

A Taekwondo Poomsae Movement Classification Model Learned Under Various Conditions

Ju-Yeon Kim*, Kyu-Cheol Cho*

*Student, Dept. of Computer Science, Inha Technical College, Incheon, Korea

*Professor, Dept. of Computer Science, Inha Technical College, Incheon, Korea

[Abstract]

Technological advancement is being advanced in sports such as electronic protection of taekwondo competition and VAR of soccer. However, a person judges and guides the posture by looking at the posture, so sometimes a judgment dispute occurs at the site of the competition in Taekwondo Poomsae. This study proposes an artificial intelligence model that can more accurately judge and evaluate Taekwondo movements using artificial intelligence. In this study, after pre-processing the photographed and collected data, it is separated into train, test, and validation sets. The separated data is trained by applying each model and conditions, and then compared to present the best-performing model. The models under each condition compared the values of loss, accuracy, learning time, and top-n error, and as a result, the performance of the model trained under the conditions using ResNet50 and Adam was found to be the best. It is expected that the model presented in this study can be utilized in various fields such as education sites and competitions.

▶ **Key words:** Pose Estimation, Machine Learning, Deep Learning, Convolutional Neural Network, Deep Convolutional Neural Network

[요 약]

태권도 겨루기의 전자호구, 축구의 VAR 등 스포츠에서 기술 발전이 고도화되고 있다. 하지만 태권도 품새는 사람이 직접 자세를 눈으로 보고 판단하며 지도하기 때문에 때로는 대회의 현장에서 판정시비가 일어난다. 본 연구는 인공지능을 이용하여 태권도 동작을 더 정확하게 판단하고 평가할 수 있는 인공지능 모델을 제안한다. 본 연구에서는 촬영 및 수집한 데이터를 전처리한 후 학습, 테스트, 검증 세트로 분리한다. 분리한 데이터를 각 모델과 조건을 적용하여 학습한 후 비교하여 가장 좋은 성능의 모델을 제시한다. 각 조건의 모델은 정확도, Precision, Recall, F1-Score, 학습 소요 시간, Top-n error의 값을 비교하였고 그 결과 ResNet50과 Adam을 사용한 조건에서 학습한 모델의 성능이 가장 우수한 것으로 나타났다. 본 연구에서 제시한 모델을 활용하여 교육 현장이나 대회 등 다양한 분야에서 활용할 수 있을 것으로 기대한다.

▶ **주제어:** 자세 분류, 머신러닝, 딥러닝, 합성곱 신경망, 심층 합성곱 신경망

-
- First Author: Ju-Yeon Kim, Corresponding Author: Kyu-Cheol Cho
 - *Ju-Yeon Kim, (kimjuyeon0912@gmail.com), Dept. of Computer Science, Inha Technical College
 - *Kyu-Cheol Cho (kccho@inhac.ac.kr), Dept. of Computer Science, Inha Technical College
 - Received: 2023. 07. 10, Revised: 2023. 10. 18, Accepted: 2023. 10. 18.

I. Introduction

인공지능 기술은 번역기, 얼굴 인식 등 우리 삶의 많은 부분에 적용되고 있다[1][2]. 다양한 분야에서 적용되는 인공지능 기술은 속도, 비용, 자동화, 정확도 등의 측면에서 유리하다. 이와 마찬가지로, 인공지능 기술은 태권도 분야에서도 적용될 수 있다.

2022년 5월 29일 기준, 세계 태권도 연맹의 가입국은 212개국으로 태권도는 세계적으로 인기있다고 할 수 있다[3]. 하지만 태권도 교육은 여전히 교육자의 시각으로 이루어지며, 같은 동작에 대해 가르치더라도 각각의 교육자마다 다른 기준으로 가르친다. 또한, 교육 현장뿐만 아니라 승패를 결정하는 대회에서도 사람의 눈으로 보고 승패나 감점 여부를 결정한다. 심판의 주관적인 판단에 따라 승패가 결정되며 때로는 판정시비나 판정불복의 결과를 가져오기도 한다[4]. 본 연구는 이런 태권도와 인공지능 기술을 결합하여 주관적인 시선과 판단보다 객관적인 판단 기준을 제시하고자 시작했다.

Human Pose Estimation은 Computer Vision 분야에서 활발히 연구되고 있는 분야 중 하나로 객체의 위치와 방향을 탐지하는 문제이며, 사람의 신체 관절인 Keypoint가 어떻게 구성되어 있는지 위치를 측정하고 추정하는 문제이다. Pose Estimation은 2D와 3D Pose Estimation으로 나눌 수 있고, Bottom-up과 Top-down 방식으로 구분할 수 있다[5].

