

제36회 서울특별시학생탐구발표대회

탐구보고서

픽셀 간의 관계를 이용한 이미지 예측 및 특징 추출

출 품 번 호

미기재

출 품 부 문

IT SW 융합

2021. 08. 07.

학 번 : 20105

이 름 : 김동환

목차

1. 탐구 주제
2. 탐구에 대한 동기와 목적
3. CVFW 개발 계획
4. CVFW 기술 원리
5. 성능 분석
6. 실용 예시
7. 탐구 소감
8. 참고 문헌
9. 부록

1. 탐구 주제

- 이미지 픽셀 간의 관계를 확인해보면 특징 관계와 이미지의 특징을 찾아낼 수 있지 않을까?

2. 탐구에 대한 동기와 목적

(1) 탐구 동기

- 나는 방학 동안 인공지능에 대해 더 공부해야겠다는 목표를 세웠었다. 그리고 CNN이라는 이미지 데이터를 집중적으로 학습하는 신경망 아키텍처를 알게 되었다. 그로 인해 ‘컴퓨터에게 이미지를 어떻게 학습시킬까?’라는 의문점을 깊이 고민하게 되었고, 나는 픽셀 간의 관계 중 특징적인 관계가 존재할 것이라고 생각했다. 그렇게 나는 특징 관계(가중치)를 이용하여 이미지 처리를 하는 CVFW(부록 참조) 이라는 프로젝트를 진행하게 되었다.

(2) 탐구 목적

- 이번 탐구로서 컴퓨터 비전 분야에 새로운 아이디어를 제시하며 기여하고, 본 탐구에서 CVFW 알고리즘으로 이미지 클래스 예측 및 특징 추출을 하는 것이다. 또한 CVFW 알고리즘이 여러 분야에서 직접적으로 활용되거나, CVFW 알고리즘에 아이디어를 덧붙여 더욱 뛰어난 성능을 보여주는 알고리즘이 개발되었으면 하는 것이 이번 탐구의 목적이다.

3. CVFW 개발 계획

1. 픽셀 간의 관계를 가중치로 확인하기
2. 가중치 그룹화 및 특징 그룹 찾기
3. 비용 함수 정의
4. 비용 함수를 이용하여 이미지 클래스 예측하기
5. 각 가중치에 가장 큰 특징 그룹들을 이용하여 모델링(특징 추출)하기

4. CVFW 기술 원리

(1) 정의

1. m은 이미지의 가로 크기, n은 이미지의 세로 크기이다.
2. W는 픽셀 간의 가중치 행렬이다.
3. P는 이미지 픽셀의 1차원 데이터이다.
(자세한 정의는 각 소제목에서 서술한다.)

(2) 픽셀 간의 가중치 구하기

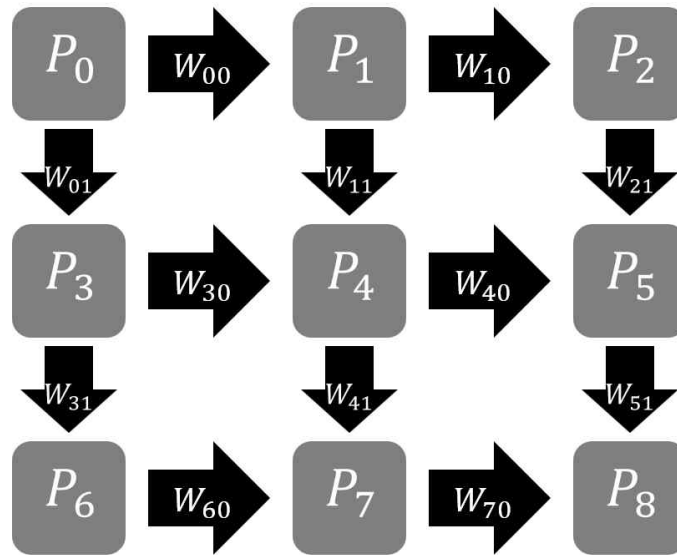
$$W_{i0} = \frac{P_{i+1}}{P_i} \quad \{(i+1) \bmod m \neq 0\}$$
$$W_{i1} = \frac{P_{i+m}}{P_i} \quad \left(\left[\frac{i}{m}\right] \neq n-1\right)$$

[그림1. 가중치 공식]

W_{i0} 은 P_i 번째 픽셀과 그 오른쪽 픽셀과의 가중치이다. (단, $i+1$ 은 m 으로 나누어떨어지지 않는다.)

W_{i1} 은 P_i 번째 픽셀과 그 아래쪽 픽셀과의 가중치이다. (단, i / m 의 정수부분은 $n - 1$ 이 아니다.)

이렇게 위 공식을 이용하여 이미지의 모든 픽셀에서의 가중치를 구한다. 그렇게 P_i 에 W_{i0} 을 곱하면 $P_{(i+1)}$ 의 값이 되고, P_i 에 W_{i1} 을 곱하면 $P_{(i+m)}$ 이 되도록 한다.



