

<https://doi.org/10.7236/JIIBC.2022.22.1.163>

JIIBC 2022-1-23

가중치 합산 기반 안면인식 특징점 저장 알고리즘 연구

Study on Weight Summation Storage Algorithm of Facial Recognition Landmark

조성욱*, 유영균*, 광광진**, 박정민***

Seonguk Jo*, Youngkyon You*, Kwangjin Kwak**, Jeong-Min Park***

요약 본 논문은 실생활 속 정제되지 않은 입력으로 인해 안면의 특징점을 추출하여 객체 인식 모델의 이상적인 성능과 속도를 보증해주지 못하는 문제점을 가중치 합산을 통한 저장 알고리즘을 통해 개선하는 방법을 소개한다. 많은 안면인식 프로세스들은 이상적인 상황에서의 정확도를 보장해주지만, 실생활에서 발생할 수 있는 수많은 방해요인에 대해서는 대처하지 못한다는 문제점이 주목받고 있으며 이는 곧 보안과 밀접하게 관련된 안면인식 프로세스에서 심각한 문제를 발생시킬 수도 있다. 본 논문에서는 사람의 고유한 특징점은 사진의 구도 등의 여러 변수가 있어도 결국 평균적인 하나의 형태를 띠는 점을 이용하여 입력으로 추출된 특징점을 여러 방해 요인에 과적합 되어있지 않은 소수의 특징점과 비교를 통해 실시간으로 빠르게, 그리고 정확하게 안면인식을 할 수 있는 방법을 가중치 합산방식을 통하여 제시한다.

Abstract This paper introduces a method of extracting facial features due to unrefined inputs in real life and improving the problem of not guaranteeing the ideal performance and speed of the object recognition model through a storage algorithm through weight summation. Many facial recognition processes ensure accuracy in ideal situations, but the problem of not being able to cope with numerous biases that can occur in real life is drawing attention, which may soon lead to serious problems in the face recognition process closely related to security. This paper presents a method of quickly and accurately recognizing faces in real time by comparing feature points extracted as input with a small number of feature points that are not overfit to multiple biases, using that various variables such as picture composition eventually take an average form.

Key Words : Face Landmark Storage Algorithm, Face Recognition, Face Landmark Detection, Weighted Sum Algorithm, Object Detection

1. 서론

딥러닝이 두각을 드러내기 시작한 이래로 전 세계적으로 많은 연구가 진행되고 있다. 딥러닝을 이용한 객체 인

식 분야에서 화두가 되는 안면인식 또한 마찬가지로 상당히 많은 연구가 진행되고 있다.

현시대의 안면인식 알고리즘은 객체 인식 모델로부터 추출된 얼굴의 특징점에 대한 거릿값을 측정하는 것으로

*준회원, 한국산업기술대학교 컴퓨터공학부

**준회원, 한국산업기술대학교 자율제어연구소

***정회원, 한국산업기술대학교 컴퓨터공학과(교신저자)

접수일자 2022년 1월 21일, 수정완료 2022년 1월 31일
게재확정일자 2022년 2월 4일

Received: 21 January, 2022 / Revised: 31 January, 2022 /

Accepted: 4 February, 2022

*Corresponding Author: jmpark@kpu.ac.kr

Dept. of computer engineering, Korea Polytechnic University, Korea

[1], 대부분의 연구는 이 특징점을 효율적으로 추출해내는 방법론에 관한 연구들이 진행되는 추세이다. 대표적인 연구로 오직 정면에서 촬영된 사진만 정상적인 특징점을 추출할 수 있었던 문제점을 보완하여 측면과 같은 정제되지 않은 이미지에서도 어느 정도 특징점을 추출할 수 있는 'Retina Face'^[2]와 같은 연구가 있다.

