

YOLOv5 기반 객체 인식과 딥러닝을 통한 아동 그림의

심리 상태 예측 모델 개발

AI 융합학부 AI 학과 20221426 정여원

성신여자대학교

## 요약

본 연구는 아동 그림을 분석하여 심리 상태를 자동으로 파악하는 알고리즘을 개발하는데 초점을 맞추고 있습니다. YOLOv5 모델을 사용하여 그림에서 나무, 사람 등의 객체를 탐지하였으며, CNN과 전이 학습 모델을 통해 탐지된 객체를 분류했습니다. 탐지된 객체의 특징을 바탕으로 랜덤 포레스트와 LightGBM 모델을 활용하여 더욱 정밀한 객체 분류를 수행하였습니다. 또한, 색상 정보 추출을 위해 히스토그램 기반 방법을 사용하여 아동 그림의 색상 분포를 분석했습니다. K-means 클러스터링을 이용하여 객체의 위치와 색상 정보를 각각 분석하였으며, 이를 통해 아동의 내면 세계를 정밀하게 이해하고, 심리적 문제를 조기에 발견하는 데 기여할 수 있는 도구를 제공하고자 합니다. 향후 연구에서는 더욱 다양한 데이터셋과 앙상블 모델을 통한 예측 성능 향상을 목표로 하고 있습니다. 본 연구는 아동의 정신 건강 평가와 치료에 중요한 기여를 할 수 있을 것으로 기대됩니다.

키워드: 아동 그림 분석, 심리 상태 예측, YOLOv5, CNN, 전이 학습, K-means 클러스터링, 랜덤 포레스트, LightGBM, 색상 히스토그램, 객체 탐지

## 1. 서론

### 1.1 연구 배경 및 동기

아동의 그림은 그들의 내면 세계를 표현하는 중요한 수단입니다. 특히, 아동들은 언어적 표현이 미숙한 경우가 많아, 그림을 통해서 감정, 욕구, 성격 등을 드러낼 수 있습니다. 이러한 그림들은 아동의 심리 상태를 직관적으로 반영할 수 있는 도구로 사용될 수 있습니다. 예를 들어, 특정 색상의 사용 빈도나 그림에 등장하는 객체의 배치 등이 아동의 감정 상태나 심리적 특징을 암시할 수 있습니다. 그림에 사용된 색상과 구성이 아동의 심리 상태를 반영할 수 있다는 점에서, 이러한 요소들을 분석하는 것은 아동의 정신 건강을 평가하고 치료하는 데 큰 의미가 있습니다. 색상은 아동의 감정과 심리 상태를 표현하는 중요한 요소이므로, 컬러 이미지를 활용한 심리 분석이 더욱 필요 해졌습니다. 이 연구는 아동 그림의 색상 정보와 더불어 객체 배치를 분석하여 심리 상태를 예측합니다. 이를 통해 아동의 정신 건강을 보다 정확하고 신속하게 평가할 수 있는 도구를 개발하고자 합니다. 이러한 도구는 심리 치료사나 교육자가 아동의 심리 상태를 이해하고 적절한 대응을 하는 데 큰 도움을 줄 수 있을 것입니다. 컬러 이미지를 활용한 분석을 통해 기존의 흑백 이미지 분석의 한계를 극복하고, 보다 풍부한 정보를 기반으로 한 심리 상태 예측 모델을 개발하는 것이 이 연구의 주요 목표입니다. 이를 통해 아동의 그림이 지닌 심리적 의미를 더욱 깊이 있게 이해하고, 이를 바탕으로 한 실질적인 응용 가능성을 제시하고자 합니다.

### 1.2 연구 목적

본 연구의 목적은 아동 그림을 분석하여 심리 상태를 자동으로 파악하는 알고리즘을 개발하고, 그 성능을 평가하는 것입니다. 이를 위해 다음과 같은 세부 목표를 설정하였습니다. 첫째, YOLOv5 모델을 사용하여 아동 그림에서 나무, 집, 사람

등 특정 객체를 탐지하고, 각 객체의 위치를 파악합니다. 이를 통해 그림에 나타난 각 객체의 위치가 지닌 심리적 의미를 해석할 수 있는 기반을 마련합니다. 둘째, 탐지된 객체를 제대로 분석하기 위해 다양한 딥러닝 및 머신러닝 기법을 적용합니다. CNN, Transfer Learning, Clustering Techniques, Clustering Techniques, Random Forest, LightGBM 등을 사용하여 객체의 특성을 분석하고, 이를 통해 객체의 심리적 의미를 도출합니다. 셋째, 색상 히스토그램 분석을 통해 그림에서 사용된 색상 정보를 추출합니다. 색상은 아동의 감정과 심리 상태를 반영하는 중요한 요소이므로, 이를 정량적으로 분석하여 아동의 심리 상태를 보다 정확하게 파악합니다. 이를 통해 아동의 정신 건강을 평가하고, 심리 치료와 교육 현장에서 활용할 수 있는 유용한 도구를 개발하고자 합니다. 이 연구의 결과는 아동의 그림을 통한 심리 상태 분석을 자동화함으로써, 아동의 정신 건강 평가를 위한 보다 효율적이고 정확한 방법을 제공할 것입니다. 또한, 개발된 알고리즘의 성능을 Precision, Recall, F1-Score 등의 지표를 통해 평가하여 그 유효성과 신뢰성을 검증할 예정입니다.