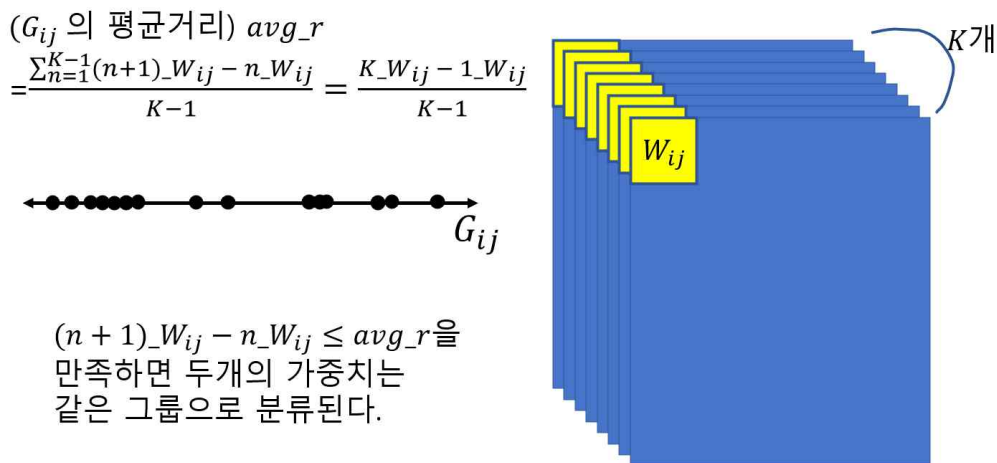
[그림2. 가중치 예시]

(3) 가중치 그룹화 및 특징 그룹 찾기

- 그룹화: 특징 그룹을 찾기 전 준비단계로, K개의 학습 데이터로 모인 K개의 W_{ij} 를 그룹화하는 것이다. ‘어떻게 그룹화를 할까?’ 아이디어를 고민하다가 밀집되어있는 가중치를 찾아야겠다고 생각했고, 그룹을 다음과 같이 정의하였다. 그림에 나온 방법으로 G_{ij} 를 그룹화한다.

정의1. G_{ij} 는 K개의 W_{ij} 모임이다.

정의2. n_W_{ij} 는 n번째 W_{ij} 이다.



[그림 3. 가중치 그룹화]

- 특징 그룹 찾기: 실질적인 학습 단계라고 볼 수 있는 특징 그룹 찾기에는 이용자가 직접 조정하는 변수가 있다. 하나의 G_{ij} 에서 허용되는 특징 그룹의 최대 개수인 FGN(Feature Group Number)과 하나의 특징그룹 안의 원소의 최소개수인 FWN(Feature Weight Number)이다. 특징 그룹을 찾는 방법은 다음과 같다. G_{ij} 에서의 특징그룹은 FW_{ij} (Feature Weight)이다. 또 FW_COUNT 는 추후 비용 함수를 정의하며 사용된다.

- 단계1. 하나의 G_{ij} 에서 원소의 개수가 많은 순으로 상위 FGN개의 그룹을 뽑는다.
- 단계2. 뽑은 그룹중 원소의 최소개수 FWN을 만족하지 않는 그룹은 제외한다.
- 단계3. 단계1, 2를 거쳐 남은 그룹을 특징그룹(FW_{ij})에 담는다.
- 단계4. 특징그룹안의 그룹들을 평균으로 변환시킨다.
- 단계5. 특징그룹의 원소의 개수가 1개 이상이라면 FW_COUNT 에 1을 더한다.

(4) 비용 함수 (cost function)

- 본 탐구에서의 비용 함수는 학습된 이미지와 비교하려는 이미지의 차이량이다. cost function은 클래스를 예측할 때 중요하게 쓰이는 함수이다. 비용 함수에 들어가는 인자는 비교하려는 이미지의 가중치이다. FW_COUNT 는 학습된 이미지의 특징그룹(FW)에 따라 cost function을 잘 조정하는 역할을 한다. 예를 들어 학습된 A이미지에서의 원소의 개수가 1 이상인 FW_{ij} 의 개수가 3개라면 $Cost(W)$ 가 낮을 확률이 굉장히 높다. (FW_{ij} 가 비어있을 경우 $C(W_{ij}) = 0$ 으로 대체되기 때문) FW_COUNT 를 이용하는 것은 이런 경우를 대비한 것이다.

