

네이버
쇼핑렌즈 구현

LookUs

목 차

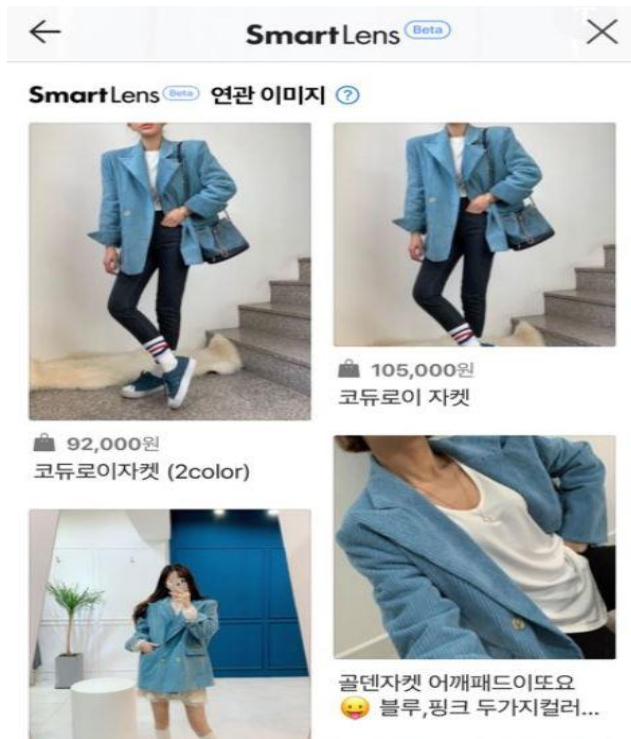
1. 프로젝트 개요
2. Look us Lens 알고리즘
3. Look us Lens 웹 구현하기
4. 프로젝트 결과

1. 프로젝트 개요

영화나 드라마, 혹은 거리에서
누군가가 입은 옷을 보고

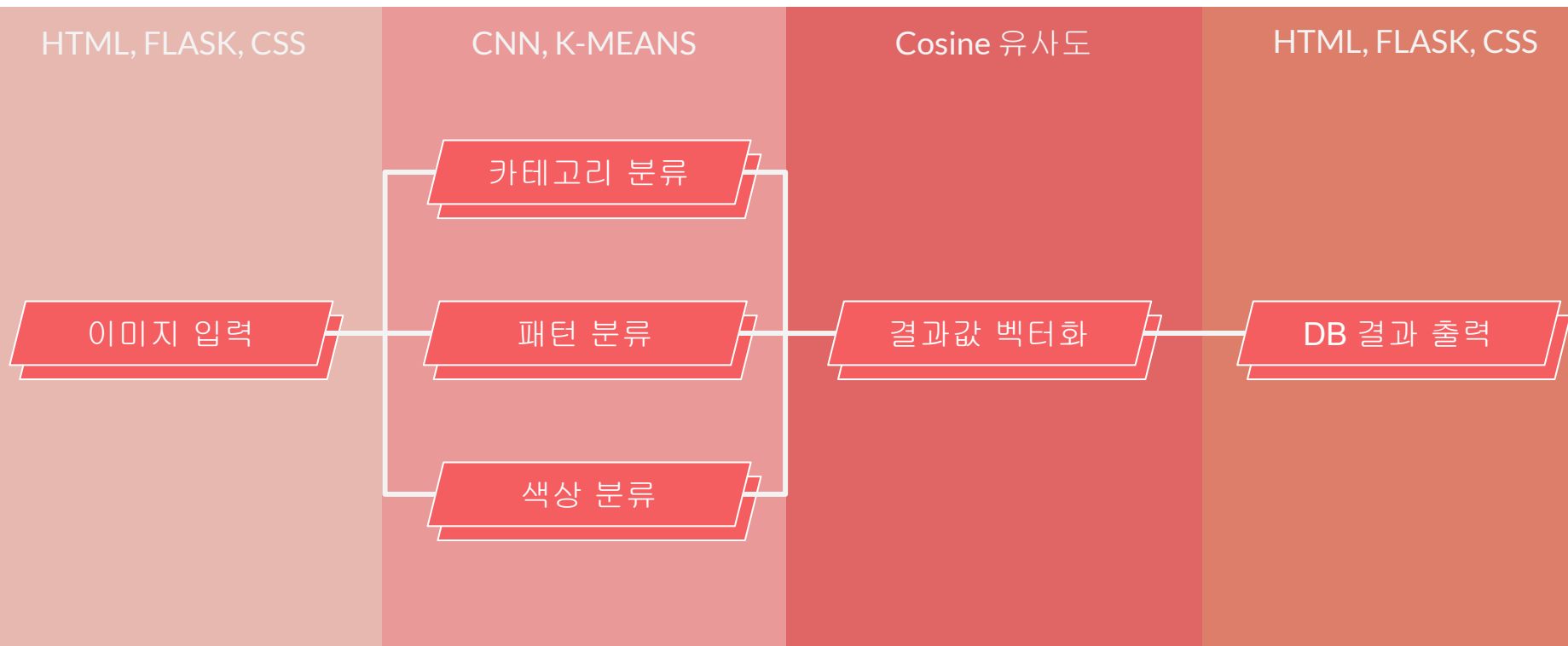
‘저 옷 어디서 팔지?’, ‘비슷한
디자인의 다른 옷 없나?’라고
궁금해 하신 적 있으신가요?

