

1. 디자인 특허 데이터 기반 이미지 유사도 분석 결과 발표

2022. 03. 25



목차

Siamese Neural Network

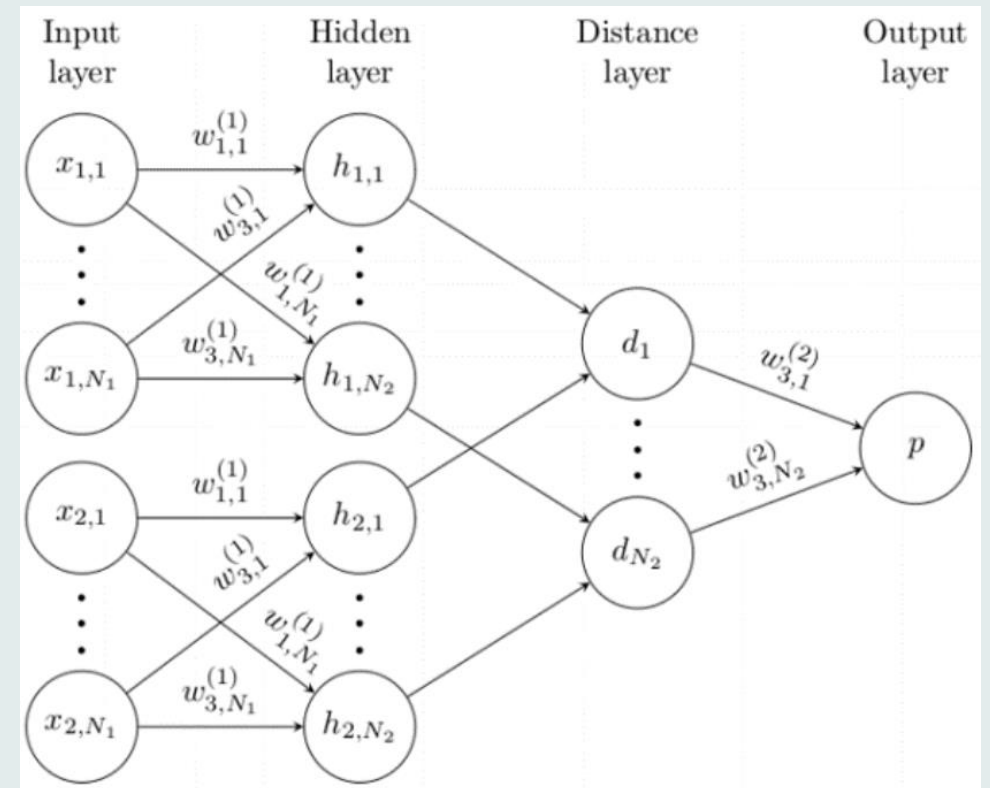
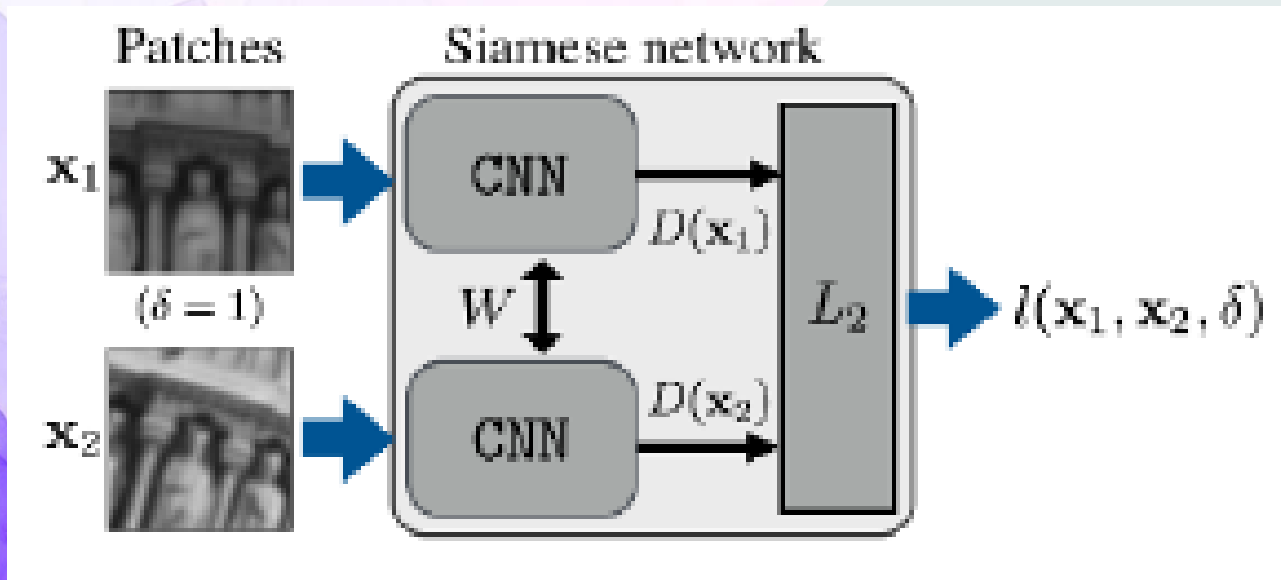
Vgg19 network