- 단일 cost function(하나의 W_{ij} 에 대한 비용함수)은 다음과 같다. 단, FW_{ij} 이 비어있다면 $C(W_{ij}) = 0$ 이다. (min은 인자로 들어온 행렬 중 최솟값을 찾는 것이다.)

$$C(W_{ij}) = \min(FW_{ij} - W_{ij})^2$$

[그림4. 단일 cost function 식]

- 전체 cost function 은 다음과 같다.

$$\begin{aligned} Cost(W) &= \frac{\sum_{i=0}^{mn-1} \sum_{j=0}^1 C(W_{ij})}{FW_COUNT} \\ &= \frac{\sum_{i=0}^{mn-1} \sum_{j=0}^1 \min(FW_{ij} - W_{ij})^2}{FW_COUNT} \end{aligned}$$

[그림5. 전체 cost function 식]

(5) 이미지 예측

- 이미지 예측은 예측하려는 이미지가 있을때 그 이미지의 클래스를 이미 학습된 이미지를 이용하여 예측하는 것이다. (본 탐구에서 얘기하는 클래스는 예측할 수 있는 범위를 얘기하며 예로 들어 마우스와 노트북을 구별해야 한다면, 마우스와 노트북이 클래스에 해당한다) 예를 들면 강아지 사진과 고양이 사진을 학습하고, 강아지 사진을 읽고, 그 사진이 강아지 사진인지 고양이 사진인지를 예측하는 것이다. 이미지 예측은 위에서 정의한 Cost Function 을 이용하여 쉽게 구할 수 있다.

정의1. 예측하려는 이미지의 가중치는 W이다.

정의2. 예측 결과값은 Predict(W) 이다.

$$Predict(W) = (Cost_1(W) \quad Cost_2(W) \quad Cost_3(W) \cdots)$$

[그림6. Predict(W) 식]

min(Predict(W))에 해당하는 클래스가 바로 예측 클래스이다.

(6) 특징 모델링

- 특징 모델링은 학습된 한 클래스에서의 가장 큰 특징들을 종합하여, 이미지로 반환하는 것이다. 예로 들면 형태는 각기 다른 가방을 학습시켰을 때는, 모양이 다른 가방 안에서도 공통된 특징을 찾아내어 반환하는 것이다. 모델링 된 데이터는 M으로 표현한다. 이 소제목에서는 가장 위쪽, 가장 왼쪽을 제외한 M_i는 두 개의 가중치로 결정됨을 인지하지 못한 채 진행하여 고민을 많이 했었다.

정의1. 이 소제목에서의 W_{ij}는 FW_{ij}에서 가장 원소의 개수가 많은 특징 그룹이다.

정의2. N_{ij}는 FW_{ij} 안의 그룹들의 원소의 최대 개수이다.

많은 고민 끝에 떠올린 아이디어는 가중치를 N_{(i-m)1} 과 N_{(i-1)0} 의 비로 계산하여 합치는 것이었다.

$$M_i = \frac{M_{i-m} * W_{(i-m)1} * N_{(i-m)1} + M_{i-1} * W_{(i-1)0} * N_{(i-1)0}}{N_{(i-m)1} + N_{(i-1)0}}$$

[그림7. 특징 모델링 식]

위의 식을 통해 이미지의 특징을 알아내었고, 특징의 유사도를 측정하여 클래스를 예측하는 방법 또한 생각이 났었다. 그렇게 모델링 된 데이터로 예측하는 방법인 modeling_predict 함수 또한 연구하였다.

방법은 간단했다. 4-5에서 소개한 Predict(W)는 그대로 사용하지만 Cost Function만 조금 다르다.

정의1. MW은 모델링된 데이터 M의 가중치이다.

정의2. W은 비교하려는 데이터의 가중치이다.

$$Cost(W) = \sum_{i=0}^{mn-1} \sum_{j=0}^1 |(MW_{ij} - W_{ij})|$$

[그림8. 모델링 된 데이터로 예측하기 위한 Cost Function 식]

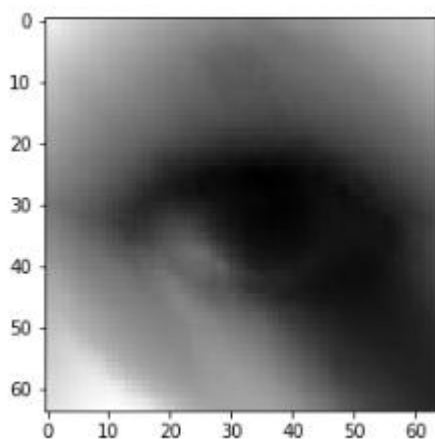
5. 성능 분석

(1) 모델링 및 정확도 측정

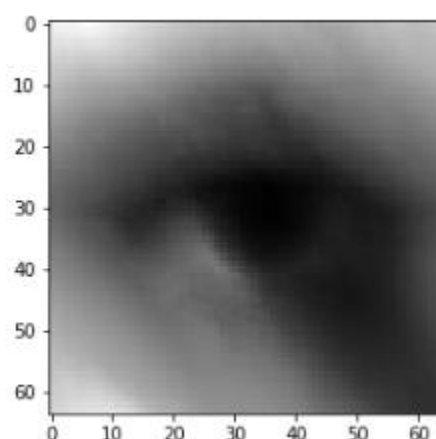
- 목차4를 바탕으로 프로그래밍 된 모듈로 정확도를 측정할 차례이다. 총 3가지 데이터셋 (학습에 필요한 이미지 데이터)을 학습하고 성능을 분석했다. 첫번째는 눈을 보고 여자인지 남자인지를 구별하는 것이었고, 두번째는 얼굴을 보고 구별하는 것이었고, 마지막 세번째는 손의 모양을 보고 어떤 모양인지 구별하는 것이었다.

