차원축소팀 보고서

주제: MNIST 데이터셋 최적의 차원 축소 및 클러스터링 유도 후 오분류 데이터 분석
소주제: CNN 특징 추출 전후 차원 축소 결과 비교
소주제: 차원 축소 기법에 따른 최적의 클러스터링 유도
소주제: 클러스터 간 데이터 유사도 분석 및 클러스터 내 오분류 데이터 유사도 분석

목차

1. MNIST 데이터셋 특징
MNIST는 숫자 0~9의 손글씨 이미지 데이터셋
각 이미지는 28픽셀 x 28픽셀의 흑백 이미지
픽셀값은 0(검정)~255(흰색). 흰색이 숫자를 나타냄
라벨은 0~9까지 총 10개

2. 생각해볼 수 있는 숫자 간 유사성
MNIST는 손글씨이기에 날려쓰면 다른 숫자끼리 서로 유사하게 보일 수 있음.
그룹 A: (3,5,8)
그룹 B: (4,9)
그룹 C: (0,9)
그룹 내의 숫자들끼리는 날려쓰면 서로 유사해 보일 수 있음을 예상할 수 있음.

3. 특징 추출 기법
CNN(Convolution Neural Network)을 특징 추출 기법으로 사용.
CNN은 이미지의 특징을 뽑아내어 분류하거나 분석하는 딥러닝 모델로 MNIST데이터셋을 약 99%의 확률로 구분 가능.
작동원리를 간단히 설명하면 작은 필터(커널)로 이미지를 스캔하며 중요한 패턴(선, 모양)을 찾고, 데이터를 점점 간단하게 줄이며(풀링), 최종적으로 원하는 결과(예: 고양이냐 개냐)를 예측.
고차원에서 CNN으로 추출한 특징은 더 낮은 차원에 임베딩하여 벡터형태로 일차원배열에 저장함.
이 프로젝트에서는 256차원에 임베딩하였음.

<CNN 예시>

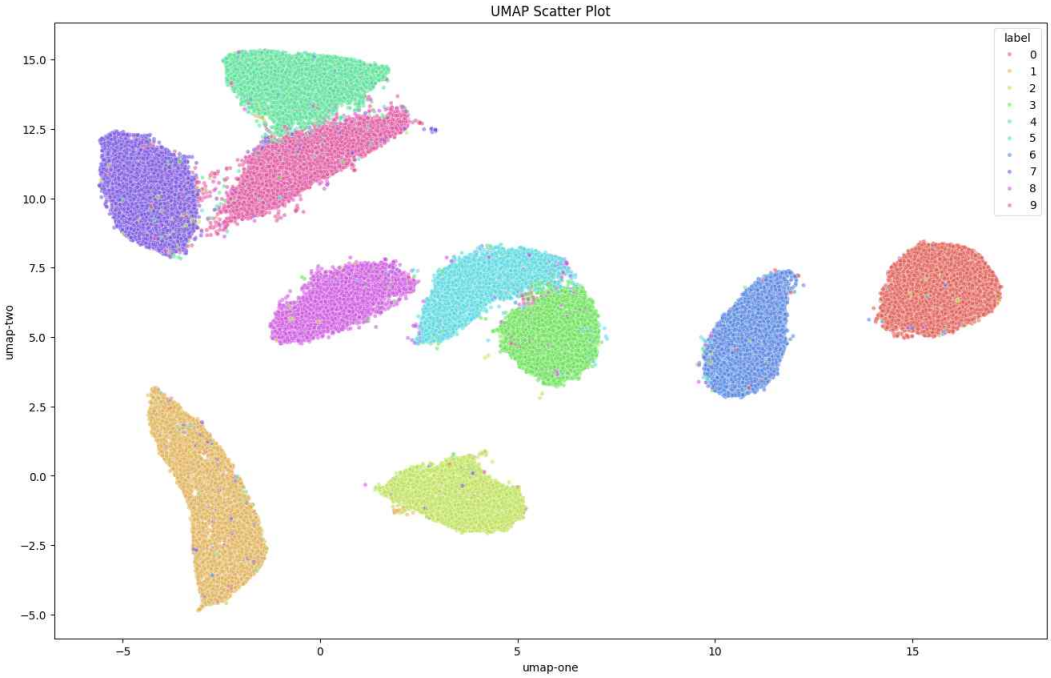
4. 차원축소 설명 및 사용한 기법
차원 축소는 데이터의 고차원 특성을 저차원으로 변환하는 과정으로 데이터의 중요한 정보를 최대한 유지하면서 분석, 시각화, 모델 학습에 필요한 계산량을 줄이기 위해 사용.

프로젝트에서 사용한 기법: UMAP, tSNE.

UMAP: 지역적 구조보단 전역적 구조를 잘 보존하며 O(NlogN)의 시간복잡도를 가짐.
T-SNE: 전역적 구조보단 지역적 구조를 잘 보존하며 O(N^2)의 시간복잡도를 가짐.

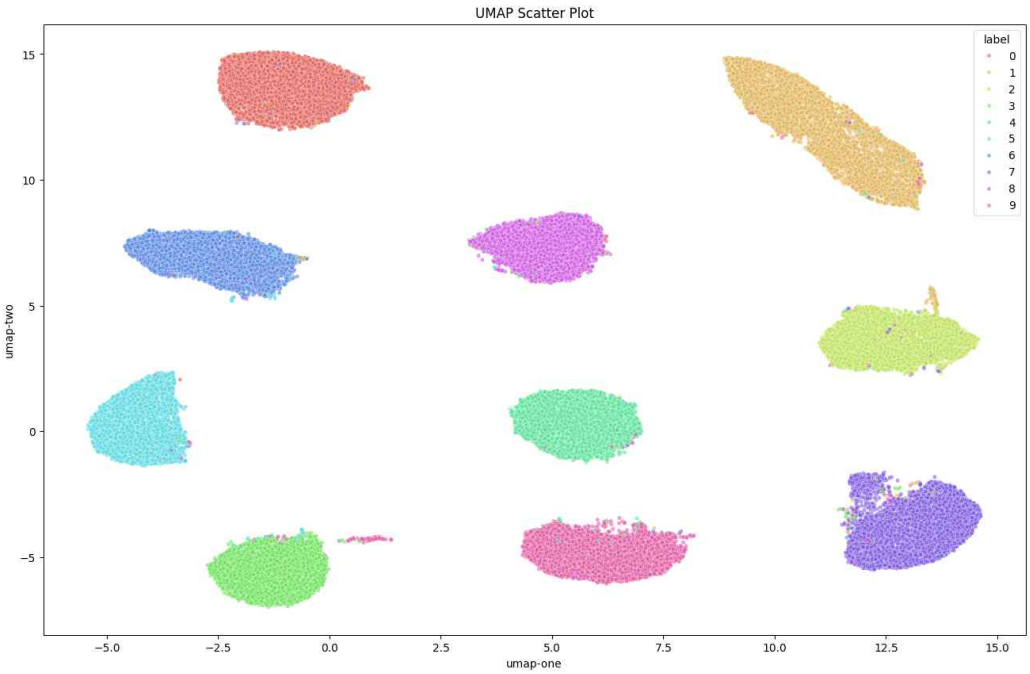
5. UMAP

5-1. 특징 추출 전 UMAP



UMAP의 특성 때문에 거리를 통한 전역적 구조는 잘 보존했으나 서로 붙어 있는 군집이 있음. 이로인해 흑백이미지일 경우 군집을 어떻게 나눠야할지 불분명함.
특징 추출 및 클러스터링을 통해 더 명확한 분리 필요.