이러한 문제를 해결하기 위해
패션아이템을 사진으로
검색하고, 유사한 상품을
찾아볼 수 있는 패션 이미지
검색 기술을 구현하려 합니다.



1. 프로젝트 개요

◆ LookUs Project Flow-Chart




2. 쇼핑 렌즈 알고리즘 구현하기

1. 이미지 수집
2. k-means 군집분석을 통한
색상추출
3. 카테고리 CNN 모델 학습
4. 패턴 CNN 모델 학습
5. 데이터베이스 - 벡터화 하기
6. 코사인 유사도로 유사 상품 출력

2-1. 이미지 수집

카테고리 분류 cnn 모델을 학습시킬 때, 패턴이 영향을 안 미치도록,
패턴 분류 cnn 모델을 학습시킬 때, 카테고리가 영향을 안 미치도록
데이터를 수집

(수집 방법) Chrome Crawler 자체 개발

< 카테고리 >	< 패턴 >	< 색상 >
<ul style="list-style-type: none">맨투맨셔츠바지치마 <div data-bbox="197 751 633 886"><div>train 900</div><div>val 100</div><div>test 200</div></div>	<ul style="list-style-type: none">민무늬큰 가로 줄무늬체크무늬세밀한 가로 줄무늬세로 줄무늬땡땡이 무늬프린팅 <div data-bbox="718 751 1155 886"><div>train 900</div><div>val 100</div><div>test 200</div></div>	

자체 크롤러 개발

```
def cloths_crawler(name) :
    # 1. dirver 경로/파일 지정
    driver = webdriver.Chrome("c://")
    # driver - 브라우저를 조정할수 있는 객체

    # 2. 이미지 검색 url
    driver.get("https://www.google.co.kr/imghp?hl=ko") # 구글에 이미지를 검색할 수 있는 창이 나온다.
```

카테고리



셔츠

티셔츠

치마

바지

패턴



민무늬

체크

큰 줄무늬

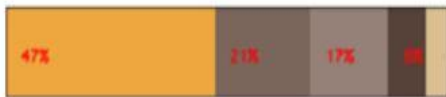
세밀 줄무늬

프린팅

땡땡이

세로줄무늬

2-2. K-means 군집분석을 통한 색상 추출



RGB 이미지의 픽셀들



K-means 알고리즘을 이용하여
픽셀들 클러스터링 하기



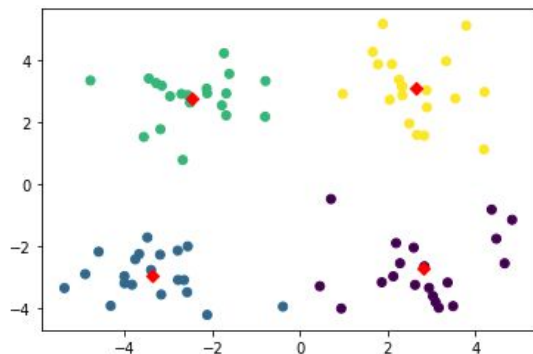
각 클러스터의 중심 좌표

< 각 클러스터의 중심 좌표 >

```
[ [213.91598059 219.43220108 230.12752849]
[ 35.09403255 75.17049858 96.93929217]
[126.88448374 138.59697661 143.92498574]
[ 5.19160505 7.87103088 11.10961288]
[ 7.0815678 124.23463983 175.61811441]]
```


kMeans 실행

```
# kMeans 실행
clt = KMeans(n_clusters = args["clusters"])
clt.fit(image)
```



각 클러스터에 해당하는 픽셀수 - 히스토그램으로 그리기

```
# 각 클러스터에 속한 픽셀 수 세고, 히스토그램 반환하는 함수
def centroid_histogram(clt):
    # 클러스터 수 파악
    # clt : k-means 클러스터링 객체
    numLabels = np.arange(0, len(np.unique(clt.labels_))+1)
    # 클러스터에 할당된 픽셀 수를 히스토그램으로 작성하기
    (hist,)=np.histogram(clt.labels_, bins = numLabels)

    # 히스토그램의 합계가 10이 되도록 정규화하기
    hist = hist.astype("float")
    hist /= hist.sum()

    # 히스토그램 return
    return hist
```

각 군집의 상대적 비율 -barplot으로 그리기

```
def plot_colors(hist, centroids):
    # 이미지에서 가장 무제한 색상을 담을 수 있는 300x50 픽셀 직사각형을 정의한다
    bar = np.zeros((50,200,3)), dtype = "uint8")
    startX = 0

    # 각 군집의 백분율과 색상에 대한 루프
    for (percent, color) in zip(hist,centroids):
        # 각 군집의 상대적 비율을 그림으로 표시하다
        endX = startX + (percent * 300)
        cv2.rectangle(bar, (int(startX), 0), (int(endX), 50),
            color.astype("uint8").tolist(), -1)
        startX = endX

    # return the bar chart
    return bar
```