1. 눈으로 성별 구별: 약 4000개의 학습 데이터로 학습하였다.

- 모델링



[그림9. female eyes class]



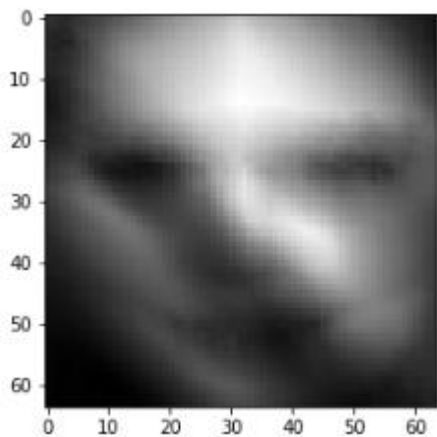
[그림10. male eyes class]

의도한대로 특징 모델링을 반환한 모습을 볼 수 있다.

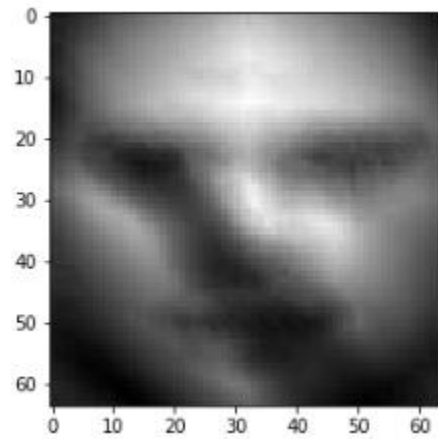
- 정확도 측정: 여러번의 FGN과 FWN의 수정 끝에 정확도를 업데이트해보았지만, 55.6% 정도로 아주 낮은 수치를 보여주었다.

2. 얼굴로 성별 구별: 약 4000개의 학습 데이터로 학습하였다.

- 모델링



[그림11. female face class]



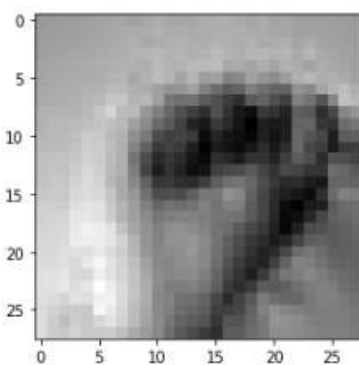
[그림12. male face class]

두번째 데이터셋 또한 의도한대로 특징을 잘 찾아낸 것 같이 판단된다.

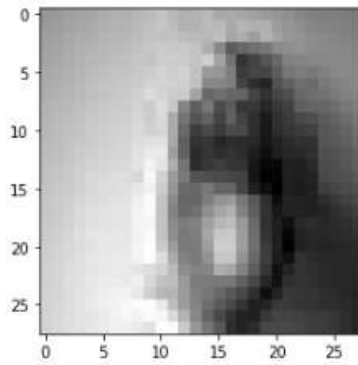
- 정확도 측정: 여러번의 FGN, FWN 수정 끝에 약 75% 정도의 높지 못한 정확도를 보여주었다.

3. 손 모양 구별: 약 27500개의 학습 데이터로 학습하였다.

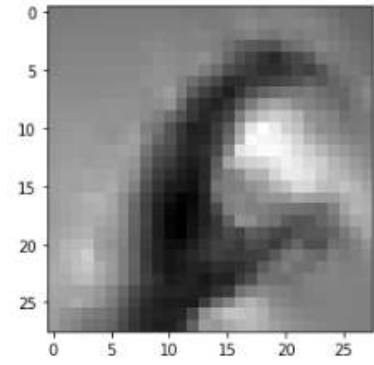
- 모델링



[그림13. rock class]



[그림14. palm class]



[그림15. C class]

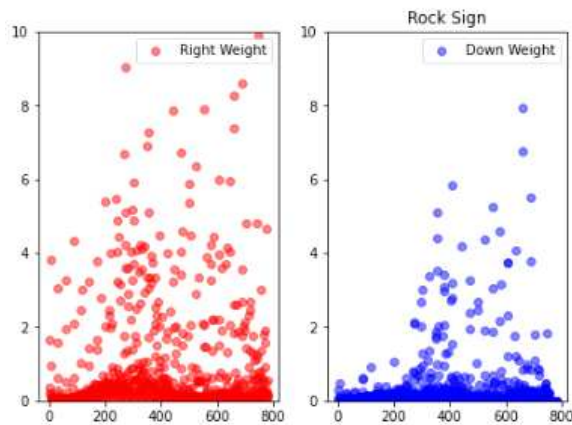
세번째 데이터셋 또한 특징을 잘 잡아주었고, 세 개의 데이터셋 모두 의도한대로 특징을 잘 잡아내었다.