'Retina Face'에서 '야생적인 사진'이라고 표현된 정제되지 않은 사진 내에서 최대한 정형화된 특징점 값을 추출하려는 연구가 많았지만, 입력 자체에 '야생적인 사진'이라는 범주를 벗어났을 때 특징점이 제대로 추출되지 않는다. 특히 머리카락, 안경과 같이 특징점 일부를 가리는 방해요인이 있을 때는 예상에서 크게 벗어난 특징점 값을 추출하는 문제점이 있었다.

이런 사진들은 전체적으로 봤을 때는 적은 비중을 띠고 있다. 하지만 안면 인식은 기존의 특징점과 새로운 특징점을 비교하는 프로세스로, 만약 예상에서 크게 벗어난 특징점이 기존의 특징점으로 저장된다면, 새로운 특징점들은 높은 정확도를 가진 특징점이어도 안면 인식이 되지 않는 결과를 보일 것이다.

그럼에도 불구하고 안면인식은 본인 인증과 같은 높은 수준의 보안이 필요한 상황에서 사용할 일이 많다. 결론적으로 여러 가지 방해요인이 있을 실생활에서 모델뿐만 아니라, 입력으로부터 모델에서 추출된 특징점을 비교할 데이터베이스 속 특징점에 대한 저장 방식 또한 중요하다는 점을 강조한다.

정리해본다면 모델의 정확도와 최적화 측면에서는 많은 발전을 이루고 있다. 그에 반해 모델을 배포하고 사용할 때 기존에 저장된 어떤 사진과 비교할 것인가에 대해서는 비교적 개발자의 재량으로 남아있는 한계점이 있다.

만약 얼굴인식을 위한 알고리즘이 $O(n)$ 이라고 가정하였을 때, 기존의 특징점을 정제하지 않은 처음의 한 특징점으로부터만 한다면, 시간 복잡도는 $O(n)$ 를 보장하겠지만 여러 가지 방해요인이 있을 상황에서 기존의 특징점이 훼손된 상황이라면 정확한 결과를 추출하지 못한다는 단점이 있다. 반면 입력받는 모든 특징점을 추후 비교를 위한 기존의 특징점으로 삼는다면 시간 복잡도는 비교할 사진 개수를 m 이라고 가정할 때 시간복잡도는 $O(n*m)$ 까지 올라갈 수 있다는 단점이 있다. 또한 비교할 기존의 특징점들이 너무 많아져 처음 보는 사람이 입력으로 들어왔을 때 다른 사람으로 인식할 수도 있다는 단점 또한 존재한다.

이런 문제점들을 해결하기 위해 본 논문은 사람의 고

유한 특징점은 여러 변수가 있어도 결국 평균적인 하나의 형태를 보인다는 특징을 이용하여 기존의 특징점과 입력받는 특징점에 적절한 가중치를 포함하여 더하는 정제방식을 사용한다. 본 논문이 제시하는 방법을 이용한다면 시간 복잡도는 $O(n)$ 와 가깝게 유지하며 낮은 사람이 들어왔을 때 잘못 인식하는 치명적인 문제를 해결하는데 도움을 줄 수 있어 실시간 안면인식 프로세스에서 모델의 속도와 정확도를 보장하려는 방법을 제시한다.

II. 관련 연구

본 장에서는 해당 연구에 필요한 기본 기술 및 개념인 안면 속 특징점 추출 방법과 추출된 특징점을 통한 인물 판별 방법, 그리고 가중치 합산 방식에 대하여 소개한다.

1. Face Landmark Detection

핵심적인 구조는 입력받은 이미지를 미리 학습된 첫 번째 Convolution Neural Network 모델을 거쳐 물체의 Bounding Box를 추출한 다음, 추출한 Box에 대한 인식을 진행하는 Key Point Detection 프로세스이다.^[3,4]

다만 모델을 만들 때 사용된 대부분의 데이터 세트가 서양인의 안면으로 학습됐다는 점과 사진을 촬영할 때의 각도 또는 표정에 따라서도 바뀔 수 있다는 점, 그리고 실시간 영상인식의 특성상 빠르고 정확하게 도출해내야 한다는 점을 미루어봤을 때 실생활에서 불특정 다수를 위한 얼굴인식을 적용하기 위한 프로세스 내에서 모델만으로는 요구사항을 실시간적으로 대처하기에 어려울 수 있다.

