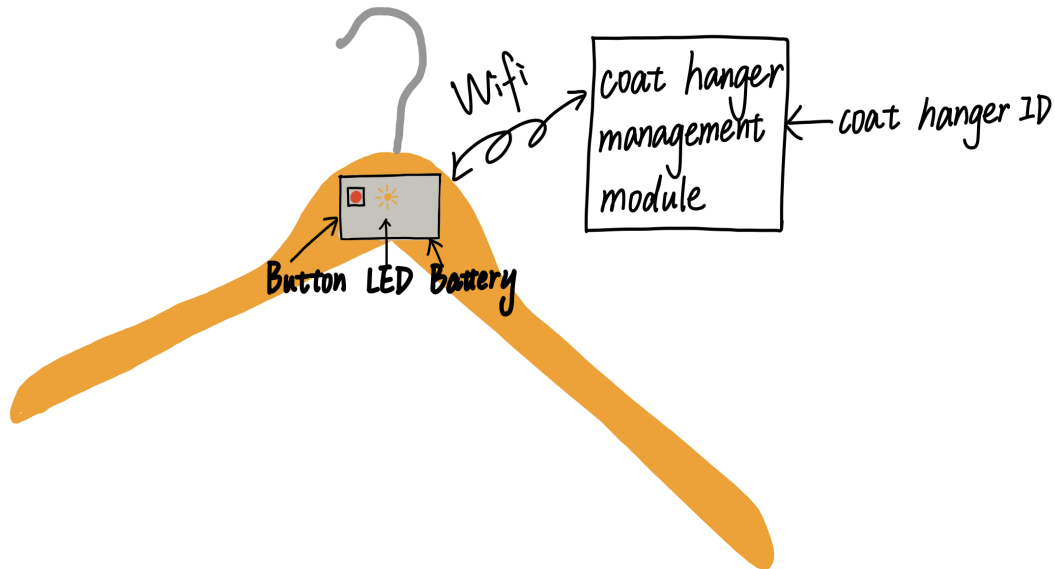


Figure 3: Physical Design for the Coat Hanger

The Coat Hanger Subsystem is designed to meet high-level requirement<sup>3</sup> (make a user-friendly interface and allow the user easily and quickly find where the chosen clothes are). This subsystem is to help users find the recommended clothes. To be more specific, when a user gets new clothes and the photo of it is inputted, the hanger management module will automatically allocate a free coat hanger for the clothes and through the hanger information stored in the database, find the hanger to communicate with. The LED attached to this coat hanger will light to tell the user its exact location and then the user can press the button to turn off the light if he/she finishes putting his/her clothes. When the user asks the wardrobe to recommend, a certain outfit is selected by the Outfit Recommendation Subsystem, the coat hangers bound to these clothes will be called and the LED will be lit to tell the user the exact locations.

For the circuit, Fig. 4 below shows a detailed schematic diagram. We choose NodeMCU (CPU:ESP8266), which allows the design of single-chip devices capable of connecting via Wi-Fi, as the Wi-Fi microchip. The input voltage allowed for ESP8266 is about 3.0V-3.6V and we choose 3.3V as our power supply. At 3.3V input, when transmitting data in Wi-Fi mode, the NodeMCU has a current of about 200mA. From *ESP8266 Low-Power Solutions*, a technical document by Espressif Inc. ESP8266 has three different sleep modes designed

to save power as shown in Fig. 5[1]. When it reaches deep-sleep mode, the current is about  $20\mu\text{A}$ . Assuming we use a battery with a capacity of 2000mAh, we can use it for about 11.4 years in deep sleep mode. In practice, the lasting time of the Wi-Fi module depends on the actual usage of users and when the battery power is exhausted, the user can easily replace it.

In this subsystem, we also have a hanger management algorithm running on the server. The algorithm should form a communication with the microchip equipped on coat hangers. At both clothes registering stage and recommendation stage, a request from Interact Subsystem will ask this algorithm to notify the chosen coat hanger, light the LED. After registering, this algorithm will be notified by hangers and should tell the Interact Subsystem that the task is complete. Considering the privacy issue, the target coat hanger information stored in the database is designed to be encrypted and this algorithm is the only one that holds both encryption and decryption.