기존 태권도 동작 분류 시스템의 경우 센서 등의 장비를 사용하여 동작 데이터를 추출하며, AI 트레이너 시스템 등의 경우에는 많은 동작을 분류하지 않고 태권도 동작같이 비슷한 동작을 분류하지 않는다. 본 연구는 카메라 외에 다른 물리적 장비를 사용하지 않고, 많은 동작을 분류하고 평가하기 위한 태권도 동작 분류 모델을 제안하는 것을 목적으로 한다. 향후 본 연구에서 제안한 모델을 활용하여 태권도 동작 평가시스템을 구축하여 실제 대회나 교육 현장에서 사용될 수 있을 것으로 기대한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련이 있는 연구에 관하여 기술한다. 3장에서는 본 연구에 필요한 데이터 전처리 과정을 기술한다. 4장에서는 학습시킨 모델의 결과를 비교하여 분석한 결과를 기술한다. 5장에서는 본 연구에 대한 결론을 기술한다.

II. Related Works

1. Pose Classification with Sensor

머신러닝을 이용한 앉은 자세 분류 연구[6]는 가속도 센서의 출력 데이터를 이용한 연구이다. 1개의 가속도 센서를 경추에 부착해 데이터를 얻는다. 가속도 센서의 출력 데이터는 주성분 분석법(Principal Component Analysis, PCA)을 사용하여 특징을 추출한다. 훈련 모델은 비지도 학습 방법인 K-Means 알고리즘과 지도 학습 방법인 SVM을 사용하며, 추출한 데이터를 사용하여 각각 훈련 모델을 생성한다. 1가지의 바른 자세와 4가지의 잘못된 자세, 총 5가지의 자세를 분류한다.

태권도 품새 검증 애플리케이션 설계 및 구현[7]과 모션 인식기술을 이용한 태권도 품새 평가시스템[8]은 Kinect 센서의 출력 데이터를 이용한 연구이다. Kinect 출력 데이터를 이용하여 사람의 스켈레톤 데이터를 추출한다. 태권도 품새 검증 애플리케이션 설계 및 구현은 사용자의 컬러 스트림, 뎁스 스트림, 스켈레톤 정보를 얻어 사용자의 동작과 품새 가이드의 동작을 비교하여 일치하면 점수를 획득하고 다음 동작으로 이동한다. 모션 인식기술을 이용한 태권도 품새 평가시스템은 스켈레톤의 각도와 연속적으로 이루어지는 시간을 이용하여 동작을 분류한다.

위 세 가지 연구는 센서를 이용하여 진행했다. 센서를 이용하여 사람의 동작을 분류하게 되면 각 스켈레톤의 데이터를 실시간으로 추적하기 쉽고, 여러 가지 출력 데이터를 이용할 수 있다는 장점이 있다. 하지만 센서를 사용한 시스템은 센서를 사용해야지만 동작을 판단할 수 있고, 대회 현장에서는 센서를 사용한 시스템의 이용이 불가능하다. 따라서 본 연구에서는 시스템을 편하게 사용하고 광범위한 현장에서 적용할 수 있게 센서를 사용하지 않고 영상과 이미지 데이터만을 사용하여 동작을 분류하였다.

2. Pose Classification into Less Classes

홈 트레이닝을 위한 운동 동작 분류 및 교정 시스템[9]은 이미지 내 사람의 스켈레톤 데이터와 OpenPose를 이용한 시스템이다. 위 시스템 알고리즘은 자세 추정, 자세 분류 그리고 자세 교정의 단계로 이루어진다. 먼저 영상 데이터의 각 프레임은 추출하고 각 프레임의 스켈레톤을 추출하여 이 정보를 기반으로 자세를 분류하고, 사용자의 자세를 교정한다. 자세는 Squat, Lunge 그리고 Dumbbell Curls로, 총 3가지의 자세로 분류하며 각 관절의 각도를 이용하여 자세를 추론하고 자세를 교정할 수 있도록 도와준다. 자세 분류 모델은 다층 퍼셉트론

(Multiple-Layer Perceptron, MLP) 기반의 분류 모델을 사용했고, 투표 알고리즘을 사용하여 영상 속 동작을 분류했다. MLP는 총 3개의 은닉층으로 구성하였다. 위 연구는 세 가지 자세로 분류한다.

PoseNet을 활용한 헬스 트레이너 웹 서비스[10]는 Google의 Teachable Machine을 활용하여 이미지의 스켈레톤 데이터를 학습한다. 위 시스템은 Client Tier, Application Tier, Data Tier로 이루어진다. 자세는 Squat, Stand, Bad Posture, Waist down more, 그리고 None으로, 총 5가지의 자세로 구분한다. 위 연구는 다섯 가지 자세로 분류한다.