2. L2 Norm

새로운 입력 x 와 기존의 데이터베이스에 있는 n 개의 y 에 대하여 특징점 각각의 거리에 대한 제곱의 평균값을 나타낸다.

$$\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}} \quad (1)$$

위 수식 (1) 은 기존 데이터베이스에 있는 특징점과 새로 입력된 특징점 사이의 거리를 측정할 때 사용되어 연산의 결과값이 작을수록 유사도가 높음을 이용한 비교 방법이다.

3. 가중치 합산 방식

데이터의 균형을 맞추어 주자 사용되는 방식이다.^[5] 예를 들어 100개의 데이터 중 A라는 라벨이 100개가 있고, B라는 라벨이 20개가 있다고 할 때 이를 그대로 혼련한다고 하면 모델은 자연스럽게 80개의 데이터가 있는 A라는 라벨에 대하여 과적합이 일어날 가능성이 크다. 이 문제를 방지하기 위하여 A에는 한 번 학습할 때 0.2만큼만 편향되게 가중치를 주게 된다면 모델은 이상적으로 A와 B를 학습할 수 있다는 이론이며 이런 가중치 합산 방식을 특징점 사이의 거리와 접목하는 것이 본 논문의 주제이다.

III. 시스템 및 알고리즘 프로세스

본 장에서는 해당 알고리즘이 적용된 시스템에서의 안면인식 특징점의 가중치 합산 방식에 대한 프로세스를 차례대로 설명한다.

1. 시스템 프로세스

〈Step 1〉 Image Preprocessing

실생활에서 촬영되는 영상에서 특징점을 추출하기 위해 영상 내 안면을 추출해야 한다.

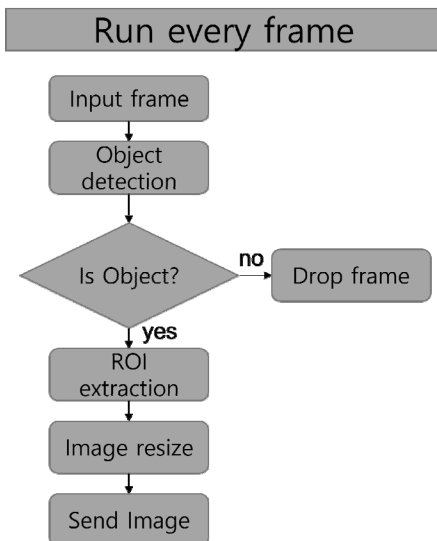


그림 1. 영상 내 안면 추출 순서도
 Fig. 1. Flowchart of facial extraction in video

그림 1의 순서도에서 보는 바와 같이 촬영된 영상을 입력값으로 받게 되면 Resnet18^[6]과 FPN^[7]이 적용된 Object Detection 모델을 통과시킨다. 그 뒤 영상 내에서 ROI(Region Of Interest)를 탐지한다. 그 다음 얼굴인식을 위한 모델은 입력 이미지 사이즈가 224x224로 고정되어있기 때문에 RoI로 추출된 이미지를 224x224로 축소 혹은 확대하여 얼굴인식을 위한 서버로 전송한다.

〈Step 2〉 Face Preprocessing

실시간 영상, 특히 마스크나 모자 등을 착용하는 사람들이 포함된 영상의 특성상, 정상적으로 특징점을 추출할 수 있는 입력이 들어올 확률이 높지 않다. 따라서 얼굴인식에 적합한 영상인지를 판별하는 Pre Recognition 과정이 필요하다.