Texture Analysis

전문가 자문 의견

Siamese Neural Network

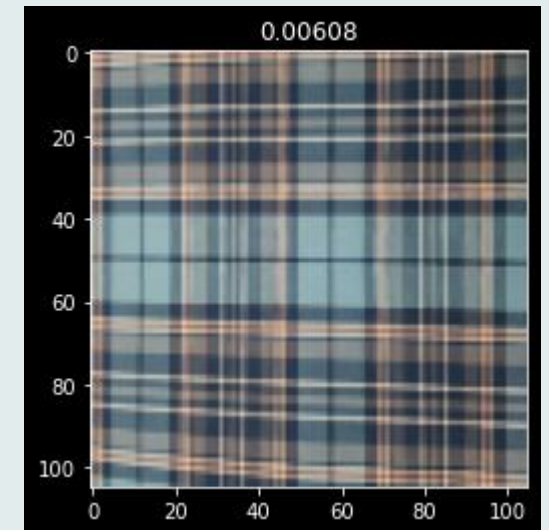
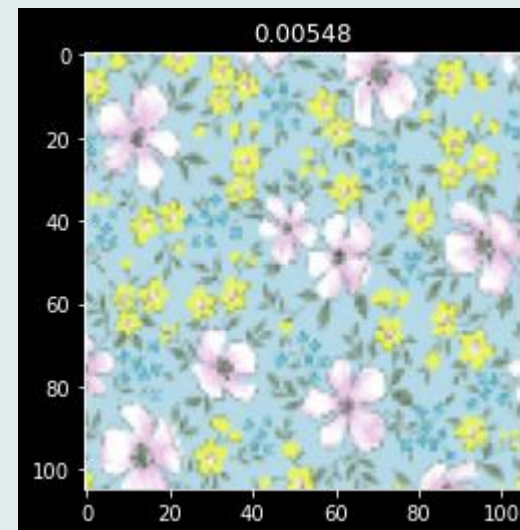
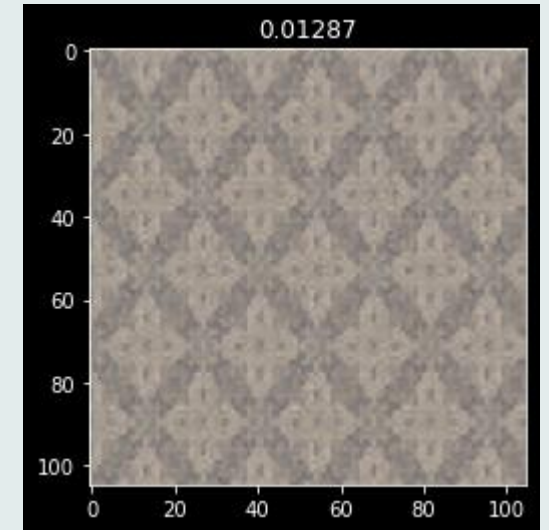
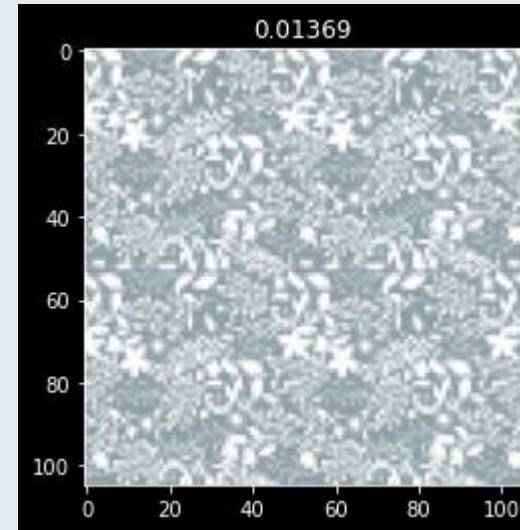
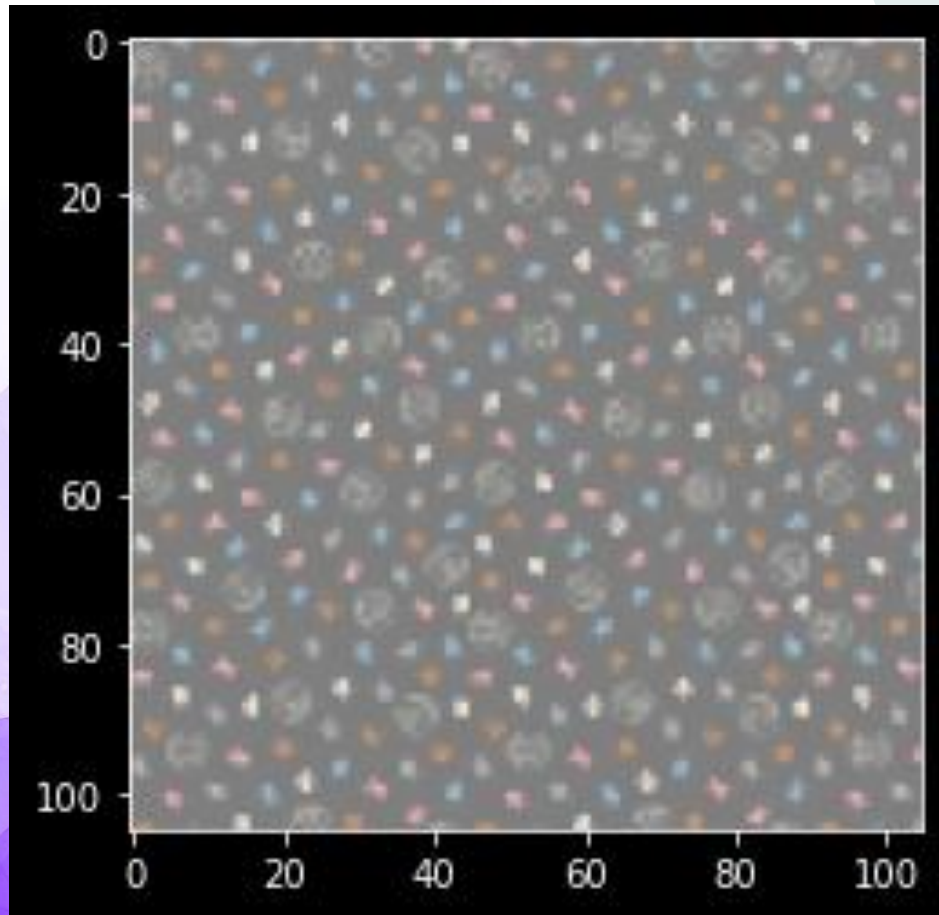
- 입력으로 들어온 2장의 그림의 유사도를 출력하는 Siamese Neural Network를 학습시키고, 특허로 등록되어 있는 패턴들과의 유사도를 계산