### 1.3 연구의 중요성

본 연구는 아동의 심리 상태를 실시간으로 분석할 수 있는 도구를 제공하여, 아동의 정신 건강 평가와 치료에 크게 기여할 수 있습니다. 특히, 아동들은 자신의 감정을 명확히 표현하는 데 어려움을 겪는 경우가 많아, 그림과 같은 비언어적 표현 수단이 심리 상태를 이해하는 데 중요한 역할을 합니다. 본 연구는 이러한 아동 그림을 심층적으로 분석함으로써 아동의 내면 세계를 보다 정확하게 파악할 수 있는 방법을 제시합니다. 특히, 본 연구에서 개발되는 알고리즘은 색상 정보와 객체의 위치 정보를 결합한 통합 모델을 통해 더욱 정밀한 심리 상태 예측이 가능합니다. 객체의 위치는 아동의 그림에서 공간적 배치를 통해 심리적 의미를 전달하며, 색상 정보는 감정

상태를 반영합니다. 이러한 다중 요소를 종합적으로 분석함으로써 아동의 심리 상태를 다각도로 이해할 수 있습니다. 이 연구는 아동의 심리적 문제를 조기에 발견하고, 적절한 치료를 제공하는 데 중요한 역할을 할 수 있습니다. 조기에 발견된 심리적 문제는 보다 효과적으로 개입할 수 있으며, 이는 아동의 장기적인 정신 건강과 발달에 긍정적인 영향을 미칠 수 있습니다. 예를 들어, 우울증이나 불안과 같은 문제를 초기 단계에서 식별하고 개입함으로써, 아동이 더 건강한 성인으로 성장할 수 있는 토대를 마련할 수 있습니다. 또한, 본 연구에서 개발된 도구는 심리 치료사나 교육자가 아동의 심리 상태를 보다 쉽게 이해하고, 이를 바탕으로 맞춤형 치료 계획을 수립하는 데 도움을 줄 수 있습니다. 이는 아동의 개별적 필요에 맞춘 보다 개인화된 접근을 가능하게 하여, 치료 효과를 극대화할 수 있습니다. 결론적으로, 본 연구는 아동 그림 분석을 통해 아동의 심리 상태를 실시간으로 평가할 수 있는 혁신적인 도구를 제공함으로써, 아동의 정신 건강 평가와 치료에 중요한 기여를 할 것입니다. 이 연구는 아동의 심리적 문제를 조기에 발견하고, 적절한 개입을 통해 아동의 전반적인 정신 건강과 삶의 질을 향상시키는 데 필수적인 역할을 할 것입니다.

#### 1.4 연구의 구성

본 연구는 아동 그림의 심리 상태를 자동으로 파악하는 알고리즘을 개발하기 위해 다음과 같은 단계로 구성됩니다: 1) 데이터 수집 및 전처리 - AI Hub 에서 무채색 아동 그림 데이터를 수집하고, 추가로 컬러 이미지를 수집하여 색상 분석을 위한 데이터셋을 구성합니다. 2) 객체 탐지 모델 개발 - YOLOv5 모델을 사용하여 아동 그림에서 특정 객체를 탐지하고 각 객체의 위치를 파악합니다. 3) 객체 분석 모델 개발 - CNN, Transfer Learning, Clustering Techniques, Random Forest, LightGBM 과 같은 다양한 딥러닝 및 머신러닝 기법을 적용하여 탐지된 객체의 특성을 분석합니다. 4)

색상 분석 - 수집된 컬러 이미지를 사용하여 색상 히스토그램을 생성하고 주요 색상을 추출합니다. K-means 클러스터링을 사용하여 색상 정보를 군집화합니다. 5) 심리 상태 예측 모델 개발 - 객체의 위치 정보와 색상 분석 결과를 종합하여 최종적으로 아동의 심리 상태를 예측합니다. 6) 성능 평가- Precision, Recall, F1-Score 등의 정량적 지표를 사용하여 개발된 알고리즘의 성능을 평가합니다. 7) 결론 도출 및 제언 - 본 연구의 주요 성과를 요약하고 아동 그림의 심리 상태 분석에 대한 결론을 도출합니다.

## 2. 문제 제시

### 2.1 아동 미술 작품의 심리적 의미

아동의 그림은 그들의 감정, 욕구, 성격 등을 표현하는 중요한 수단입니다. 언어적 표현이 미숙한 아동들은 그림을 통해 자신이 느끼는 다양한 감정을 드러낼 수 있으며, 이는 심리 상태를 파악하는 중요한 단서가 됩니다. 아동이 사용하는 색상과 그림의 구성 요소는 무의식적으로 내면 심리를 반영하는 경우가 많습니다. 예를 들어, 아동이 빨간색을 많이 사용하는 경우, 이는 에너지가 넘치고 활발한 성격을 나타낼 가능성이 큼니다. 빨간색은 흥분, 열정, 강한 에너지와 관련이 있습니다. 반면에, 어두운 색상을 주로 사용하는 아동은 불안감이나 스트레스를 나타낼 수 있습니다. 검정색이나 회색과 같은 어두운 색상은 두려움, 슬픔, 우울감 등을 반영할 수 있습니다. 또한, 다른 색상들도 각각의 심리적 의미를 지닙니다. 파란색: 차분함, 안정, 신뢰를 나타내며, 때로는 우울함을 반영할 수 있습니다. 초록색: 평화, 성장, 균형을 상징하며, 자연과의 연결을 나타낼 수 있습니다. 노란색: 행복, 창의성, 낙천성을 의미하며, 밝고 긍정적인 감정을 표현합니다. 보라색: 상상력, 영성, 신비로움을 나타내며, 감수성과 창의력을 반영할 수 있습니다. 그림의 구성 요소 또한 중요한 심리적 의미를 지닙니다. 예를 들어, 그림에 자주 등장하는 특정 객체나 그 객체들의 배치는 아동의 내면 욕구나

