

<https://doi.org/10.7236/JIIBC.2023.23.6.1>  
JIIBC 2023-6-1

## CNN 기술을 적용한 침수탐지 학습모델 개발

# Development of a Flooding Detection Learning Model Using CNN Technology

김동준\*\*, 최유진\*\*, 박경민\*\*, 박상준\*\*, 이재문\*, 황기태\*\*, 정인환\*\*

Dong Jun Kim\*\*, YU Jin Choi\*\*, Kyung Min Park\*\*, Sang Jun Park\*\*,  
Jae-Moon Lee\*, Kitae Hwang\*\*, Inhwan Jung\*\*

**요약** 본 논문은 인공지능 기술을 활용하여 일반 도로와 침수 도로를 분류하는 학습모델을 개발하였다. 다양한 데이터 증강기법을 사용하여 학습 데이터의 다양성을 확장하며, 여러 환경에서도 좋은 성능을 보이는 모델을 구현하였다. CNN 기반의 Resnet152v2 모델을 사전 학습모델로 활용하여, 전이 학습을 진행하였다. 모델의 학습 과정에서 다양한 파라미터 튜닝 및 최적화 과정을 거쳐 최종 모델의 성능을 향상하였다. 학습은 파이선으로 Google Colab NVIDIA Tesla T4 GPU를 사용하여 구현하였고, 테스트 결과 시험 데이터 세트에서 매우 높은 정확도로 침수상황을 탐지함을 알 수 있었다.

**Abstract** This paper developed a training model to classify normal roads and flooded roads using artificial intelligence technology. We expanded the diversity of learning data using various data augmentation techniques and implemented a model that shows good performance in various environments. Transfer learning was performed using the CNN-based Resnet152v2 model as a pre-learning model. During the model learning process, the performance of the final model was improved through various parameter tuning and optimization processes. Learning was implemented in Python using Google Colab NVIDIA Tesla T4 GPU, and the test results showed that flooding situations were detected with very high accuracy in the test dataset.

**Key Words** : Flooded Road, Deep Learning, Image Classification, CNN

### 1. 서 론

도시화와 기후 변화로 인한 갑작스러운 폭우는 도로 침수와 같은 문제를 발생시키고 있다. 침수로 인한 문제는 사회적, 경제적으로 막대한 손실을 초래하며 해를 거듭하며 증가하고 있다. 2023년, 미흡한 대응으로 14명

의 사상자를 낸 '오송 지하차도 침수 사고'는 그 심각성을 말해 준다.

최근 인공지능 기술은 혁신적으로 발전해왔다. 이는 단순히 기술적인 발전을 넘어서 사회의 거의 모든 분야에서 그 효용성을 발휘하고 있다. 최근 몇 년 동안 다양한 산업 분야에서 인공지능의 적용 사례가 눈에 띄게 증

\*정회원, 한성대학교 컴퓨터공학부(교신저자)

\*\*정회원, 한성대학교 컴퓨터공학부

접수일자 2023년 10월 3일, 수정완료 2023년 11월 13일

게재확정일자 2023년 12월 8일

Received: 3 October, 2023 / Revised: 13 November, 2023 /

Accepted: 8 December, 2023

\*Corresponding Author: jmlee@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

가하고 있으며, 특히 딥 러닝 기반의 이미지 인식 기술은 다양한 분야에서 높은 성능을 보여주고 있다.

이 기술을 바탕으로, 본 연구에서는 CNN 기반의 Resnet152v2 모델<sup>[1]</sup>을 활용하여 '침수탐지 딥 러닝 모델'을 개발하였다. 모델은 이미지 분류 기술(Image Classification)<sup>[2, 3]</sup>을 통해 일반 도로와 침수 도로 이미지의 패턴을 학습하여, 다양한 환경에서 정상상황과 침수 상황을 신속하게 구별한다.

2장에서는 관련 연구를 소개하며, 3장에서는 침수탐지 모델의 구현과정을 상세하게 설명한다. 4장에서는 학습과 검증 과정, 그리고 그 결과를 분석하고, 마지막으로 5장에서 본 연구의 결론과 향후 연구 방향에 대해 논한다.