- 정확도 측정: 특징 그룹으로 클래스를 예측하는 방법을 이용하여 최대 92.9%로 꽤나 높은 수치를 보여주었다. 그리고 특징 모델링 된 데이터로 유사도를 측정하여 예측하는 방법으로는 약 95.7%로 많이 높은 수치를 보여주었다. 이렇게 이 두 개의 예측 방법을 비교해보았는데, 그 부분은 5-3에서 소개되었다.

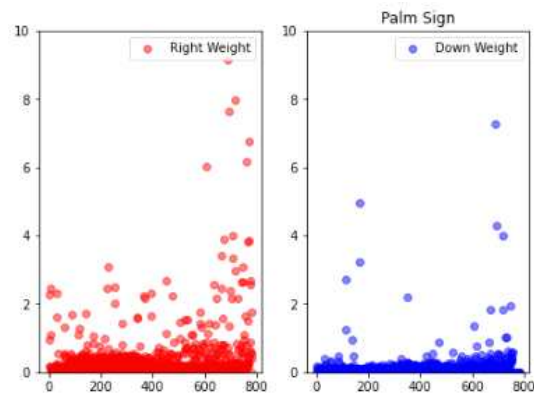
(2) 정확도별 데이터셋 가중치 표준편차 분석

- 5-1에서 보았듯이 정확도가 잘 나와주었던 데이터셋과 잘 나와주지 못했던 데이터셋들이 있었다. 그래서 이 데이터셋들의 가중치를 구하여 각 가중치마다 표준편차를 구해 시각화한 것이다.

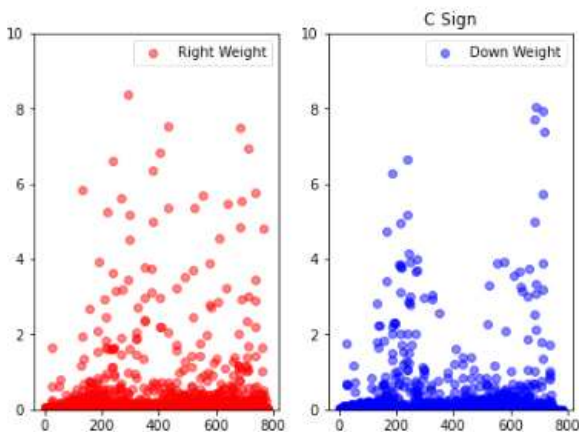
가장 처음으로는 정확도가 제일 잘 나와주었던 손 모양 구별 데이터셋을 시각화했다. 처음 예상은 정확도가 높게 나왔으므로 표준편차가 적을 것이고 많이 원만할 것으로 예상하였다. 빨간색의 산점도는 오른쪽 방향의 가중치이고, 파란색의 산점도는 아래쪽 방향의 가중치이다.



[그림 16. rock standard deviation]



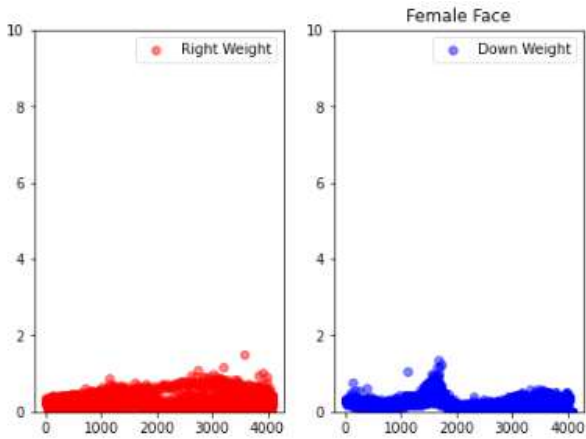
[그림 17. palm standard deviation]



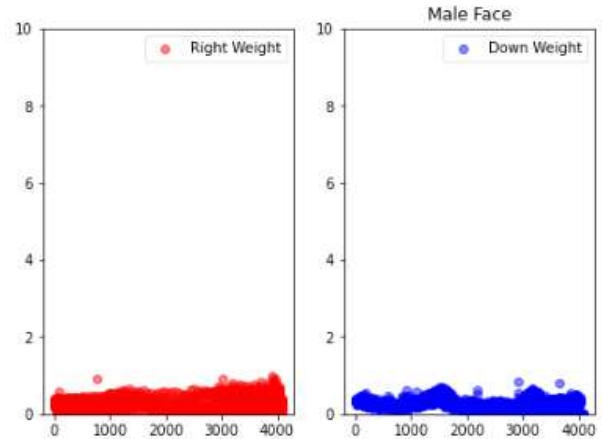
[그림 18. C standard deviation]

시각화를 해보니 예상외로 표준편차가 여러 곳에 분포해 있는 것을 볼 수 있었다. 그렇지만 산점도가 뭉쳐서 여러곳에 분포해있는것이 아닌 간격을 두고 떨어져있는, 노이즈 데이터로 판단할 수 있을 만한 산점도들이었다. 이 점을 숙지하고 다른 데이터셋들을 시각화해보았다.