관심사를 반영할 수 있습니다. 아동이 자주 그리는 가족 구성원이나 특정 사물은 그들이 느끼는 애정, 관심, 두려움 등의 감정을 표현할 수 있습니다. 따라서, 아동 미술 작품의 심리적 의미를 분석하는 것은 아동의 내면 세계를 이해하고, 심리적 상태를 평가하는 데 매우 중요한 역할을 합니다. 본 연구에서는 이러한 심리적 의미를 보다 체계적이고 정량적으로 분석하기 위해, 객체 탐지와 색상 분석을 결합한 알고리즘을 개발하고자 합니다. 이를 통해 아동의 정신 건강을 평가하고, 필요한 경우 적절한 개입과 지원을 제공할 수 있는 기반을 마련하고자 합니다.

## 2.2 컬러 이미지 분석의 필요성

기존의 연구에서는 주로 흑백 이미지를 사용하여 아동의 심리 상태를 분석해왔습니다. 그러나 흑백 이미지는 색상 정보를 반영하지 못하므로, 아동의 심리 상태를 정확하게 분석하는 데 한계가 있습니다. 색상은 아동의 감정과 심리 상태를 표현하는 중요한 요소로, 흑백 이미지 분석만으로는 이러한 정보를 충분히 활용할 수 없습니다. 컬러 이미지를 사용하면, 그림에 사용된 색상을 분석하여 아동의 심리 상태를 더 정확하게 파악할 수 있습니다. 색상은 아동의 감정, 성격, 욕구 등을 표현하는 데 중요한 역할을 합니다. 예를 들어, 밝고 따뜻한 색상은 긍정적이고 활발한 감정을 나타내는 반면, 어두운 색상은 부정적이고 불안한 감정을 나타낼 수 있습니다. 이러한 색상 정보를 분석함으로써 아동의 심리 상태를 보다 정밀하게 이해할 수 있습니다. 컬러 이미지 분석의 구체적인 필요성은 다음과 같습니다. 첫째, 정확한 감정 파악입니다. 아동이 사용하는 색상은 그들의 현재 감정 상태를 반영합니다. 컬러 이미지를 분석하면 아동의 긍정적, 부정적 감정 상태를 더 정확하게 파악할 수 있습니다. 둘째, 심리적 변화 감지입니다. 시간이 지남에 따라 아동의 그림에서 색상 사용의 변화는 심리적 상태의 변화를 나타낼 수 있습니다. 컬러 이미지 분석을 통해

이러한 변화를 감지하고, 조기에 개입할 수 있습니다. 셋째로, 개인화된 평가입니다. 각 아동은 색상에 대해 개인적인 선호와 의미를 가질 수 있습니다. 컬러 이미지 분석을 통해 아동 개개인의 색상 사용 패턴을 이해하고, 개인화된 심리 평가를 제공할 수 있습니다. 마지막으로 종합적 분석입니다. 색상 정보는 그림의 구성 요소와 결합하여 더 종합적인 심리 분석을 가능하게 합니다. 예를 들어, 특정 색상이 특정 위치에 반복적으로 사용된다면, 이는 특정 감정이나 생각의 패턴을 나타낼 수 있습니다. 따라서, 컬러 이미지를 활용한 심리 상태 분석은 아동의 정신 건강 평가에 중요한 역할을 할 수 있습니다. 본 연구에서는 컬러 이미지 분석을 통해 아동 그림의 색상 정보를 정량적으로 분석하고, 이를 바탕으로 아동의 심리 상태를 보다 정확하게 예측하는 알고리즘을 개발하고자 합니다. 이를 통해 아동의 정신 건강을 평가하고, 필요한 경우 적절한 개입과 지원을 제공할 수 있는 기반을 마련하고자 합니다.

## 2.3 기존 연구의 한계와 개선 방향

기존 연구의 한계는 다음과 같습니다. 첫째, 흑백 이미지를 사용하여 색상 정보를 분석하지 못한다는 점입니다. 색상은 아동의 감정과 심리 상태를 표현하는 중요한 요소임에도 불구하고, 흑백 이미지 분석만으로는 이러한 정보를 충분히 활용할 수 없습니다. 흑백 이미지는 아동의 그림에서 색상에 따른 감정 표현을 반영하지 못하여 심리 상태를 정확하게 분석하는 데 한계가 있습니다. 둘째, 객체 탐지와 색상 분석을 독립적으로 수행하여, 통합된 분석 결과를 제공하지 못한다는 점입니다. 기존 연구에서는 객체 탐지와 색상 분석을 개별적으로 수행하여, 두 가지 정보가 결합된 종합적인 심리 상태 분석을 제공하지 못했습니다. 이는 아동의 그림에서 객체의 배치와 색상 사용이 심리적 의미를 가지는 경우, 그 관계를 충분히 반영하지 못하게



합니다. 본 연구에서는 이러한 한계를 극복하기 위해 다음과 같은 개선 방향을 제시합니다.