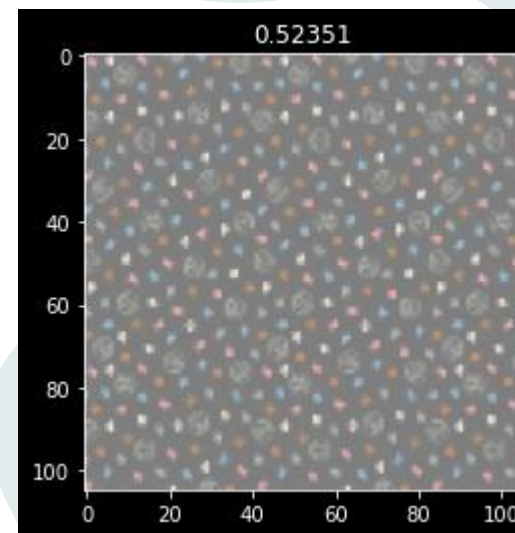
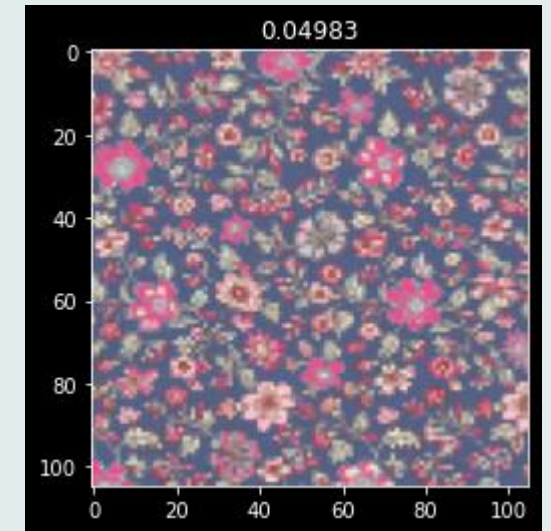
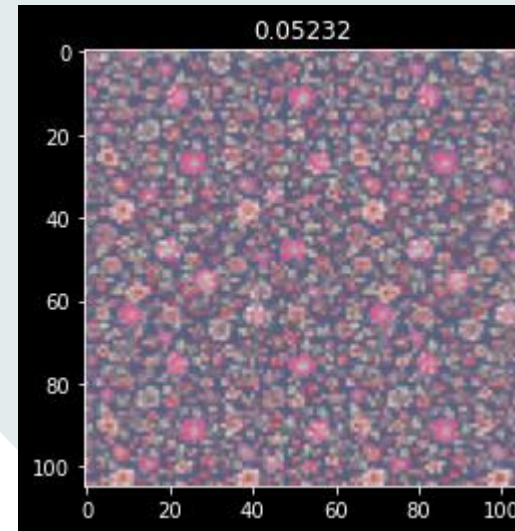
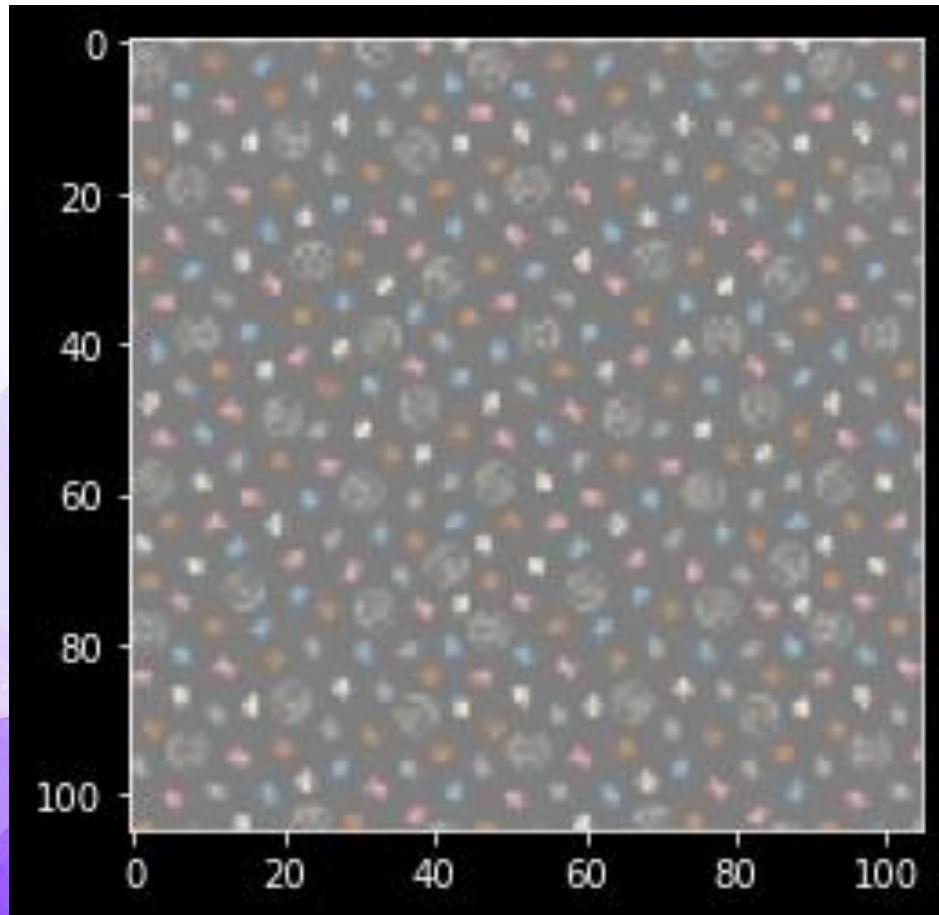
Siamese Neural Network