Requirements	Verification
<ol style="list-style-type: none"> <li>1. The coat hanger can hold up to 2 kilograms (A piece of clothing is generally between 500g-2000g).</li> <li>2. Implement reliable communication between coats hangers and server (control delay within 2s and achieve above 90% accuracy).</li> </ol>	<ol style="list-style-type: none"> <li>1. Do a small load-bearing experiment. Apply two kilograms of force evenly to a coat hanger to see if it breaks.</li> <li>2. Within 3 minutes, sent about 10 messages. Measure the delay time and see if the LED lit up accurately. The above tests may be carried out several times, and the delay time and accuracy should be recorded.</li> </ol>

### Tolerance Analysis

The power supply we choose is about 3.3V, which is generally used for our Wi-Fi module, ESP8266. The input voltage allowed for ESP8266 is about 3.0V - 3.6V, so to meet the power transmission specification, the voltage input should be within  $\pm 0.3\text{V}$  ( $\pm 9.1\%$  error). There are also some errors in the impedance of the resistors and LEDs, but these small errors can be ignored because they will only influence the brightness of the led but not the operation of the circuit.

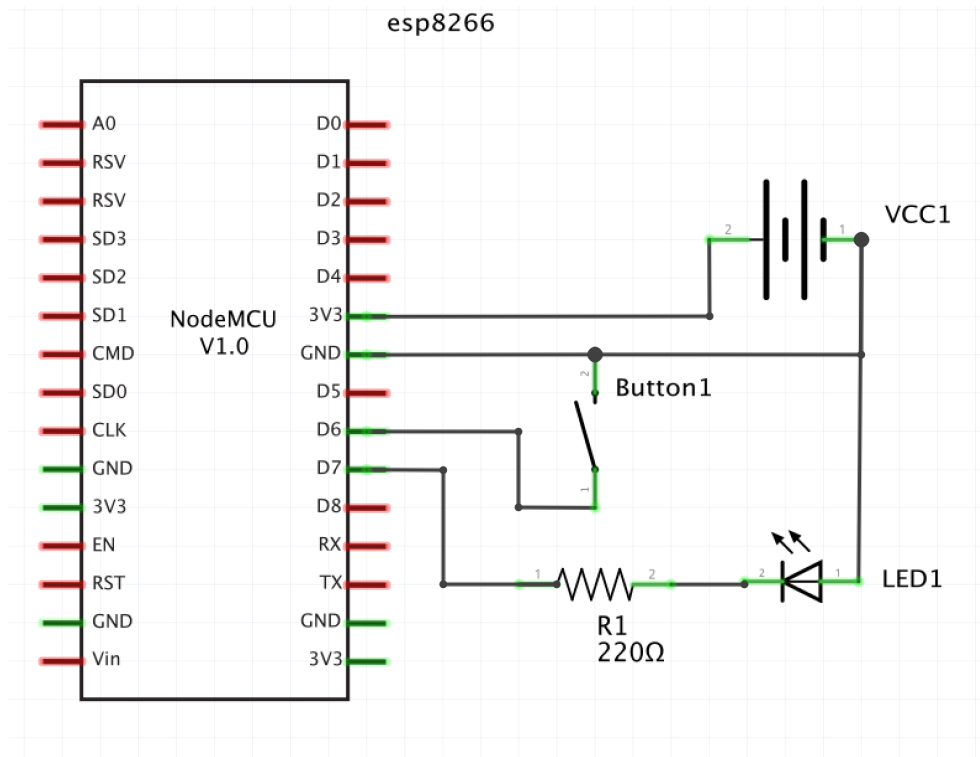


Figure 4: The Schematic Diagram for Circuit

		Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi Connection		on	on	off
RTC		on	on	on
CPU		on	pause	off
Substrate Current		15mA	0.4mA	~20 $\mu$ A
Average Current	DTIM=1	16.2 mA	1.8mA	-
	DTIM=3	15.4 mA	0.9mA	-
	DTIM=10	15.2mA	0.55mA	-

RTC: Real-Time Clock

DTIM (Delivery Traffic Indication Message): The frequency at which packets are sent wirelessly when using a wireless router

Figure 5: Three Sleep Patterns Comparison from Espressif Inc. (ESP8266 Producer)



### 2.2.2 Clothes Photo Info Extraction Subsystem

This subsystem serves as one of our core algorithms running on the server, is designed to meet the first high-level requirement. The first task of this algorithm is to get the clothes landmark that serves as a bounding box. The second task is to predict clothes category and attributes. Every clothes is assigned with one category like jeans and several attributes such as floral, striped and knit. In the extraction stage, this algorithm will generate category and attributes for each clothes image registered by Interact Subsystem and store a unique image index with its category and attributes in our database, the category will be stored as a category index and attributes will be stored as a string of 1/0, with each one represents one attribute.