1. 컬러 이미지를 사용한 색상 정보 분석

-아동의 그림에서 사용된 색상을 분석하여, 각 색상이 반영하는 감정과 심리 상태를 정밀하게 파악합니다. 이를 통해 아동의 심리적 특성을 보다 명확하게 이해할 수 있습니다.

2. YOLOv5 모델을 사용한 객체 탐지

-YOLOv5 모델을 사용하여 아동 그림에서 나무, 집, 사람 등 특정 객체를 탐지하고, 각 객체의 위치를 파악합니다. 이는 그림의 구성 요소와 배치가 심리적 의미를 가지는 경우, 이를 분석하는 데 도움을 줍니다.

3. 객체의 위치 정보와 색상 정보를 결합한 통합 모델 개발

-객체 탐지와 색상 분석을 결합하여, 아동의 심리 상태를 종합적으로 예측하는 통합 모델을 개발합니다. 탐지된 객체의 위치 정보와 색상 정보를 함께 분석함으로써, 아동의 그림이 지닌 심리적 의미를 더욱 정확하게 해석할 수 있습니다.

4. 정확한 심리 상태 예측

-CNN, Transfer Learning, Clustering Techniques 등을 사용하여 객체의 특성을 분석하고, 개별 모델의 분석 결과를 앙상블 기법을 통해 결합합니다. 이를 통해 보다 정확하고 신뢰성 있는 예측을 제공합니다.

이러한 개선 방향을 통해, 본 연구는 아동 그림의 심리 상태를 더욱 정밀하고 종합적으로 분석할 수 있는 새로운 접근법을 제안합니다. 이를 통해 아동의 정신

건강을 평가하고, 필요한 경우 적절한 개입과 지원을 제공할 수 있는 기반을 마련하고자 합니다.

### 3. 연구 방법

#### 3.1 데이터 수집

##### 3.1.1 AI Hub 데이터 수집

본 연구에서는 AI Hub 에서 제공하는 아동의 흑백 그림 데이터를 수집합니다. AI Hub 는 다양한 분야의 데이터를 제공하는 플랫폼으로, 아동의 미술 활동 데이터도 포함하고 있습니다. 이 데이터셋을 선택한 이유는 다음과 같습니다. AI Hub 데이터셋은 다양한 연령대(7~13 세)와 성별의 아동 7,000 명으로부터 수집한 56,000 건의 그림으로 구성되어 있어, 다양한 아동의 감정과 심리 상태를 반영하는 데 유용합니다. 또한, 집, 나무, 여자 사람, 남자 사람의 4 가지 주요 객체로 분류된 바운딩박스 라벨링이 포함되어 있어 심리 상태 분석에 필요한 정보를 제공합니다. 이는 원본 이미지 데이터에 대한 라벨링 전처리 시간을 줄여주어 연구 속도를 향상시킬 수 있습니다. 이 데이터셋은 56,000 건의 아동 그림 이미지(JPG 형식)와 각 이미지에 대한 바운딩박스 라벨링 데이터(JSON 형식)로 구성되어 있습니다. 메타데이터에는 이미지 ID, 수집 기관명, 수집일, 이미지 파일 경로, 라벨 파일 경로, 아동의 나이, 성별 등이 포함됩니다. 어노테이션 데이터는 객체 분류명, 바운딩박스 정보(좌표, 가로/세로 길이 등)를 포함하고 있으며, 이는 집, 나무, 여자 사람, 남자 사람에 대한 세부 객체 정보를 제공합니다. 이 데이터를 통해 아동 그림의 주요 객체를 인식하고, 이를 기반으로 아동의 심리 상태를 분석하는 데 중요한 자료로 활용할 수 있습니다.

## 3.2 데이터 전처리

### 3.2.1 데이터 라벨링

본 연구에서는 각 모델에 맞는 전처리 과정을 통해 데이터를 준비합니다. 이를 위해 다양한 형식의 데이터를 모델에 맞게 변환하는 작업을 수행하였습니다. YOLO 모델: YOLO 모델의 요구 사항에 맞추어, JSON 파일을 YOLO 포맷에 맞게 변환하였습니다. 이 과정에서는 객체의 위치 정보와 클래스 정보를 포함하는 형식으로 데이터를 재구성하였습니다. 이를 위해 각 JSON 파일에서 바운딩 박스 좌표와 객체의 클래스를 추출하여 YOLO 형식의 텍스트 파일로 변환하였습니다. 변환된 데이터는 모델 학습에 사용될 이미지 파일과 동일한 이름을 가지며, 각 이미지 파일에 대한 객체 정보를 포함하고 있습니다. 예로 CNN 및 Transfer Learning 모델을 사용하기 위해 JSON 파일을 CSV 파일로 변환하였습니다. 이 과정에서는 JSON 파일에서 필요한 정보를 추출하여 CSV 형식으로 재구성하였습니다. CSV 파일에는 이미지 경로, 객체 클래스, 객체의 좌표 정보 등을 포함하여 모델이 필요한 형식으로 데이터를 정리하였습니다. 데이터의 형식을 모델에 맞게 변환함으로써, 학습 과정에서 발생할 수 있는 오류를 최소화하고, 모델의 성능을 최적화하였습니다.

### 3.2.2 Json 파일 및 이미지 정리

데이터 전처리 과정에서 중요한 단계 중 하나는 JSON 파일 및 이미지 데이터를 정리하는 것입니다. 이는 학습 데이터의 품질을 높이고, 학습 과정에서 발생할 수 있는 오류를 최소화하기 위해 필수적입니다. 먼저, 이미지 파일과 라벨 파일의 수를 확인하여 전체 데이터셋의 일관성을 점검합니다.