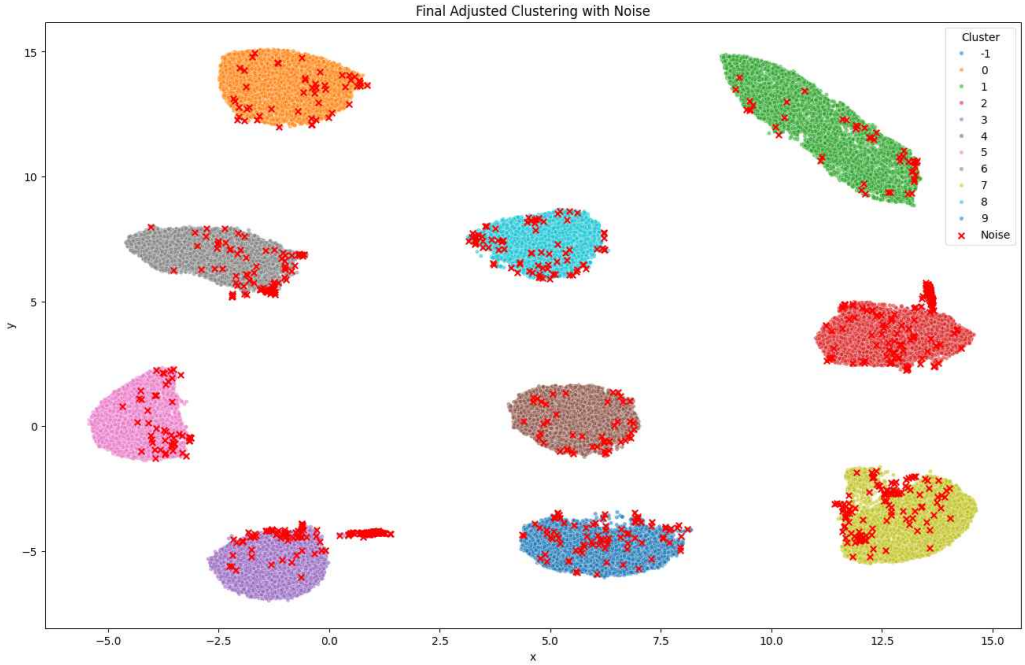
5-2. 특징 추출 후 UMAP



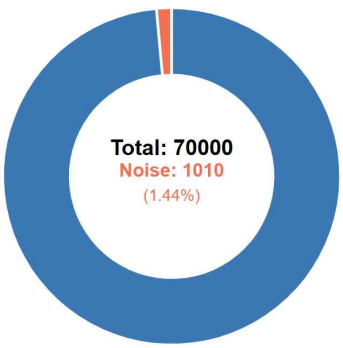
특징 추출 전보다 전역적 구조와 군집간의 구분이 확실해졌음. 하지만 군집내에 소수의 다른

라벨의 데이터가 위치하여 100% 확실하게 분류하지는 못함.
데이터들이 전역적 구조와 지역적 구조의 보존성이 높아 K-MEANS 알고리즘을 적용하여 클러스터링하는 것이 좋음.

5-3. 특징 추출 후 UMAP결과를 K-MEANS로 클러스터링



특징 추출 후 UMAP결과를 K-MEANS로 클러스터링한 후 잘못 분류된 데이터를 빨간색x로 표시. 클러스터링 전에도 이미 군집끼리 잘 분류되었기에 전과 후의 차이가 크지 않음.



총 7만개의 데이터 중에 1010개의 데이터가 잘못 분류됨.
약 0.0144%의 확률로 잘못 분류함.

5-4. 궁금한 점 및 해결방안

궁금한점:

“왜 잘못 분류되는걸까? 노이즈와 노이즈가 속한 군집의 대표(실제) 이미지 간에 뭔가 다른 부분이 있지 않을까? cnn으로 추출한 특징을 서로 비교할 수 있는 지표가 있지 않을까?”라는 질문을 던질 수 있음.

해결방안:

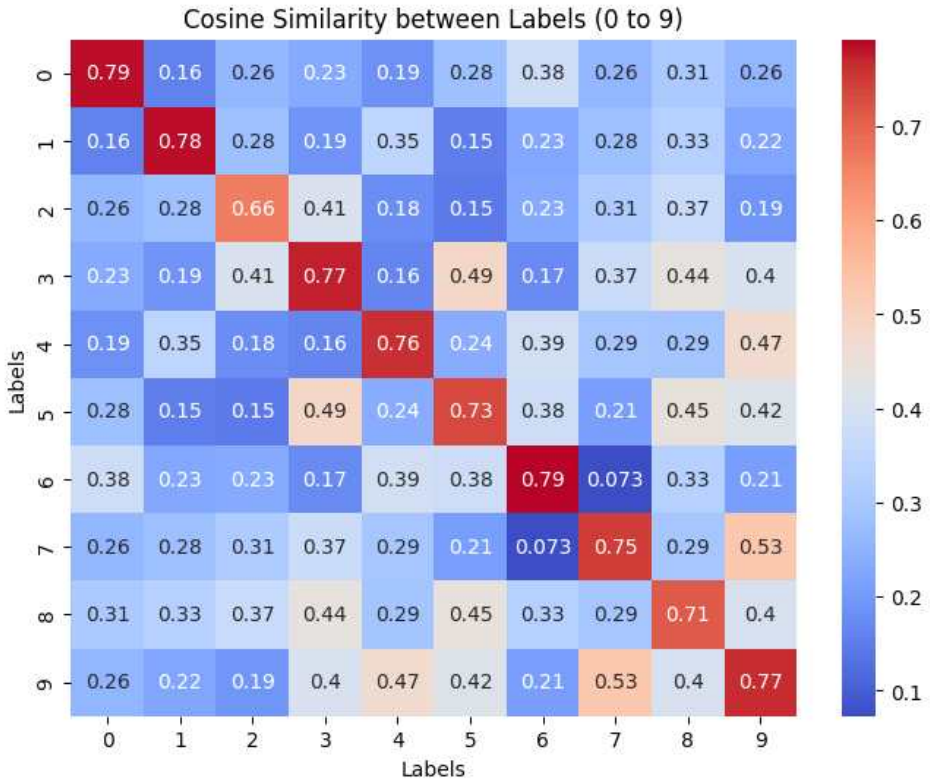
CNN은 고차원 특징을 더 낮은 차원에 임배딩하여 벡터형태로 일차원배열에 저장됨.

