

## 스크린 사용 여부 및 사용 디바이스 감지를 위한 머신러닝 모델 성능 비교

황상원<sup>1</sup> · 김동우<sup>1</sup> · 이주환<sup>1</sup> · 강승우<sup>2\*</sup>

### Performance Comparison of Machine Learning Models to Detect Screen Use and Devices

Sangwon Hwang<sup>1</sup> · Dongwoo Kim<sup>1</sup> · Juhwan Lee<sup>1</sup> · Seungwoo Kang<sup>2\*</sup>

<sup>1</sup>Graduate Student, Department of Computer Engineering, Graduate School, KOREATECH, Cheonan, 31253 Korea

<sup>2\*</sup>Assistant Professor, School of Computer Science and Engineering, KOREATECH, Cheonan, 31253 Korea

#### 요약

일상생활에서 디지털 스크린을 오랜 시간 사용하면 눈의 피로, 안구 건조, 두통 등 컴퓨터 시각 증후군을 경험하게 된다. 컴퓨터 시각 증후군을 예방하기 위해서는 스크린 사용 시간을 제한하고 수시로 휴식을 취하는 것이 중요하다. 최근 스마트폰에서는 스크린 사용 시간을 알 수 있도록 도와주는 다양한 애플리케이션이 존재한다. 하지만, 사용자는 스마트폰 스크린뿐만 아니라 데스크탑, 노트북, 태블릿 등 다양한 스크린을 보기 때문에 이러한 앱만으로는 한계가 있다. 본 논문에서는 color, IMU, lidar 센서 데이터를 이용하여, 사용 중인 스크린 디바이스를 감지하는 머신러닝 기반 모델을 제안하고 여러 가지 모델의 성능을 비교한다. 성능 비교 결과 신경망 기반 모델이 전통적인 머신러닝 모델보다 높은 F1 스코어를 보였다. 신경망 기반 모델에서는 MLP, CNN 기반 모델이 LSTM 기반 모델보다 높은 스코어를 보였으며, 전통적인 머신러닝 모델에서는 RF 모델이 가장 우수했으며, 다음으로는 SVM 모델이었다.

#### ABSTRACT

Long-term use of digital screens in daily life can lead to computer vision syndrome including symptoms such as eye strain, dry eyes, and headaches. To prevent computer vision syndrome, it is important to limit screen usage time and take frequent breaks. There are a variety of applications that can help users know the screen usage time. However, these apps are limited because users see various screens such as desktops, laptops, and tablets as well as smartphone screens. In this paper, we propose and evaluate machine learning-based models that detect the screen device in use using color, IMU and lidar sensor data. Our evaluation shows that neural network-based models show relatively high F1 scores compared to traditional machine learning models. Among neural network-based models, the MLP and CNN-based models have higher scores than the LSTM-based model. The RF model shows the best result among the traditional machine learning models, followed by the SVM model.

**키워드**: 컴퓨터 시각 증후군, 스크린 사용 감지, 스크린 디바이스 분류, 머신러닝

**Keywords**: Computer Vision Syndrome, Screen Use Detection, Screen Device Classification, Machine Learning

Received 29 January 2020, Revised 14 February 2020, Accepted 5 March 2020

\* Corresponding Author Seungwoo Kang(E-mail:swkang@koreatech.ac.kr, Tel:+82-41-560-1406)

Assistant Professor, School of Computer Science and Engineering, KOREATECH, Cheonan, 31253 Korea

Open Access <http://doi.org/10.6109/jkiice.2020.24.5.584>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서론