색상추출에서의 문제상황



해결

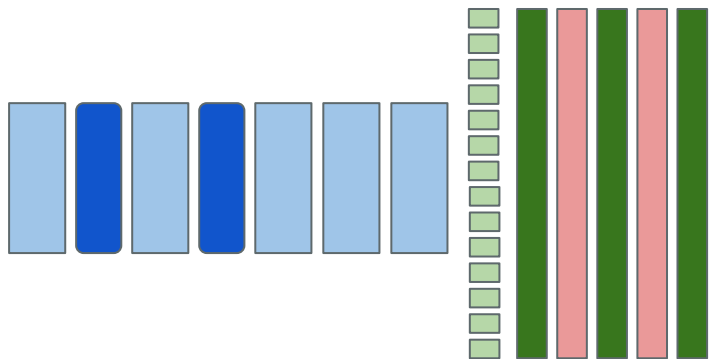


k-means의 군집수 1로 한 후
최대 비율 값을 추출하니
빨간색의 특징이 사라졌다.

k-means의 군집수 5로 바꿨더니
색깔 픽셀들이 평균을 띄며
좀 더 명확한 색상추출이
가능해졌다.

2-4. 카테고리 분류하는 cnn 모델 1

사용된 CNN layer



셔츠



티셔츠



치마



바지

conv2D

max_pooling

dense

drop_out

```
In [17]: model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 56, 56, 96)	34944
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
conv2d_1 (Conv2D)	(None, 27, 27, 256)	614656
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_2 (Conv2D)	(None, 13, 13, 384)	885120
conv2d_3 (Conv2D)	(None, 13, 13, 384)	1327488
conv2d_4 (Conv2D)	(None, 13, 13, 256)	884992
flatten (Flatten)	(None, 43264)	0
dense (Dense)	(None, 4096)	177213440
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4)	16388
=====		
Total params: 197,758,340		
Trainable params: 197,758,340		
Non-trainable params: 0		

```
# Hyper parameters
img_h = 224 # height
img_w = 224 # width
input_shape = (img_h, img_w, 3)
```

```
# 1. CNN Model layer
print('model create')
model = Sequential()
```

```
# Convolution layer1
model.add(Conv2D(96, kernel_size=(11, 11), activation='relu', strides=4, padding='same',
                 input_shape=input_shape))
model.add(MaxPool2D(pool_size=(3, 3), strides=2, padding='valid'))
# Convolution layer2
model.add(Conv2D(256, kernel_size=(5, 5), activation='relu', strides=1, padding='same'))
model.add(MaxPool2D(pool_size=(3, 3), strides=2, padding='valid'))
# Convolution layer3 : maxpooling() 제외
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu', strides=1, padding='same'))
model.add(Conv2D(384, kernel_size=(3, 3), activation='relu', strides=1, padding='same'))
model.add(Conv2D(256, kernel_size=(3, 3), activation='relu', strides=1, padding='same'))
```

```
# Flatten layer : 3d -> 1d
model.add(Flatten())
```

```
# DNN hidden layer(Fully connected layer)
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.2))
```

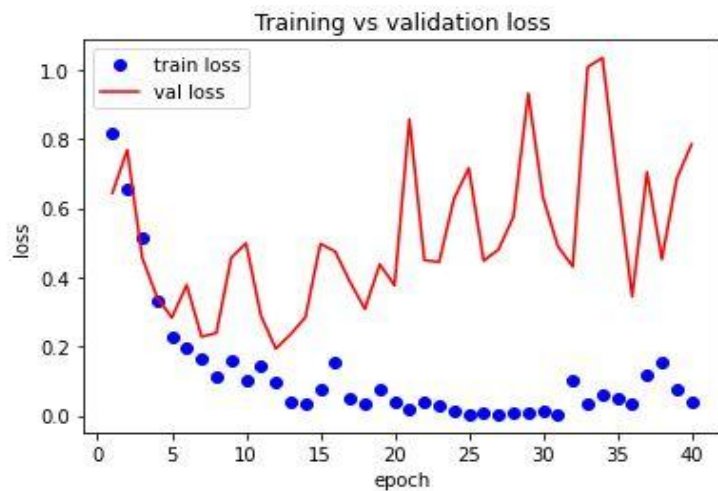
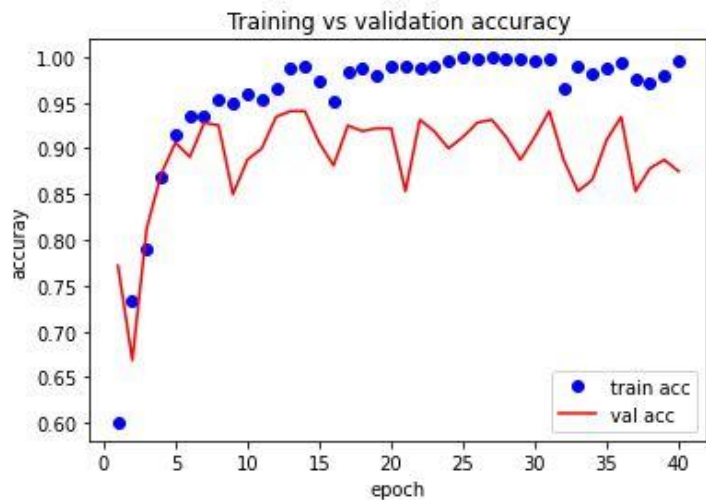
```
# DNN Output layer
model.add(Dense(4, activation='softmax'))
```

모델 환경설정에 따른 accuracy 비교

< 최적의 parameter >	
loss	sparse_categorical_crossentropy
metrics	['sparse_categorical_accuracy']
epochs	40
optimizer	adam
batch_size	32

모델 학습과정

accuracy
-92.3%



예측이 틀린 이미지 관찰 + 시사점

```
[ ] metrics.confusion_matrix(y_true, y_pred)
```

	셔츠	치마	바지	티셔츠
셔츠	78	0	2	0
치마	0	70	1	9
바지	2	3	69	6
티셔츠	0	6	1	73

치마와 바지에서 예측률이
낮았음



따라서, 데이터 베이스의 예측이
틀린 이미지를 보고자 함.