We use FashionNet Fig. 6 as our extraction model[2]. The network accepts images with input dimension 224x224, the first four layers(conv1-conv4) have the same structure as VGG-16[3], (conv5\_pose-fc7\_pose) works as a landmark extractor, then both output from conv4 and fc7\_pose are fed into pool5\_local which serves as a landmark pooling layer, together with a global pooling achieved by conv5\_global, give the prediction of category and attributes.

We use a large-scale data set, DeepFashion [2] to train our model. This data set contains more than 280,000 images and is divided into 50 categories with 1000 attributes labeled for each image. To achieve less training cost and higher prediction accuracy, we adjust the category and attributes. The detailed strategy shows as follows: For category, the number of categories is decreased to 20 as Fig. 7, we delete and merge the category which contains samples less than 1000, and those serves at home such as onesie. For attributes, we delete those attributes that only be positive in less than 500 samples. Correlations between attribute pairs are computed and for two attributes with high correlation, only one should be retained. Here we define the correlation between  $attribute_i$  and  $attribute_j$  as

$$C(i, j) = \frac{N_i \cap N_j}{N_i \cup N_j}$$

, where  $N_i/N_j$  denotes the number of image with  $attribute_i/attribute_j$  is positive.

For the training process, we will use the mmfashion [4], which is a fashion analysis toolbox developed by Multimedia Lab, CUHK. Here we also define the loss function used in our model in Fig. 6. First, a  $L_2$  regression loss is used for landmarks

$$L_{landmarks} = \sum_{j=1}^{|D|} \|v_j \cdot (\hat{l}_j - l_j)\|_2^2$$

where  $D$ ,  $\hat{l}_j$ , and  $v_j$  denote the number of training samples, the ground truth of the landmarks of the j-th sample and visibility of the landmark of j-th sample. Second, a 1-of-K softmax loss is used for category as

$$L_{category} = \frac{1}{D} \sum_{j=1}^D (-\log p_i)$$

where  $D$  and  $p_i$  denote the number of training samples and the softmax function. Third, the attribute is predicted by a weighted cross-entropy loss

$$L_{attributes} = \sum_{j=1}^{|D|} (w_{pos} \cdot \mathbf{a}_j \log p(\mathbf{a}_j | \mathbf{x}_j) + w_{neg} \cdot (1 - \mathbf{a}_j) \log(1 - p(\mathbf{a}_j | \mathbf{x}_j)))$$

where  $\mathbf{x}_j$  and  $\mathbf{a}_j$  denote the  $j$ -th image and attribute labels,  $w_{pos}$  and  $w_{neg}$  are two coefficients determined by the ratio of the number of positive and negative samples in our training set[2].

Requirements	Verification
1. The prediction of top 5 category accuracy should reach 50%. 2. The prediction of top 5 Attributes recall should reach 30%.	1. Test the category prediction use our test set, for every one sample in our test set, if the true category label appear in the top 5 prediction, we consider the prediction as successful, otherwise failed. 2. Test the attributes prediction use our test set, the top 5 recall here define as $\frac{TP}{5}$ .

### Tolerance Analysis

1. One challenge here is that the accuracy of both category and attributes prediction in the clothes recognition field is still at a low level. The definition of clothing categories is vague, and the labeling of attributes varies from person to person, which makes it difficult to annotate data sets according to a certain standard, the low quality of the data set makes the model hard to learn. The consequence of poor prediction accuracy is that the wrong category and attributes will be fed into our Outfit Recommendation Subsystem and further affect recommendation performance. The solution we have to increase the accuracy here is that we give top5 or even top 10 prediction results of category and attributes to users and users can make a further correction, choose the result they agree and send to our Outfit Recommendation Subsystem so that the recommendation performance will not be affected.

2. The second challenge is that the quality of images in our data set varies. People wearing the clothes always have different poses and face directions, which makes it hard to detect the clothes accurately, and further influences the prediction and recommendation performance. Our solution here is to suggest users upload their photos maintaining good posture so the extraction can be achieved with high accuracy.

