**<인공지능 프로젝트 최종보고서>**

**[코디를 부탁해]**

**B611128 유창조**

**B711217 황규빈**

**B811100 신주은**

**B811086 서민성**

1. **프로젝트 개요**

MZ세대, 최신 유행하는 패션이 빠르게 변하고 옷으로 자신의 개성을 표현하는 시대에 우리는 매일 아침 어떤 옷을 입어야 할지 고민한다. 또한 오늘의 포인트 아이템을 결정했다고 하더라도 이와 어울리는 최적의 코디는 무엇일지 고민하며 수많은 옷들을 대응해보게 된다. 이를 통해 들어가는 시간과 노력을 단축시킴으로써 원하는 스타일의 코디를 빠르게 추천해줄 수 있는 서비스를 제공할 수 있다면 어떨까 하는 생각에서 코디를 부탁해 프로젝트를 고안하게 되었다.

코디를 부탁해는 프로젝트 명에서부터 알 수 있듯이 사용자에게 최적의 코디를 추천해주는 서비스이다. 이 서비스는 사용자가 옷 상품 사진을 넣었을 때 우리가 구현해낸 각각의 알고리즘을 통하여 옷의 색깔을 파악하고 그 색을 토대로 사용자가 선호하는 스타일의 코디를 찾아내어 총 다섯 가지의 코디를 추천해준다. 핵심은 특정 색깔과 어울리는 특정 스타일의 코디를 도출해낸다는 것이다. 최종적으로 프로그램을 돌렸을 때 사용자가 만족할 만한 코디를 얻을 수 있도록 하는 것이 목표이다.

이 프로젝트를 통해 사용자들은 바쁜 현대 사회 속 데일리 코디를 고민하는 시간을 줄일 수 있으며 특정 아이템에 매치 시킬 옷이 없을 때 손쉽게 어울리는 아이템을 찾아내어 구매할 수 있다. 또한 여러가지 스타일의 코디를 구경할 수가 있기 때문에 본인이 시도해보지 못했던 색다른 스타일에도 관심을 가지며 도전해 볼 수 있게 될 것이다. 이를 통해 자신에게 맞는 스타일을 확립하여 더욱 패션과 가까워질 수 있다.

1. **구현 환경**

IDE : Visual Code, Pycharm

Library : numpy, scikit-learn, openCV, matplotlib, json, Image, glob, random

1. **사용 알고리즘**

**K-means**

* 알고리즘 설명

K-means 알고리즘은 unsupervised learning의 한 예로 KNN과 달리 input에 대한 output이 주어지지 않은 알고리즘이다. K-means 알고리즘은 K개의 군집으로 묶어, 레이블에 대한 값을 비교하는 것이 아닌 정보를 얻어내는 것에 활용할 수 있다.

K-means 알고리즘은 3가지 순서로 진행된다. 첫 번째는 centroid 지정하기, 두 번째로는 데이터들을 centroid와 가까운 쪽에 맞게 할당하기, 마지막으로 centroid를 새로 지정하여 그룹화 되어있는 데이터들의 중심 centroid로 업데이트해준다. 이 과정을 반복하여 centroid의 변화가 없을 시 알고리즘을 모두 수행하게 된다.

첫 번째로 centroid는 임의로 설정하여 잡는다. K값에 따라서 k개의 centroid를 임의로 설정하는데, 이 때 feature의 크기에 맞게 centroid를 설정한다. 예를 들어, x y좌표에 따른 K-means를 수행하고자 한다면 centroid는 임의의 x, y 좌표를 지닌 값이 되는 것이다.

두 번째로 centroid를 설정하였다면, 이 centroid에 가까운 데이터들을 그룹화하는 과정이다. 이는 거리측정 방식에 따라 다르며 대표적으로는 Euclidean 거리측정법을 사용한다. 이는 좌표 사이의 차이의 제곱들을 합하고 이를 루트를 씌워 좌표의 거리들을 측정한다. 예를 들어, (3,1) (4,2)의 두 좌표의 거리를 구한다면 로 input으로 넣은 데이터들과 새로운 데이터 사이의 거리를 측정하여 가장 가까운 거리에 존재하는 데이터들이 무엇인지 확인하고자 하였다. 각 데이터 마다 centroid와의 거리를 측정한 후, 가장 가까운 k개의 centroid에 대한 인덱스를 기억하여 이를 저장한다. 이를 통해서 우리는 centroid 마다 가까운 데이터들의 그룹을 형성할 수 있게 된다.