코사인유사도는 벡터간의 유사도를 측정하는 방법으로, 각도를 기반으로 두 벡터간의 유사도를 판단하며 벡터의 크기가 아닌 벡터의 방향에 초점을 맞춤.

손글씨는 같은 숫자라도 스케일(크기)가 다르기에 벡터의 크기보다는 벡터의 방향에 초점을 맞추는 것이 적절함.

따라서 코사인유사도라는 지표를 사용하는 것이 적절.

0~9까지의 데이터에서 각각 50개씩 샘플을 뽑아 다른 숫자의 샘플들과 코사인유사도를 계산.

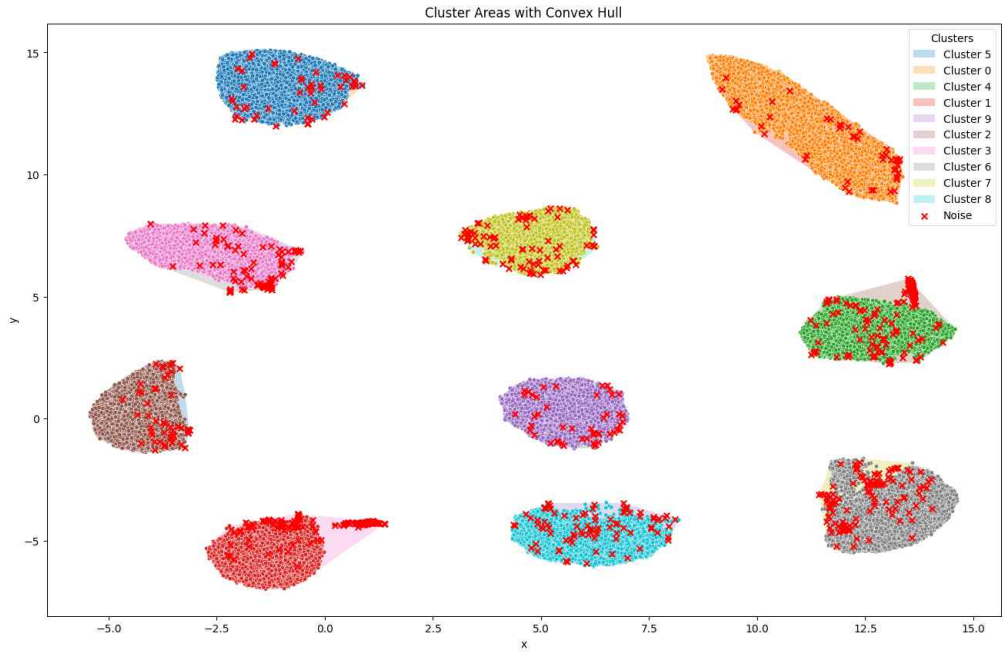


1에 가까울수록 유사도가 높으며 0.8정도면 높은 유사도를 가지고 있다고 볼 수 있음.

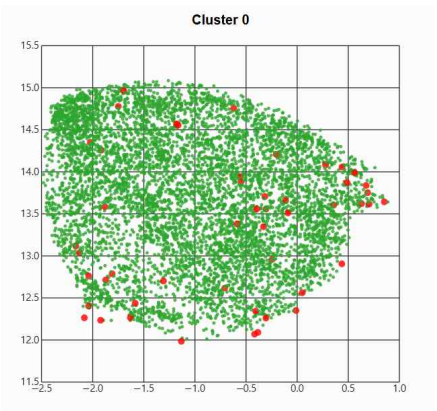
2를 제외하고는 모두 0.8에 근접하며 2도 자기 자신과의 유사도가 가장 높음.

따라서 노이즈와 노이즈가 속한 군집의 대표이미지간의 유사도가 낮게 나올 것으로 예상됨.

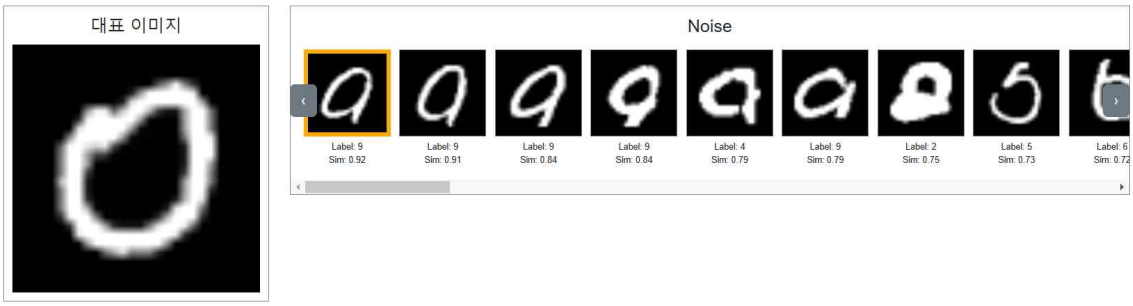
5-5. 해결 결과



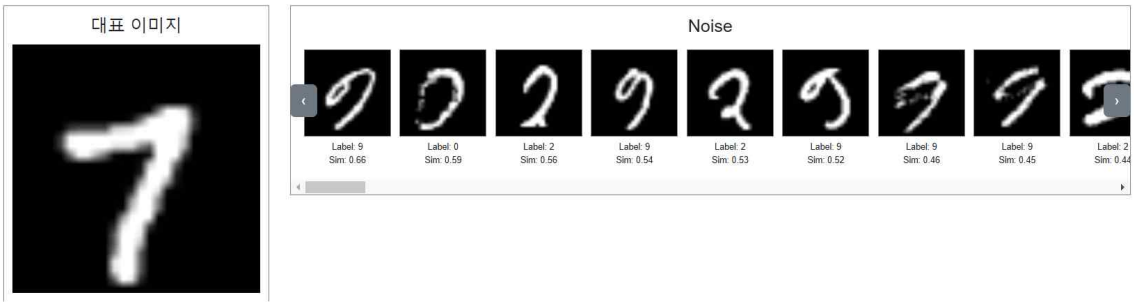
노이즈와 노이즈가 속한 군집의 대표이미지를 비교하기 위해 Convex Hull을 통해 자동으로 군집간의 영역을 설정.