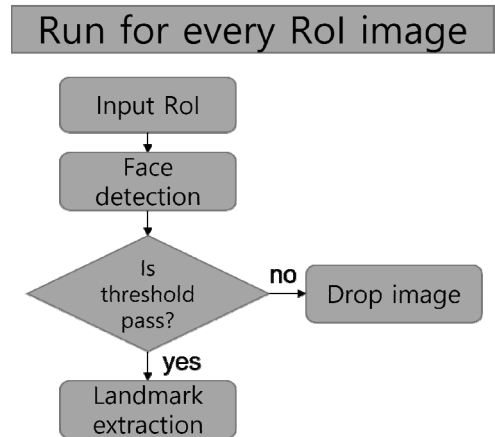


그림 2. Rol 이미지 검증 순서도
 Fig. 2. Rol Image Verification Flowchart

그림 2의 순서도에서 보는 바와 같이 임계점에 따라서 특징점을 필터링 하는데, 그 이유는 예를 들면 다음과 같다. 모델은 특징점과 정확도를 반환하는데, 정확도가 30%라고 생각하면 이는 정확한 특징점이라고 볼 수 없다. 따라서 정확도는 주관적인 기준으로 정해야하지만, 본 논문이 사용된 시스템에서는 80%의 정확도를 설정했을 때 가장 잘 결과를 도출해냈다.

〈Step 3〉 Wild Landmark Filter

위의 RetinaNet에서도 언급했듯이 '야생의' 특징점들을 잘 걸러내야 한다.



그림 3. 특징점 필터 순서도
Fig. 3. Flowchart of landmark filter

그림 3의 순서도에서 보는 바와 같이 첫 번째 분기에서 임시적으로 저장된 특징점들을 L2-Norm을 통하여 거리를 측정한다. 그 다음 거리가 임계값 이하일 때, 즉 두 특징점이 ‘비슷하다’ 라는 수준이 되었을 때 특정 조건에 맞는 특징점들을 얼굴인식 전 최종 특징점으로 결정한다.

두 번째 조건식에서 충분한 특징점들이 모였는가 검사하는데, 본 논문에서는 5개로 설정하였을 때 가장 높은 성능을 보였다.

〈Step 4〉 Face Recognition by Weighted Sum

Step1~3을, 즉 전처리를 거친 특징점을 가지고 본 논문의 핵심 주제인 가중치 합산 알고리즘을 이용한다. 이는 다음 알고리즘 구조에서 자세하게 서술한다.

2. 알고리즘 구조

시스템 프로세스의 Step 4부분을 보강하는 부분이며 본 논문의 핵심인 가중치 합산방식 저장 알고리즘을 설명한다.

〈Step 1〉 Warm-Up

Bag of Tricks for Image Classification with Convolutional Neural Networks^[8]의 관점에 따르면 일정 시행 n회전 평균적인 특징점이 형성되기 전에는 결

과가 입력에 대하여 민감하다는 취약점이 있다. 따라서 n회 이전의 기간 동안 입력받은 특징점 간의 거릿값을 계산하여 특정 임계값 t보다 적은 거리에 있는 특징점 간에는 평균 연산을 하고, 임계값 보다 먼 거리에 있는 특징점 간에는 각각의 노드를 만들어 최대 n개의 노드를 저장한다.

Algorithm 1: Warm-up

```

Result: Landmark nodes list
nodeList ← Null;
condition ← True;
while condition do
    landmark ← input;
    for nodelist do
        if L2Norm(node, landmark) < k then
            node.count ← node.count + 1;
            node ← (node + landmark)/node.count;
        else
            nodeList.append(landmark);
        end
    end
end
  
```

그림 4. Warm-up 알고리즘 의사코드
Fig. 4. Pseudo code of Warm-up Algorithm

특정 시행 n 전까지 수행하는 Warm-Up 과정으로 입력받은 Landmark들로 노드를 만든다. 그 다음 생성된 노드들을 list로 만들어서 저장한다. 이 과정에서 비슷한 거리에 있는 노드들은 합치는 과정을 거친다. 이 과정에서 합성 방법은 단순히 평균치를 구하는 방식으로 구한다. 본 논문에서는 특정 시행 n을 10회로 설정하였다.