스마트폰, 컴퓨터 모니터, 태블릿 등 다양한 디지털 스크린을 사용하는 것은 매우 일상적인 행동으로 우리의 생활에서 떼려야 뗄 수 없는 일이다. 이러한 디지털 스크린을 오래 보는 것은 사람들의 건강상의 문제를 일으키기도 하는데 대표적인 것이 컴퓨터 시각 증후군이다 [1]. 컴퓨터 시각 증후군은 컴퓨터 모니터와 같은 디지털 스크린을 오래 보게 될 때 사용자에게 나타날 수 있는 다양한 눈 관련 질환을 종합적으로 이르는 말로서 이의 대표적인 증상은 눈이 침침해지거나 눈의 피로를 느끼며, 안구가 건조해지고 두통, 목 또는 등의 통증을 느끼는 것 등이 있다. 컴퓨터 시각 증후군 예방을 위해서는 스크린의 적절한 밝기와 주변 환경의 적절한 조명을 유지하는 것, 바른 자세로 스크린을 보는 것, 스크린을 보는 시간을 적절히 제한하는 것, 스크린을 보는 도중 수시로 스크린 보기를 멈추고 쉬는 것이 중요하다. 미국검안협회는 컴퓨터 시각 증후군을 완화하기 위한 방법 중 하나로 20-20-20 규칙을 지키기를 권장한다 [2]. 20-20-20 규칙은 스크린을 20분 동안 보면, 20초 이상을 20피트 떨어진 먼 곳을 보면서 눈이 휴식을 취할 수 있도록 해야 한다는 것이다.

스크린을 보는 시간을 제한하고 적절하게 휴식을 취할 수 있도록 하기 위해서는 사용자의 스크린 사용 여부를 감지하고 스크린 사용 시간을 알 수 있어야 한다. 최근 아이폰이나 안드로이드 스마트폰에는 스크린 타임 [3], 디지털 웰빙 [4]과 같은 앱을 통해 사용자의 스마트폰 사용 시간을 집계하여 보여주고 일정 사용 시간을 초과하면 스마트폰 사용을 제한할 수 있는 기능을 제공하고 있다. 하지만 디지털 웰빙은 안드로이드 스마트폰에만 국한되어 있어 다양한 종류의 스크린 디바이스를 사용하는 것이 일상적인 상황에서 정확하게 스크린 사용 여부를 감지하는 것에 한계가 있다. 스크린 타임의 경우에는 아이폰만이 아니라 아이패드, 맥 등 여러 종류의 디바이스에 종합적으로 적용이 가능하지만, 애플의 디바이스만 대상으로 한다는 점에서 한계가 있다.

본 논문에서는 다중 센서 융합 기반으로 스크린 사용 여부 및 사용 디바이스를 감지하는 방법을 제안한다. 기존 연구 [5-7]에서 초점을 둔 스크린 사용 감지만 아니라 사용하는 스크린 디바이스의 종류를 분류하는 것을 목적으로 한다. 이를 위해 본 논문에서는 다양한 머신

러닝 모델을 활용하여 스크린 디바이스 분류기를 학습하고 성능을 비교하였다. 여기서 사용된 모델은 총 6가지로 Support Vector Machine, Random Forest, k-Nearest Neighbors와 같은 전통적인 머신 러닝 모델과 Multi-Layer Perceptron, Convolutional Neural Network, Long Short-Term Memory와 같은 신경망 기반 모델이다.

F1 점수를 평가지표로 하여 10겹 교차 검증(10-fold cross validation)을 적용한 결과 전반적으로 신경망 기반 모델의 성능이 다른 모델의 성능보다 나은 결과를 보였다. 신경망 모델 중에서는 MLP 모델을 이용한 분류가 가장 높은 F1 점수를 보였다. 전통적 머신 러닝 모델 3가지 중에서는 RF 모델을 이용한 분류가 나머지 모델에 비해 우수하였고, k-NN 모델 기반 분류는 매우 낮은 점수를 보였다.

본 논문은 다음과 같이 구성되었다. 다음 섹션에서는 센서 기반 스크린 사용 감지 관련 연구와 센서 기반 행동 인식에 딥러닝 모델을 적용한 연구에 대해서 논의한다. 섹션 3에서는 스크린 사용 디바이스 분류를 위해 사용한 모델과 모델 학습을 위해 사용한 데이터에 대해서 기술한다. 섹션 4에서는 모델의 평가 결과를 소개하고 섹션 5에서 결론을 맺는다.