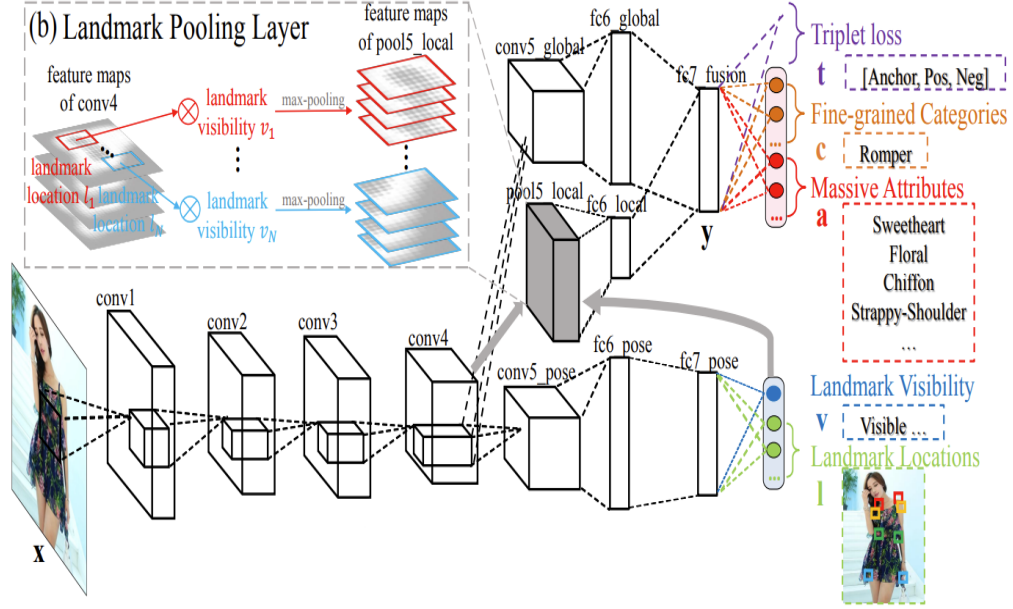


Figure 6: FashionNet

Upper-cloth (20)	Anorak, Blazer, Blouse, Bomber, Button-Down, Cardigan, Flannel, Halter, Henley, Hoodie, Jacket, Jersey, Parka, Peacoat, Poncho, Sweater, Tank, Tee, Top, Turtleneck	Upper-cloth (9)	Blazer, Blouse, Coat, Hoodie, Jacket, Parka, Sweater, Tank, Tee
Lower-cloth (14)	Capris, Chinos, Culottes, Cutoffs, Gauchos, Jeans, Jeggings, Jodhpurs, Joggers, Leggings, Sarong, Shorts, Skirt, Sweatshorts, Trunks	Lower-cloth (9)	Chinos, Culottes, Cutoffs, Jeans, Jeggings, Joggers, Shorts, Skirt
Full-cloth (16)	Caftan, Cape, Coat, Coverup, Dress, jumpsuit, Kaftan, Kimono, Nightdress, Onesie, Robe, Romper, Shirtdress, Sundress	Full-cloth (2)	Dress, Romper (Jumpsuit)

Figure 7: Category Change

### 2.2.3 Outfit Recommendation Subsystem

This subsystem serves as the core algorithm to meet the second high-level requirement: give some valuable suggestion (at least 50% of the recommendation should make sense for at least 50% test user). It should also include some randomness to make the recommendation flexible and allow the unsatisfied user a second chance.

The workflow of the algorithm is shown in Fig. 8. The user specification input is obtained from the Interact Subsystem. Then based on the prior knowledge in the first Bayesian network, a globe combination (e.g. upper clothes + lower clothes + coat) is chosen. Then the system will decide which specific clothes to choose one by one. The first clothes (the upper cloth in Fig. 8) is chosen based on the user specification and another Bayesian network, while the following clothes will be chosen based on the previous clothes just chosen. To avoid style conflict, the algorithm will tend to choose clothes similar in style to the previous one.