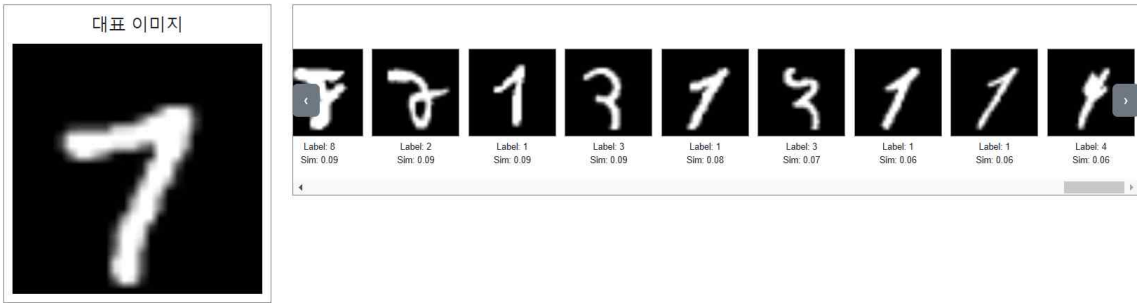
군집을 클릭하여 해당 군집의 데이터만 왼쪽의 사진처럼 선택하여 노이즈와 대표이미지간의 코사인 유사도를 비교하고 유사도가 높은 순서대로 시각화함.



위의 사진에서 볼 수 있듯이 숫자 0 군집에 속해있는 아랫부분이 잘린 9는 0과 매우 흡사하게 생겨 유사도가 0.92로 매우 높음.

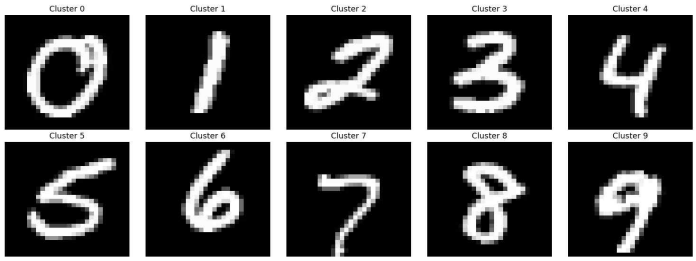


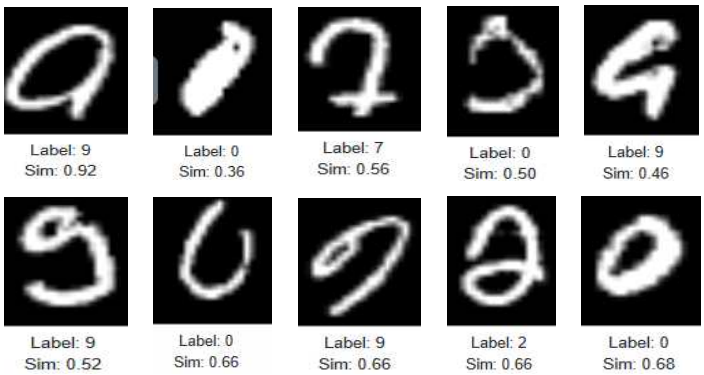
숫자 7 군집에 속한 숫자 9 또한 윗부분의 동그라미 부분이 너무 작아서 7과 흡사하게 생겨 유사도가 0.66으로 꽤 높음.



숫자 7 군집에서 유사도가 가장 낮은 건 4로 삼지창처럼 생겨서 눈으로 봤을 때는 유사할 수도 있으나 유사도는 0.06으로 매우 낮음.

(각 숫자별 대표 이미지와 가장 유사도가 높은 노이즈 정리)

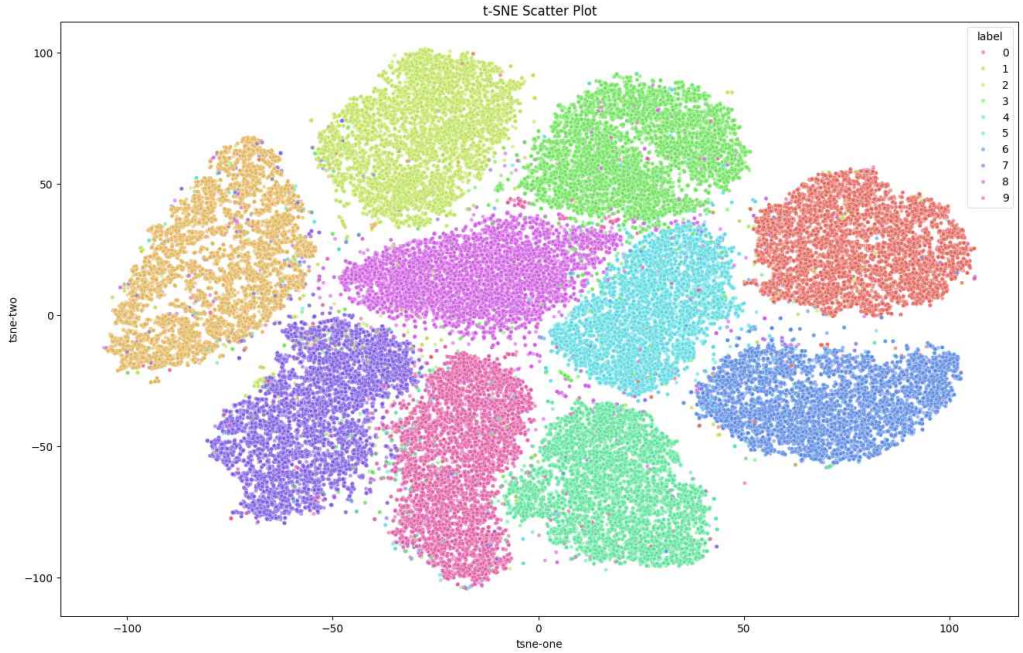




위의 결과를 보면 잘못 분류된 데이터들이 반드시 유사도가 낮은 건 아니지만 대부분이 0.6 이하로 유사도가 낮음을 알 수 있음.