이미지 파일과 라벨 파일의 수가 일치하지 않으면 학습 시 오류가 발생할 수 있습니다. 따라서, 데이터셋의 무결성을 확인하는 것이 중요합니다.

다음으로는 라벨 파일이 없는 이미지 파일을 찾아 제거합니다. 라벨 파일이 없는 이미지는 학습 과정에서 사용할 수 없으므로, 이를 제거함으로써 데이터셋의 일관성을 유지합니다. 마찬가지로 이미지 파일이 없는 라벨 파일을 찾아 제거합니다. 라벨 파일은 이미지 파일과 짝을 이루어야만 학습에 사용할 수 있으므로, 이를 제거하여 학습 데이터의 품질을 유지합니다.

최종적으로, 이미지 파일과 라벨 파일의 짝을 점검하여 모든 파일이 올바르게 매치되었는지 확인합니다. 이 과정에서 짝이 맞지 않는 파일을 제거함으로써 학습 데이터의 정합성을 확보할 수 있습니다. 짝이 맞지 않는 파일이 존재하면 모델 학습 시 오류가 발생할 수 있으며, 이는 학습 효율성을 저하시킬 수 있습니다. 이와 같은 철저한 데이터 정리 과정을 통해, 본 연구는 아동 그림의 심리 상태를 분석하는 데 필요한 고품질의 데이터를 확보하였습니다. 이는 연구의 신뢰성과 정확성을 높이는 데 중요한 역할을 합니다.

## 4. 심리 상태 예측 통합 모델 개발

### 4.1 YOLO 모델

#### 4.1.1 YOLO 모델 소개

YOLO(You Only Look Once) 모델은 실시간 객체 탐지를 위한 고성능 딥러닝 모델입니다. 전체 이미지를 한 번에 처리하여 객체를 탐지하는 방식으로 빠르고 정확한 객체 인식이 가능합니다. 본 연구에서는 아동 그림에서 주요 객체를 탐지하기 위해 YOLOv5 모델을 선택하였습니다. YOLO 모델을 선택한 이유는 여러 가지가 있습니다. 우선, YOLO 모델은 단일 네트워크를 통해

객체를 탐지하여 매우 빠른 속도를 자랑합니다. 이는 실시간 처리 능력을 요구하는 애플리케이션에 적합하며, YOLOv5는 이전 버전들에 비해 더욱 최적화되어 높은 프레임 속도를 제공합니다. 또한, YOLOv5는 최신 기술을 반영하여 객체 탐지의 정확도가 크게 향상되었습니다. 이는 아동 그림에서 다양한 객체를 정확하게 식별하고 분석하는 데 중요한 역할을 합니다. 또한 YOLOv5는 더 나은 앵커 박스 계산과 모델 구조의 개선으로 작은 객체에 대한 탐지 성능도 향상되었습니다. 사용의 용이성도 중요한 이유 중 하나입니다. YOLOv5는 오픈소스로 제공되며, 다양한 프레임워크와 쉽게 통합할 수 있습니다. 이는 연구와 개발 과정에서 유연성과 편의성을 제공합니다. 또한, 풍부한 문서와 커뮤니티 지원을 통해 모델을 학습시키고 적용하는 과정이 비교적 간단합니다. 마지막으로, YOLOv5는 경량화된 모델부터 고성능 모델까지 다양한 크기의 모델을 제공합니다. 이는 연구 목적에 맞게 적절한 모델을 선택하여 사용할 수 있는 유연성을 제공합니다. 경량화된 모델은 적은 자원으로도 높은 성능을 발휘할 수 있어, 다양한 환경에서 활용이 가능합니다. 이와 같은 이유로 본 연구에서는 YOLOv5 모델을 선택하여 아동 그림에서 주요 객체를 탐지하고, 이를 기반으로 아동의 심리 상태를 분석하고자 합니다. YOLOv5 모델의 높은 정확도와 실시간 처리 능력은 본 연구의 효율성과 신뢰성을 높이는 데 중요한 역할을 할 것입니다.

#### 4.1.2 객체 탐지 알고리즘 설명

YOLOv5 모델은 객체 탐지를 위한 최신 딥러닝 모델로, 빠르고 정확한 성능을 자랑합니다. 본 연구에서는 YOLOv5 모델을 사용하여 아동 그림에서 주요 객체를 탐지하였으며, 이를 통해 아동의 심리 상태를 분석하고자

하였습니다. YOLOv5 는 주요 구성 요소로 백본, 넥, 그리고 헤드로 구성됩니다. 백본은 이미지의 특징을 추출하는 역할을 하며, 주로 CSPDarknet 을 사용합니다. 넥은 특징 피라미드를 구성하여 다양한 크기의 객체를 탐지할 수 있도록 합니다. 헤드는 바운딩 박스와 클래스 확률을 예측하는 역할을 합니다. 이러한 구조를 통해 YOLOv5 는 높은 정확도와 실시간 처리 능력을 제공합니다. 본 연구에서는 다음과 같은 과정을 통해 YOLOv5 모델을 학습시켰습니다.