## II. 관련 연구

### 1. CNN 기술

CNN(Convolutional Neural Network)은 딥 러닝 분야에서 이미지 인식 및 분류에 주로 사용되는 알고리즘이다. 즉, 컴퓨터 비전에서 일반적으로 사용되는 딥 러닝 신경망 아키텍처라 할 수 있다. CNN은 다양한 필터를 통해 이미지의 지역적 특징을 추출하며, 이러한 특징들을 기반으로 고수준의 패턴을 학습한다. 1980년대 손으로 쓴 숫자를 인식하는 시스템 LeNet-5를 기점으로 CNN의 기본 개념이 도입되었으며, 2010년대 여러 개의 컨볼루션 레이어와 완전 연결 레이어가 적용된 AlexNet이 ILSVRC (ImageNet Large Scale Visual Recognition Challenge)<sup>[4]</sup>에서 우승하며 CNN에 관한 본격적인 연구가 시작되었다. 현재는 이미지 분류, 이미지 생성, 객체 검출 등 다양한 분야에서 핵심 기술로 사용되며 그 범위를 넓혀가고 있다.

### 2. Resnet152v2 모델

Resnet152v2<sup>[1]</sup>는 Residual Network의 변형으로, 레이어가 많아져 깊어진 신경망에서 발생할 수 있는 경사 소실 문제를 해결하기 위해 고안되었다. 152개의 레이어로 구성되어 있어, 복잡한 패턴의 학습에 유용하며, 기존 Resnet152에서 일부 레이어의 순서와 활성화 함수의 위치를 변경하여 성능을 개선한 변형 버전이다. Resnet152v2는 스킵 연결(skip connection)을 활용하여 그래디언트의 흐름을 개선하며, 이를 통해 더 깊은 네트워크에서도 안정적인 학습이 가능하다. 깊은 네트워크

구조로 인해 복잡한 이미지 패턴까지도 높은 정확도로 인식이 가능하며, 이미지 관련 분야에서 널리 사용되고 있다.

### 3. 구글 코랩 노트북

코랩 노트북은 구글이 제공하는 클라우드 기반 Jupyter 노트북 서비스<sup>[5]</sup>이다. 이 서비스는 다량의 데이터 기반 작업에 높은 연산 능력을 지닌 GPU와 TPU를 무료로 제공해주어, 딥 러닝 및 데이터 연구자들에게 인기가 많다. 협업 관련 도구와 플러그인을 제공하여 협업에 용이하며, 별도의 라이브러리와 패키지 설치 없이 다양한 머신러닝 프레임워크를 사용할 수 있다. 제공 리소스로는 NVIDIA의 Tesla T4<sup>[6]</sup> 및 A100과 V100 GPU<sup>[7]</sup>가 있으며, 딥 러닝 작업에서 중요한 벡터 및 행렬 연산에 특화된 TPU가 있다. 또한, Google의 클라우드 기반 파일 저장 시스템인 구글 드라이브(Google Drive)와 연동하여 작업 과정에서 생성된 데이터와 결과물 관리에 높은 효율성을 보인다. 딥 러닝에 최적화된 작업환경 제공, 다양한 GPU 리소스 제공, 효율적인 협업환경 제공을 통해 개인, 팀 단위 연구자들 사이에서 널리 사용되고 있다.

## III. 침수탐지 학습모델 구현

이미지 분류 기술(Image classification)은 주어진 이미지의 정보 및 패턴을 분석하여, 어떤 카테고리에 속하는지 판단하는 인공지능 기술이다. 본 연구에서는 이미지 분류 기술을 이용하여, 일반 도로 표면과 침수 도로 표면을 분류하는 침수탐지 학습모델을 구현하였다.