영상 데이터에서의 CRNN(Convolutional Recurrent Neural Network) 기반 자세 인식[11]은 사용자가 취한 체조 동작이 어느 동작인지 알려주는 시스템이다. 합성곱 신경망(Convolutional Neural Network, CNN)과 순환 신경망(Recurrent Neural Network, RNN)이 연결된 하나의 end-to-end CRNN 형태의 신경망을 사용했다. 휴대폰 카메라로 촬영한 영상 데이터에서 추출한 이미지 데이터를 사용하며 CNN으로는 Google의 Inception-V3 Pretrained Model, RNN으로는 GRU(Gated Recurrent Unit)를 사용했다. 위 연구는 열여섯 가지 자세로 분류한다.

위 세 가지 연구는 적은 클래스로 자세를 분류한다. 적은 클래스로의 분류는 많은 특징의 학습이 필요하지 않다. 본 연구에서는 위 세 가지의 연구보다 더 많은 64가지의 클래스로 자세를 분류하기 때문에 Teachable Machine을 사용하지 않았고 더 많은 특징을 학습할 수 있는 심층 합성곱 신경망을 사용하여 모델을 설계했다.

III. Taekwondo Image Classification Using Vgg16_BN And ResNet50

본 연구는 모델 중 Vgg16_bn과 ResNet50, Optimizer 중 SGD와 Adam을 사용하여 학습 조건을 다르게 설정하며 어느 조건에서의 학습이 태권도 동작 분류에 적합한지 제시한다.

1. System Architecture

그림 1은 본 시스템의 구조이다. 사용자 단계에서는 카메라로 사람의 동작을 촬영하면 실시간으로 영상이 전처리되어 일정 주기마다 해당 동작의 클래스와 정확도 정보를 출력한다. 시스템은 동작 분류 모델로 출력한 결과를 사용자에게 반환한다. 모델 학습 단계에서는 전처리된 데이터를 저장하여 학습 데이터를 생성한다. 생성된 데이터는 동작 인식 모델에 주기적으로 전이학습을 진행한다. 입출력 데이터는 각 단계에 어떤 데이터가 필요한지를 나타낸다. 본 연구에서는 데이터 전처리, 모델 학습 그리고 모델 평가 단계를 구현한다.

본 시스템은 AI HUB에 공개되어있는 ‘고양시 태권도 데이터[12]’와 ‘2023 태권도 품새 경기규칙[13]’을 바탕으로 작성한 촬영 리스트를 참고하여 총 두 사람의 데이터를 사용한다. 위 데이터를 학습하여 총 64가지의 동작으로 분류하는 모델을 생성한다.

사용자가 사람의 동작을 촬영하여 데이터가 생성되면 해당 데이터를 전처리하여 동작 분류를 진행하여 사용자에게 동작의 분류와 정확도를 보여준다. 이를 위해 동작 분류를 위한 모델을 선정하고 사용하는 과정을 다룬다. 본 연구는 데이터 수집 및 전처리, 모델 선정 및 학습, 모델 정확도 비교의 주요 단계로 구성된다.

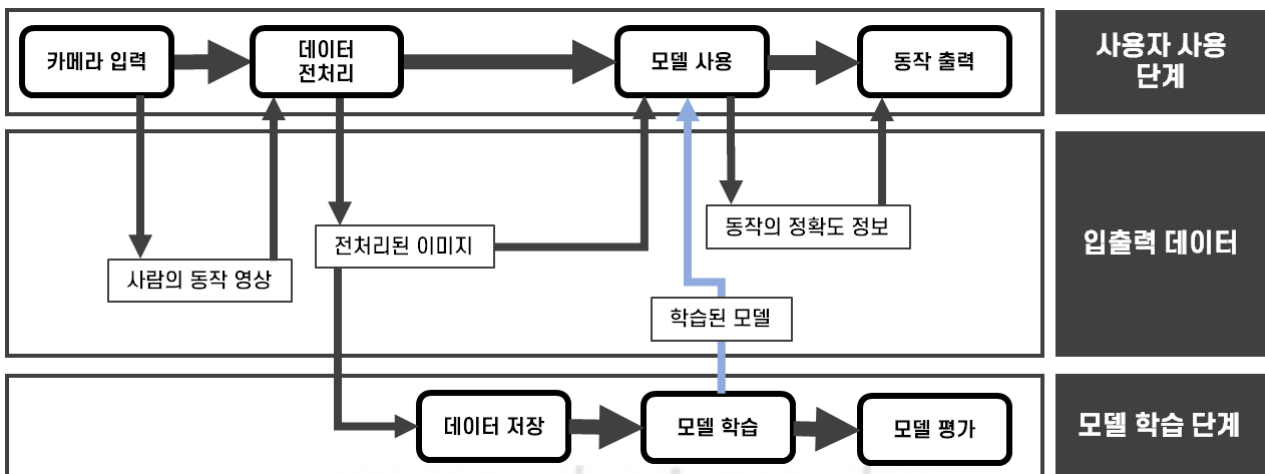


Fig. 1. System Architecture

2. Data Preprocessing



Fig. 2. Data Preprocessing

그림 2는 데이터 전처리의 전체 과정을 나타낸 것이다. 먼저 학습에 필요한 이미지를 준비한다. 이미지는 사람의 동작 부분만 잘라낸 후 사용할 모델의 Input 크기인 $224 \times 224(h \times w)$ 로 해상도를 가공한다. 해상도를 변경한 후 이미지 오류 등으로 해상도가 다른 이미지는 제거한다. 오류가 없는 이미지는 학습, 테스트, 검증 세트로 분리한다.