〈Step 2〉 Online Learning

$$w = \frac{1}{Count_{node} + 1} \tag{2}$$

위 수식 (2)는 간단하게 가중치를 설정하는 식이다.

$$wLandmark = w * landmark_{input} + (1 - w) * landmark_{node} \tag{3}$$

위 수식 (3)은 이를 이용하여 가중치가 부여된 특징점을 구하는 식이다.

Algorithm 2: Online Learning

```

Result: Weighted Landmark nodes list
best ← Null;
bestNode ← Null;
landmark ← input;
for node in nodelist do
    distance ← L2Norm(node, landmark);
    if (distance < k) and (distance < best) then
        best ← distance;
        bestNode ← node;
    end
end
if best == Null then
    return;
end
w ← 1/bestNode.count;
landmark ← (1 - w) * bestNode.landmark + w * landmark;
bestNode.Landmark ← landmark;
    
```

그림 5. Online Learning 알고리즘 의사코드
 Fig. 5. Pseudo code of Online Learning Algorithm

그림 5의 알고리즘 프로세스의 4번째 줄 반복문은 가장 가까운 노드를 찾는 과정이다. 이 과정중에 k값을 거치는데, 이는 임계점 값으로, 상황마다 다르지만 본 논문은 구현한 프로세스에서는 0.3으로 설정했다.

결과적으로 bestNode를 추출하고, 만약 임계점을 넘지 못했다면 그 상태로 알고리즘을 종료한다.

bestNode가 정해졌다는 가정 하에 14번째 줄을 시행하는데, 이 계산은 수식 (2)와 수식 (3)을 참고한다.

해당 과정을 거쳐 좀 더 강력하고 응집력 있는 노드들의 리스트가 완성된다.

〈Step 3〉 Success Rate-based Weight Storage

이 과정은 선택적으로 수행하는 단계이다. 그 이유는 바로 아래 후술하겠지만, 모든 노드들을 가중치를 더해서 합친다. 이로 인해서 얻을 수 있는 장점은 처음 보는 사람을 오해할 가능성이 줄어든다는 점이고, 그에 따른 단점은 이미 데이터베이스 내에 존재하는 사람을 예측할 때 예상치 못한 야생적인 입력이 들어왔을 때 정확도가 낮다는 점이다. 방법은 아래와 같다.

Step 1과 2를 수행하면 최대 n개의 노드와 각 노드에 대한 호출 횟수들이 존재한다.

$$w = \frac{Count_{node}}{Count_{all}} \quad (4)$$

수식 (4)는 모든 노드에 대하여 각 노드에 대한 가중치를 설정하는 수식이다.

Algorithm 3: Success Rate-based Weight Storage

```

Result: Normalized Landmark
countList ← Null;
allCount ← 0;
for node in nodelist do
    allCount ← allCount + node.count;;
    landmarkList.append(node.count);
end
finalLandmark ← 0;
for node, count in nodelist, countList do
    weightedNode ← node.weight * (count/allCount);
    finalLandmark ← finalLandmark + weightedNode.landmark
end
    
```

그림 6. Success Rate-based Weight Storage 알고리즘 의사코드
 Fig. 6. Pseudo code of Success Rate-based Weight Storage

그림6의 3번째 줄은 수식 (4)를 기반으로 가중치를 설정하는 과정이다. 다음 모든 노드에 각각의 가중치를 곱한 뒤 하나의 특징점으로 정제한다. 이렇게 한 사람의 입력에 대한 평균적인 특징점이 생성된다.

해당 과정을 모두 거친 뒤에는 Step 2를 적용하여 특징점을 갱신하는 방식으로 학습해도 되고, 이미 충분하다고 판단되면 학습을 중단하여도 된다.