```
validation_generator.class_indices # {'가디건': 0, '점퍼': 1, '티셔츠': 2}
labels = ["셔츠", "치마", "바지", "티셔츠"]
```

성공여부

```
for i in range(1300):
    if y_true[i] == y_pred[i]:
        print(i, "success :", labels[y_true[i]])
    else:
        print(i, "fail : real({}) -> pred({})".format(labels[y_true[i]],
                                                         labels[y_pred[i]]))
```

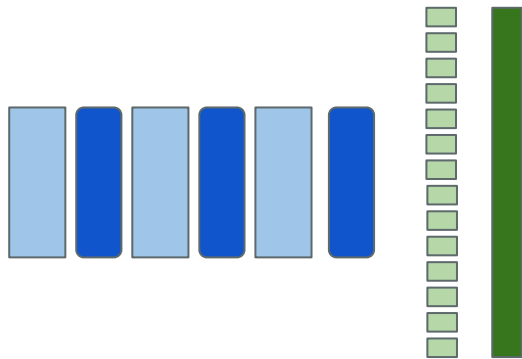
```
46 success : 셔츠
47 success : 셔츠
48 success : 셔츠
49 success : 셔츠
50 success : 셔츠
51 fail : real(셔츠) -> pred(바지)
52 success : 셔츠
53 success : 셔츠
54 success : 셔츠
55 success : 셔츠
56 success : 셔츠
57 success : 셔츠
58 fail : real(셔츠) -> pred(바지)
59 success : 셔츠
60 success : 셔츠
61 success : 셔츠
62 success : 셔츠
63 success : 셔츠
64 success : 셔츠
65 success : 셔츠
```



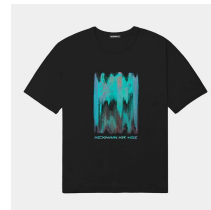
- 길이가 긴 치마를 바지로 예측하는 경향이 있다.
- 특징이 분명하지 않은 셔츠를 치마로 예측하는 경향이 있다.
- 이를 해결하기 위해 데이터셋을 조정

2-4. 패턴(무늬) 분류하는 cnn 모델 2

사용된 CNN layer



conv2D
max_pooling
dense



패턴 분류에서의 문제상황 :
팔 부분을 세로줄무늬로 인식
주름 부분을 세로 줄무늬로 인식



해결 : 적절한 위치로 crop



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 73, 73, 32)	896
max_pooling2d (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_1 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 17, 17, 64)	0
conv2d_2 (Conv2D)	(None, 15, 15, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 256)	1605888
dense_1 (Dense)	(None, 7)	1799
Total params: 1,700,935		
Trainable params: 1,700,935		
Non-trainable params: 0		

카테고리 분류모델보다 적은 노드
수로 높은 성능을 보였다.

```
# Hyper parameters
img_h = 75 # height
img_w = 75 # width
input_shape = (img_h, img_w, 3)

# 1. CNN Model layer
print('model create')
model = Sequential()

# Convolution layer1
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
                 input_shape = input_shape))
model.add(MaxPool2D(pool_size=(2,2)))