## II. 관련 연구

최근 여러 가지 센서를 이용하여 스크린 사용을 감지하는 방법을 제안한 연구가 있었다. 논문 [5]는 착용형 카메라를 사용하여 일상생활 중 디지털 스크린을 보는 지 감지하는 연구를 진행하였다. 카메라를 사용한 스크린 사용 탐지는 최근의 비전 인식 기술의 급속한 발전으로 정확하게 인식하기 수월하다는 장점이 있다. 하지만, 카메라로 영상을 녹화하는 것은 프라이버시 침해에 민감한 이슈가 있고, 이미지 센서를 이용한 지속적인 영상 캡처와 비전 처리는 연산 비용이 크기 때문에 일상생활에서 장시간 지속적인 사용이 어려운 단점이 있다.

논문 [6]은 스마트 안경에 컬러 센서를 부착하여 사용자의 스크린 사용 여부를 파악하는 시스템을 제안하였다. 하지만, 컬러 센서만을 사용해서 스크린 사용 여부를 탐지할 경우 IMU나 Lidar 센서를 부가적으로 더 사용하는 것에 비해 정확도가 떨어질 수 있다는 점이 이전 논문 [7]에서 실험적으로 확인되었다.

논문[7]에서는 IMU 센서, 컬러 센서, lidar 센서의 다중 센서 융합 기술과 SVM 분류기를 이용하여 사용자의 스크린 사용을 탐지하였다. 이를 통해 다양한 상황에서도 정확하게 사용자의 스크린 사용을 감지를 할 수 있었다. 하지만, 이 연구에서는 스크린 사용 여부만을 감지하는 것에 초점을 맞추어 어떤 종류의 스크린 디바이스를 사용하는지를 구분하는 것은 다루지 않았다.

기존의 센서 데이터를 이용한 사용자 행동 인식 기술을 제안하는 연구들은 특징 추출과 분류를 따로 진행하는 전통적인 머신 러닝 모델을 주로 사용하였다. 최근에는 딥러닝 모델 연구가 활발해지면서 센서를 통해 수집된 데이터에 딥러닝 모델을 적용하여 활용하는 사례가 증가하고 있다. 논문[8]과 [9]는 사용자의 다양한 행동을 인식하는 방법으로 CNN과 LSTM이 결합된 모델을 설계하고 구현하였다. 논문[10]은 CNN을 기반으로 한 행동 인식 모델을 만들고 F1 스코어로 모델을 평가하였다.

### III. 스크린 사용 디바이스 분류 모델

앞서 언급한 것처럼 일상생활 중에 다양한 스크린 디바이스를 사용하게 된다. 가장 쉽게 자주 접하는 디바이스는 스마트폰일 것이고, 컴퓨터로 업무를 수행하는 사람들의 경우 데스크탑이나 노트북 모니터도 많이 볼 것이다. 이 외에도 태블릿이나 텔레비전과 같은 디지털 스크린을 보는 경우도 있다.

본 논문에서는 사용자가 다양한 스크린 디바이스를 사용할 수 있는 상황에서 어떤 디바이스를 보고 있는지를 감지할 수 있는지, 그리고 어떤 모델이 이에 가장 좋은 성능을 보이는지 평가해보고자 한다. 여기서 대상으로 하는 스크린 디바이스는 데스크탑 모니터, 노트북 모니터, 스마트폰 총 3가지로 정하였다. 그리고 스크린을 보지 않는 상황을 따로 두었다. 따라서 본 논문에서 제시하는 스크린 사용 디바이스 분류 모델은 총 4개의 클래스를 구분하는 다중 클래스 분류 모델이 된다.

다중 클래스 분류 모델에 사용될 수 있는 다양한 머신 러닝 모델을 적용하여 그 성능을 비교해보기 위해 널리 사용되는 전통적인 머신 러닝 모델(SVM, RF, k-NN)과 신경망 기반 모델(MLP, CNN, LSTM)을 이용하여 분류기를 학습하였다. SVM, RF, k-NN 모델은 sklearn 기계 학습 라이브러리를 사용하였고, MLP, CNN, LSTM 모

델은 keras 신경망 라이브러리를 사용하였다. 각 모델에는 실험적으로 도출된 적절한 파라미터 값을 적용하였다. 아래에서 별다른 언급이 없는 파라미터 값은 디폴트 설정 값을 적용하였다.