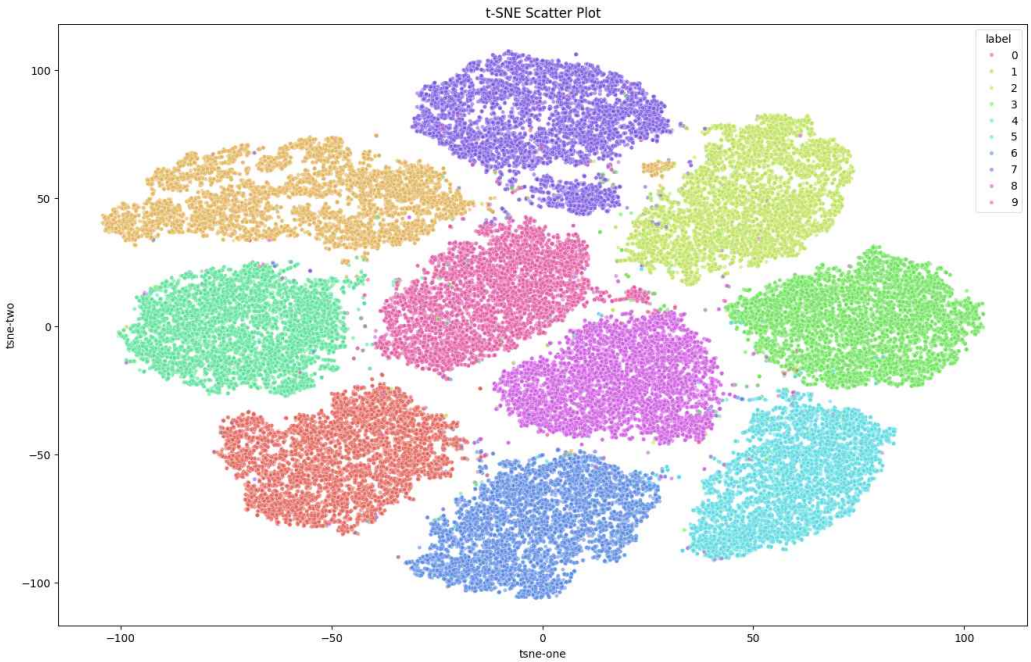
6. T-SNE

6-1. 특징 추출 전 T-SNE



T-SNE의 특성 때문에 지역적 구조는 잘 유지되지만 군집간의 거리를 통한 전역적 구조는 유지되지 않음. 이로인해 군집 사이에 위치한 점들이 많아서 흑백 이미지일 경우 군집사이의 점들은 어느 군집에 속하는지 알 수 없음.
특징 추출 및 클러스터링을 통해 더 명확한 분리 필요.

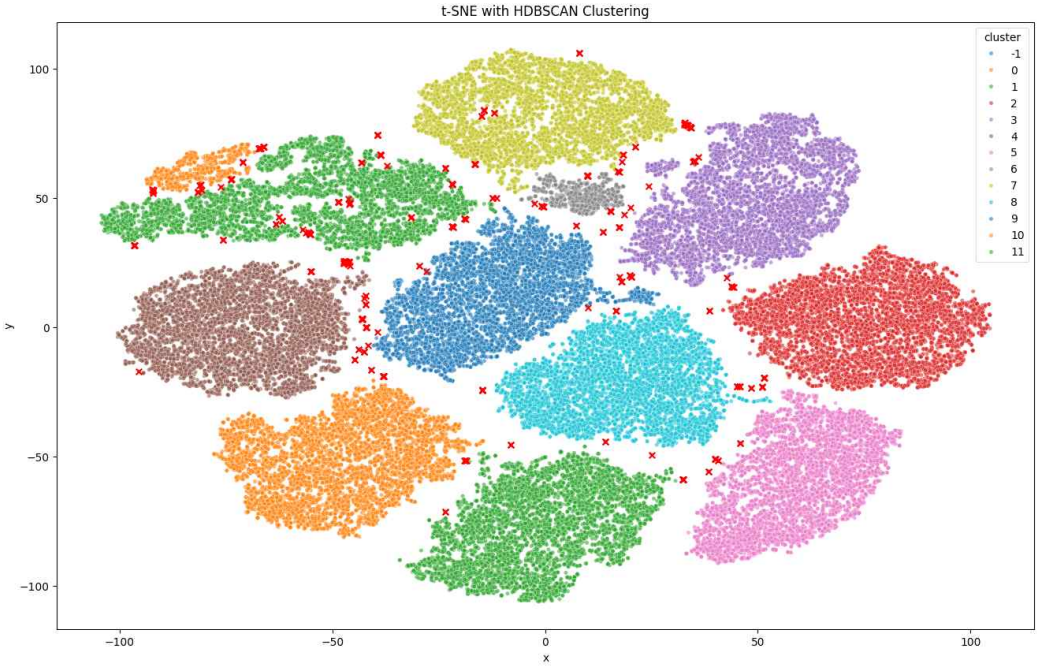
6-2. 특징 추출 후 T-SNE



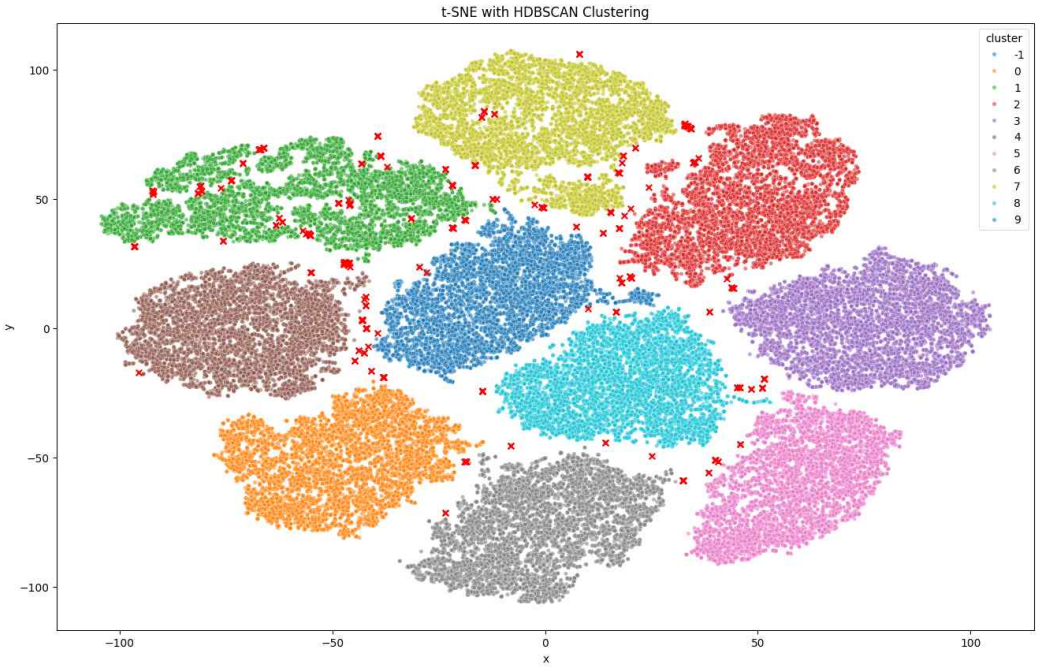
특징 추출 전보다 군집사이의 점들이 확연히 줄어들었음. 하지만 여전히 사이에 위치한 점들

이 있고 자신의 군집에 위치하지 않은 데이터가 있음. 거리를 통한 전역적 구조도 UMAP만큼 보존하지는 못함. T-SNE도 UMAP처럼 100% 확실하게 분류하지는 못함. HDBSCAN으로 클러스터링 하는게 좋음.