# Convolution layer2
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

# Convolution layer3 : maxpooling() 제외
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

# Flatten layer : 3d -> 1d
model.add(Flatten())

# DNN hidden layer(Fully connected layer)
model.add(Dense(256, activation = 'relu'))

# DNN Output layer
model.add(Dense(7, activation = 'softmax'))

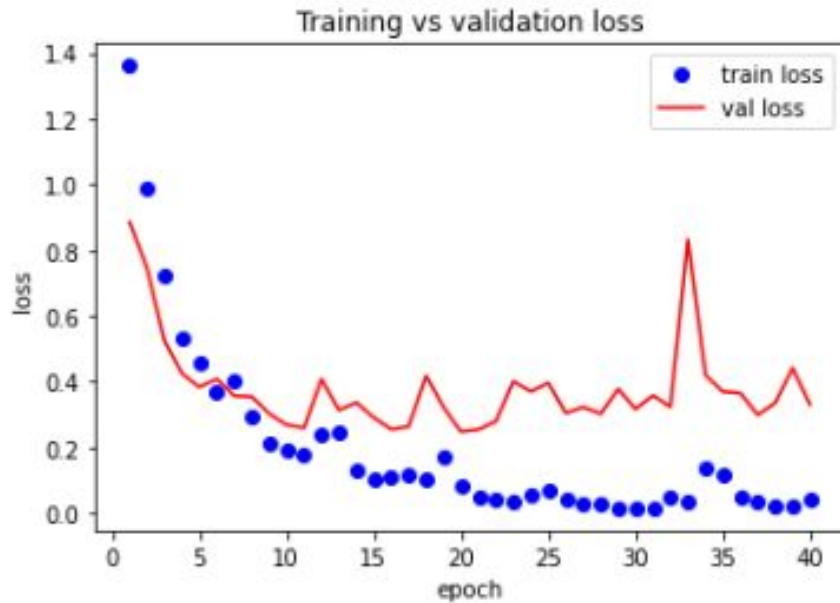
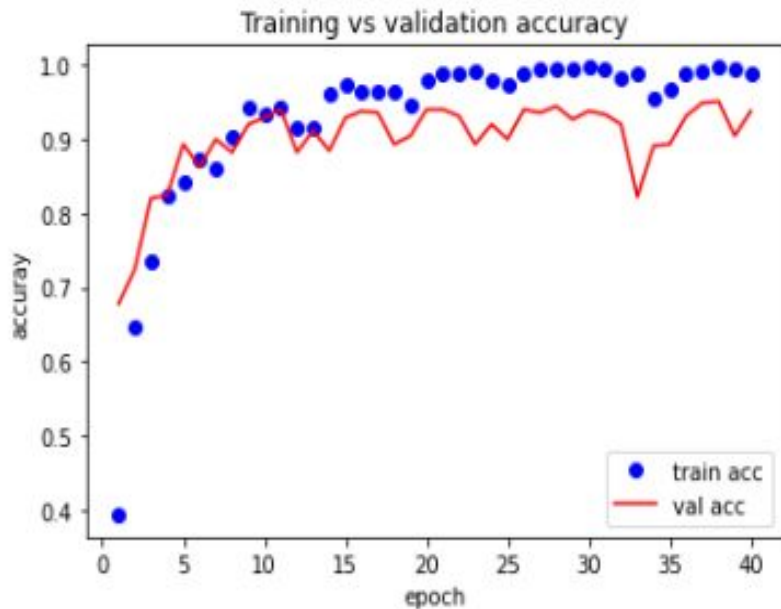
# model training set : Adam or RMSprop
model.compile(optimizer = 'adam',
              #loss = 'binary_crossentropy', # integer(generator가 integer로 읽어들임) + 이항분류
              #loss = 'categorical_crossentropy' # y:원핫인코딩
              loss = 'sparse_categorical_crossentropy', # Y=integer + 다항분류
              metrics = ['sparse_categorical_accuracy'])
```

모델 환경설정에 따른 accuracy 비교

< 최적의 parameter >	
loss	sparse_categorical_crossentropy
metrics	['sparse_categorical_accuracy']
epochs	40
optimizer	adam
batch_size	32

모델 학습과정

accuracy
93%



예측이 틀린 이미지 관찰 + 시사점

	민 무늬	빅 스트라이프	체크	도트 무늬	프린팅	세로 줄무늬	스트라이프
민 무늬	58	0	0	1	0	1	0
빅 스트라이프	1	51	2	1	0	0	5
체크	0	0	56	0	0	4	0
도트 무늬	0	1	2	55	2	0	0
프린팅	1	4	4	3	50	1	1
세로 줄무늬	0	0	3	0	1	56	0
스트라이프	0	7	0	0	0	0	53

프린팅 무늬에서 예측률이 낮았음

가로 줄무늬들끼리 혼동하는
경우가 많음

정확도

0.9047

```
validation_generator.class_indices
labels = ["민무늬", "큰줄무늬", "체크", "도트", "프린팅", "세로", "가는 줄무늬"]
# 성공여부
for i in range(1300):
    if y_true[i] == y_pred[i]:
        print(i, "success :", labels[y_true[i]])
    else:
        print(i, "fail : real({}) -> pred({})".format(labels[y_true[i]],
                                                         labels[y_pred[i]]))
```

```
254 success : 프린팅
255 success : 프린팅
256 success : 프린팅
257 success : 프린팅
258 success : 프린팅
259 fail : real(프린팅) -> pred(큰줄무늬)
260 success : 프린팅
261 success : 프린팅
262 success : 프린팅
263 fail : real(프린팅) -> pred(민무늬)
264 success : 프린팅
265 success : 프린팅
266 success : 프린팅
267 fail : real(프린팅) -> pred(도트)
268 success : 프린팅
269 fail : real(프린팅) -> pred(큰줄무늬)
270 success : 프린팅
271 success : 프린팅
272 success : 프린팅
273 success : 프린팅
```



- 프린팅 무늬 특성상 옷의 가운데 부분 이미지에 도트무늬이거나 줄무늬가 있는 것은 도트나, 줄무늬로 인식하였다.

2-5. 데이터베이스에 저장



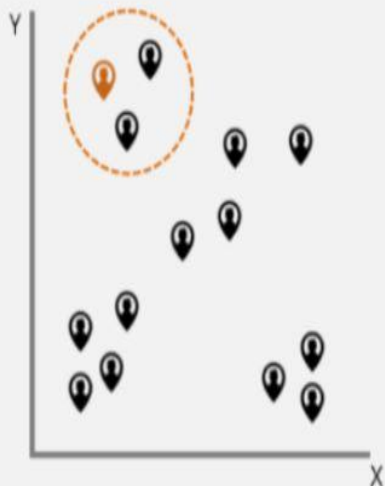
각 이미지를 for문을 돌려
카테고리 구분 모델, 패턴 구분 모델, 색상추출
함수에 넣어 각 값들을 데이터베이스에 삽입한다.

id	category	pattern	color_R	color_G	color_B
1	0	1	125.	126.5	134.3
2					
...					

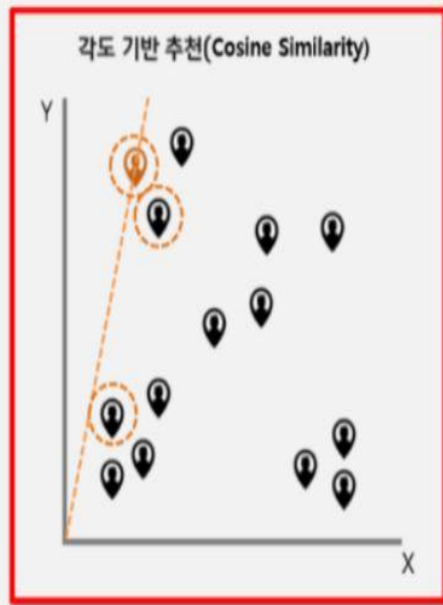
2-6. 코사인 유사도로 가장 가까운 색 이미지 출력

유사도 측정 방식

거리 기반 추천(Euclidean distance)



각도 기반 추천(Cosine Similarity)