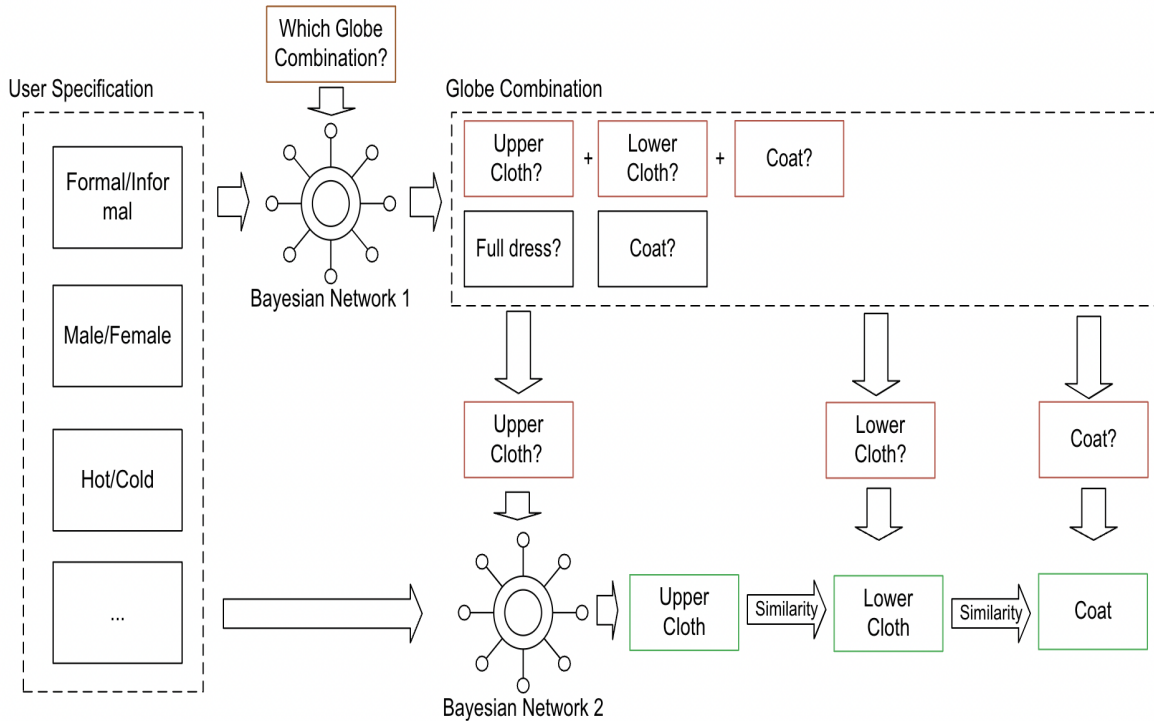


Figure 8: Recommendation Process

The reason we use the Bayesian network in our design is that what to wear is a decision out of people's belief. Due to this subjective nature, we decided to take a subjective Bayesian view[5] to design our algorithm. Our target is to integrate some "consensus knowledge" into our system so that the users can save themselves from spending much time and attention on that. For example, black clothes and wool materials are not recommended on hot days is a "consensus knowledge". If our system can apply that for the user, they can spend more time and attention on their personal preference. One philos-

ophy to keep in mind during our design process is that our system should never go too far: we should not interfere with users' preferences at any stage of the algorithm.

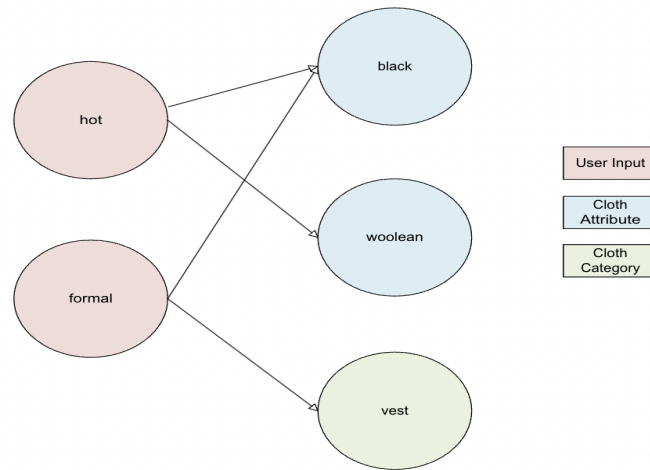


Figure 9: Bayesian Network Example

As we can see in our example Fig. 9, the dependence edges in the network capture those “consensus knowledge” that is widely accepted by most people (e.g. hot can affect whether to choose black cloth, formal can affect whether to choose a vest, etc.). However, the edge in the network should be far less than a complete graph. On the one hand, this choice makes the computation more efficient. On the other hand, this is a reflection of our design philosophy described before: we should not interfere with the users' preferences.