그 다음으로는 약 75%로 좋지 못한 정확도를 보여준 데이터셋을 시각화해보았다.



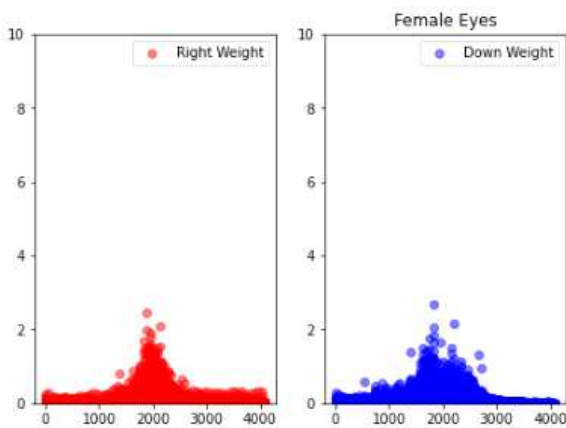
[그림19. female face standard deviation]



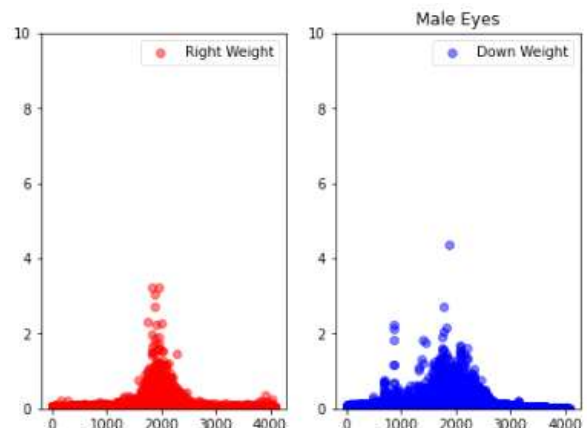
[그림20. male face standard deviation]

아래쪽 방향으로 가는 가중치에서 x축 1000 ~ 2000 부분에서 가중치가 뭉쳐서 위로 솟아오르는 형태를 보였다.

마지막으로는 55.6%로 제일 낮은 정확도를 보여준 데이터셋을 시각화하였다.



[그림21. female eyes standard deviation]



[그림22. male eyes standard deviation]

x축 1000 ~ 3000 부분이 뭉쳐서 위로 솟아오르는 모습을 보여주면서, 예상대로 표준편차가 원만하게 아래쪽에 분포해있지 않은 모습을 볼 수 있었다.

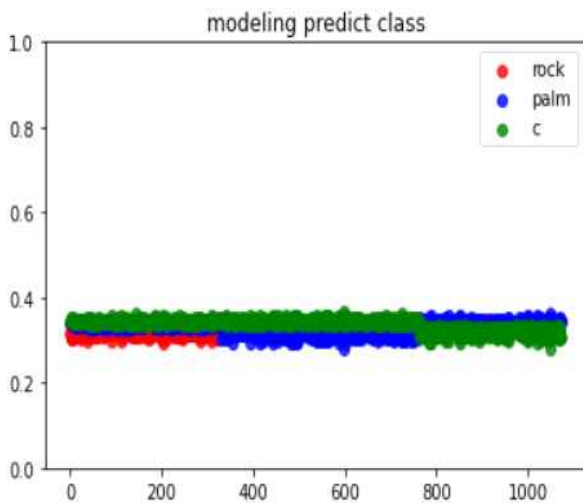
그렇게 세 번의 시각화를 하여 다음과 같이 결론을 지었다.

- 표준편차를 시각화하였을 때 노이즈 데이터를 제외한 나머지 부분이 비교적 원만하게 아래쪽에 분포해 있을수록 좋은 정확도를 보여준다.