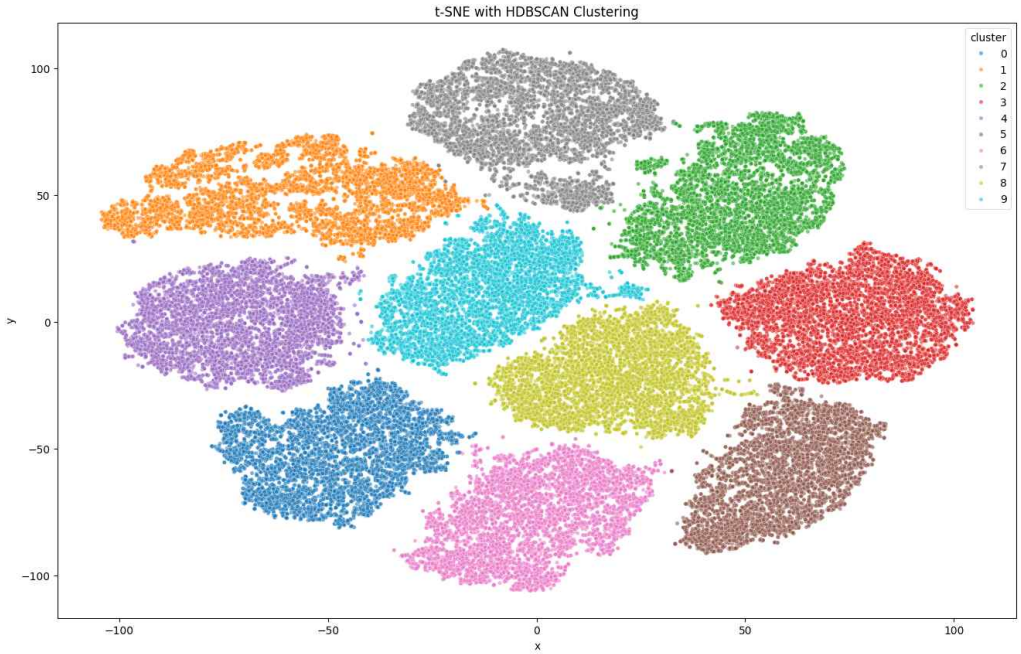
6-3. 특징 추출 후 T-SNE결과를 HDBSCAN으로 클러스터링



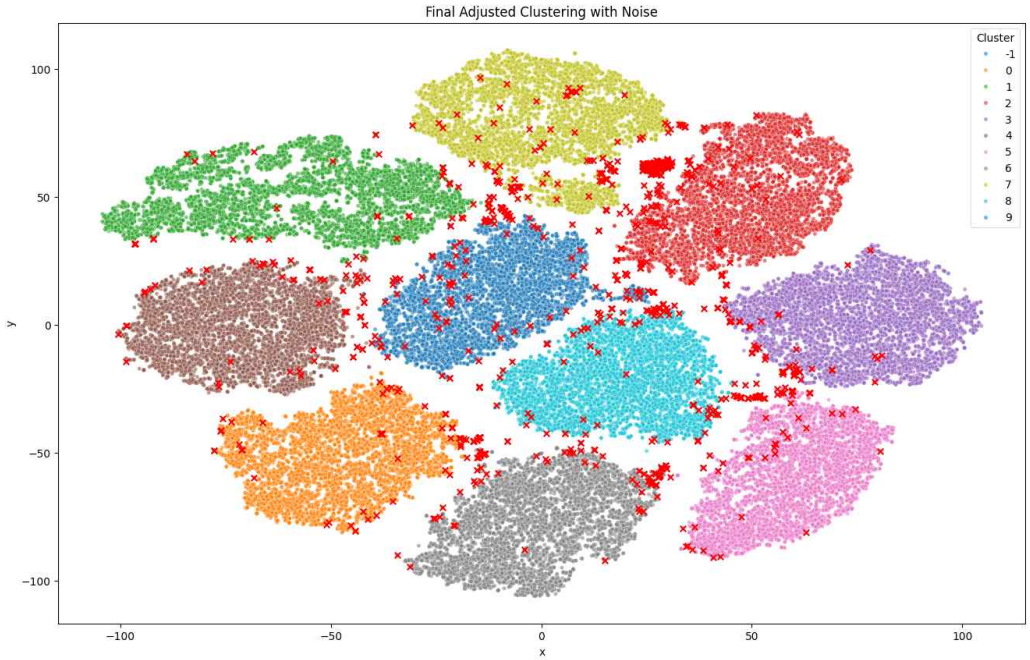
HDBSCAN으로 클러스터링하면 노이즈를 포함하여 총 13개의 군집을 만들어줌.
위 사진에서 각 군집마다 제일 많은 수의 숫자로 군집 이름을 변경(아래사진).

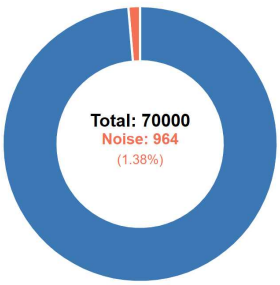


노이즈를 포함한 11개의 군집이 만들어지고, 노이즈를 가장 가까운 군집으로 편입시키면 아래 사진처럼 10개의 군집이 만들어짐.



위 사진에서 자신의 군집에서 가장 분포가 높은 숫자가 아닌 다른 숫자 데이터는 모두 노이즈 처리.