#### 3.1. 전통적 머신 러닝 모델

##### 3.1.1. Support Vector Machine(SVM)

4차원 이상의 데이터 셋에 대해서 클래스 분류를 하기 위해서는 초평면 이상의 환경에서 분류가 필요하여, 분류하기가 상당히 어렵다. SVM은 고차원에서의 선형 경계를 저차원 환경에서 비선형 경계로 바꾸어서 클래스를 분류 할 수 있도록 학습하는 모델이다. 클래스 사이의 결정 경계를 확실히 하기 위해, 서포트 벡터라는 클래스 사이 경계에 위치한 데이터 포인트와 새로운 데이터 포인트 사이의 거리를 측정하여 클래스를 예측하도록 학습한다. SVM에서는 C와 gamma, 두 가지의 하이퍼 파라미터에 따라 성능에 차이를 보인다. 본 논문에서는 이전 논문[7]의 하이퍼 파라미터 값을 따라 C는 100, gamma는 0.001로 지정하였다.

##### 3.1.2. Random Forest(RF)

RF는 무작위의 특성을 가지는 여러 개의 결정 트리를 생성하고, 각 트리의 특성 중요도를 취합하여 멀티 클래스를 예측하도록 학습하는 모델이다. RF 모델을 통해 생성된 여러 개의 결정 트리는 각각 다른 특성을 통하여 학습이 진행되기 때문에, 다양한 가능성을 고려할 수 있고, 더 넓은 시각으로 데이터를 바라볼 수 있는 모델이다. RF 모델은 학습을 위해 필요한 결정 트리의 생성 개수를 지정해야 한다. 본 논문에서는 결정 트리 생성 개수에 대한 하이퍼 파라미터를 경험적으로 90개로 지정하였다.

##### 3.1.3. k-Nearest Neighbors(k-NN)

k-NN 모델은 가장 간단한 머신 러닝 모델 중 하나이다. k-NN 모델의 클래스 예측은 예측해야 할 데이터 포인트와 이웃 데이터 포인트 간 유클리디안 거리를 측정 한 후, 가장 가까운 k개의 이웃 데이터 포인트 중에서 가장 많이 차지하고 있는 클래스로 예측하게 된다. 본 논문에서는 k개의 이웃 수를 경험적으로 5로 지정하였다.

### 3.2. 신경망 기반 모델

#### 3.2.1. Multi-Layer Perceptron(MLP)

MLP는 분류 문제를 해결할 수 있는 간단한 신경망으로서 은닉층을 이용해서 데이터들의 특성을 찾아내어 가중치를 스스로 학습하는 모델이다. 신경망을 구성하는 것이 다른 모델들에 비해 간단하고 학습 시간, 실행 시간 역시 짧은 것이 장점이다.

이 논문에서 사용된 네트워크는 총 1개의 입력층 8개의 은닉층 1개의 출력층으로 구성되어있다. 활성화 함수로는 입력층과 가까운 5개 층은 ELU 함수를, 출력층과 가까운 3개 층은 ReLU 함수를 이용하였다. 다중 분류를 위해서 손실 함수는 categorical cross-entropy 함수를 사용하였다. 초기값은 he 초기값[11]을 사용하였다. adam optimizer을 사용하여 손실 함수의 최소값을 찾기 쉽도록 적용하였다.

#### 3.2.2. Convolution Neural Network

CNN은 최근 광범위하게 활용되는 딥러닝 모델로 주로 이미지 분류 혹은 이미지 생성 기법에서 많이 사용되는 모델이다. 최근 CNN은 이미지 관련 연구가 아닌 시계열 센서 데이터를 이용한 행동 인식 연구에서도 사용되고 있다. 일반적으로 DNN에 비해 연산량과 파라미터의 개수가 많지만 높은 정확도를 가지는 것이 장점이다.

본 연구에서 사용된 CNN 아키텍처는 convolution 계층 4개, pooling 계층 2개, dropout 계층 1개로 구성되었다. Convolution 계층의 필터 수는 입력층과 가까운 2개의 계층에서는 100개, 출력층과 가까운 2개의 계층에서는 160개, 커널의 크기는 4개의 계층 모두 10으로 설정하였다. Dropout 계층의 비율은 0.5로 지정하였다. MLP와는 다르게 활성화 함수는 모두 ReLU를 사용하였다. 기울기 감소 문제가 심하게 발생하지 않기 때문에 계산이 용이한 ReLU 함수를 이용하여 모델을 구성하였다. 오버피팅을 방지하기 위해 dropout 계층을 출력층 직전에 배치하였으며 2개의 convolution 계층 다음에는 pooling 계층을 사용하였다.