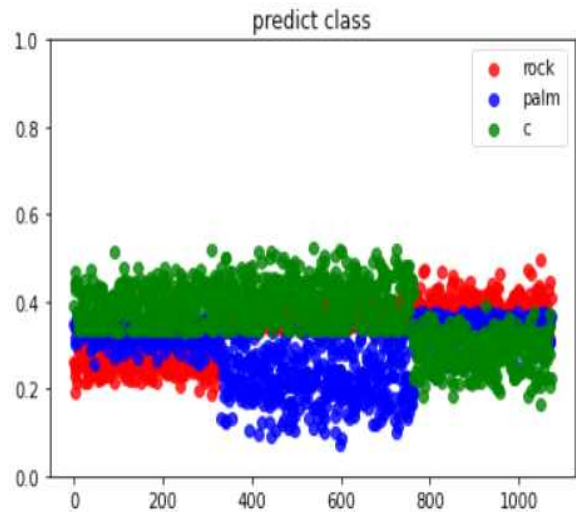
(3) 예측 방법에 따른 장단점

- CVFW 기술에는 총 두 가지의 예측 방법이 있었다. 첫번째 예측 방법은 학습된 이미지의 FW를 이용한 방법(이하 'predict class')이며, 두번째 예측 방법은 특징 모델링된 데이터를 유사도를 확인하여 이미지를 예측하는 방법(이하 'modeling predict class')이다. 성능 분석을 했을 때에 두 예측 방법의 차이가 있었고, 본 소제목에서는 이를 시각화하여 분석해보았다. 분석으로 사용한 데이터셋은 손 모양 구별 데이터셋이었다.

1. cost 값의 분포 비교



[그림23. modeling predict class cost]



[그림24. predict class cost]

- $x = a$ 일때의 데이터들은 a번째의 이미지로 예측했을때의 cost 들이다.

빨간색 산점도는 rock 클래스의 산점도이고, 파란색 산점도는 palm 클래스, 초록색 산점도는 c 클래스이다. 먼저 나는 최종 예측 클래스가 0에 가까울수록 안정성이 있는 모델이라고 판단했다. 왼쪽의 modeling predict class 그래프를 보면 같은 x에서의 세 산점도가 전부 비슷한 위치에 있다는걸 알 수 있다. 세 산점도가 전부 비슷한 위치에 있다는 것은 최종 예측 클래스가 0에 많이 가깝지 않다는 의미로 안정성이 많이 떨어진다고 판단하였다. 그 반면 오른쪽의 predict class 그래프를 보면 같은 x에서의 세 산점도가 modeling predict class에 비해 많이 떨어져있는 모습을 볼 수 있다. 이에 안정성은 predict class가 더 높다고 판단하였다. 하지만 예측 속도는 modeling predict class가 약 1.53배(학습 이미지마다 다름) 더 빨랐으며, 학습속도 면에서 modeling predict class는 FGN과 FWN의 영향을 받지 않고 항상 일정하여 따로 수정을 해줄 부분이 없기 때문에 훨씬 빨랐다.

- predict class 장단점

장점: 안정성이 modeling predict class에 비해 많이 높다.

단점: 여러번의 FGN, FWN 수정으로 학습 속도가 많이 느려질 수 있고, 예측 속도 또한 modeling predict class에 비해 느리다.

- modeling predict class 장단점

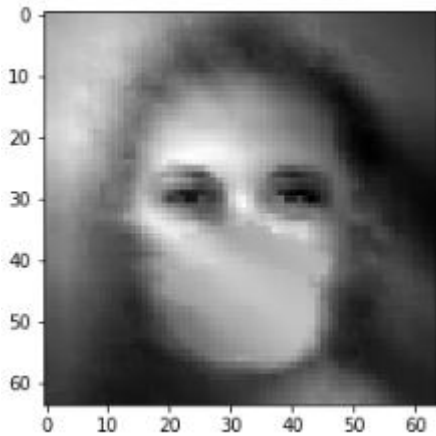
장점: 학습속도와 예측속도가 많이 빨라 실시간 예측 같은 상황에서 용이하게 사용될 수 있다.

단점: 안정성이 많이 떨어져서 실제 적용했을 때 예상과 벗어나는 결과를 불러일으킬 수 있다.

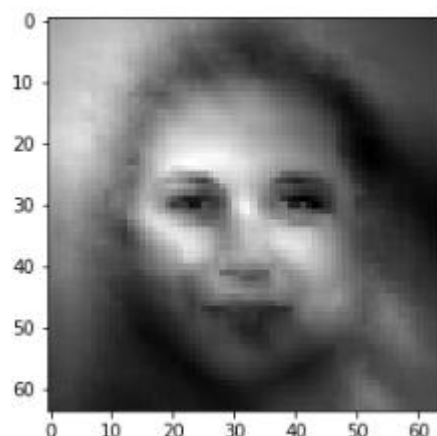
6. 실용 예시

- 실용 예시로는 실시간으로 마스크 착용 여부를 확인하는 모델을 만들었다. 요즘 코로나 시대에 용이하게 쓰일 수 있는 프로그램이라고 생각하였기 때문이다. 실시간으로 예측을 하는 것이어서 예측 방법은 modeling predict class 방법을 이용하였다. 학습 데이터의 개수는 6001개였다.

- 특징 모델링: 학습된 모델에서 클래스들을 모델링한다.