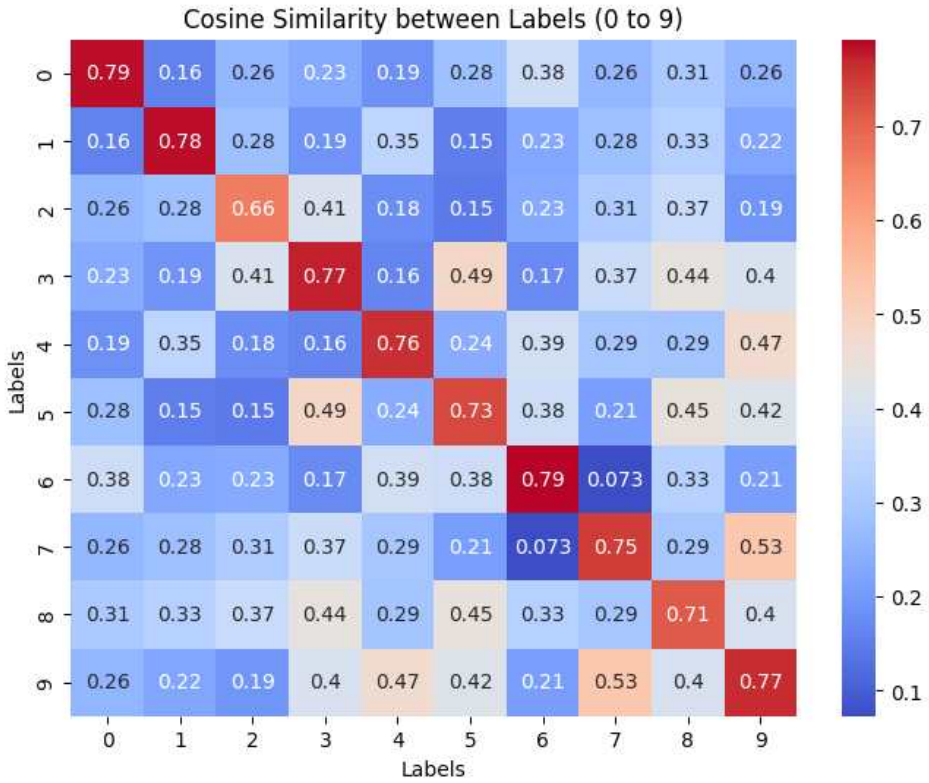


총 7만개의 데이터 중에 964개의 데이터가 잘못 분류됨.
약 0.0138%의 확률로 잘못 분류함.

6-4. UMAP에서와 같은 방식으로 궁금한 점 해결 및 해결 결과
궁금한점:
“왜 잘못 분류되는걸까? 노이즈와 노이즈가 속한 군집의 대표(실제) 이미지 간에 뭔가 다른 부분이 있지 않을까? cnn으로 추출한 특징을 서로 비교할 수 있는 지표가 있지 않을까?”라는 질문을 던질 수 있음.

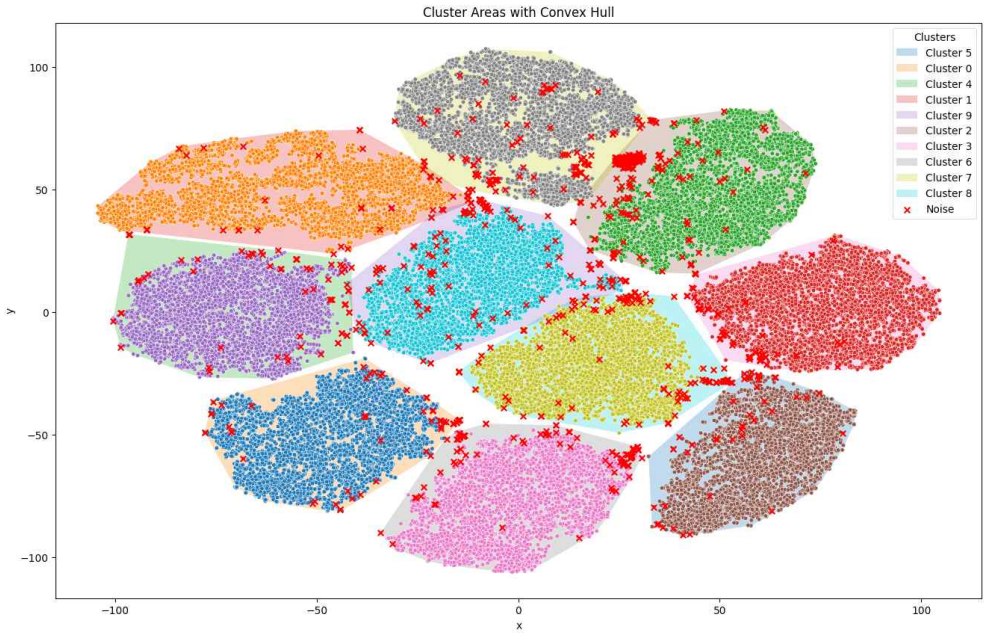
해결방안:
CNN은 고차원 특징을 더 낮은 차원에 임배딩하여 벡터형태로 일차원배열에 저장됨.
코사인유사도는 벡터간의 유사도를 측정하는 방법으로, 각도를 기반으로 두 벡터간의 유사도를 판단하며 벡터의 크기가 아닌 벡터의 방향에 초점을 맞춤.
손글씨는 같은 숫자라도 스케일(크기)가 다르기에 벡터의 크기보다는 벡터의 방향에 초점을 맞추는 것이 적절함.
따라서 코사인유사도라는 지표를 사용하는 것이 적절.

0~9까지의 데이터에서 각각 50개씩 샘플을 뽑아 다른 숫자의 샘플들과 코사인유사도를 계산.

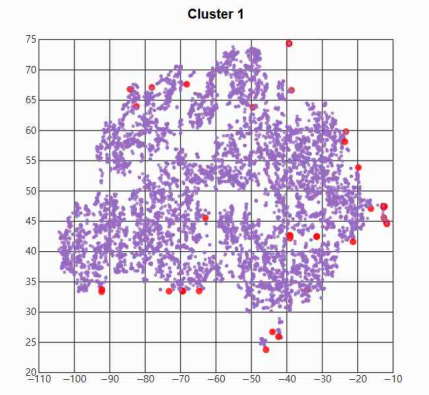


1에 가까울수록 유사도가 높으며 0.8정도면 높은 유사도를 가지고 있다고 볼 수 있음.
2를 제외하고는 모두 0.8에 근접하며 2도 자기 자신과의 유사도가 가장 높음.
따라서 노이즈와 노이즈가 속한 군집의 대표이미지간의 유사도가 낮게 나올 것으로 예상됨.