```
#####  
##### 3. find  
#####  
  
# 1. input_1 = input_color_find  
# 1) 코사인 유사도  
dist={}  
for i in range(1300) :  
    data = list(rgb.iloc[i])  
    d = cos_sim(input_color, data)  
    dist[d] = i  
  
a = sorted(dist.items(), reverse = True)  
  
# 2. input_2 = input1_category_find  
category_dist=[]  
  
for _, i in a :  
    if category[i] == input_category_num :  
        category_dist.append(i)  
  
# 3. input_3 = input2_pattern_find  
final_index=[]  
  
for i in category_dist :  
    if pattern[i] == input_pattern_num :  
        final_index.append(i)  
  
final_index[:5]  
  
for i in final_index[:5] :  
    image = cv2.imread(f"C:\\~~~~~({i}).jpg")  
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
    plt.imshow(image)  
    plt.axis("off")  
    plt.show()
```



코사인 유사도로 색상순으로 데이터 베이스 정렬



그 중에서 카테고리가 일치하는 것 색상순대로



그 중에서 패턴이 일치하는 것 색상순대로 추출

TOP 6 추출



3. FLASK

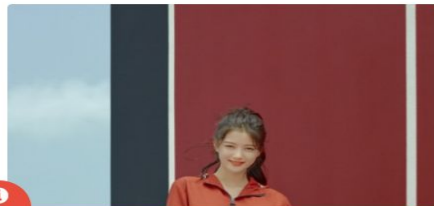
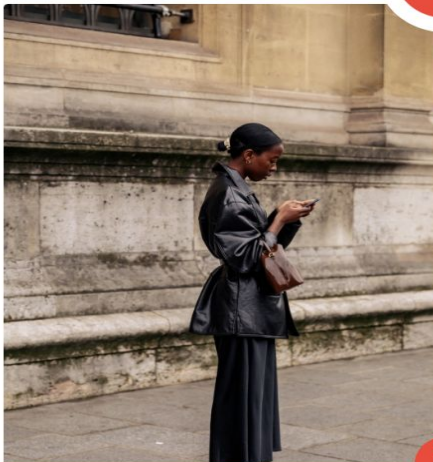
웹 구현하기

[작업 환경]

- 파이썬 : Flask
- html, css

Home 화면