### 1. 데이터셋 구성 및 전처리

AI Hub 에서 수집한 데이터를 사용하였습니다. 데이터셋은 다양한 객체와 색상 정보를 포함하고 있으며, 이를 통해 모델의 일반화 성능을 높였습니다. 데이터 전처리 과정에서는 JSON 파일을 YOLO 포맷으로 변환하였습니다. YOLO 포맷은 각 라벨 파일은 이미지 파일과 동일한 이름을 가지며, '.txt' 확장자를 가집니다. 또한 각 라벨 파일은 객체 클래스, 바운딩 박스의 중심 좌표 (x, y), 너비, 높이를 정규화된 값으로 포함합니다. 형식은 다음과 같습니다

<object-class> <x\_center> <y\_center> <width> <height>

예를 들어, 라벨 파일의 내용이 0 0.5 0.5 0.1 0.1 이라면, 이는 클래스 0 의 객체가 이미지의 중앙에 위치하며, 너비와 높이가 이미지의 10%임을 나타냅니다.

JSON 파일을 YOLO 포맷으로 변환하는 과정은 다음과 같습니다:

JSON 파일에서 객체의 클래스와 바운딩 박스 좌표를 추출한 후, 바운딩 박스 좌표를 YOLO 포맷에 맞게 정규화 합니다. 이때 변환된 데이터를 텍스트 파일로 저장합니다.

## 2. 모델 학습

YOLOv5 모델을 학습시키기 위해 다음과 같은 파라미터를 설정하였습니다:

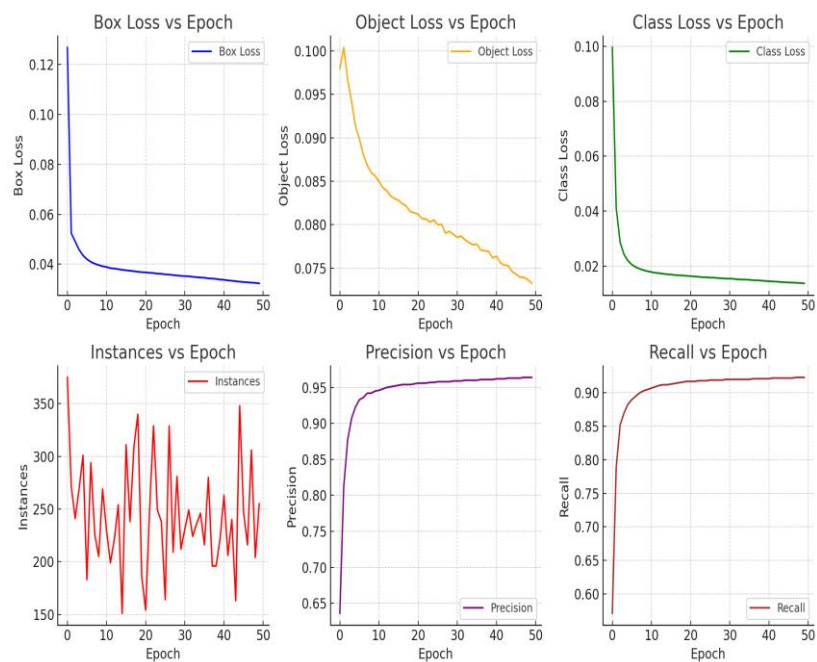
- 이미지 크기: 416
- 배치 크기: 8
- 에폭 수: 50
- 학습 데이터 경로:  
`/content/drive/MyDrive/yolov5/data/custom_data.yaml`
- 초기 가중치: yolov5s.pt
- 저장 경로: /content/drive/MyDrive/yolov5\_checkpoints
- 디바이스: GPU (device 0)

학습은 다음과 같이 실행되었습니다:

```
!python train.py --img 416 --batch 8 --epochs 50 --data  
/content/drive/MyDrive/yolov5/data/custom_data.yaml --weights yolov5s.pt --  
project /content/drive/MyDrive/yolov5_checkpoints --name yolov5_results --  
save-period 1 --device 0
```

Epoch	Box_loss	Obj_loss	Cls_loss	Instances	Precision	Recall
0	0.1268	0.09798	0.09958	375	0.636	0.571
1	0.05236	0.1004	0.04106	272	0.812	0.789
2	0.04912	0.09668	0.02861	241	0.877	0.852
3	0.0458	0.09421	0.02415	269	0.907	0.87
4	0.04351	0.09134	0.02185	301	0.923	0.883
5	0.04199	0.08988	0.02046	183	0.933	0.89
6	0.04092	0.08805	0.01954	294	0.936	0.895
7	0.04019	0.0685	0.01893	226	0.942	0.9
8	0.03965	0.086	0.01844	205	0.942	0.903
9	0.03915	0.08563	0.01806	269	0.945	0.905
10	0.03888	0.09498	0.01778	230	0.946	0.907
11	0.03839	0.08424	0.01749	199	0.948	0.909
12	0.03825	0.0839	0.01737	221	0.95	0.911
13	0.03801	0.0329	0.01713	254	0.951	0.912