[그림25. with mask modeling]



[그림26. without mask modeling]

왼쪽은 마스크를 쓴 with mask 클래스의 특징 모델링이고, 오른쪽은 마스크를 쓰지 않은 without mask 클래스의 특징 모델링이다. 원하는대로 왼쪽은 마스크를 썼고, 오른쪽은 마스크를 쓰지 않은 모습을 확인할 수 있다.

- 실제 모델을 실시간 마스크 착용 여부 프로그램에 적용

먼저 학습된 모델을 modeling predict class 방법을 이용하여 정확도를 측정했을 때 약 98.6% 로 상당히 높은 정확도를 보여주었다. 하지만 modeling predict class의 안정성 문제로 실제 적용을 했을 때는 98.6%보다 낮을 것이라고 예상하였다.

마스크 착용 여부 프로그램은 실시간 캠 화면 안에 네모 박스가 있고, 그 네모 박스 안에 맞게 얼굴을 위치시키면 마스크 착용 여부에 따라 네모 박스의 테두리 색깔이 변한다. 검정색일 때는 마스크를 쓰지 않은 상황, 하얀색일 때는 마스크를 쓴 상황으로 설계하였다.



[그림27. without mask test]



[그림28. with mask test]

얼굴의 위치를 네모 박스 안에 조금이라도 잘 맞추지 못할 경우에는 정확도가 현저하게 떨어졌지만, 얼굴의 위치를 정확하게 맞추었을 때는 98.6%의 정확도라고 할 수 있을만큼의 정확도를 보여주었다. 또한 예측 속도가 빠른 덕에 실시간으로 예측함에도 불구하고 영상이 끊기는 현상은 크게 신경 쓰지 않아도 될 정도였다. 내가 생각한 실제 활용 가능한 예시는 마스크를 써야하는 건물의 입구이다. 마스크를 쓰고 있다고 예측되면 건물 출입이 가능하고, 마스크를 쓰지 않고 있다고 판단되면 건물 출입이 불가능한 상태가 되는 것이다.

실제 위의 상황에 활용되었을 때의 환경과 비슷하게 하기 위해 화면 왼쪽에서 걸어와 마스크 여부 판단을 예측했을 때는 대부분 잘 예측하였다.

7. 탐구 소감

- 개인적으로 이번 탐구는 정말 재미있었고, 의미있었던 탐구였다. 이번 탐구의 개발 코드와 자료들은 오픈소스로 공개할 예정이며, 앞으로 여러 사람들이 내 아이디어를 빌리거나, 덧붙여 여러 발전이 이루어졌으면 하는 바람이다. 그리고 이미지 예측은 이미 딥러닝 분야에서 많이 개발되었고 발전된 분야이다. 하지만 직접적인 이미지의 특징을 추출하는 기술은 본 적이 없었다. 그런 상황에서 나는 내 아이디어를 직접 탐구하고 개발하여 실현시켜 각기 다른 이미지 안에서의 특징추출을 성공했을 때 정말 뿌듯했다. 또한 그 성능도 뛰어났다고 생각한다. 탐구하는 기간동안 특징 모델링을 처음 성공했을 때는 나도 모르게 눈물이 떨어졌다. 그 눈물에는 33%는 뿌듯함, 33%는 성취감, 남은 34%는 그동안 탐구하며 열심히 노력한 스스로에게 수고했다는 의미를 담고 있었다. 본 탐구에서는 직접적인 지식에 관해 배운 점도 많지만, 탐구 경험을 늘리는 점에서도 굉장히 의미있었고, 많이 성장하였다. 한달이 약간 넘는 기간동안 매일 같은 주제에 대해 생각하며 아이디어를 떠올리고, 탐구한다는 것은 정말 흥미로운 일이었다. 본 탐구는 오랫동안 기억될 것이다.

8. 참고 문헌

(1) 이용한 데이터셋

1. hand sign - <https://www.kaggle.com/ash2703/handsignimages?select=Test>
2. male, female face - <https://www.kaggle.com/cashutosh/gender-classification-dataset>
3. male, female eyes - <https://www.kaggle.com/pavelbiz/eyes-rtte?select=maleeyes>
4. face mask - <https://www.kaggle.com/prasoonkottarathil/face-mask-lite-dataset>

(2) 사용한 패키지

Cv2 - John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), [DOI:10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) ([publisher link](#))

Matplotlib - John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), [DOI:10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) ([publisher link](#))

Numpy - Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357 - 362 (2020). DOI: [0.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). ([Publisher link](#)).

tqdm - <https://github.com/tqdm/tqdm>

9. 부록

(1) 용어와 정의

- CVFW: 이미지의 픽셀 간의 관계를 확인하여 특징을 추출하고, 이미지를 학습하는 새로운 방법의 컴퓨터 비전 분야 알고리즘으로 Computer Vision Feature Weight의 약자이다. 이 알고리즘은 본 탐구에서 새로 개발된 알고리즘의 이름이다.

(2) 개발 테스트 코드

- <https://github.com/SimplePro/CVFW>