### 1. 학습 환경

데이터 세트 정제 작업 및 침수탐지 모델의 학습 작업은 구글 코랩 노트북에서 진행하였다. GPU는 NVIDIA Tesla T4를 사용하였으며, 프로그래밍 언어는 Python (3.10.12)을 사용하였다. 딥 러닝 프레임워크는 Pytorch (2.0.1)를 선택하였다. Python과 호환성이 좋고, 학습 및 추론 속도가 빠르며 코드의 유연성과 직관성이 뛰어나 본 연구에 적합하다고 판단하여 Pytorch를 채택하였다. 전이 학습을 위한 사전 학습모델은 Resnet152v2<sup>[1]</sup>를 기반으로 학습을 진행하였다. 데이터 처리 및 이미지 변환 작업을 위해 여러 외부 라이브러리를 사용하였다. 주요 사용 라이브러리는 데이터 분석 및 처리를 위한

Pandas(1.5.3), 이미지 처리와 변환을 위한 Torchvision (0.15.2)과 Pillow(8.4.0), 수학적 계산을 위한 Numpy (1.25.2), 그리고 이미지 처리를 위한 opencv(4.8.0.74) 와 Imgaug(0.4.0) 등의 라이브러리를 사용하였다.

## 2. 학습 데이터 준비

일반 도로 표면과 침수 도로 표면을 구분하는 이미지 분류 모델 생성을 위해, 데이터 제공 플랫폼인 Kaggle의 'Road Classification images'<sup>[8]</sup>, 'Roadway Flooding Image Dataset'<sup>[9]</sup> 데이터를 활용했다. 더불어, 구글 웹을 통해 이미지 데이터를 추가하여 기본 이미지 데이터 세트를 구축했다. 학습에 필요한 이미지는 도로 표면이기에, PhotoScape 툴을 이용하여 필요한 정보만을 추출하였다. 그림 1은 표면 추출 과정의 예시이다. 추출 결과, 725장의 일반 도로 이미지와 182장의 황토 침수 도로 이미지, 173장의 비 황토 침수 도로 이미지를 확보하였다. 침수 도로는 황토가 섞인 황토 침수 도로와 황토가 섞이지 않은 비 황토 침수 도로 2가지 카테고리로 구분하였다.



그림 1. 도로 표면 추출 예시  
 Fig. 1. Example of road surface extraction

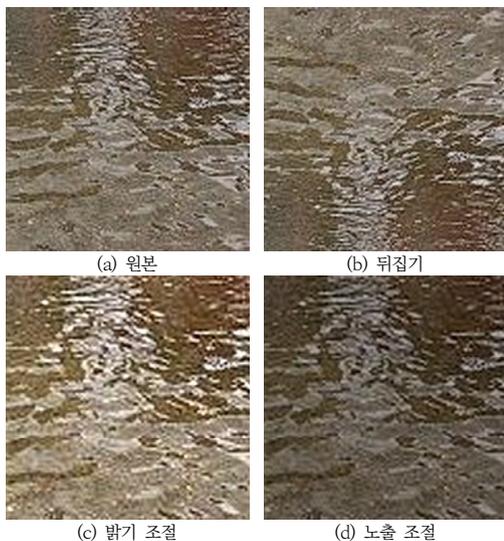


그림 2. 이미지 증폭에 대한 예  
 Fig. 2. Example of image augmentation

모델의 성능을 향상하기 위해 학습 데이터의 양과 다양성을 확장하는 데이터 증강기법<sup>[10]</sup>을 적용하였다. 우선, 이미지 데이터를 320 \* 320 픽셀 크기로 조정하여 이미지 크기에 대한 통일성을 부여하고, 신경망에 입력되는 크기를 일정하게 유지하였다. 상황에 따라 다양하게 연출되는 도로 표면 상태를 반영하기 위해, 뒤집기 (Flip), 밝기 조절(Brightness control), 노출 조절 (Exposure control), 회전(Rotation) 데이터 증강 (Data Augmentation)기법을 적용하였다. 이를 통해, 최종적으로 일반 도로 2,175장, 황토 침수 도로 1,456장, 비 황토 침수 도로 1,384장의 이미지 데이터를 확보하였다. 그림 2는 원본 이미지(a)에 대한 뒤집기(b), 밝기 조절(c) 및 노출 조절(d)에 대한 예이다.