Requirements	Verification
1. At least 50% of the test user should be satisfied with the recommendation result.	1. Find a group of volunteers, teach them how to use the system, then let them imagine a scenario (e.g. today is hot, you will need to go to an interview). Ask the volunteers to input their specification to the system through the user interface. Record if they find the result they want (they can ask the system to regenerate the result until they get tired with that). If more than 50% the user are satisfied with the result in the end, we regard this subsystem as a successful implementation.

## Tolerance Analysis

1. The result of the algorithm can only be judged subjectively by the test user. And it's challenging to make sure the user is satisfied with our recommendation. To deal with that, we introduce randomness to our algorithm and allow the algorithm to be ran multiple times. If each recommendation has 30% probability to make the user satisfied, with 10 alternative recommendations the user only have 2.82% probability to reject all the recommendations.

## 2.2.4 Interact Subsystem

This subsystem, which contains a database running in our server and a web tool on a smart tablet like a pad, is used for human-machine interaction. The connection between the web and our server will base on the HTTP request. This subsystem is mainly designed to satisfy the high-level requirement that users should easily and quickly find where the chosen clothes are.

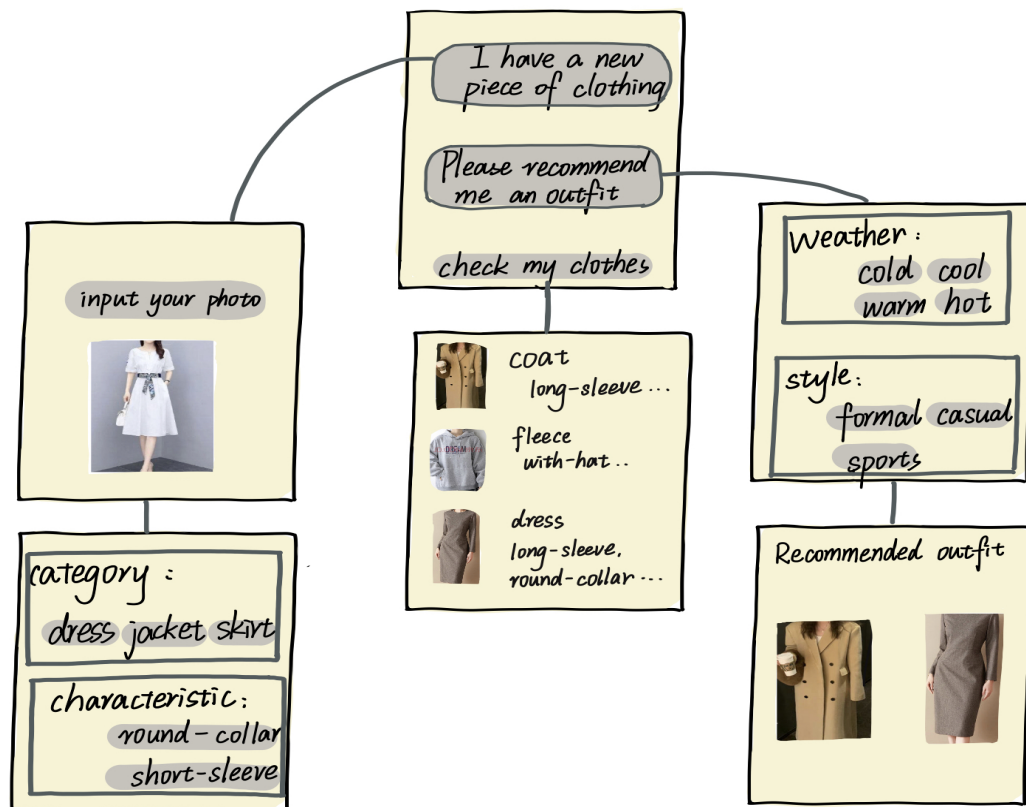


Figure 10: Interact with Users

We design the functions of this subsystem as the following three scenarios as Fig. 10: The first scenario is for the user to register new clothes, our web will support image upload from pad (the image should be taken beforehand and stored in pad), then the predicted clothes category and attributes by our Clothes Photo Info Extraction Subsystem will be shown to the user, the user then chooses the most conformed category and attributes,