마지막으로 centroid를 새로 지정해준다. 이전 단계에서 거리를 측정한 후 centroid에 대한 가까운 그룹들을 형성할 수 있었다면, 이 그룹 안에서 평균 지점을 계산하여 이를 centroid로 활용하는 것이다. 따라서 우리가 알고리즘의 제목에서 볼 수 있듯이 K개의 군집에 대한 평균 centroid를 측정함으로써 이를 k개의 군집의 대표 centroid를 구할 수 있게 된다.

위 과정들을 진행하고 난 후에는 다시 첫 번째 과정부터 수행을 반복한다. Centroid의 값이 변하지 않을 때까지 새로운 군집을 계속해서 형성해본 뒤 이에 대한 centroid의 값이 변하지 않게 되었을 때 보다 더 정확한 군집의 형태와 centroid 값을 얻을 수 있을 것이다.

* 장점 및 단점

K-means의 장점은 간단하다는 점이다. 이러한 간단한 알고리즘을 통해서 대규모 정보에도 적용시키고 군집을 얻어내는 것이 가능하다. 그렇기 때문에 실행시간도 짧은 편이며 기초적인 알고리즘이다. 또한 앞서 소개했던 것과 같이 unsupervised learning으로 자료에 대한 사전 정보 없이 결과를 얻을 수 있다. 사전 정보 없이 알고리즘을 수행하는 것으로 특정 변수에 대한 역할 정의 또한 필요하지 않아 관찰 데이터 간의 거리만을 분석에 필요한 input으로 사용하면 된다.

이에 반해 K-means는 여러 단점들도 존재하며, 첫 번째로는 k값에 선정이 필요하다는 점이다. 이는 뒤이어 소개할 KNN과 같은 맥락으로 k값에 따라서 원 데이터 구조에 적합하지 않는 군집 수를 얻게 된다면 좋지 않은 결과를 얻을 수 있다. 또한 K-means는 unsupervised learning이기 때문에 target에 대한 label을 직접 선정해야 한다. 이는 비교할 수 있는 사전 정보에 대한 결과가 없기 때문에 결과 분석에 어려움을 가진다는 점이 단점이다. 마지막으로는 가중치와 거리계산에 따라서 군집 형성에 따른 결과가 달라진다는 점이다. Input이 거리 계산으로만 이루어지기 때문에 거리 계산이 필요하나 이는 중심이 되는 centroid에 따라서 거리에 대한 결과 또한 달라진다. 그렇기 때문에 처음 적정 거리에 대한 정의가 필요하며 이에 따라 가중치 또한 정의가 필요하다.

* 선정 이유

K-means를 사용하게 되면 색상에 대한 군집을 분류할 수 있다. 사진에 대한 색상 군집을 분류하였을 때 여러 군집들 중 가장 주된 군집을 확인한 후 대표 색을 정할 수 있을 것이라고 판단하였다. 이에 따라 이미지의 RGB 데이터를 클러스터링하게 된다면 여러 군집을 분류할 수 있을 것이라 생각하여 K-means를 이용하였다.

* 활용 방법

openCV를 활용한 이미지의 RGB 데이터를 추출한 후, 이를 여러 색상에 대한 RGB 군집을 K-means를 이용하여 분류한다. K 값에 따라서 다양한 색의 군집을 구할 수 있을 것이므로 여러 K값에 따라 이를 적용시켜보고 이미지의 대표 색을 정하기 위해 어떠한 k값을 적용시켜야 할지 알고리즘을 구현하는 과정에서 적용시켜보고자 한다.

**KNN**

* 알고리즘 설명

KNN 알고리즘은 Supervised learning의 한 예로 input과 input에 대한 결과인 output의 쌍을 입력 받고 모델을 학습시킨 후 새로운 데이터를 입력 받았을 때의 그 데이터를 예측하는 알고리즘이다. 또한 이 knn 알고리즘은 classification 또는 regression을 통해 추론할 수 있는 알고리즘이다.

 KNN 알고리즘은 3가지 순서로 구현이 가능하다. 첫 번째는 거리측정, 두 번째는 k개의 가까운 이웃들을 구하기, 마지막은 k개의 이웃들을 통해 가장 많이 발견된 데이터를 통해 추론하는 과정이다.