LookUs의 제작자들이 선정한 각기 다른 16개의 테마를 주제로 선정한 룩을 경험해보세요. 룩의 이미지를 클릭하면 테마와 상세설명을 조회할 수 있습니다.



LookUs' Special Look Editions

LookUs의 제작자들이 선정한 각기 다른 16개의 테마를 주제로 선정한 룩을 경험해보세요. 룩의 이미지를 클릭하면 테마와 상세 설명을 조회할 수 있습니다.

USE AI PERSONAL COORDINATOR

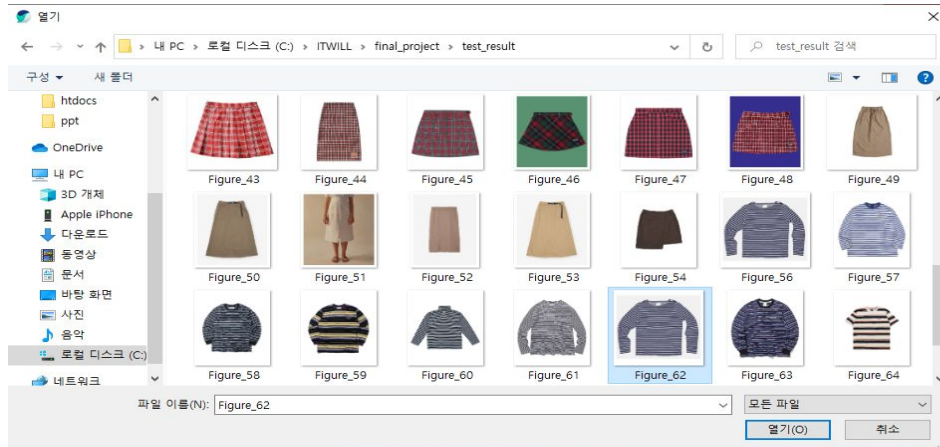
AI COORDINATOR는 당신이 찾고자 하는 스타일의 옷을 AI 알고리즘을 통하여 SEARCH 합니다. 옷의 이미지를 입력해주세요.

파일 선택 선택된 파일 없음

제출

Search 화면

AI COORDINATOR는 당신이 찾고자 하는 스타일의 옷을 AI 알고리즘을 통하여 SEARCH합니다. 옷의 이미지를 입력해주세요.



거친 질감, 짙은 배색의 탱아

흔히 말하는 '정정 패션'은 본디 광부들을 위해 내구성이 뛰어나게 제작된 데님 재질의 옷에서 비롯되었다. 청바지와 청자켓은 거친 질감에서부터 규율에 얽매이지 않길 원하는 탱아의 느낌을 자아낸다. 남자들이라면 한번쯤, 거친 자유로움을 맘껏 뽐내고 싶은 욕구가 있지 않을까. Lookus가 소개하는 두 번째 룩의 키워드는 '러프한 자유'이다.

- GOZNUK -

• Top
청자켓, 검정 티셔츠

• Bottom
데님 디스트로이드 진

• Shoe
반스 어센틱 체크 보드

USE AI PERSONAL COORDINATOR

AI COORDINATOR는 당신이 찾고자 하는 스타일의 옷을 AI 알고리즘을 통하여 SEARCH합니다. 옷의 이미지를 입력해주세요.

파일 선택 선택된 파일 없음

제출

Processing 화면



PROCESSING...

Find 화면

LookUs AI 가 추천하는 결과는 다음과 같습니다.

- 이미지 & 쇼핑몰 링크



Lookus AI 가 추천하는 결과는 다음과 같습니다.



Shopping List

• Item 1 [쇼핑하러 가기!](#)

• Item 2 [쇼핑하러 가기!](#)

• Item 3 [쇼핑하러 가기!](#)

• Item 4 [쇼핑하러 가기!](#)