2.1 Data Collection

학습을 위한 태권도 영상 이미지 데이터 수집 방법으로 두 가지 방법을 이용했다.

첫 번째는 AI HUB에 공개되어있는 ‘고양시 태권도 데이터[12]’이다. ‘고양시 태권도 데이터’는 328.17GB의 크기로 약 900,000장의 이미지로 구성된다. 총 64가지의 동작으로 분류되어 있고 각 동작의 처음, 중간 그리고 끝 동작이 포함되어 있다. 위 데이터는 품새 동작 데이터로 이루어져 있으며 태권도 동작을 인식하여 분류하거나, 태권도 자세를 추정하기 위한 데이터로 본 연구에서 진행하고자 하는 동작 분류에 적합하여 위 데이터를 선택했다.

두 번째는 직접 촬영이다. ‘2021 태권도 품새 경기규칙[13]’을 바탕으로 품새 동작 리스트를 작성하고 작성한 리스트 중 네 가지 동작을 선정했다. 선정한 동작의 두 사람 데이터를 촬영·수집했다.

2.2 Image Capture

2.1의 데이터 수집 단계에서 수집한 데이터 영상을 학습을 위한 이미지 형태로 만들기 위해 각 영상의 프레임을 추출하여 이미지 데이터로 변환했다. 표 1은 촬영한 영상으로부터 이미지를 추출하여 저장하는 Pseudo Code이다. 동작의 손실을 최소화하며 이미지를 추출하기 위해 5프레임당 1장의 이미지가 추출되게 하였으며 1초당 약 6장, 20초 내외의 영상마다 약 110장에서 120장의 이미지가 추출됐다.

2.3 Crop Image with YOLOv4 and Resolution

Normalization

‘고양시 태권도 데이터’의 원본 이미지 해상도는 $1920 \times 1080(h \times w)$ 이고, 직접 촬영하여 추출한 이미지의 해상도

Table 1. Pseudo Code of Image Capture

Input: DL is a list of video data
S is a flag to successful reading of video
C is the sequence number of the saved image
VC is a reading one image
IM is reading the video

Output: captured images

begin:

```

1: DL ← [ video data list ]
2: for dli in DL do
3:   S ← true
4:   C ← 1
5:   VC ← read dli
6:   while S do
7:     S, IM ← read VC
8:     if S = false do
9:       break
10:    end if
11:    if frame of VC % 5 = 0 do
12:      save IM
13:      C ← C + 1
14:    end if
15:  end while
16:  release VC
17: end for
end
  
```

는 $1080 \times 1920(h \times w)$ 이다. 원본 이미지는 동작을 인식하는데 불필요한 영역이 크기 때문에 사람의 동작을 제외한 이미지의 불필요한 여백 부분을 잘라내고, $224 \times 224(h \times w)$ 로 정규화하는 과정이 필요하다. 이미지를 잘라내는 과정에서 이미지를 일괄적인 좌표로 잘라내게 되면 학습에 필요한 사람의 동작도 잘릴 수 있으므로 사람이 동작을 취하는 필요한 영역만 잘라서 저장한 후 학습에 필요한 이미지 해상도인 $224 \times 224(h \times w)$ 로 정규화한다. 표 2는 YOLOv4를 이용해 사람을 검출하여 사람의 동작만 잘라낸 후 $224 \times 224(h \times w)$ 로 정규화하는 Pseudo Code이다.

Table 2. Pseudo Code of Crop Image with YOLOv4

Input: R is detecting a person
S is target size of image change
IL is a list of image data
IM is a reading one image
IMR is a crop of image

Output: resized and cropped image

begin:

```

1: S ← (224, 224)
2: IL ← [ image data list ]
3: for ili in IL do
4:   IM ← read ili
5:   R ← detecting person in IM with YOLOv4
6:   IMR ← crop IM with R
7:   resize IMR with S
8:   save IMR
9: end for
end
  
```

2.1-2.3의 과정을 거친 이미지 데이터의 전처리 전·후 결과는 그림 3과 같다.