- 특허 이미지셋은 특허를 받을 만한 고유한 특징이 있는 패턴
- 다이텍의 이미지셋은 독특한 패턴인지 특허를 받은 패턴과 유사한 패턴인지에 대한 라벨이 없으므로, 테스트 용도로 적합하지 않음
- 결국 특허 이미지셋을 분할하여 학습과 테스트를 진행해야 함
- 평가 방법은 테스트셋으로 선택된 각각의 이미지들을 고유의 클래스로 생각하고 그 이미지의 다른 부분을 crop한 부분들을 같은 클래스로 분류하고 다른 이미지의 부분들은 다른 클래스로 분류하는지 평가해야 함

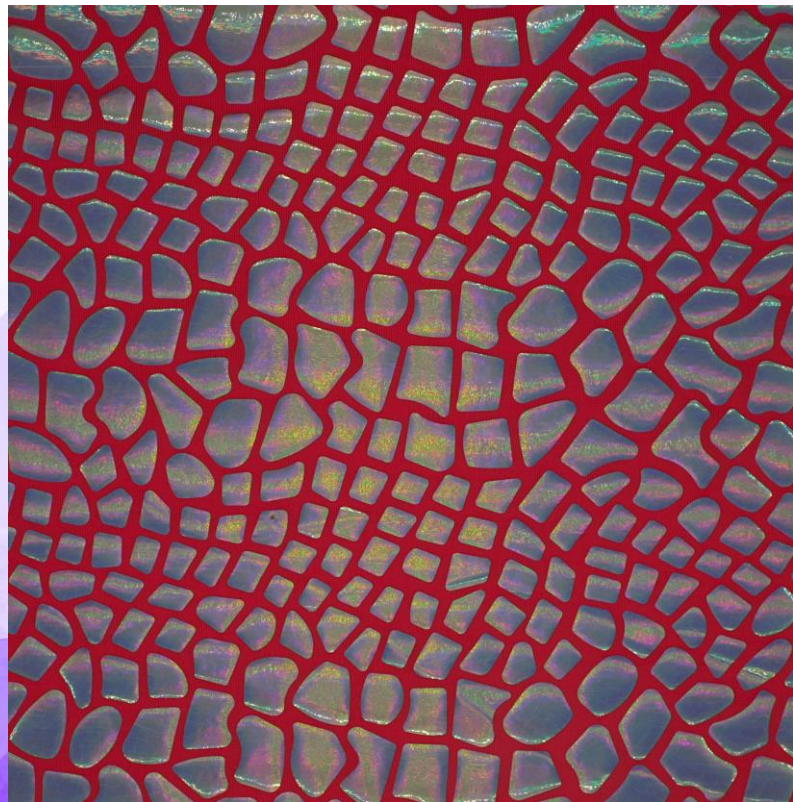