 먼저 첫 번째 거리를 측정하는 방법을 살펴보자. 거리를 측정하는 방법에도 다양한 방법이 있지만 대표적인 방법으로 Euclidean의 거리측정법을 사용하였다. 이에 대한 측정법은 앞서 K-means 알고리즘을 수행한 방식과 동일하게 데이터 들과의 거리를 측정한 후 이를 기억해준다.

 다음은 두 번째 순서로 k개의 가까운 이웃들이 무엇일지 확인하는 것이다. 앞서 측정한 거리들을 바탕으로 가장 가까운 데이터들이 무엇일지 확인한 후 가까운 순서대로 k개의 이웃들을 찾아준다. 이 k 값에 따라서 새로운 데이터의 결과를 정확하게 추론이 가능한지에 여부를 찾을 수 있는데 먼저 k는 짝수보다는 홀수로 설정하는 것이 좋다. 이는 짝수에 경우 가까운 데이터들의 결과의 비율이 동률이 되어 새로운 데이터의 결과를 예측하기 난감한 상황이 발생할 수 있기 때문에 홀수로 설정하는 것이 좋다. 또한 k값이 너무 작거나 클 때에도 문제가 발생하는데 너무 작다면 이상치 하나에도 큰 영향을 받아 결과가 좌지우지 될 것이며 너무 클 때에는 미세하게 차이 나는 결과를 확인하지 못하고 잘못 분류하게 될 것이다. 따라서 k값을 적절하게 설정하고 정확한 분류를 위해 도출하는 과정이 필요하다.

 마지막으로 세 번째는 k개의 가까운 이웃 데이터들을 바탕으로 새로운 데이터의 결과를 추론하는 과정이다. 이는 앞서 가까운 k개의 이웃 데이터들을 확인하여 이들의 결과들 중 가장 많이 발견된 label 즉 target class를 확인하고 이를 새로운 데이터의 결과로 추론하는 것이다. 하지만 이러한 majority vote를 실행하는 과정에 문제점이 있을 수 있는데 이는 가장 가까운 거리와 결과가 같을 수 있음에도 가장 많이 발견된 label에 따라서 결과 값이 잘못 추론될 수 있다는 점이다. 따라서 이러한 majority vote를 하는 과정에 가중치(weight)를 적용하여 가까운 거리의 점들은 더 큰 가중치를 적용시켜 보다 더 정확한 결과를 찾을 수 있도록 weighted majority vote를 이용하여 분류를 진행하는 것 또한 좋은 방법이다.

majority vote:

weighted majority vote:

* 장점 및 단점

 KNN의 장점은 알고리즘 자체가 구현이 간단하다는 것이다. 앞서 설명한 세 가지 순서를 통해 결과를 추론할 수 있기 때문에 구현이 간단하다는 것뿐만 아니라 다른 알고리즘들의 기반이 되기도 한다. 또한 학습시키는 과정 또한 굉장히 빠르다는 것이다. Supervised learning으로 input에 대한 결과 값이 함께 적용되어 있기 때문에 그저 새로운 데이터를 추가하는 것으로 학습이 가능하다. 그리고 분류된 값을 모두 검사하여 높은 정확도를 지닐 뿐만 아니라 k개의 데이터를 선택함에 따라 오류데이터는 제외하고 기존 데이터를 활용하여 가정을 하지 않는 점에서 장점을 지닌다.

 하지만 여러 단점들도 존재한다. 데이터가 많아지게 된다면 모든 데이터들을 비교해야 하기 때문에 처리시간이 느려지며 메모리를 많이 사용하게 된다는 점에서 단점이 존재한다. 또한 다양한 거리측정법이 존재하는 만큼 거리측정법마다 다른 성능을 지니게 된다. 그리고 이러한 거리를 측정하게 되면서 데이터들의 범위의 차가 크게 된다면 거리측정법에 미치는 영향이 커질 수 있기 때문에 데이터들의 범위를 조정하는 표준화, 정규화 과정이 필요할 수 있다. 또한 k를 선택함에 있어서도 어려운 점이 존재한다는 것을 뽑을 수 있다.

* 선정 이유

앞서 K-means를 통해 RGB에 대한 색깔 추출을 하게 되었을 때 3크기의 리스트 형식을 반환할 수 있다. 이 리스트를 이용하여 무슨 색일지 라벨링을 할 것이기 때문에 20개의 색상에 대한 여러 RGB 데이터가 있는 json을 이용하여 추측하고자 하였다. 따라서 3 크기의 값들을 다른 좌표들과 비교를 통해 색깔 데이터를 얻을 수 있을 것이라고 생각하게 되었다. 이에 따라 input의 수도 작고 비교하는 RGB 데이터의 수도 적었기 때문에 KNN을 사용하기에 적합하였으며 정확도가 어느 정도 측정 될 수 있는지 확인하고자 하였다.