#### 3.2.3. Long Short-Term Memory(LSTM)

LSTM은 Recurrent Neural Network (RNN) 모델 중의 하나로 이전 상태의 결과가 다음 계층에 영향을 주는 구조로 되어있다. 순차적으로 들어오는 데이터 처리에 적합한 모델로 계속해서 들어오는 센서 데이터 처리에 높은 성능을 낼 수 있다.

본 연구에서 스크린 디바이스 분류를 위해 사용한 LSTM 네트워크는 2개의 LSTM 계층과 2개의 dropout 계층으로 이루어져 있고, 출력층 전 계층으로 완전 연결 계층을 추가하였다. dropout 계층을 추가하여 오버피팅을 방지하였다.

### 3.3. 데이터 셋

모델의 학습 및 평가를 위해 이전 연구 [7]에서 사용된 데이터 셋을 활용하였다. 이 데이터 셋은 10명의 참가자(남:9, 여:1)로부터 수집된 데이터로서 랩 환경에서 수집된 것이다. 여기에는 참가자가 다양한 조건에서 스크린을 사용하는 도중 총 3가지 센서에서 수집된 데이터가 포함되어 있다. 사용된 센서는 color 센서, IMU 센서, lidar 센서로 샘플링 레이트는 각각 26Hz, 80Hz, 66Hz이었다. 데이터 수집 조건으로는 다음과 같은 것이 있었다.

- **스크린 디바이스:** 참가자는 3가지 스크린 디바이스(데스크탑 모니터, 노트북 모니터, 스마트폰)를 사용하였다.
- **콘텐츠:** 스크린에서 참가자가 보는 콘텐츠로는 정적인 것과 동적인 것 두 가지가 있었다. 정적인 콘텐츠로는 뉴스나 커뮤니티 게시판을 보는 것이었고, 동적인 콘텐츠는 영화를 보거나 게임을 하는 것이었다.
- **조도:** 어두운 환경과 밝은 환경 두 가지 조건을 두었다.
- **스크린 미사용:** 스크린을 보지 않는 상황에서도 데이터가 수집되었는데, 여기에는 책을 읽는 것과 휴식을 취하는 것 두 가지가 있었다. 휴식을 취하는 중에는 디지털 스크린을 보는 것을 제외한 활동을 자유롭게 할 수 있었다.

데이터 수집은 총 16가지 시나리오(3(스크린 디바이스) x 2(콘텐츠) x 2(조도) + 2(스크린 미사용) x 2(조도) = 16)에 대해서 이루어졌다.

전통적인 머신 러닝 모델에서는 미가공(raw) 데이터에서 특징(feature) 데이터를 추출하여 이를 모델 학습에 사용하는 것이 일반적이다. 학습에 사용할 특징 데이터 추출을 위하여 본 논문에서는 이전 논문[7]에서 제시한 특징 추출과 차원 축소 방법을 사용하였다. 앞서 기술한 SVM, RF, k-NN 모델을 기반으로 스크린 사용 디바이스 분류기를 학습할 때는 이와 같은 특징 추출과 차원 축소 과정을 거쳐 준비된 데이터를 이용하여 학습이 이루어졌다.