Siamese Neural Network



Siamese Neural Network



Vgg19 network

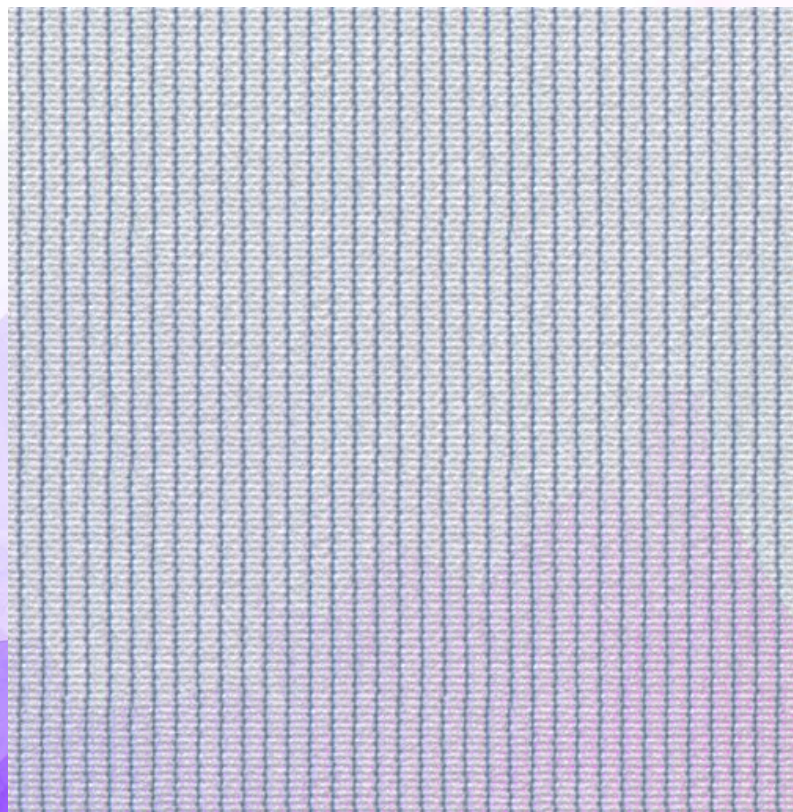


Train image

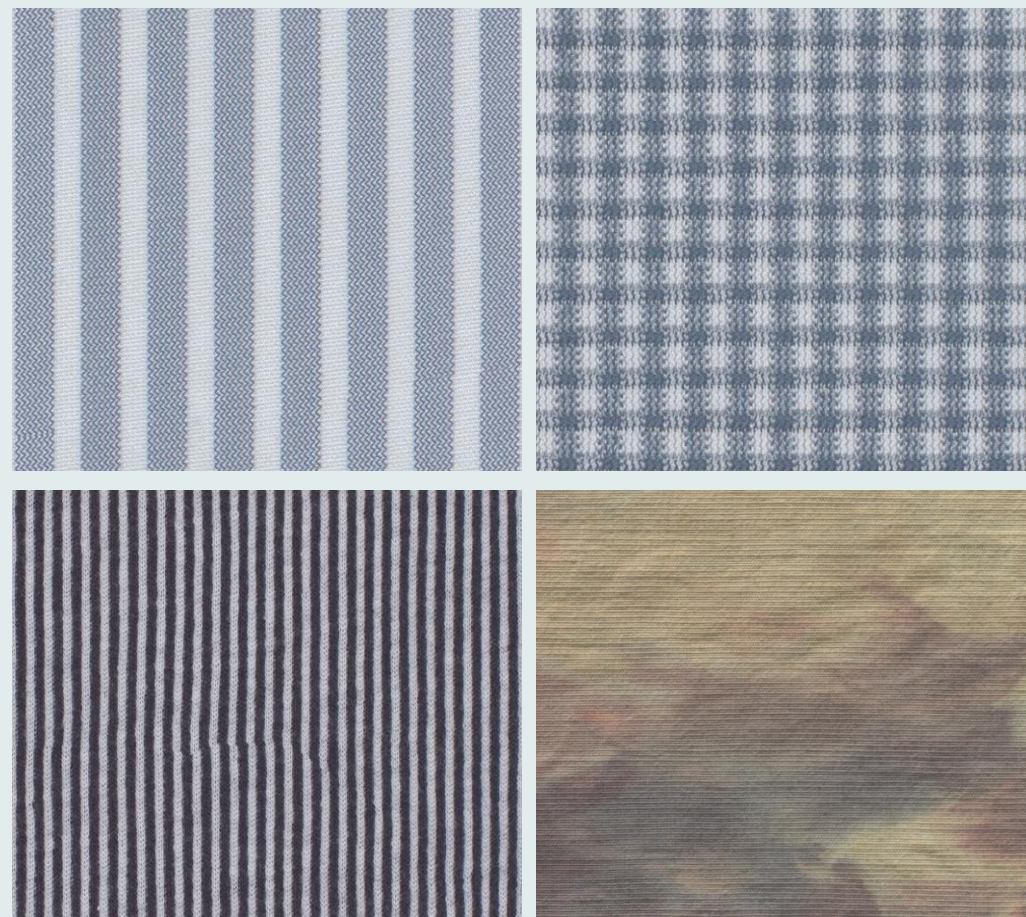


Test image

Vgg19 network



Train image



Test image

Texture Analysis

GLRLM(Grey-Level Run-Length Matrix)