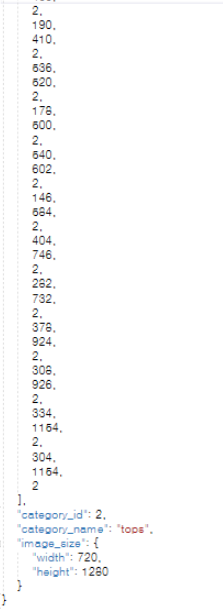
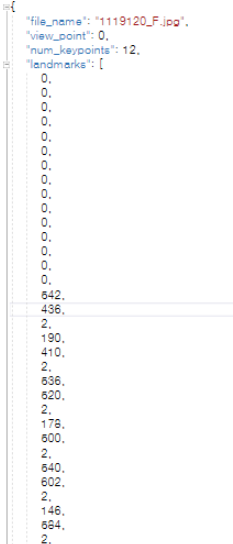
* 활용 방법

KNN의 활용을 위해 앞서 K-means을 먼저 사용하여 상품사진에 대한 RGB 데이터를 확보한다. 3 크기의 리스트를 얻게 되면, 이를 준비해둔 color.json 파일에 있는 20개의 target데이터로 되어 있는 약 500개 정도의 color RGB 값과 비교한다. 이 때 가중치를 두었을 때와 k값에 따라서 어떻게 정확도가 달라질 수 있는지, 또한 정확도가 제대로 나오지 않았다면 어떻게 개선할 수 있을지 고민하고자 한다.

1. **데이터**

**상품 사진 data set**

* 약 4,000개 사진, json 형식의 상품 사진 정보



**코디 사진 data set**

* 약 50,000개의 사진(8개의 스타일로 이루어진 폴더)
* json 형식의 코디 사진 정보



(출처 : https://aihub.or.kr/aidata/30755, https://aihub.or.kr/aidata/7988)

**Input**

* 상품 사진
* 원하는 코디 스타일 입력
* RGB 값 : 0~255사이의 값으로 이루어진 크기 3의 리스트
* Category\_id : 1(아우터), 2(상의), 3(하의)

상품 사진을 통해 RGB값 추출 후 json파일의 라벨링된 RGB 값, category\_id 사용

**Output**

* 5개의 랜덤한 코디 사진 파일

Ex)

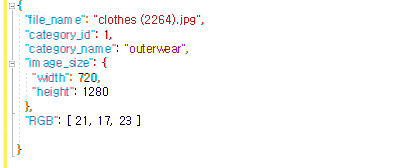
벽, 사람, 의류, 실내이(가) 표시된 사진

자동 생성된 설명

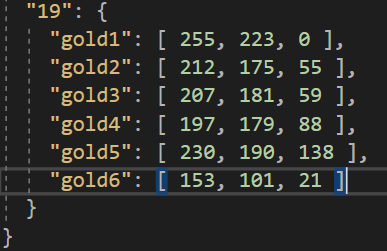
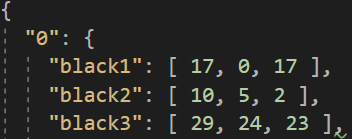
텍스트이(가) 표시된 사진

자동 생성된 설명

1. **추가적인 가공 데이터**



기존의 데이터에서 Input에 사용되지 않는 데이터를 삭제하고 K-means 알고리즘을 통해 추출한 RGB값을 추가하였다.

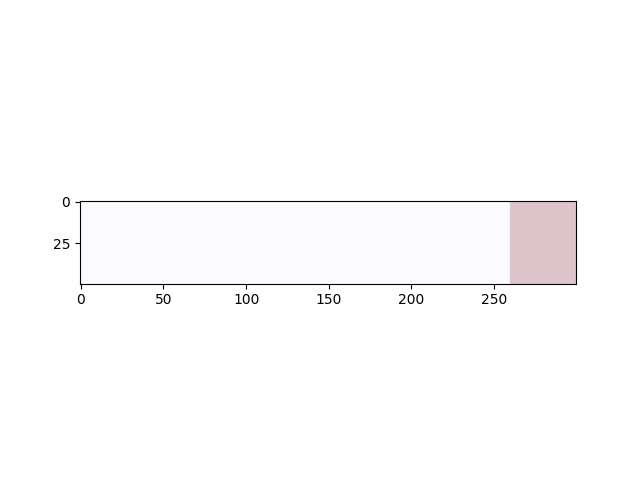


추출한 RGB값을 통해 색상을 판별할 수 있도록 추가적으로 21가지의 색상의 RGB값을 포함한 JSON 파일을 만들었다.