증강기법을 통한 데이터 확장은 imgaug와 numpy 라이브러리를 통해 구현하였다. 그림 3은 증강기법을 구현한 Python 코드의 일부이다.

```
import imgaug.augmenters as iaa
import numpy as np

def augment_images(images):
    # 모든 이미지를 320x320 크기로 변경
    resize = iaa.Resize((320, 320))
    # 수평 뒤집기
    fliplr = iaa.Fliplr(1.0)
    # 수직 뒤집기
    flipud = iaa.Flipud(1.0)
    # 밝기 조절 (-30% ~ +30%)
    multiply = iaa.Multiply((0.7, 1.3))
    # 노출 조절 (-10% ~ +10%)
    contrast = iaa.LinearContrast((0.9, 1.1))

    augmented = []
    for image in images:
        image_resized = resize(image=image)
        for angle in [0, 90, 180]:
            rotate = iaa.Rotate(angle)
            rotated_image = rotate(image=image_resized)
            # 증강 기법 적용
            augmented.append(fliplr(image=rotated_image))
            augmented.append(flipud(image=rotated_image))
            augmented.append(multiply(image=rotated_image))
            augmented.append(contrast(image=rotated_image))

    return np.array(augmented)
```

그림 3. 데이터 증강 구현 코드  
 Fig. 3. Data augmentation implementation code

증강을 마친 전체 데이터 세트를 훈련(train), 검증(validation), 테스트(test) 세트로 분할 하였다. 모델 학습에 사용되는 훈련 세트는 80%, 학습 중 모델 성능을 평가하여 하이퍼 파라미터를 유동적으로 조정하고 일반화 능력을 향상하는 검증 세트는 10%, 학습이 완료된 모델의 최종 성능을 평가하는 테스트 세트에는 10%의 비

율로 분할 하였다.

다량의 이미지 데이터 세트와 해당하는 레이블 정보를 학습 과정에서 효율적으로 추출하기 위해 csv파일을 생성하였다. csv파일의 각 행은 하나의 이미지 데이터를 나타내며, filename(이미지 파일 이름), common\_road(일반 도로), mud\_flooded\_road(황토 침수 도로), nonmud\_flooded\_road(비 황토 침수 도로) 네 개의 열(column)로 구성되어 있다. 이미지별로 해당하는 클래스 정보를 one-hot 인코딩 형식의 레이블로 제공하여 다중 클래스 분류 작업에 효율성을 높였다. 그림 4는 csv 파일의 예시이다.

	A	B	C	D
1	filename	common_road	mud_flooded_road	nonmud_flooded_road
2	common_road_301_1.jpg	1	0	0
3	nonmud_flooded_road_5_0.jpg	0	0	1
4	mud_flooded_road_14_6.jpg	0	1	0
5	mud_flooded_road_40_4.jpg	0	1	0
6	nonmud_flooded_road_12_5.jpg	0	0	1

그림 4. csv 구성 예시

Fig. 4. Example of csv configuration

### 3. 학습 진행

모델 학습의 안정성과 성능 향상을 위해 데이터 전처리를 진행하였다. Pytorch의 torchvision.transforms 모듈을 이용하여 이미지 데이터를 Tensor 형태로 변환하였고, 각 채널에 대해 정규화를 진행하여 데이터의 분포를 안전화 시켰다. 그 후, ResNet152v2 모델의 사전 학습된 가중치를 가져와 본 연구에 맞게 재정의하였다. 모델의 기존 출력 계층을, '일반, 황토 침수, 비 황토 침수' 총 3개의 클래스를 분류하는 완전 연결 계층으로 변경하였다. 또한 드랍 아웃(Dropout) 기법을 이용하여 과적합 현상을 방지하였다. 드랍 아웃 비율로 0.3을 설정하여 각 학습 단계에서 신경망의 30% 뉴런이 비활성화되고, 이는 모델이 특정 뉴런에 과도하게 의존하는 것을 방지하여 일반화 성능을 개선한다.