• Item 5 [쇼핑하러 가기!](#)

• Item 6 [쇼핑하러 가기!](#)

Shopping 화면

쇼핑몰 링크를 따라가 쇼핑을 즐기세요!

SOUTH KOREA ▾ 고객센터 < 카톡에서 YOOX를 친구로 추가하세요! >

회원가입 로그인





여성 남성 키즈 / DESIGN+ART


YOOX

Q ♥ 🛒

신상품 디자인어 의류 슈즈 액세서리 8 BY YOOX 콜라보레이션 YOOXYGEN SALE 클리어런스

홈 / 남성 / 진 & 데님 / 진 / CITIZENS OF HUMANITY





CITIZENS OF HUMANITY
진

₩179,000 (KRW 215,050.61) **US\$ 148.00 (KRW 177,807.21)**

YOOX 가격

분류

사이즈

31

사이즈 가이드

라벨에 표시된 사이즈와 일치합니다.

마지막 상품 구매가능

쇼핑백에 담기

DREAM BOX에 담기

혼용품

100% 코튼

상세정보:

데님, 가죽 디테일, 로고, 슬리드 얼러, 미디엄 워싱, 바턴 포켓, 노멀 웨이스트, 프론트 플러저, 버튼 & 지퍼, 멀티 포켓, 통풍성 비설유 자재 함유

지수

밑단 너비 22 cm

배송 기간 & 비용

공유

상품 코드: 42759716JA

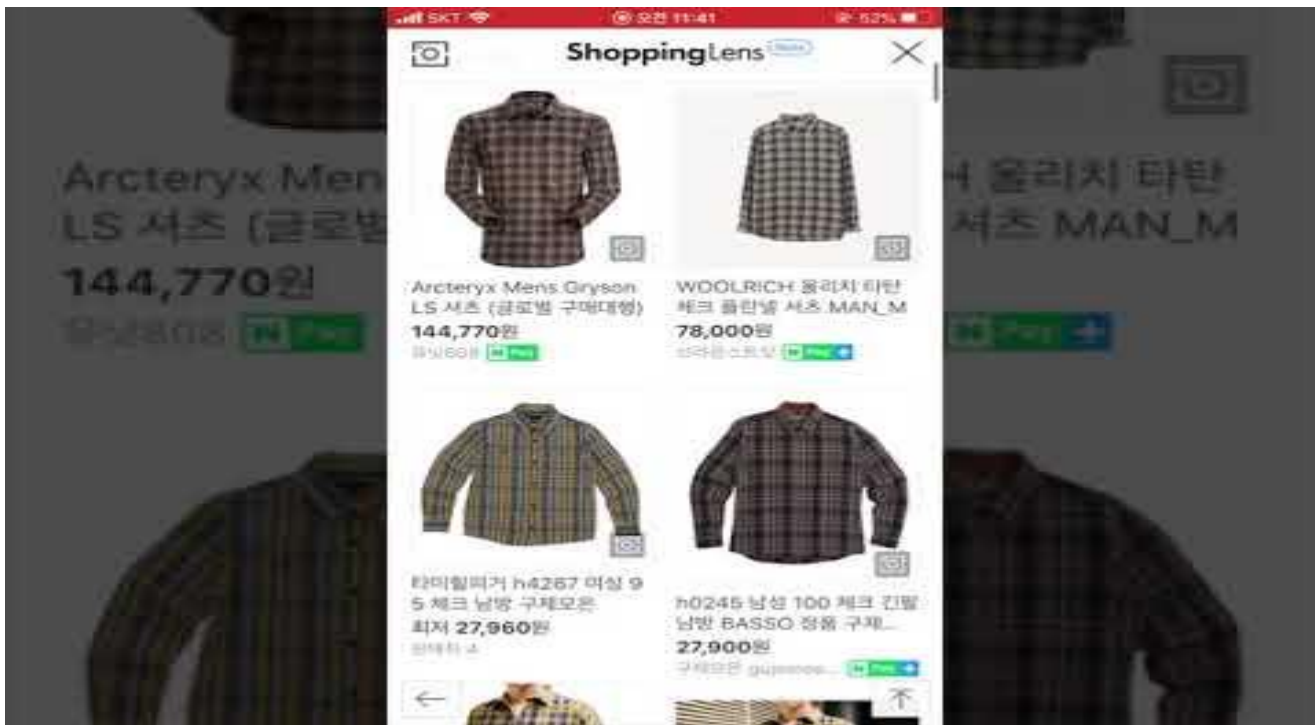
www.yoox.com의 육단을 기다리세요

고객님을 위한 선물!

4. 프로젝트 시연

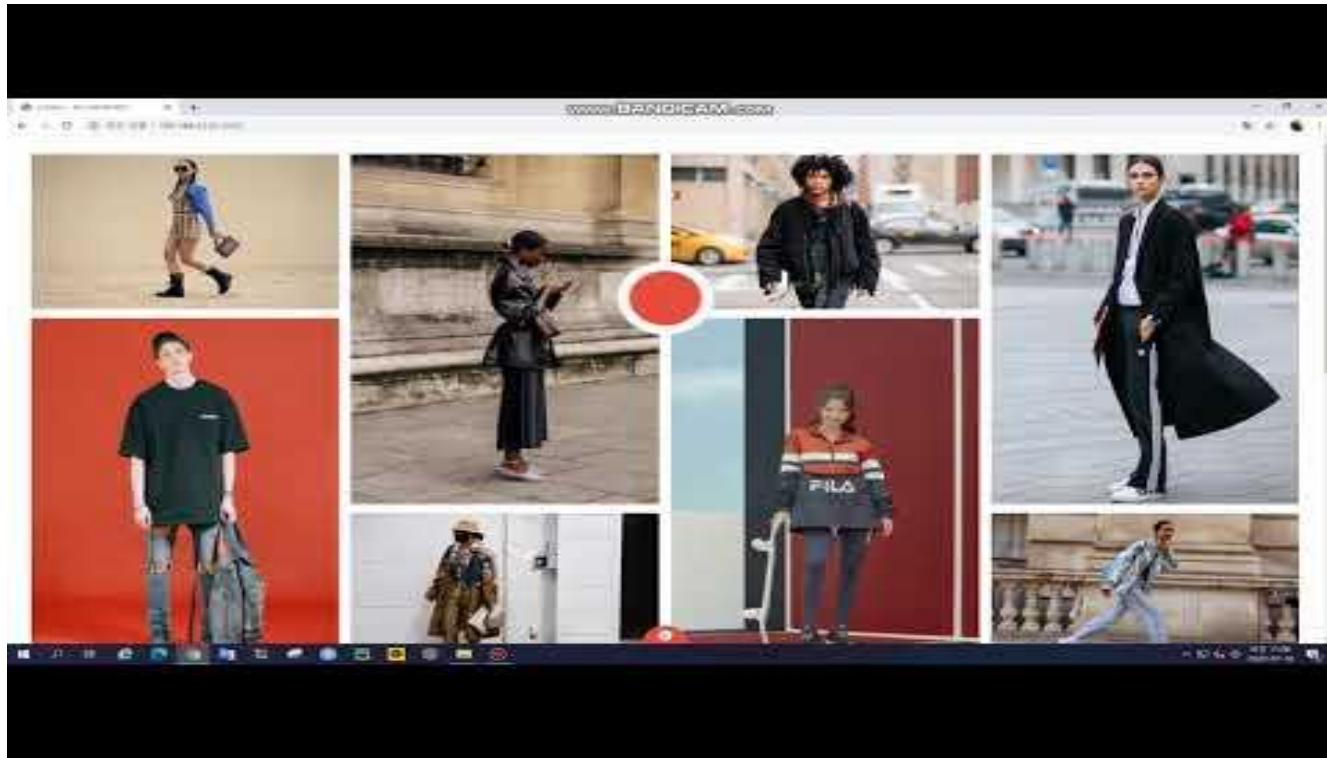
네이버의 쇼핑렌즈(아래 재생)

프로젝트를 진행하기전 주제선정에 많은 영감을 준 네이버의 쇼핑렌즈를 모바일앱에서 진행해본 영상입니다.



Look Us의 쇼핑렌즈(아래 재생)

LOOK US 만의 느낌으로 감각적이게 구현한
쇼핑렌즈 웹 시연 영상입니다.



5. 향후 계획

보완점

- 카테고리 와 패턴의 종류 다양화
- DB 확장
- 입고 있는 사진도 정확하게 분류할 수 있도록 이미지 부분 인식 기술 적용
- 일시적으로만 연결 가능한 Flask를 대체하기 위하여 아파치 서버와 연동

상업적 활용 가능성

개인 DB가 있는 쇼핑몰에 도입

사용자가 이미지 입력을 하면 쇼핑몰 DB에 있는 제품 추천