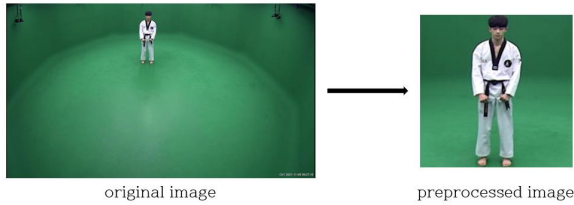


Fig. 3. Preprocessed Result Image

2.4 Split Data

모든 전처리 과정을 거친 이미지 데이터는 과적합을 방지하기 위해 학습, 테스트, 검증 세트로 데이터를 분리한다. 본 연구에서 사용하는 데이터는 900,000장 이상으로 충분한 학습 데이터를 확보하면서도 모델의 일반화 성능을 평가하고 검증 및 테스트하는 데에 충분한 양의 데이터를 분리하기 위해 8:1:1 비율로 분리한다. 표 3은 데이터를 분리하는 Pseudo Code이다.

Table 3. Pseudo Code of Data Split

```

Input:      DS is a dataset
              DS_S is a length of DS
              TR_S is a size of train set
              VL_S is a size of validation set
              TS_S is a size of test set
Output:   dataset of randomly shuffled
begin:
1: DS ← dataloader to data
2: DS_S ← length of DS
3: TR_S ← integer of DS_S * 0.8
4: VL_S ← integer of DS_S * 0.1
5: TS_S ← DS_S - TR_S - VL_S
6: random split and save with TR_S, VL_S, TS_S
end

```

3. Model Selection and Summary

태권도 동작은 다양한 자세와 동작으로 구성되어 있다. 태권도 동작 분류 모델은 다양한 동작의 특징을 추출하고 학습하기 때문에 다양하고 복잡한 동작을 학습할 수 있는 모델을 선정하였다.

3.1 Vgg16_bn

Vgg16은 깊은 신경망 구조, 네트워크 내 동일한 필터 크기 사용 그리고 간단한 구조와 효율성 등의 특징들을 가지고 있다. LeNet-5와 AlexNet에 비교하여 깊은 신경망 구조로 되어 있어서 복잡한 동작 패턴과 특징을 학습하는데 유리하다[14]. GoogLeNet의 Inception Module과 AlexNet의 드롭아웃 레이어와 비교하면 Vgg16은 보다 단

순한 구조이다. 단순한 구조는 학습과 추론 단계에서 높은 효율성을 가진다. 배치 정규화는 학습 도중 각 레이어의 입력을 정규화하여 학습 과정을 안정화하고 성능을 향상시켜 과적합을 줄이고 일반화 성능을 향상한다. 따라서 본 연구에서는 깊은 신경망 구조, 단순한 구조 그리고 배치 정규화를 사용하는 Vgg16_bn을 선택했다. Vgg16_bn은 16개의 layer를 가지며 $224 \times 224 \times 3(h \times w \times c)$ 의 Input, Convolution Layer, Batch Normalization Layer, Max Pooling Layer 그리고 Fully Connected Layer로 구성된다[14]. Vgg16_bn은 사전 학습된 1000개의 클래스로 분류하기 때문에 Vgg16의 모델 구조만 가져와서 사용하며, 64개의 클래스로 분류하기 위하여 기존 모델의 클래스 개수를 64개로 변경하였다.

3.2 ResNet50

ResNet50은 깊은 신경망 구조, Residual Network 그리고 계층적 특징 추출 등의 특징을 가지고 있다. 다른 알고리즘에는 사용하지 않는 Residual Network는 깊은 신경망 구조를 가져도 Gradient Vanishing, Exploding Gradient 등의 문제를 해결해서 복잡한 동작 패턴 학습에 유리하다[15]. 그리고 여러 개의 레이어를 건너뛰고 특징을 추출할 수 있는 Skip Connection은 더 정확한 패턴 인식과 분류를 가능하게 한다. 따라서 본 연구에서는 Residual Network와 Skip Connection의 특징을 가지는 ResNet50을 선택했다. ResNet50은 50개의 layer를 가지며 $224 \times 224 \times 3(h \times w \times c)$ 의 Input, Convolution Layer, Max Pooling Layer, Residual Block Layer, Global Average Pooling Layer 그리고 Fully Connected Layer로 구성된다[15]. Vgg16_bn과 마찬가지로 사전 학습된 ResNet50은 1000개의 클래스로 분류하기 때문에 ResNet50의 모델 구조만 가져와서 사용하며, 64개의 클래스로 분류하기 위해 기존 모델의 마지막 Fully Connected Layer를 새로운 Connected Layer로 교체하였다.