이 과정은 Step 2로부터 충분히 많은 시행이 일어난 뒤에 진행해야한다. 본 논문에서는 Step 2에서 100회 이상의 시행이 일어난 뒤 진행했다.

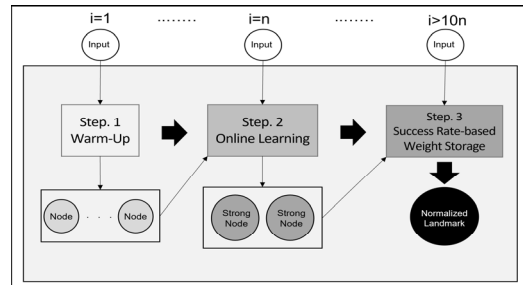


그림 7. 알고리즘 프로세스 시각화
 Fig. 7. Algorithm process visualization

그림 7은 최종적으로 본 논문의 알고리즘을 시각화했다.

IV. 실험 및 결과

본 장에서는 1) 특징점 전처리를 했을 때와 안했을 때의 정확도와 처리속도, 2) 한 가지 특징점으로 비교하는

경우, 알고리즘을 적용하여 비교하는 경우, 모든 특징점을 비교하는 경우 세 가지의 정확도와 처리속도에 대한 성능평가를 실행한다.

1. 성능 평가

가. 전처리 유무

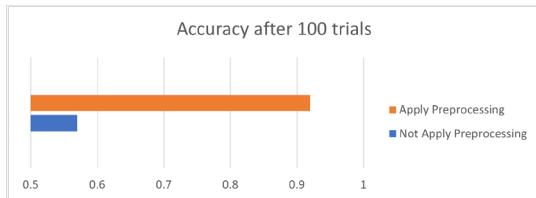


그림 8. 전처리 유무에 따른 100번 시행 후 정확도
Fig. 8. Accuracy after 100 trials with Preprocessing

그림 8은 전처리 유무에 따른 정확도를 비교한 것이다. 이는 본 논문의 알고리즘을 적용한 상태이며, 적용을 했음에도 불구하고 전처리를 하지 않은 상태에서는 57%로 낮은 정확도가 나왔고, 적용을 했을 때는 92%의 비교적 높은 정확도가 나왔다.

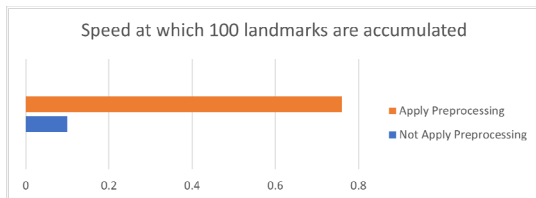


그림 9. 전처리 유무에 따른 100개의 특징점 축적 속도
Fig. 9. Speed at which landmarks are accumulated with Preprocessing

그림 9는 전처리 유무에 따른 100개의 특징점이 축적되는 속도를 비교한 것이다. 전처리를 하지 않는 것이 7~8배정도 빠른 속도를 가졌지만, 앞서 서술한 듯이 낮은 정확도를 보였다.

나. 세 가지 상황에 대한 비교

비교를 하기 전, 한 사람에 대한 100개가 넘는 사진을 구하기가 힘들어 비교 대상에 대한 표본이 10명 남짓으로 부족하다. 이는 본 논문의 취약점이며 수천, 수만의 데이터가 입력됐을 때의 정확도는 더 낮아질 것으로 예상된다. 인당 100장~120장의 입력이 무작위로 들어갔다.