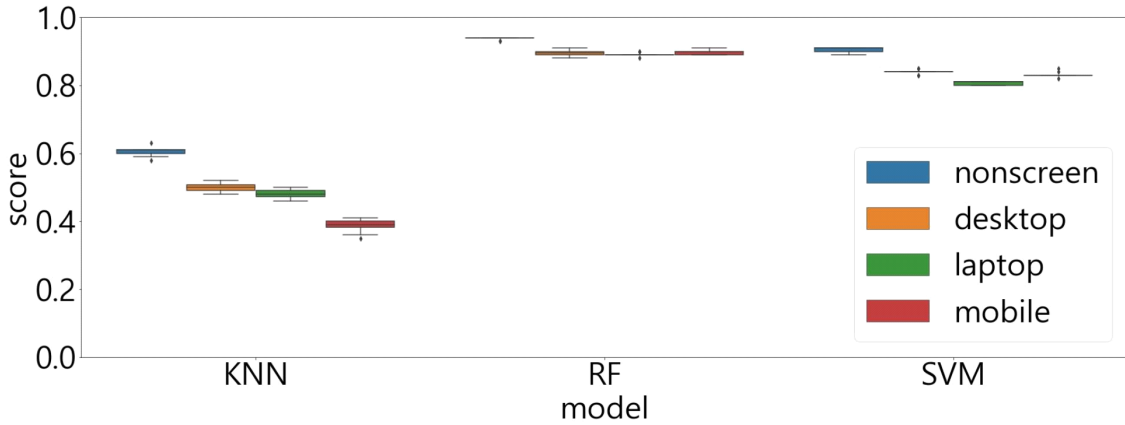


Fig. 1 The F1 score results of conventional machine learning models

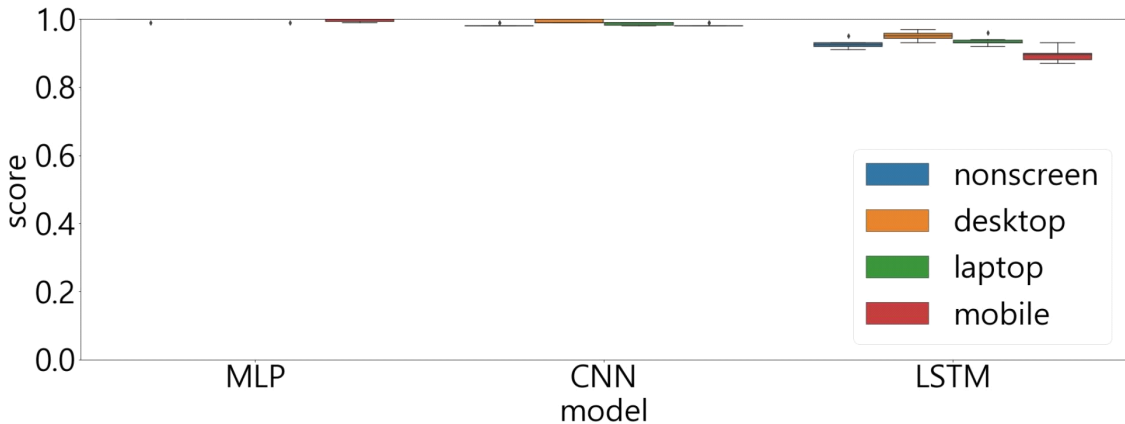


Fig. 2 The F1 score results of neural network-based models

신경망 기반 모델은 특징 추출과 차원 축소 과정이 없이 미가공 센서 데이터를 분류기를 학습하는데 그대로 사용하였다. 그런데 위에 언급한 것처럼 각 센서의 샘플링 레이트가 다르기 때문에 분류기의 입력으로 사용될 수 있도록 3가지 센서 데이터를 15Hz로 다운샘플링 하여 레이트를 동일하게 맞춰주었다.

#### IV. 평가

##### 4.1. 평가 방법

분류 모델 성능을 평가하기 위한 지표로는 F1 스코어를 이용하였다. F1 스코어는 분류 모델의 정확도를 나타내는 척도로서 사용되는 대표적인 평가지표 중 하나로,

정밀도(precision)와 재현율(recall)의 조화평균 값으로 계산된다. F1 스코어가 높을수록 성능이 높다고 할 수 있다.