finally these data will be stored in our database. The second scenario is for the user to get a recommendation, the user needs to choose from our predefined requirements, such as weather conditions. Then, several outfits will be shown and the user can choose one they like best, then the data of chosen clothes will be stored in our database, later accessed by the algorithm in the Coat Hanger Subsystem. The third scenario is that the user can view all the registered clothes, we do not allow the user to do additions, modifications, or deletions which ensure the security and stability of the data.

```

CREAT TABLE img_info(
    img_id int(10) unsigned NOT NULL AUTO_INCREMENT ,
    img_name varchar(50) DEFAULT NULL ,
    img_content varchar(255) DEFAULT NULL ,
    img_category int(10) DEFAULT NULL ,
    img_attributes varchar(100) DEFAULT NULL,
    PRIMARY KEY (img_id)
)

CREAT TABLE img_hanger_bind(
    img_id int(10) unsigned NOT NULL AUTO_INCREMENT ,
    is_bind int(10) unsigned NOT NULL,
    hanger_id int(10) unsigned ,
    img_name varchar(50) DEFAULT NULL ,
    PRIMARY KEY (img_id),
    FOREIGN KEY (hanger_id) REFERENCES hanger_status(hanger_id)
)

CREAT TABLE hanger_status(
    hanger_id int(10) unsigned NOT NULL AUTO_INCREMENT ,
    hanger_address varchar(50) DEFAULT NULL,
    is_bind int(10) unsigned NOT NULL,
    hanger_id int(10) unsigned ,
    hanger_status int(10) NOT NULL ,
    PRIMARY KEY (hanger_id)
)

```

Figure 11: Schemas

Database in this subsystem is responsible for all the data interaction, which contain data stored by the Clothes Photo Info Extraction Subsystem and data accessed by the Outfit Recommendation Subsystem. Here we define three tables in SQL, with their schemas in Fig. 11. The table "img\_info" stores image with its category and attributes generated by the Outfit Recommendation Subsystem; table "img\_hanger\_bind" gives the corresponding binding of clothes and our coat hanger; table "hanger\_status" record all the coat hanger, if they have already bound, a encrypted communication address will be recorded, the algorithm in the Coat Hanger Subsystem owns the secret key which can achieve encryption and decryption.

We will use Django to manage the web and database, the advantage of using Django is that it turns our database into a "Model Class" after connecting and we can either use the native SQL query to access the database or work on the "Model Class" just like a python dictionary. We design four pages described as follows: the navigation page serves as the initial interface, provides users with access to the registration page, recommendation page and collection page; the registration page provides the function of image upload and prediction selection; the recommendation page provides the function of requirement selection and recommendation viewing; the collection page is for the user to manage all the registered clothes.

Requirements	Verification
<ol style="list-style-type: none"> <li>1. The web tool should be user-friendly and our users can quickly learn how to use it in 15 minutes.</li> <li>2. The web and database should be stable and not easy to break.</li> </ol>	<ol style="list-style-type: none"> <li>1. Find a group of volunteers, test whether they can get familiar with both registering and recommending without any instruction.</li> <li>2. For web tool, test its stability using scripting tools, either detect the script and deny the access or sustain 100 visits per second; for database, test it by writing illegal data into each table, if all requests are denied, we can prove that database is also stable.</li> </ol>

### **Tolerance Analysis**

1. A database running on a server is always in danger of being hacked, which further cause User privacy leakage. To deal with possible privacy issues, we use the AES algorithm to encrypt data such as image content and hanger address. Only the algorithm in Coat Hanger Subsystem has the secret key and can decrypt.



## 3 Cost and Schedule

### 3.1 Cost Analysis

Graduated from UIUC, the average salary of an CompE student is \$99,145/year[6] and for ME student, the average salary is \$79,117/year[7]. With average person works 2080 hours/year. For our team, equipped with two CompE students and one ME students, we will have a salary of  $\$99,145/2080 \times 2 + \$79,117/2080 = \$133.37/hour$  in total. Except the very early weeks for brainstorming, we assume 10 weeks of working time, with 10 hours working time for each person each week, we have a total labor cost:

$$3 \times \$133.37/hours \times 10weeks \times 10hours/week = \$40,010$$

Our parts costs are listed below, \$4.08 for each.