IV. Experiment

실험을 통해 본 연구에서 설정한 조건: VS(Vgg16_bn, SGD), VA(Vgg16_bn, Adam), RS(ResNet50, SGD), RA(ResNet50, Adam)에 대해서 학습 소요 시간, 정확도, Precision, Recall, F1-Score 그리고 Top-N Error를 비교하여 제시하고자 한다. 데이터는 직접 수집한 Image 데이터와 AI HUB에 공개되어있는 Image 데이터를 사용하

였다. 모든 조건에서 epoch는 100, 과적합을 방지하기 위한 early stop을 5로 설정해 5회 이상 성능이 향상하지 않으면 학습을 종료되게 설정했다. Batch size는 클수록 학습 시간이 단축되기 때문에 본 실험을 진행하는 환경에서 가능한 가장 큰 Batch Size인 128로 설정했다. 각 조건에서 학습 과정에 차이를 두고 성능을 비교하기 위해 Model과 Optimizer를 다르게 설정하여 실험을 진행했다. 학습 조건은 표 4와 같다.

Table 4. Training Conditions

cond.	VS	VA	RS	RA
epoch	100			
early stop	5			
batch size	128			
model	vgg16_bn		resnet50	
optimizer	SGD	Adam	SGD	Adam

1. Training Time

학습 소요 시간은 모델이 주어진 데이터를 학습하는데 소요되는 시간을 측정한 것이다. 모델의 학습 시간이 짧을수록 실시간 환경에서 사용이 적합하다.

Table 5. Training Time

	VS	VA	RS	RA
time(h)	6.52	5.73	1.84	1.95

표 5는 각 조건에서 데이터를 학습할 때 1 epoch 당 평균 소요 시간이다. ResNet50을 사용한 모델이 Vgg16_bn을 사용한 모델보다 훈련 속도가 빠르다.

2. Comparison of VS, VA, RS and RA

모델의 성능 평가 지표로는 Accuracy, Precision, Recall, F1-Score, Top-N Error를 사용한다. Accuracy는 전체 데이터 중에서 모델이 올바르게 예측한 데이터의 비율로 나타낸다. Precision은 분류 모델이 True로 예측한 데이터 중 실제로 True인 데이터의 비율을 나타낸다. Recall은 실제 정답이 True인 것 중에 분류 모델이 True로 예측한 데이터의 비율이다. F1-Score는 정밀도와 재현율의 조화평균이다. F1-Score는 정밀도와 재현율이 이룰 때 큰 값을 가진다. Top-N Error는 모델이 정확한 예측을 하지 못하고 상위 N개의 클래스 중 하나에 예측을 포함하지 못한 비율을 나타낸다. Top-1 Error는 모델의 예측 결과 중 가장 확률이 높은 클래스와 실제 레이블이 일치하지 않는 비율을 나타내며, Top-5 Error는 모델의 예측 결과의 상위 5개 확률의 클래스 중 실제 레이블

이 포함되지 않은 비율을 나타낸다. Accuracy, Precision, Recall, F1-Score는 값이 클수록 좋은 모델로 평가하고, Top-N Error는 값이 작을수록 좋은 모델로 평가한다. Accuracy, Precision, Recall, F1-Score를 구하는 식은 (1)(2)(3)(4)와 같다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

TP, TN, FP, FN에 대한 정의는 다음과 같다.

- TP(True Positive): 실제 A 동작을 A로 예측
- TN(True Negative): 실제 A가 아닌 동작을 A가 아닌 동작으로 예측
- FP(False Positive): 실제 A가 아닌 동작을 A라고 예측
- FN(False Negative): 실제 A 동작을 A가 아닌 동작으로 예측