이 논문에서 사용된 데이터 셋에 대한 분류 모델의 일반화 성능 평가를 위해, 널리 사용되는 교차 검증 방법의 하나인 10겹 교차 검증(10-fold cross validation) 방법을 이용하였다. 이를 위해 전체 데이터 셋을 임의적으로 섞은 후 10개의 세트로 분할한다. 분할한 10개의 세트 중 9개는 학습 데이터로 사용하고, 나머지 1개는 테스트 데이터로 사용하여 평가를 진행한다. 이 과정을 중복되지 않게 총 10번(fold) 진행한다. 각 fold마다 테스트 데이터에 의해 테스트 된 결과가 F1 스코어로 나오게 된다. 아래 결과에서는 각 모델의 F1 스코어를 박스 플롯으로 도식화하여 나타내고 10 fold의 평균값을 같이 보

**Table. 1** The F1 score results of conventional machine learning models

	Nonscreen	Desktop	Laptop	Mobile	Average
k-NN	0.605	0.498	0.481	0.387	0.492
RF	0.938	0.894	0.891	0.897	0.905
SVM	0.905	0.84	0.806	0.832	0.845

**Table. 2** The F1 score results of neural network-based models

	Nonscreen	Desktop	Laptop	Mobile	Average
MLP	0.999	1	0.999	0.997	0.998
CNN	0.982	0.994	0.987	0.982	0.986
LSTM	0.925	0.95	0.933	0.893	0.925

고한다. 각 분류 모델의 학습과 테스트는 Intel i7-7700 CPU와 NVIDIA Geforce 1060 GPU가 탑재된 컴퓨터를 이용하여 수행되었다.

#### 4.2. 평가 결과

그림 1에서 볼 수 있는 것처럼 세 가지의 전통적인 머신러닝 모델을 비교했을 때 가장 좋은 결과를 보인 모델은 RF 모델이다. RF의 평균 F1 스코어는 0.905이고, SVM의 평균 F1 스코어는 0.845로 RF보다 낮은 성능을 보였다. 마지막으로 k-NN 모델은 나머지 두 모델보다 상대적으로 상당히 낮은 성능을 보였고, 평균 F1 스코어는 0.492이다. RF 모델은 전체적으로 스코어가 다른 모델에 비해 높고 디바이스 간의 차이도 작아서 좋은 성능을 보이는 것으로 평가할 수 있다. 반면, k-NN 모델의 경우 평균 스코어도 낮을 뿐더러 디바이스에 따라서 상대적으로 차이가 크게 나타나는 것도 성능이 좋지 않은 것으로 볼 수 있다. 표 1에서 볼 수 있듯이, 각 클래스 별 스코어를 살펴보면, RF 모델은 스크린 미사용 0.938, 데스크탑 스크린 0.894, 노트북 스크린 0.891, 스마트폰 스크린 0.897이다. SVM 모델은 스크린 미사용 0.905, 데스크탑 스크린 0.84, 노트북 스크린 0.806, 스마트폰 스크린 0.832이다. k-NN 모델은 스크린 미사용 0.605, 데스크탑 스크린 0.498, 노트북 스크린 0.481, 스마트폰 스크린 0.387이다.

그림 2에서 보듯이 신경망 기반 모델 중에서는 MLP의 F1 스코어가 다른 모델에 비해 다소 더 높았다. MLP의 평균 F1 스코어는 0.998이고, CNN의 평균 F1 스코어는 0.986으로 MLP가 조금 높은 점수를 보였다. 상대적으로 이 둘보다 떨어지는 성능을 보인 것은 LSTM 모델

이었다. LSTM의 평균 F1 스코어는 0.925 정도가 나옴을 확인할 수 있었다. 하지만 이는 RF 모델보다 좋은 결과이다. 표 2에서 볼 수 있듯이, 각 클래스 별 스코어를 살펴보면, MLP는 스크린 미사용 0.999, 데스크탑 스크린 1.000, 노트북 스크린 0.999, 스마트폰 스크린 0.997로 높은 수치가 나왔다. CNN은 스크린 미사용 0.982, 데스크탑 스크린 0.994, 노트북 스크린 0.987, 스마트폰 스크린 0.982로 평균값은 MLP에 비해 조금 떨어지는 수치였다. LSTM은 스크린 미사용 0.925, 데스크탑 스크린 0.95, 노트북 스크린 0.933, 스마트폰 스크린 0.893으로 두 모델보다 상대적으로 낮게 나왔다.