손실 함수로는 다중 클래스 분류 작업에 널리 사용되는 교차 엔트로피를 선택하였다. 옵티마이저는 Adam을 사용하였으며, L2 정규화를 통해 과적합을 방지하였다. 추가로, ReduceLROnPlateau 스케줄러를 통해 학습 도중 학습률을 동적으로 조정하여 학습의 효율성을 높였다. 그림 5는 최소의 검증 손실을 가지는 학습 모델을 구하는 알고리즘이다.

파라미터는 배치 크기(Batch), 학습률(Learning rate), 에포크(Epochs) 값들을 조정하며 그림 5의 알고리즘을 3번 진행하였다. 32 배치 크기, 30 에포크, 0.001 학습

```
# 학습을 진행할 때 사용할 파라미터들을 설정
batch_size, epochs, learning_rate 초기화

# 데이터 세트 로딩
train_loader, val_loader, test_loader =
DataLoader(batch_size)

# ResNet152 모델을 사전 학습된 가중치와 함께 불러오기
model = resnet152(pretrained=True).to(device=GPU)

for epoch in range(epochs):

    model.train() # 모델을 학습 모드로 설정
    train_loss = 0.0
    for images, labels in train_loader:
        train_loss += calculateTrainLoss(model, images, labels)

    # 에포크별 평균 학습 손실 계산
    avg_train_loss = train_loss / len(train_loader)

    model.eval() # 모델을 평가 모드로 설정
    val_loss = 0.0
    for images, labels in train_loader:
        val_loss += calculateTrainLoss(model, images, labels)

    # 에포크별 평균 검증 손실 계산
    avg_val_loss = val_loss / len(val_loader)
    val_losses.append(avg_val_loss)

    # 최소 검증 손실 모델을 저장
    if avg_val_loss < best_val_loss:
        best_val_loss = avg_val_loss
        torch.save(model.state_dict(), 'best_model_weights.pth')
```

그림 5. 최소 검증 손실 학습모델을 구하는 알고리즘

Fig. 5. Algorithm to find the minimum verification loss learning model

률을 설정하여 1차 학습을 진행한 결과, 후반부 에포크에서는 정확도의 상승이 멈추며, 손실 값이 올라가는 현상을 보였다. 2차 학습에선 이를 개선하기 위해 에포크를 25로, 배치 크기를 64로 조정하였다. 후반부 에포크의 학습상황은 개선되었지만, 여전히 소량의 손실 값 증가 현상을 보였다. 또한, 학습 중반부에 손실 값이 진동하는 현상이 나타났다. 3차 학습에는 손실 값 안정화를 위해 배치 크기를 16으로 줄이고, 에포크를 20으로 설정하였다. 또한, 유의미한 정확도 상승 폭을 보이지 못한 2차 학습의 결과를 반영하여 더 정밀한 학습을 위해 학습률을 0.0001으로 감소시켰다. 그 결과, 손실 값 진동 현상은 개선되었고 정확도는 최고 수치를 도출하였다. 훈련 손실(Train Loss)과 검증 손실(Validation Loss)은 각각 0.0032, 0.0076 값을 기록했으며 정확도는 0.998 값을 도출하였다. 이 결과는 모델이 훈련 데이터에 충분히 적응하면서도, 검증 데이터에서도 높은 성능을 보이고 있다는 것을 의미한다. 그림 6은 3차 학습에서 에포크가 진행됨에 따른, 손실 값의 변화를 나타낸 그래프이다.

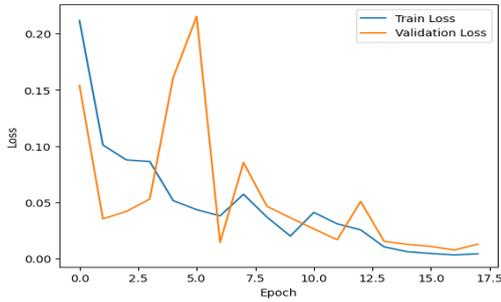


그림 6. 손실 값의 변화  
 Fig. 6. Change of loss values

#### IV. 모델 검증

표 1은 학습 과정에서 에포크별로 변화하는 모델의 성능을 정밀도(Precision), 재현율(Recall), F1 Score 지표를 통해 보인다. 정밀도와 재현율 간의 균형을 관찰하였을 때, 두 지표 모두 에포크가 진행됨에 따라 점차 증가하는 모습을 보였다. 이는 모델이 학습에 진행됨에 따라, 각 클래스의 이미지를 잘 분류하고 있음을 나타낸다. 특히, 정밀도와 재현율의 조화평균 지표인 F1 Score가 점차 증가하는 추세를 보임으로써, 정밀도와 재현율이 동시에 향상되고 있음을 확인할 수 있다.