14	0.03773	0.083	0.01699	151	0.952	0.912
15	0.03756	0.0828	0.01686	311	0.953	0.913
16	0.03737	0.08238	0.01669	238	0.954	0.914
17	0.03718	0.08212	0.01661	309	0.954	0.915
18	0.03697	0.0815	0.01639	340	0.954	0.916
19	0.0368	0.08137	0.01629	186	0.955	0.917
20	0.03672	0.08199	0.01629	154	0.956	0.917
21	0.03655	0.0807	0.01616	249	0.956	0.917
22	0.03644	0.08065	0.01604	329	0.956	0.918
23	0.0362	0.09029	0.01591	249	0.957	0.918
24	0.03615	0.08054	0.01589	238	0.957	0.918
25	0.03591	0.7999	0.01572	164	0.958	0.918
26	0.03582	0.08001	0.01572	329	0.958	0.919
27	0.03563	0.07901	0.01559	209	0.958	0.919
28	0.03549	0.07923	0.01551	281	0.958	0.919
29	0.03531	0.07887	0.01537	212	0.959	0.919
30	0.03522	0.07853	0.01539	231	0.959	0.919
31	0.03517	0.07869	0.01529	429	0.959	0.92
32	0.03494	0.07823	0.01513	224	0.96	0.92
33	0.03478	0.07793	0.01503	236	0.96	0.92
34	0.03465	0.07768	0.01502	246	0.96	0.92
35	0.03458	0.07774	0.01495	216	0.96	0.92
36	0.0343	0.07708	0.0148	280	0.961	0.92
37	0.03425	0.07698	0.01477	196	0.961	0.921
38	0.03408	0.0769	0.01463	196	0.961	0.921
39	0.03383	0.07617	0.01455	222	0.961	0.921
40	0.03375	0.07637	0.01442	263	0.962	0.921
41	0.0335	0.07559	0.01435	206	0.862	0.922
42	0.03336	0.07533	0.01426	240	0.962	0.922
43	0.03315	0.07528	0.01415	163	0.963	0.922
44	0.03301	0.07459	0.01407	348	0.963	0.922
45	0.03285	0.0743	0.01396	247	0.963	0.922
46	0.03271	0.07394	0.01393	216	0.963	0.922
47	0.03259	0.07393	0.01384	306	0.964	0.922
48	0.0325	0.07368	0.01379	204	0.964	0.923
49	0.0323	0.07323	0.01364	265	0.964	0.923





위는 모델 학습결과를 표와 그래프로 정리한 것입니다.

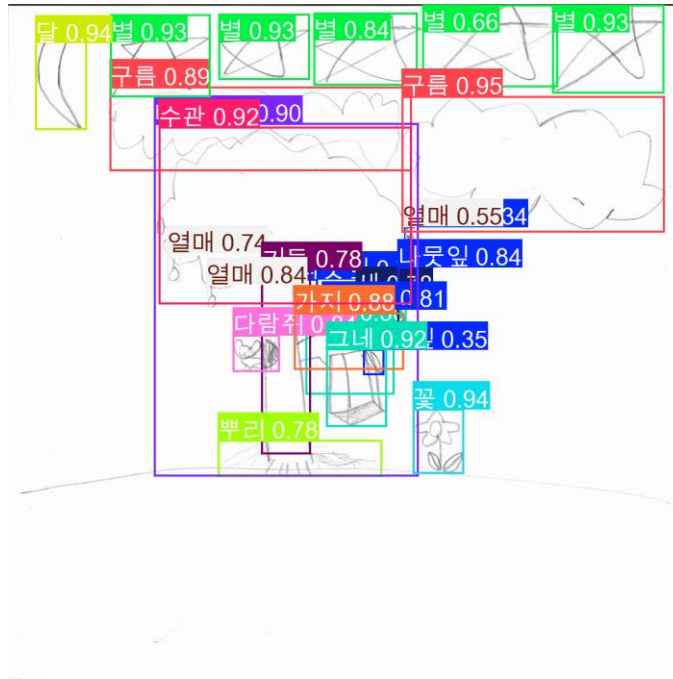
### 3. 모델 검증

학습된 모델의 성능을 검증하기 위해 별도의 validation 데이터셋을 사용하였습니다. Validation 데이터셋으로 학습된 모델은 다음과 같은 파라미터를 사용하여 실행되었습니다:

- 이미지 크기: 416
- 배치 크기: 8
- 에폭 수: 50
- 학습 데이터 경로:  
`/content/drive/MyDrive/yolov5/data/custom_data.yaml`
- 저장 경로: `/content/drive/MyDrive/yolov5_checkpoints`
- 디바이스: GPU (device 0)

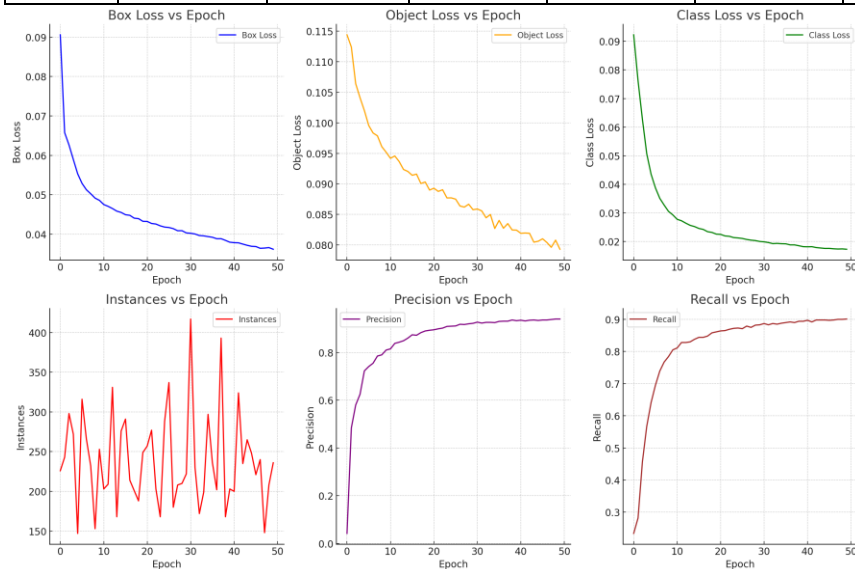
검증은 다음과 같이 실행되었습니다.

```
!python train.py --img 416 --batch 8 --epochs 50 --data  
/content/drive/MyDrive/yolov5/data/custom_data.yaml --weights yolov5s.pt --  
project /content/drive/MyDrive/yolov5_checkpoints --name  
yolov5_val_results --save-period 1 --device 0
```