2.1 Accuracy

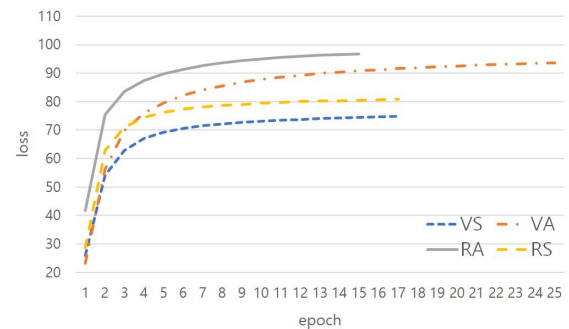


Fig. 4. Train Accuracy

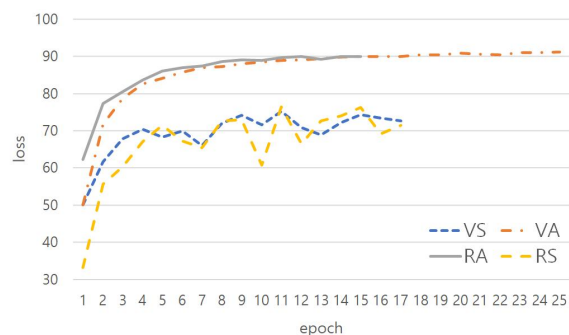


Fig. 5. Validation Accuracy

그림 4는 표 1에서 설정한 각 조건에 대한 Train Accuracy 값을 나타낸다. 위 그래프에서 보면 Accuracy가 증가하는 방향으로 수렴하며 마지막 Epoch의 Accuracy는 RA:96.77, VA:93.57, RS:80.74, VS:74.83이다. 이 점을 볼 때 SGD보다 Adam, 같은 Optimizer를 사용했을 때 Vgg16_bn보다 ResNet50을 사용한 모델의 성능이 좋다고 할 수 있다. 그림 5는 표 4에서 설정한 각 조건에 대한 Validation Accuracy 값을 나타낸다. RA와 VA는 일정하게 증가하지만, RS와 VS는 Accuracy 변동의 폭이 크다. 이를 볼 때 RS와 VS에서 Overfitting이 발생했다는 것을 알 수 있다. 마지막 epoch의 Accuracy는 RA:89.94, VA:91.10, RS:71.49, VS:72.61이다.

Accuracy 그래프를 비교·분석한 결과 SGD를 사용한 모델은 학습 과정에서 Overfitting이 발생하였다. RA는 Train Accuracy가 높고, VA는 Validation Accuracy가 높다. 따라서 Accuracy의 경우 SGD보다 Adam을 사용한 모델의 성능이 좋다고 볼 수 있다.

2.2 Precision, Recall, F1-Score

Table 6. Precision, Recall, F1-Score

	VS	VA	RS	RA
precision	0.7754	0.9200	0.7835	0.9050
recall	0.7354	0.9169	0.7435	0.9051
f1-score	0.7418	0.9181	0.7485	0.9041

표 6은 각 조건에서 측정한 Precision, Recall, F1-Score 값이다. RS, VS보다 RA와 VA 조건에서 상대적으로 높은 Precision, Recall, F1-Score 값을 가진다. 이를 볼 때 RA와 VA 조건에서 True 예측의 신뢰도가 높으며, 실제 True를 잘 감지한다고 할 수 있다.

2.3 Top-N Error

Table 7. Top-N Error

	Top-1 Error	Top-5 Error
VS	22.3892	2.5336
VA	6.8641	0.5096
RS	21.3977	2.3362
RA	8.1930	0.6175

표 7은 각 조건에서 측정한 Top-N Error 값이다. Top-1 Error와 Top-5 Error 둘 다 VA→RA→RS→VS 순으로 Top-N Error가 낮은 것을 볼 수 있다. 모델의 구조에 따른 차이는 볼 수 없지만, SGD를 사용한 모델보다 Adam을 사용한 모델의 성능이 좋다고 할 수 있다.

3. Performance Comparison

지금까지의 성능 평가 지표의 수치들을 비교하면 표 8과 같다.

Table 8. Performance Comparison

	VS	VA	RS	RA
time(h)	6.52	5.73	1.84	1.95
accuracy	72.61	91.10	71.49	89.94
precision	0.7754	0.9200	0.7835	0.9050
recall	0.7354	0.9169	0.7435	0.9051
f1-score	0.7418	0.9181	0.7485	0.9041
top-1 err	22.39	6.86	21.40	8.19
top-5 err	2.53	0.51	2.34	0.62

Accuracy, Precision, Recall, F1-Score, Top-N Error의 부분에서 SGD보다 Adam을 사용하는 것이 성능이 좋고, 학습 소요 시간은 Vgg16_bn보다 ResNet50을 사용하는 것이 빠르다. 학습 소요 시간을 제외한 모든 지표에서 RA보다 VA가 성능이 약 0.01에서 2 정도 우수하지만, 학습 소요 시간에서는 VA보다 RA가 약 3배 더 빠르다.

따라서 본 연구에서는 모든 성능 비교 지표를 종합해보면 RA(Resnet50, Adam) 조건에서 우수한 성능이 나타나는 것을 볼 수 있어 Resnet50과 Adam을 사용한 분류 모델을 권장한다.

V. Conclusions

본 연구는 태권도와 인공지능 기술을 결합한 시스템과 기술로 태권도의 동작을 정확하게 판단하고 태권도를 흥미롭게 배우고자 하는 관점에서 시작되었다. 태권도 동작 분류에 적합한 학습 조건을 찾기 위해 모델 구조 중 vgg16_bn과 ResNet50, Optimizer 중 SGD와 Adam을 사용하여 조건을 다르게 설정하였고 손실 값, 정확도, Top-N Error를 비교하고 분석하였다. 본 연구에서는 성능이 가장 우수한 모델로 ResNet50과 Adam을 사용하는 모델을 제시한다. 본 연구에서는 ResNet50과 Adam을 사용했을 때 태권도 동작 분류 정확도가 높고, 태권도 동작 분류에 가장 적합한 것을 볼 수 있으며, SGD를 사용했을 때 Overfitting이 발생하는 것을 볼 수 있었다. 따라서 드롭아웃, 데이터 추가 등의 방법으로 Overfitting을 해결하고 여러 개의 모델을 활용하는 앙상블 기법을 적용하면 Overfitting을 방지하면서 더욱 높은 정확도를 가진 모델을 만들어 낼 수 있을 것이다. 추후 본 모델은 태권도 교육을

현장이나 대회 등의 방면에서 사용될 수 있다. 또한, 본 모델을 사용하는 동안 태권도 수련자들의 태권도 동작 데이터를 저장하고 해당 데이터를 전이학습해서 정확도를 더 향상하고 태권도 동작의 수련뿐만 아니라 동작의 잘못된 부분을 수정할 수 있는 기능까지 추가할 수 있을 것으로 기대된다.