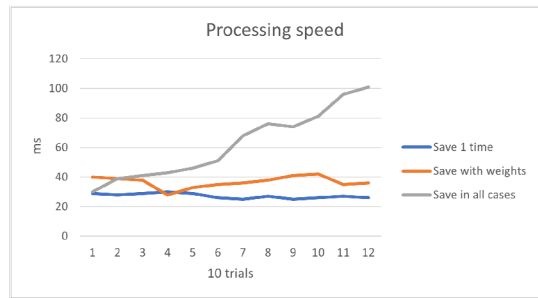


그림 10. 3 가지 경우의 처리속도
Fig. 10. Processing speed in 3 cases

그림 10은 입력으로 받은 사람들에 대하여 처음의 1회만 저장했을 때와, 가중치 저장 방식, 그리고 모든 경우를 저장한 경우에 대한 처리 속도의 평균을 나타낸 표이다. 처음 1회 저장방식과 가중치 저장방식은 35ms 부분에서 형성되었고, 모든 경우를 저장하는 경우는 선형적으로 계속 늘어나 100장이 넘어갔을 때 2.5배정도의 처리속도를 보였다. 더 많은 시행이 들어간다면 선형적으로 더 많이 상승할 것이라고 예상된다.

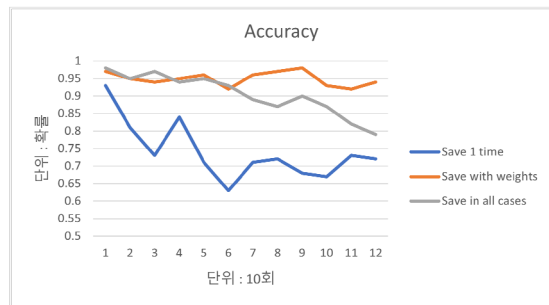


그림 11. 3가지 경우의 정확도
Fig. 11. Accuracy in 3 cases

그림 11은 입력으로 받은 사람들에 대하여 처음의 1회만 저장했을 때와, 가중치 저장 방식, 그리고 모든 경우를 저장한 경우에 대한 정확도의 평균을 나타낸 표이다. 1회만 저장하는 경우는 입력이 많아질수록 다른 사람으로 오해하는 경우가 많았다. 이 경우 때문에 실험이 지속될수록, 특히 아직 학습하지 않은 안면의 특징점일 경우 기하급수적으로 낮은 정확도를 보였다.

모든 경우를 저장했을 때에는 시행이 길어질수록 시간 복잡도도 높아지고, 다른 사람으로 오해하는 빈도가 급격하게 늘어났다.

반면 가중치를 저장했을 때에는 평균적인 값이 아닐 때에 잘 잡아내지 못하였지만 입력이 충분히 쌓인 상태

에서도 높은 정확도를 보여줬다. 이는 가중치 합산 알고리즘을 통하여 처음 보는 사람을 오해하는 빈도를 상당히 줄여주어 이 결과가 도출되었다고 추측한다.

V. 결 론

얼굴인식은 본인 인증과 같은 높은 수준의 보안이 필요한 상황에서 사용할 일이 많다. 촬영 각도, 모자, 마스크 등 여러 가지 방해 요인이 있을 실생활에서 이런 노이즈 때문에 보안과 관련된 치명적인 결함이 발생할 경우가 생길 가능성이 존재한다.

본 논문은 해당 문제를 다루기 위하여 사람의 고유한 특징점은 여러 변수가 있어도 결국 평균적인 하나의 형태를 띠는 점을 이용하여 입력받는 특징점에 가중치를 준 정제방식을 사용해 서술한 문제점을 해결하기 위한, 즉 가중치로 정제된 특징점으로 시간 복잡도는 처음 1회만 저장했을때와 가깝게 유지하며 기존의 방법들로 유지했을 때 가장 큰 문제점이었던 낯선 사람을 잘못 인식하는 치명적인 문제를 해결하려는 개선점을 통해 실시간 얼굴인식 프로세스에서 모델의 속도와 정확도를 보장하려는 방법을 제시한다.