Input : 이미지, 각도

Output : GLRLM

0	0	0	1	1
1	1	1	2	2
1	2	1	3	2
2	2	2	3	2
0	2	0	3	3

, angle : $0^\circ \Rightarrow$

level \ length

0

1

2

3

	1	2	3
0	2	0	1
1	2	1	1
2	4	1	1
3	2	1	0

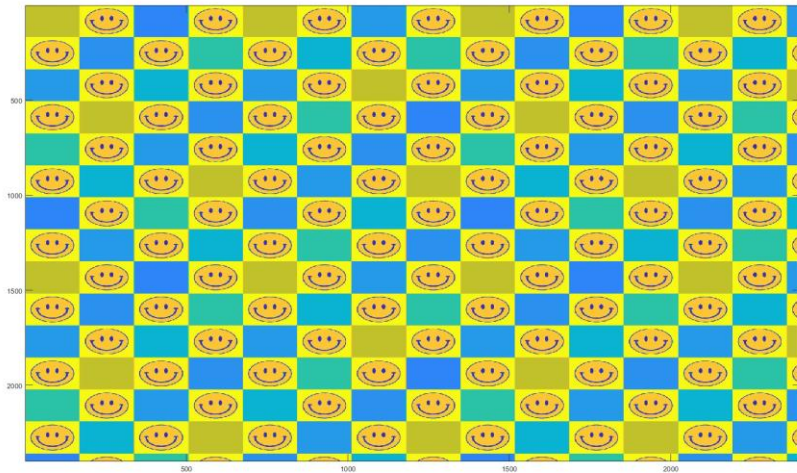
Texture Analysis

Features of GLRLM

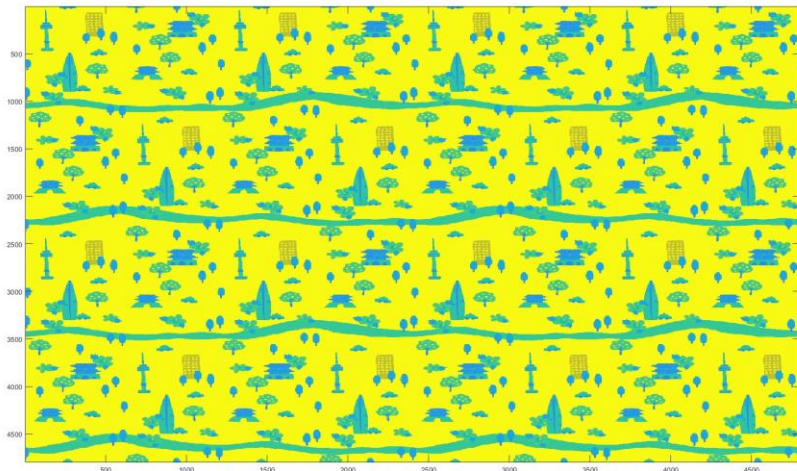
각 방향 ($0^\circ, 45^\circ, 90^\circ, 135^\circ$)에 대하여
만든 GLRLM의 feature들을 구한
후 평균을 사용함

수식	이름
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{P(i,j)}{j^2}$	Short Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N P(i,j) \cdot j^2$	Long Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{P(i,j)}{i^2}$	Low Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N P(i,j) \cdot i^2$	High Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{P(i,j)}{i^2 \cdot j^2}$	Short Run Low Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{P(i,j) \cdot i^2}{j^2}$	Short Run High Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N \frac{P(i,j) \cdot j^2}{i^2}$	Long Run Low Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{i=1}^M \sum_{j=1}^N P(i,j) \cdot i^2 \cdot j^2$	Long Run High Gray-Level Run Emphasis
$\frac{1}{n_r} \sum_{j=1}^N \left(\sum_{i=1}^M p(i,j) \right)^2$	Run Length Non-uniformity
$\frac{1}{n_r} \sum_{i=1}^M \left(\sum_{j=1}^N p(i,j) \right)^2$	Gray-Level Non-uniformity
$\frac{n_r}{n_p}$	Run Percentage