REFERENCES

- [1] Jianri Li, Jong-Hyeok Lee, "RA Survey on Neural Machine Translation", Communications of the Korean Institute of Information Scientists and Engineers, Vol. 33, No. 10, pp. 40-47, October 2015.
- [2] Yeongseo Ha, Jihee Park, Jaechang Shim, "Comparison of Face Recognition Performance Using CNN Models and Siamese Networks", Journal of Korea Multimedia Society, Vol. 26, No. 2, pp. 413-419, February 2023. DOI: 10.9717/kmms.2023.26.2.413
- [3] WTF HomePage, <http://www.worldtaekwondo.org/about-wt/about.html>
- [4] Won Chul Bing, Yeo Tak Yoon, Sun Jang Lee, "A study on ethical conflict experience of Taekwondo coach by referee's authority", The Journal of the Korean Society for the Philosophy of Sport, Dance, & Martial Arts, Vol. 17, No. 2, pp. 107-121, June 2009.
- [5] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1302-1310, Honolulu HI, USA, July 2017. DOI: 10.1109/CVPR.2017.143
- [6] Sangyong Ma, Sangpyo Hong, Hyeon-min Shim, Jang-Woo Kwon, Sangmin Lee, "A Study on Sitting Posture Recognition using Machine Learning", The transactions of The Korean Institute of Electrical Engineers, Vol. 65, No. 9, pp. 1557-1563, September 2016. DOI: 10.5370/KIEE.2016.65.9.1557
- [7] Won Joo Lee, Byung Woon Jang, Ho Sang Yu, "A Design and Implementation of Taegwondo Poomsae Verification Application", Proceedings of the Korean Society of Computer Information Conference, Vol. 24, No. 2, pp. 37-38, July 2016.
- [8] Hae-Jong Joo, Hoe-Sun Han, Yoon-Hee Park, Sang-Soo Kim, Moon-Taek Cho, "A Study on the Taekwondo Poomsae Rating System using Motion Recognition Technology", Proceedings of the Korea Academia-industrial cooperation Society Conference, (), (), pp. 212-214, November 2014.
- [9] Jae Min Kang, Seongsu Park, Yun Soo Kim, Jin Kyu Gahm, "Pose Classification and Correction System for At-home Workouts", Journal of the Korea Institute of Information and Communication Engineering, Vol. 25, No. 9, pp. 1183-1189, September 2021. DOI: 10.6109/jkiice.2021.25.9.1183
- [10] Lee Sang-min, Kim Dohoon, "Health Trainer Web Services with PoseNet", Proceedings of KIIT Conference, Vol. 2021, No. 6, pp. 729-733, June 2021.
- [11] Se-Min Lim, Byeong-Cheol Chae, Soo-Bin Lim, Jooyoung Park, and Hyeong-Cheol Oh, "CRNN-based Human Posture Recognition in Video Data", Proceedings of the Korean Institute of Information Scientists and Engineers, Vol. 202, No. 7, pp. 726-728, (), 2020.
- [12] Taekwondo data in Goyang, <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realM&dataSetSn=71259>
- [13] Korea Taekwondo Association HomePage, <https://koreataekwondo.co.kr/ebook/2023/p/index.html?reload=1675306869680#page=1>
- [14] Simonyan, Karen, Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, (), (), (), 2014. DOI: <https://doi.org/10.48550/arXiv.1409.1556>
- [15] He, K., Zhang, X., Ren, S., Sun, J, "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, (), (), pp. 770-778, June, 2016. DOI: <https://doi.org/10.1109/cvpr.2016.90>

Authors



Ju-Yeon Kim received the A.S. degree in Dept. of Computer Science from Inha Technical College, Korea, 2023 respectively. Ms. Kim entered Dept. of Computer Science of Inha Technical College in 2023 as a B.S.

degree and is currently attending. She is interested in the Artificial Intelligence.



Kyu-Cheol Cho received the B.S., M.S. and Ph.D. degrees in Computer Science and Information Engineering from Inha University, Korea, in 2005, 2007 and 2013, respectively. Dr. Cho joined the faculty of the Department

of Computer Science at Inha Technical College, Incheon, Korea, in 2016. He is currently a assistant professor in the Department of Computer Science, Inha Technical College. He is interested in cloud computing, green IT and web programming.