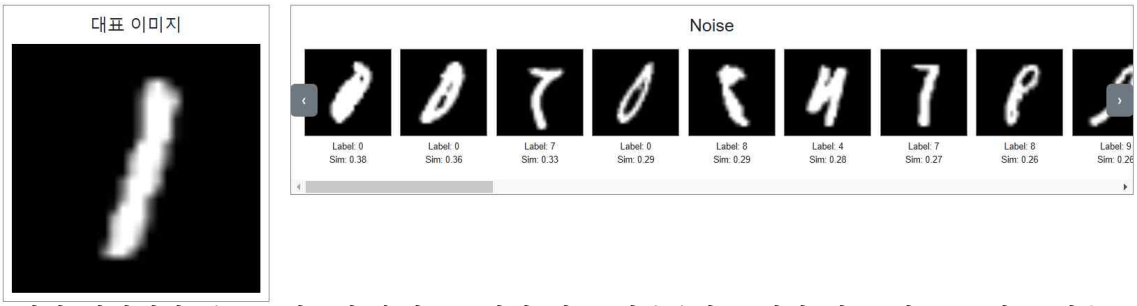
6-5. 해결 결과



노이즈와 노이즈가 속한 군집의 대표이미지를 비교하기 위해 Convex Hull을 통해 자동으로 군집간의 영역을 설정.

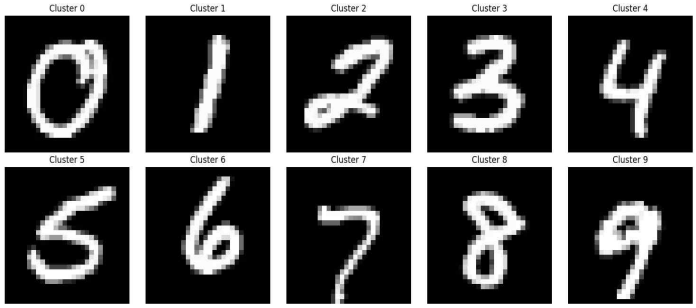


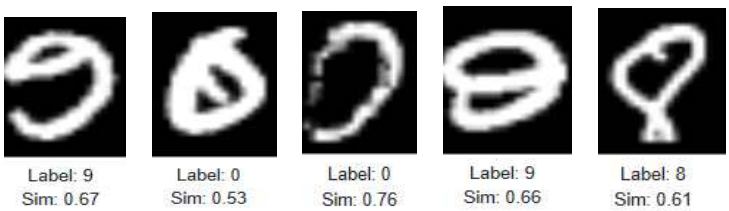
군집을 클릭하여 해당 군집의 데이터만 왼쪽의 사진처럼 선택하여 노이즈와 대표이미지간의 코사인 유사도를 비교하고 유사도가 높은 순서대로 시각화함.



위의 사진에서 볼 수 있듯이 숫자1 군집에 있는 윗부분의 구멍이 아주 작은 0 같은 경우 1과 유사하게 보임. 하지만 유사도가 0.38로 높지는 않음.

(각 숫자별 대표 이미지와 가장 유사도가 높은 노이즈 정리)





위의 결과를 보면 UMAP의 결과와 비슷함.
잘못 분류된 데이터들이 반드시 유사도가 낮은 건 아니지만 대부분이 0.6 이하로 유사도가 낮음을 알 수 있음.

7. 결론

MNIST데이터셋은 사람이 쓴 손글씨로 그 크기와 모양이 제각각이므로 특징을 추출하지 않고 차원축소하면 {4,9}, {3,5,8}, {0,9} 집합 내의 숫자들끼리는 서로 고차원에서 붙어있기 때문에 완벽하기 분리할 수 없음.

따라서 특징 추출 후 고차원에서 미리 서로를 완벽하게 분리한 후 차원축소 해야함.

CNN이 99%의 확률로 MNIST를 분류하는 것으로 알려져있어 CNN을 사용.

MNIST는 지역적구조보다는 분류가 더 중요함으로 전역적구조를 잘 유지하는 UMAP이 T-SNE보다 더 깔끔하게 분리함.

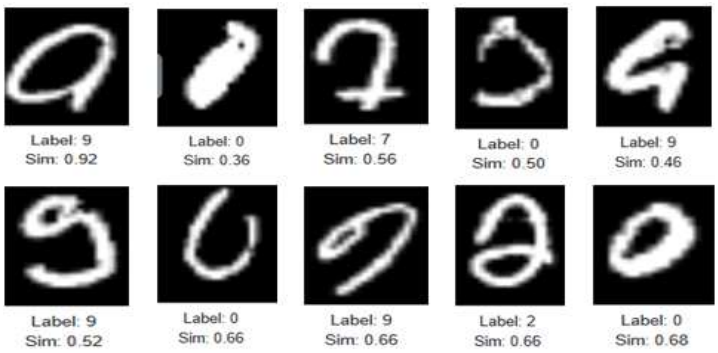
특징 추출 후 차원축소 결과 UMAP은 전역적 구조를 잘 유지하며 군집이 명확하게 구분되어 K-MEANS로 클러스터링하였고 T-SNE는 그렇지 않아 HDBSCAN으로 클러스터링함.

클러스터링 결과 UMAP은 0.0144%, T-SNE는 0.0138%의 확률로 잘못 분류.

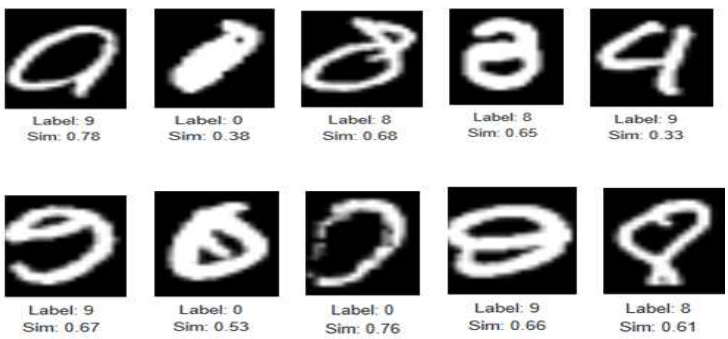
잘못 분류된 데이터(노이즈)와 해당 데이터가 속한 군집의 대표이미지간의 상관관계가 있을 것으로 가정하고 CNN으로 추출한 특징을 통해 비교하기 위해 코사인유사도로 비교.

아래의 사진에서 볼 수 있듯이 노이즈와 대표이미지간의 코사인유사도가 가장 높은 경우 0.9가 넘는 것도 있으나 대부분의 노이즈가 0.65 이하에 분포함.

<UMAP에서 대표이미지와 가장 유사도가 높은 노이즈. 순서대로 0~9까지의 군집.>



<T-SNE에서 대표이미지와 가장 유사도과 높은 노이즈. 순서대로 0~9까지의 군집.>



노이즈라고 해서 반드시 코사인유사도가 낮진 않으나 유사하다고 볼 수 있는 기준인 0.8 근처의 값을 가지는 경우는 숫자0 군집과 UMAP에서 숫자7 군집밖에 없음.

따라서 왼쪽의 이미지같은 소수의 데이터를 제외한 대부분의 데이터는 코사인유사도가 낮고, 노이즈인지 판단할 수 있음.