Texture Analysis

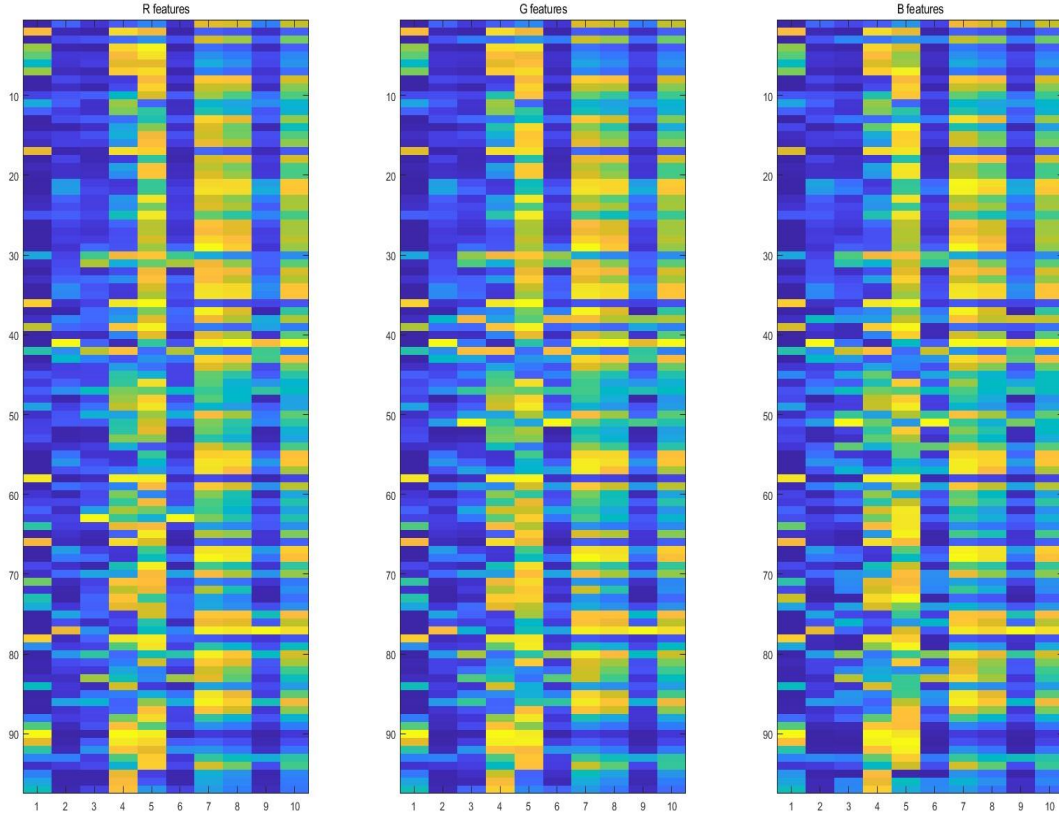


RGB로 이루어지지 않아
전처리 대상에서 제외된
파일



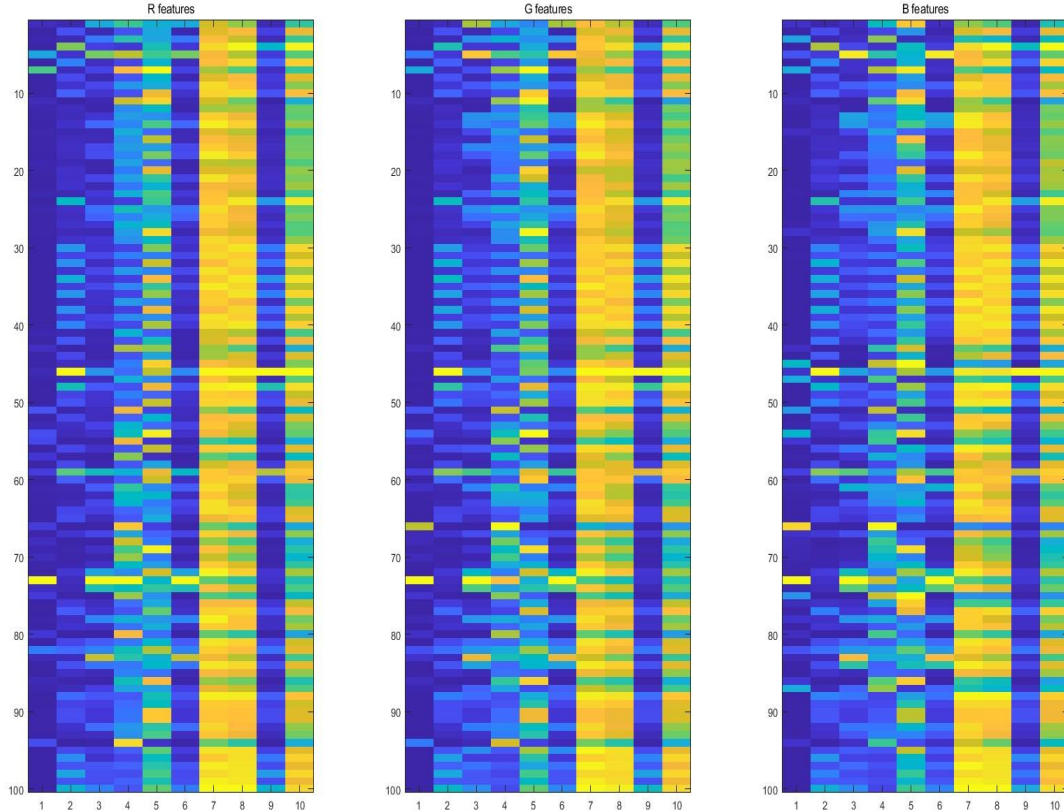
Texture Analysis

Train image : 각 이미지를
RGB층으로 나눈 후 각 층
별로 feature를 계산



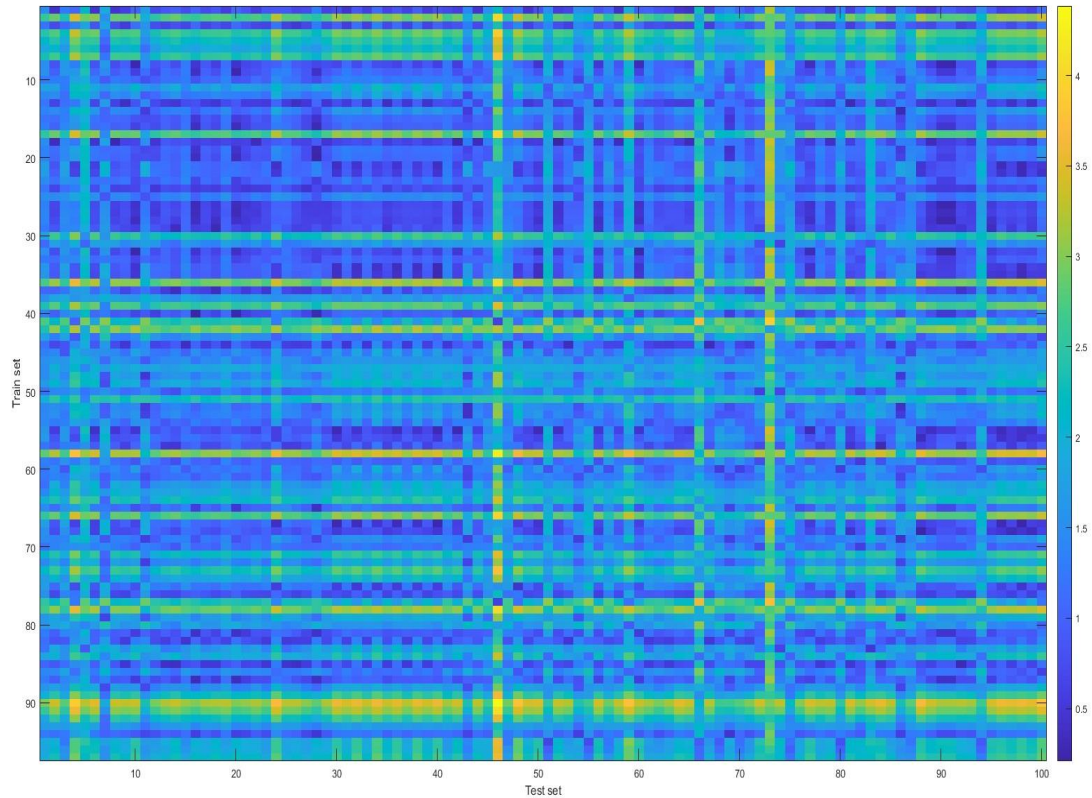
Texture Analysis

Test image : 각 이미지를
RGB층으로 나눈 후 각 층
별로 feature를 계산



Texture Analysis

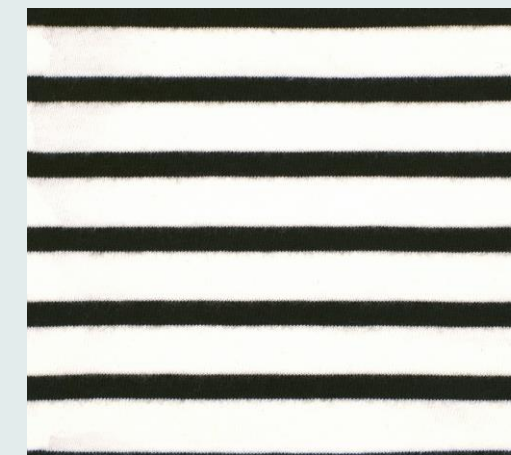
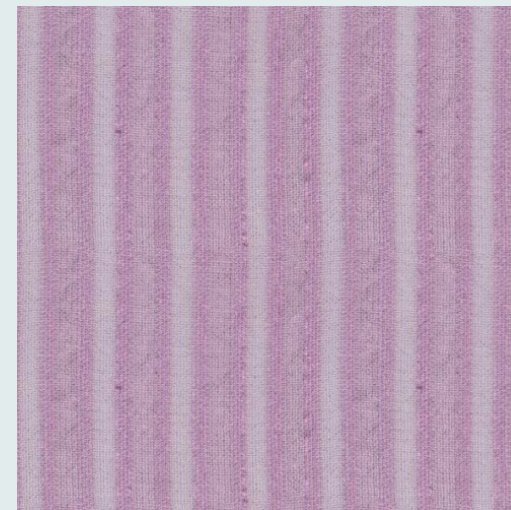
Train과 test 사이의 거리
계산



Texture Analysis



Train image



Test image



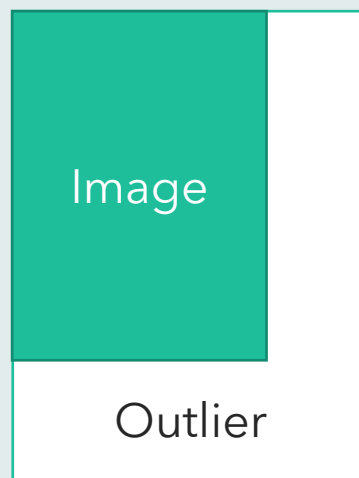
전문가 자문 의견

전문가 자문 의견 1

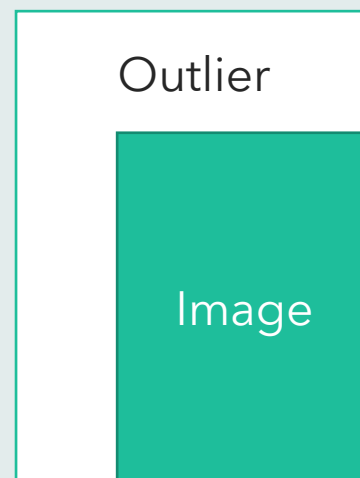
Data preprocessing

- 이미지의 테두리(1픽셀 단위)에서의 값들을 이용해 outlier 여부 판단
- Outlier의 여부가 판단되면 해당하는 부분을 제거

Ex)