표 1. 정밀도, 재현율, F1 Score 변화  
 Table 1. Change of precision, recall, F1 score

Epoch	Precision	Recall	F1 score
1	0.9787	0.9780	0.9783
5	0.9881	0.9880	0.9880
10	0.9980	0.9980	0.9980
15	0.9995	0.9990	0.9992
20	0.9980	0.9980	0.9980

시험 데이터 세트에 대한 손실 값과 정확도 값을 측정 한 결과, 0.0178 손실 값과 99.21%의 정확도를 보였다. 이는 모델이 테스트 데이터에 대해 매우 정확한 분류를 수행했다는 것을 의미하며, 일반화 능력이 우수하여 다양한 환경에서의 침수 감지에 효과적으로 활용될 수 있음을 확인할 수 있다. 추가로, 혼동행렬(confusion matrix)을 통한 검증을 진행하였다. 그 결과, 대부분의 이미지가 올바르게 분류되었으며, 소수의 이미지만 오 분류 되었음을 확인하였다. 특히, 일반 도로에 대한 예측

은 완벽하게 이루어졌다. 황토 침수 도로와 비 황토 침수 도로에 대한 예측에서 각각 3, 1개의 오류가 발생했으나 전체적으로 높은 정확도를 보였다. 이러한 결과는 모델의 신뢰성과 안정성을 더욱 강조하며, 실제 환경에서의 높은 성능을 기대하게 한다.

표 2. 시험 데이터 세트에 대한 혼돈 행렬  
 Table 2. Confusion matrix for testing data set

	예측 일반 도로	예측 황토 침수 도로	예측 비황토 침수 도로
실제 일반 도로	218	0	0
실제 황토 침수 도로	2	144	1
실제 비황토 침수 도로	1	0	138

#### V. 결론

본 논문에서는 Resnet152v2 모델을 기반으로 한 CNN의 전이 학습 방법을 통해 ‘침수탐지 딥러닝 모델’을 개발하였다. 데이터 세트 구축 단계에서 적용한 증강 기법은 제한된 데이터 범위와 다양성을 확장 시키는 데 중요한 역할을 하였으며, 이를 통해 다양한 환경에서도 침수상황을 정확히 탐지할 수 있는 모델을 성공적으로 구현하였다.

그러나 본 모델에는 몇 가지 한계점이 존재한다. 첫째로, 모델은 도로의 전체적인 상황을 파악하는 것이 아니라 도로의 일부 표면만을 타겟으로 모니터링해야 하는 제약 조건이 있다. 이러한 특성 때문에, 전반적인 도로 상황을 완벽히 파악하기 어려울 수 있다. 둘째로, 자동차나 다른 이물질과 같은 외부 변수가 개입될 경우, 모델의 분류 결과의 정확성이 불안정해질 수 있다. 이러한 변수들은 예측되지 않은 방식으로 모델의 분류 결과에 영향을 줄 수 있으므로, 실제 환경에서의 적용 시 이러한 한계점을 고려해야 한다. 이러한 문제를 해결하기 위해, 이미지 분할(Image Segmentation) 방식의 모델을 도입함으로써 전반적인 도로의 상황을 더욱 정밀하게 파악하거나 더 다양한 데이터 세트를 구축해 여러 변수에 대응하는 모델을 개발하는 것이 향후 목표이다.