1. **프로그램 진행 과정**

우리 프로그램은 2가지 단계로 이루어져 있다. 첫 번째 단계는 상품사진에서 색을 추출하는 단계이다. 두 번째 단계는 첫 번째 단계를 통해 추출된 색과 미리 사용자가 입력한 스타일에 맞는 코디를 추천해주는 단계이다..

첫 번째 단계에서는 input으로 상품 사진을 사용한다. 입력된 상품 사진에서 K-means를 활용해 사진의 색 군집을 추출해낸다.



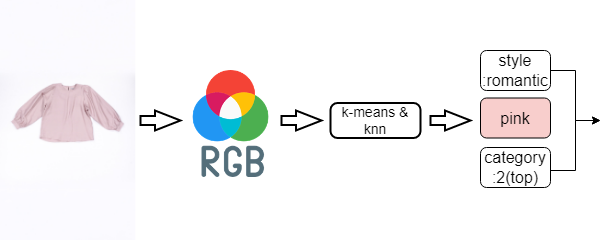
상품 사진 배경이 흰 색이라 흰 색이 가장 많은 비율을 차지하고 있다. 따라서 우리는 흰 색을 제외하고 가장 많은 비율을 차지하고 있는 색을 주요 색깔로 지정하기로 결정했다.

추출한 색의 RGB값을 상품 사진 json파일에 추가하고, 미리 선정한 20가지 색으로 분류하기 위해 kNN알고리즘을 사용하였다..

20가지 색에 어떤 RGB값들이 있는지 데이터를 수집하여 json파일로 만들었고 상품 사진 RGB값과 20가지 색 각각의 RGB값들 사이의 거리를 계산하고, k=3으로 설정하여 가장 유사한 색이 무엇인지 결정하였다.

미리 사용자에게 입력 받은 스타일, 앞선 알고리즘으로 도출된 색상, 기존 상품사진 json파일에 있던 상의/하의/원피스 정보를 이용하여 가장 어울리는 코디 사진을 랜덤으로 5개 제공한다.

1. **요약 Flow Chart**

****

1. **알고리즘 소스코드와 설명**

**Kmeans.py**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 화면, 검은색, 은색이(가) 표시된 사진

자동 생성된 설명

**Knn.py**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Main.py**

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

앞서 얻은 정보들을 활용하여 category\_id, 색깔과 같은 코디 데이터 찾아서 fashion\_data에 저장

1. **결과 및 분석 (최종 output 코디 사진 도출 및 정확도 분석)**

프로그램을 작동시키면 되면 얻을 수 있는 output은 다음과 같다. 약 ~개의 데이터 중 원하는 상품사진을 골라 경로를 입력한 후 실행시킨다. 고른 사진은 다음과 같이 핑크색의 맨투맨(상의)상품 사진 데이터 이다.



상품 사진을 넣고 K값에 따라서 어떠한 결과를 얻을 수 있는지 다음과 같이 확인할 수 있었다. 흰색 부분의 배경 부분이 제외된 나머지 군집의 RGB데이터와 함께 그의 비율을 10배한 것을 비율로써 갖도록 출력하였다. 이러한 비율은 k값이 커짐에 따라 다양한 군집들이 얼마만큼의 비율을 차지하고 있는지 확인하고 이에 대해 각 RGB 값이 KNN을 통해 얻은 label에 가중치를 두기 위함이다.

첫번째로는 K-means가 2일 때와 KNN의 값이 5일 때 결과를 확인해보았다. 원하는 스타일인 ‘로맨틱’에 있는 폴더 내에서 [253, 179, 184]의 RGB값과 KNN을 통해 수행하여 얻은 핑크색 옷의 사진 5개를 확인할 수 있었다. 이 때 출력된 결과와 첫번째 사진의 예시는 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명 벽, 실내, 바닥, 사람이(가) 표시된 사진

자동 생성된 설명

K-means와 KNN을 수행함으로써 RGB 값의 분포를 얻을 수 있었으며 이 때 K-means에서 군집의 개수 2로 설정하여 프로그램을 돌렸을 때 상품사진에서 가장 눈에 띄었던 핑크색의 색을 대표색으로 정할 수 있었다. 또한 원하는 상의 데이터가 핑크색인 코디 데이터 사진도 제대로 잘 수행되어 추천받을 수 있음을 확인하였다.