or



전문가 자문 의견 1

Similarity analysis via feature

- Autoencoder를 이용해 feature를 추출한 후, PCA를 통해 유의미한 성분에 대한 유사도 검증을 이용한 방법 제시

전문가 자문 의견 2

Deep Learning

영상은 structure와 texture로 구성

해당 디자인 표현의 특성은 structure에 대한 정보의 의존성보다 texture에 대한 의존성이 높음

- 따라서 주어진 영상으로부터 texture와 structure를 구분하기 위한 deep learning 모델을 활용
- 빠른 시간 내에 대용량 데이터와의 유사도 비교를 위하여 deep binary hashing을 통해서 유사도를 비교한 후 정량적으로 표현할 수 있음

전문가 자문 의견 3

Algorithm using CNN and Autoencoder

- CNN과 autoencoder를 결합한 이미지 특성 파악 알고리즘을 개발하는 것을 추천
- 데이터를 구성하는 주요 feature를 찾아내는 방법으로 autoencoder를 활용
- 이미지 데이터 분석을 위한 CNN 계열 알고리즘 적용

전문가 자문 의견 4

주기분석을 통한 패턴 구분

- 이미지의 주기적 성향을 furrier transform과 같은 방법으로 주기별로 분할하여 표현
- 고주파, 저주파, 중주파로 나누어 패턴 성질을 분할하고 이를 사용해 패턴을 구분