## V. 결론

본 논문에서는 color, IMU, lidar 센서 데이터를 이용하여, 스크린 사용 여부 및 사용 중인 스크린 디바이스를 감지하는 머신러닝 기반 모델을 제안하고 다양한 모델의 성능을 비교하였다. 3개의 전통적인 머신러닝 모델(RF, SVM, k-NN)과 3개의 신경망 기반 모델(MLP, CNN, LSTM)을 이용하여 스크린 사용 디바이스 분류기를 학습하고, 데스크탑 모니터, 노트북 모니터, 스마트폰, 스크린 미사용, 4개의 클래스를 분류하도록 하였다. 성능 비교 결과 신경망 기반 모델이 전통적인 머신러닝 모델보다 높은 F1 스코어를 보였다. 신경망 기반 모델에서는 MLP, CNN 기반 모델이 LSTM 기반 모델보다 상대적으로 높은 스코어를 보였으며, 전통적인 머신러닝 모델에서는 RF 모델이 가장 우수했으며, 다음으로는 SVM 모델이었다. 기존 연구에서는 주로 스크린

사용 여부만을 감지하는 것에 초점을 맞추었으나, 본 논문에서는 사용 중인 스크린 디바이스의 종류까지 구분할 수 있고 신경망 기반 모델의 경우 향상된 성능을 나타냄을 보인 것으로 의의가 있다.

### ACKNOWLEDGEMENT

This paper was supported by the Education and Research Promotion Program of KOREATECH (2018).

### REFERENCES

[ 1 ] Computer Vision Syndrome [Internet]. Available: <https://www.aoa.org/patients-and-public/caring-for-your-vision/protecting-your-vision/computer-vision-syndrome?sso=y>.

[ 2 ] The 20-20-20 Rule [Internet]. Available: <https://opto.ca/health-library/the-20-20-20-rule>.

[ 3 ] Screen Time [Internet]. Available: <https://support.apple.com/ko-kr/HT208982>.

[ 4 ] Digital Wellbeing [Internet]. Available: <https://www.android.com/digital-wellbeing/>.

[ 5 ] Y. C. Zhang and J. M. Rehg, "Watching the TV Watchers," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, Article 88, Jun. 2018.

[ 6 ] F. Wahl, J. Kasbauer, and O. Amft, "Computer Screen Use Detection Using Smart Eyeglasses," *Frontiers in ICT*, 4:8, May 2017.

[ 7 ] C. Min, E. Lee, S. Park, and S. Kang, "Tiger: Wearable Glasses for the 20-20-20 Rule to Alleviate Computer Vision Syndrome," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, Oct. 2019.

[ 8 ] T. Okita and S. Inoue, "Recognition of multiple overlapping activities using compositional CNN-LSTM model," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pp. 165-168, Sep. 2017.

[ 9 ] Y. Yuki, J. Nozaki, K. Hiroi, K. Kaji, and N. Kawaguchi, "Activity Recognition using Dual-ConvLSTM Extracting Local and Global Features for SHL Recognition Challenge,"

in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1643-1651, Oct. 2018.

[10] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, Article 74, Jun. 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026-1034, 2015.



황상원(Sangwon Hwang)

2019년 한국기술교육대학교 컴퓨터공학부 학사  
2019년-현재: 한국기술교육대학교 대학원  
컴퓨터공학과 석사과정  
※관심분야: IoT/모바일 시스템, 모바일/웨어러블/  
유비쿼터스 컴퓨팅, 딥러닝



김동우(Dongwoo Kim)

2018년 한국기술교육대학교 컴퓨터공학부 학사  
2018년-현재: 한국기술교육대학교 대학원  
컴퓨터공학과 석사과정  
※관심분야: IoT/모바일 시스템, 모바일/웨어러블/  
유비쿼터스 컴퓨팅



이주환(Juhwan Lee)

2019년 한국기술교육대학교 컴퓨터공학부 학사  
2019년-현재: 한국기술교육대학교 대학원  
컴퓨터공학과 석사과정  
※관심분야: IoT/모바일 시스템, 모바일/웨어러블/  
유비쿼터스 컴퓨팅



강승우(Seungwoo Kang)

2010년 한국과학기술원 전산학과 공학박사  
2015년-현재: 한국기술교육대학교 컴퓨터공학부  
조교수  
※관심분야: IoT/모바일 시스템, 모바일/웨어러블/  
유비쿼터스 컴퓨팅