이때 K-means의 K값이 증가하게 되면 다양한 색의 군집을 얻을 수 있을 것이라고 생각하였으며 K값이 4일 때의 경우는 어떠한 결과를 얻을 수 있을지 수행해본 결과 다음과 같은 결과를 얻을 수 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명 벽, 사람, 의류, 실내이(가) 표시된 사진

자동 생성된 설명

기존에 핑크색으로 결과를 얻었던 K=2일 때의 결과와 달리 K=4일 때의 군집을 형성하였을 때는 다른 결과임을 확인할 수 있었다. 이는 출력된 결과를 확인해본 결과 가장 많이 지배적으로 나왔을 것으로 예상한 흰색 배경의 RGB를 제외하고 K-means를 수행했음에도 ‘화이트’ 로 라벨링 되어지는 RGB의 값이 지배적으로 나왔음을 확인할 수 있었다. 각 군집의 비율을 계산하였을 때도 핑크색의 비율을 다 더한 것보다 큰 것을 확인할 수 있었다.

이를 통해 알 수 있었던 사실은 첫번째로 K-means를 수행하였을 때 K의 값이 작을수록 더 의도한 대표색을 통한 코디 데이터를 출력할 수 있었다는 것이다. 다음 사진과 같이 K의 값이 증가될수록 RGB값이 다양해질 수 있었고, 사진 안에서 의도하지 않은 RGB값을 얻어내게 되면서 너무 많이 군집이 형성된다는 점을 확인할 수 있었다.

plotting 된 사진 넣기

두번째로 color.json을 이용하여 RGB 값과 KNN을 수행하였을 때의 정확도 개선이 필요하다고 판단되었다. 20개의 색깔 라벨링을 구분하는 RGB 경계선이 상당히 모호했기 때문에 소수점 단위의 RGB 값의 데이터가 있을 만큼 다양한 양의 RGB 데이터가 필요하였다. 이는 600개로 구성한 RGB 값에 따른 20개의 색깔 라벨링이 기준 색깔에 다양한 RGB값을 가짐으로써 다른 기준의 색깔에 대한 RGB값과 비슷해 진다는 점을 확인할 수 있었다. 따라서 RGB값을 기존보다 더 단순하게 대표 기준 색상에 맞는 RGB 값의 분포가 밀집되도록 color.json을 수정할 수 있었다.

위 내용들을 확인한 후 상품 사진에 대한 색깔 라벨링과 KNN을 수행한 이후 군집의 비율을 통해서 찾아낸 색깔 라벨링 간의 비교를 하였다. K-means의 K가 각각 2와 5의 값으로 구한 RGB값과 KNN의 K가 각각 3, 5, 9일 때의 알고리즘을 수행한 결과의 색깔 라벨링이 상품 사진 색깔 라벨링과 같은 지 비교하고 정확도를 계산해보았다. 이 때 약 3700개의 상품사진을 넣어 계산하였다.

1. **진행 과정 및 역할 분담**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | WEEK 1 | WEEK 2 | WEEK 3 | WEEk 4 |
| 유창조 | 첫 발표 | 추출한 RGB값  json 파일에 저장 백업 | | 데이터 가공  마무리 작업 진행 |
| 황규빈 |  | K-means 알고리즘 구현  PPT2 제작 | KNN알고리즘  구현 | 알고리즘 class화  Main 파일 추가 |
| 신주은 |  | RGB json 데이터 가공 및 정리 | 중간 발표 | Color 값 조정  정확도 개선 작업 |
| 서민성 |  | 20개의 색깔 RGB 데이터 구성 |  | 최종 발표 및  보고서 취합 정리 |
| TEAM | 기획  PPT1 제작 | K-means를 통해 RGB값 추출 | KNN을 통한 색깔 라벨링  Output 코디 사진 결과 분석 | 최종 보고서 준비  PPT3 제작 |

1. **예상했던 난관에 대한 실제 상황 및 예상 외의 난관**

인공지능 첫 발표 때 우리가 예상했던 난관으로는 데이터 가공 문제와 영어가 아닌 한글 데이터가 호환이 잘 될지에 대한 불확실성 등이 있었다. 두 가지 모두 기획 단계에서의 예상이었기 때문에 매우 추상적인 예측이었다. 데이터 가공 문제의 경우 팀원들과 지속적인 회의를 하며 실현 가능한 아이디어를 내었고 구체적으로 단계를 밟아감에 따라 자연스레 해결되었다. 한글 데이터는 아무 문제 없이 딱 맞게 적용되었다.