그러나 본 논문에서 제시한 알고리즘은 평균적인 값을 너무 강건하게 유지하여, 입력으로 기준을 벗어난 이상치가 들어왔을 경우 반응을 하기 힘들다는 단점이 있다. 최종 특징점 노드를 여러 개로 구성하는 로직을 구현하였는데 이 경우 다른 방법들과 비슷하게 처음 보는 사람을 이미 데이터베이스 속인 누군가로 인식하는 경우가 잦았다. 전처리를 통하여 이를 보완하기도 했지만 그럼에도 불구하고 입력이 점점 많아지면서 완벽하지 못한 결과가 나오기도 했다. 추후 많은 입력에도, 그리고 평균값을 벗어난 입력에 대해서도 반응할 수 있는 저장 방식을 연구하고 개선해나갈 예정이다.

References

- [1] W. Mei, and W. Deng. "Deep face recognition: A survey." *Neurocomputing* Vol. 429, pp. 215-244, 2021. DOI: <https://doi.org/10.1016/j.neucom.2020.10.081>
- [2] Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., & Zafeiriou, S. "Retinaface: Single-stage dense face localisation in the wild." *arXiv* 1905.00641. 2019. DOI: <https://arxiv.org/abs/1905.00641>

- [3] S. J. Kim and J. J. Lee, "A Study on Face Recognition Using Directional Face Shape and SOFM." *The Journal of The Institute of Internet, Broadcasting and Communication*, Vol. 19, No. 6, pp.109-116, 2019. DOI: <https://doi.org/10.7236/IIBC.2019.19.6.109>
- [4] S., J., Kim, and J., J., Lee "A Study on Face Recognition using Support Vector Machine." *The Journal of The Institute of Internet, Broadcasting and Communication*, Vol.16, No.6, pp. 183-190, 2016. DOI : <https://doi.org/10.7236/IIBC.2016.16.6.183>
- [5] Marler, R. T., & Arora, J. S. "The weighted sum method for multi-objective optimization new insights." *Structural and multidisciplinary optimization*, Vol.41 No.6, pp. 853-862, 2010. DOI: <https://doi.org/10.1007/s00158-009-0460-7>
- [6] He, K., Zhang, X., Ren, S., & Sun, J. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016
- [7] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. "Feature pyramid networks for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125, 2017.
- [8] He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. "Bag of tricks for image classification with convolutional neural networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 558-567, 2019.

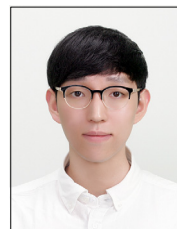
저 자 소 개

조 성 욱(준회원)



- 2015년 ~ 현재 : 한국산업기술대 컴퓨터공학과 학사과정
- 주관심분야 : Artificial Intelligence Research, Artificial Intelligence Engineering

유 영 균(준회원)



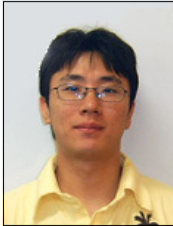
- 2016년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 학사과정
- 주관심분야 : Web Application, Data Analysis, Cloud Computing

곽 광 진(준회원)



- 2021년 ~ 현재 : 한국산업기술대학교 자율제어연구실 박사 후 과정
- 2021년 : 한국산업기술대학교 스마트팩토리융합과 박사
- 2016년 : 건국대학교 컴퓨터공학과 석사
- 2008년 : 건국대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, IoT, Smart Factory

박 정 민(정회원)



- 2014년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 부교수
- 2012년 ~ 2014년 : 한국전자통신연구원 선임연구원
- 2008년 ~ 2011년 : 동양미래대 조교수
- 2009년 : 성균관대학교 컴퓨터공학과 박사
- 2005년 : 성균관대학교 컴퓨터공학과 석사
- 2003년 : 한국산업기술대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, Software Engineering

※ “이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기본연구지원사업임 (NRF-2021R1F1A1063634)”
※ “본 연구는 2021년도 한국산업기술대학교 연구년 교수 연구비에 의해 연구되었음.”