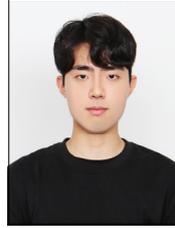
최종적으로 이 연구는 침수상황 탐지에 이미지 분류 기술을 적용하여 새로운 접근 방식을 제시하였다. 이를 통해, 침수상황의 신속한 감지 및 대응 시스템의 효율성을 높일 수 있는 발판을 마련하였다고 볼 수 있다.

## References

- [1] Hidayat, Taopik, Daniati Uki Eka Saputri, and Faruq Aziz. "MEAT IMAGE CLASSIFICATION USING DEEP LEARNING WITH RESNET152V2 ARCHITECTURE", Jurnal Techno Nusa Mandiri, Vol. 19, No. 2 pp.131-140, 2022.
- [2] Do-Eun Kim, Hee-Jin Kong, Ji-Hu Woo, Jae-Moon Lee, Kitae Hwang, Inhwan Jung, "Development of Intelligent CCTV System Using CNN Technology", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 23, No. 4, pp.99-105, 2023.
- [3] Dong Jun Kim, Yu Jin Choi, Kyung Min Park, Ji Hyun Park, Jae-Moon Lee, Kitae Hwang, In Hwan Jung, "Detection of Smoking Behavior in Images Using Deep Learning Technology", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 23, No. 4, pp.99-105, 2023.
- [4] Verma, Gaurav, et al. "Performance evaluation of deep learning compilers for edge inference." 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2021.
- [5] Bisong, Ekaba, and Ekaba Bisong. "Google colabatory", Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners, pp.59-64, 2019.
- [6] Jahanshahi, Ali, et al. "Gpu-nest: Characterizing energy efficiency of multi-gpu inference servers." IEEE Computer Architecture Letters, Vol.19, No. 2, pp.139-142, 2022.
- [7] Martineau, Matt, Patrick Atkinson, and Simon McIntosh-Smith. "Benchmarking the nvidia v100 gpu and tensor cores", European Conference on Parallel Processing, Cham: Springer International Publishing, 2018.
- [8] Gresenz, Gabriela, Jules White, and Douglas C. Schmidt. "An off-road terrain dataset including images labeled with measures of terrain roughness", 2021 IEEE International Conference on Autonomous Systems (ICAS). IEEE, 2021.
- [9] Sazara, Cem, Mecit Cetin, and Khan M. Iftekharuddin. "Detecting floodwater on roadways from image data with handcrafted features and deep transfer learning", 2019 IEEE intelligent transportation systems conference (ITSC). IEEE, 2019.
- [10] Han, Dongmei, Qigang Liu, and Weiguo Fan. "A new image classification method using CNN transfer learning and web data augmentation", Expert Systems with Applications, 95, pp.43-56, 2018.

## 저 자 소 개

### 김 동 준(비회원)



- 한성대학교 컴퓨터공학부 재학
- 딥러닝, 기계학습, 컴퓨터 비전

### 최 유 진(비회원)



- 한성대학교 컴퓨터공학부 재학
- 딥러닝, iOS

### 박 경 민(비회원)



- 한성대학교 컴퓨터공학부 재학
- 딥러닝, 게임 프로그래밍, iOS

### 박 상 준(비회원)



- 한성대학교 컴퓨터공학부 재학
- 인공지능, 모바일 시스템

### 이 재 문(정회원)



- 한양대학교 전자공학과(학사)
- 한국과학기술원 전기및전자공학과 (석사)
- 한국과학기술원 전기및전자공학과 (박사)
- 경력 : 한국통신 연구개발단
- 관심분야 : 기계학습, 게임프로그래밍, 감성컴퓨팅

황 기 태(정회원)



- 서울대학교 컴퓨터공학과 학사  
서울대학교 컴퓨터공학과 석사  
서울대학교 컴퓨터공학과 박사
- 경력  
University of Callifornia, Irvine  
방문교수  
University of Florida 방문 교수
- 주관심분야 : 모바일 시스템

정 인 환(정회원)



- KAIST 정보및통신공학과 박사
- 삼성전자 시스템사업부 수석연구원
- 한성대학교 컴퓨터공학과 교수
- 주관심분야 : 망관리, 멀티미디어신, IoT

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음