실제로 우리가 난관이라 여겨졌던 부분은 첫 번째, 흰색 배경 제외의 문제가 있었다. 패션 상품 사진은 흰색 배경에 상품이 들어가있는데 이 때문에 RGB 흰색 값이 가장 많을 수밖에 없었다. 우리는 대표 색을 추출해야 했는데 사실 꽤나 간단하게 가장 많이 나오는 색상을 제외함으로써 해결할 수 있었다. 이 때 상품의 색이 화이트라면 문제가 생기지 않을까에 대해 의문을 제기하기도 했는데 우리가 보는 색깔과 RGB의 실제 값은 천차만별이기 때문에 아무리 순백의 상품이라도 배경 값(255,255,255)과 같을 수는 없었다.

두 번째 난관으로는 색깔을 추출하고서 그 색의 라벨링을 하는 과정이 있었다. 우리의 첫 접근 방법은 RGB 색상의 범위를 정하여 우리가 원하는 20가지 색상으로 경계를 나누는 것이었다. 하지만 실제로 이를 나누는 과정에서 세가지 색의 조합으로 만들어지는 색상에 범위를 나눈다는 것은 불가능하다고 판단했다. 그렇다면 라벨링을 어떻게 하는 것이 좋을까 고민을 하던 중 나온 아이디어가 직접 색깔 json 파일을 만들어 KNN 알고리즘을 적용시키기였다. (자세한 내용은 목차 5 추가적인 가공 데이터에 나와있음.) 이를 통해 색깔 라벨링을 성공적으로 마칠 수 있었다.

마지막 가장 큰 난관으로 output 데이터 도출 과정에서의 머신러닝 적용시키기가 있었다. 비록 색깔 추출과 라벨링 과정에서 Kmeans와 KNN 알고리즘을 사용하기는 하였으나 최종 코디 도출 시 미리 라벨링 되어있는 스타일을 통해 결과를 내다보니 전체적으로 봤을 때 이 프로그램이 과연 인공지능이 맞는가에 대한 어찌 보면 가장 원시적인 의문이 등장한 것이다. 이는 중간발표 때 교수님의 가장 큰 지적사항이기도 했다. 우리는 이를 해결하기 위하여… (해결 못함… 더 보기…)

1. **프로젝트를 마치며 소감**

수업을 듣기만 하는 것이 아닌 실제 프로젝트를 진행해보며 깨달아가는 과정에서 수많은 것을 배우고 직접 느낄 수 있었다. 특히 기존 과제를 통해 KNN 알고리즘을 접한 적이 있었는데 과제에서는 미리 주어진 데이터셋을 이용하였다면 이번 프로젝트는 직접 상품 데이터를 찾아보며 주체적으로 진행해볼 수 있었고 결과를 도출해내는 과정에서 KNN을 사용하게 되었는데 실제 사례에서 적용될 수 있다는 점이 신기했다. 또한 이번 프로젝트를 통해 KNN과 K-means를 동시에 사용해보는 기회를 가졌는데 KNN은 지도형 학습(Supervised learning)으로 이미 입력 및 출력값이 있는 데이터를 사용하는 반면 K-means는 비지도 학습으로 라벨링 되어있는 데이터가 존재하지 않는다. 군집을 클러스터링하는 과정에서 이러한 두 알고리즘의 차이점을 더욱 명확히 느낄 수 있었다.

부담이 큰 팀 프로젝트였지만 팀원들끼리 브레인스토밍을 하고 회의를 하고 또 각자 맞춰가는 과정에서 혼자라면 절대 해내지 못했을 거대한 프로젝트를 기획하고 진행해볼 수 있었다. 또한 막연하게만 느껴지던 인공지능이라는 분야의 개념을 알아갈 수 있었고 실생활에 활용 가능한 알고리즘에 대해 깊게 이해함으로써 인공지능에 더 관심을 가지는 계기가 되었다.