Part	Cost
ESP8266 Wi-Fi microchip	\$1.83
5mm LED	\$0.05
DS-211/213 button	\$0.16
2.8AH Li-ion battery×2 (AA Portable Power; Nanfu)	\$0.47
raw coat hanger	\$1.57
Total	\$4.08

We plan to produce 3 coat hanger for demo, so the final total cost is

$$\$40,010 + 3 \times \$4.08 = \$400,22.24$$

### 3.2 Schedule

week	Bo pan	Chuqin Zhang	Shuhong Xiao
3/21	Derive bayesian model used in recommendation subsystem. Produce design document.	Read manual and try some demo on ESP8266. Produce design document.	Process data set, fix some bugs for training code for extraction subsystem. Produce design document.
3/28	Analysis attributes in data set, reduce the quantity.	Django, HTML, CSS learning.	label new attributes special for our system.
4/4	Train extraction model on GPU.	Set up database and tables.	Train extraction model on GPU.
4/11	Write code for bayesian network.	Establish hardware communication.	Test and tune recommendation result.
4/18	Begin with web design.	Produce and assemble coat hangers.	Begin with web design
4/25	Continue with web design.	Write management code in server for coat hanger, demo it.	Continue with web design.
5/2	Finish web design.	Coat hanger test.	Test the individual function of each subsystem.
5/9	Combine all the subsystem, debug.	Prepare for demo and presentation.	Prepare for demo and presentation.
5/16	Conduct environmental testing.	Begin writing final report.	Begin writing final report.
5/23	Finish writing final report.	Finish writing final report.	Finish writing final report.

## 4 Ethics and Safety

### 4.1 Ethics

According to the ACM Code of Ethics, term 1.6 “ensuring the accuracy of data, understanding the provenance of the data... The retention and disposal periods for that information should be clearly defined, enforced, and communicated to data subjects...” [8], computing professionals should always take special care for privacy when merging data collections. For our project, to train our intelligent system and make it to recognize the characteristics of clothing, we must use a huge data set. The data set we want to use in our project collected by MMLAB (The Chinese University of Hong Kong), which is available for non-commercial research purposes and whose images are obtained from the Internet [4]. Therefore, this public data set can be obtained for educational and research purposes and it is reliable and accurate to use. Moreover, for the photos input by user, we will keep them highly confidential to prevent users’ privacy from being leaked. When transmitting data, we will also take some encryption measures.

Secondly, by the IEEE Code of Ethics, Section I Policy 5, “to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others.”[9] Our group team welcome all the different opinion and criticism from all kind of groups such as sponsors, teaching assistance and classmates.

Moreover, if our group can successfully build this intelligent system, we should pay more attention to copyright issue. From the ACM Code of Ethics, term 1.5, “Computing professionals should therefore credit the creators of ideas, inventions, work... respect copyrights, patents... and other methods of protecting authors’ works.”[8] To protect the intelligent property of this project, we will save our work carefully and turn to legal protection if necessary.

### 4.2 Safety

The IEEE Code of Ethics, Section I Policy 1 “To hold paramount the safety, health, and welfare of the public... ” is important when considering the use of some electrical devices. For example, for our battery use, which is mounted on ESP8266, operates at about 3.3V and it is generally safe to use in our daily life. Moreover, the Wifi-enabled part of ESP8266 will not interfere with the operation of other wireless devices nearby.

## References

- [1] E. Inc. “Esp8266 low-power solutions.” (2019), [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/9b-esp8266-low\\_power\\_solutions\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf) (visited on 03/24/2022).
- [2] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [4] X. Liu, J. Li, J. Wang, and Z. Liu, “MMFashion: An Open-Source Toolbox for Visual Fashion Analysis,” in *ACM Multimedia 2021, Open Source Software Competition*, 2021.
- [5] S. Russell, S. Russell, P. Norvig, and E. Davis, *Artificial Intelligence: A Modern Approach*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, 2010, ISBN: 9780136042594. [Online]. Available: <https://books.google.com.hk/books?id=8jZBksh-bUMC>.
- [6] G. Communications. “Salary averages.” (2020), [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages> (visited on 03/24/2022).
- [7] —, “Employment and salary.” (2020), [Online]. Available: <https://mechse.illinois.edu/graduate/graduate-degree-programs/master-engineering-mechanical-engineering/employment-and-salary> (visited on 03/24/2022).
- [8] ACM. “ACM Code of Ethics and Professional Conduct.” (2018), [Online]. Available: <https://www.acm.org/code-of-ethics> (visited on 03/09/2022).
- [9] IEEE. “IEEE Code of Ethics.” (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 02/08/2020).