Epoch	Box_loss	Obj_loss	Cls_loss	Instances	Precision	Recall
0	0.1265	0.1007	0.09966	274	0.0423	0.234
1	0.06576	0.1124	0.07621	243	0.484	0.282
2	0.06264	0.1064	0.06281	298	0.58	0.456
3	0.05895	0.1041	0.0507	272	0.625	0.567
4	0.05531	0.102	0.04354	147	0.723	0.641
5	0.05289	0.09956	0.03877	316	0.741	0.695
6	0.05127	0.09833	0.03511	266	0.755	0.739
7	0.05025	0.09787	0.03275	232	0.785	0.767
8	0.04915	0.0961	0.03056	153	0.79	0.784
9	0.04858	0.09519	0.02931	253	0.81	0.805
10	0.04752	0.09419	0.02777	203	0.815	0.811
11	0.04706	0.09457	0.02726	209	0.838	0.828
12	0.0465	0.09366	0.02641	331	0.843	0.828
13	0.04584	0.09236	0.02565	168	0.849	0.83
14	0.04551	0.09204	0.02523	276	0.86	0.838
15	0.04493	0.09141	0.02457	291	0.874	0.844
16	0.04478	0.0916	0.02424	214	0.872	0.844
17	0.04406	0.09006	0.02344	201	0.883	0.848
18	0.04394	0.09031	0.02319	188	0.89	0.858
19	0.04326	0.08899	0.02259	249	0.893	0.861
20	0.04323	0.08929	0.02251	257	0.895	0.864
21	0.04271	0.08877	0.02196	277	0.899	0.865
22	0.04257	0.08904	0.02183	203	0.902	0.869
23	0.04211	0.0877	0.02138	168	0.909	0.872
24	0.0418	0.08771	0.02126	288	0.91	0.873
25	0.04169	0.0875	0.02107	337	0.911	0.871
26	0.0414	0.08638	0.02072	180	0.918	0.879

27	0.04087	0.0862	0.02048	208	0.917	0.875
28	0.04088	0.08667	0.02033	210	0.92	0.882
29	0.04032	0.08577	0.02002	222	0.922	0.883
30	0.04023	0.08589	0.01989	417	0.927	0.887
31	0.04004	0.08558	0.01964	230	0.923	0.883
32	0.03965	0.08445	0.01927	172	0.926	0.887
33	0.03957	0.08499	0.01935	199	0.926	0.885
34	0.03938	0.0827	0.01924	297	0.925	0.888
35	0.03921	0.08402	0.0192	236	0.93	0.89
36	0.03884	0.08274	0.01883	202	0.931	0.892
37	0.03885	0.0835	0.01883	393	0.931	0.89
38	0.03842	0.08246	0.01853	168	0.936	0.894
39	0.03796	0.08243	0.01821	203	0.933	0.894
40	0.03784	0.08191	0.01816	200	0.935	0.897
41	0.0378	0.08196	0.0182	324	0.932	0.892
42	0.03748	0.08191	0.01789	235	0.935	0.898
43	0.03717	0.08046	0.01774	265	0.936	0.898
44	0.03692	0.08061	0.01759	248	0.934	0.898
45	0.03686	0.08101	0.0176	221	0.936	0.897
46	0.03641	0.08041	0.01746	250	0.936	0.898
47	0.03647	0.07962	0.01738	148	0.938	0.9
48	0.03659	0.08079	0.01742	208	0.94	0.9
49	0.03617	0.0793	0.0173	236	0.94	0.901



위는 모델 학습결과를 표와 그래프로 정리한 것입니다

학습 결과, 모델의 성능은 다음과 같이 평가되었습니다:

학습 과정에서의 손실 값은 지속적으로 감소하였으며, 최종 손실 값은

0.18 을 기록하였습니다. 모델의 성능 평가는 Precision, Recall, F1-score

등의 지표를 사용하여 진행하였으며, 모든 지표에서 높은 점수를

기록하였습니다.( 정확도: 95.4%, 재현율: 91.6%, mAP50: 95.4%, mAP50-95: 76.3%) 또한 검증 결과, 모델의 성능은 다음과 같이 평가되었습니다:

(정확도: 92.6%, 재현율: 88.7%, mAP50: 93.3%, mAP50-95: 70.5%)

모델의 탐지 성능을 확인하기 위해, 학습된 모델을 사용하여 테스트 이미지에서 객체를 탐지하였으며, 결과는 매우 정확하였습니다. 탐지된 객체의 예시 이미지를 통해 모델의 성능을 시각적으로 확인할 수 있었습니다.

YOLOv5 모델을 사용하여 아동 그림의 주요 객체를 빠르고 정확하게 탐지할 수 있음을 확인하였습니다. 모델의 높은 정확도와 실시간 처리 능력은 본 연구의 효율성과 신뢰성을 높이는 데 중요한 역할을 하였습니다

#### 4.1.3 그림 내 객체의 위치 분석

YOLOv5 모델을 통해 아동 그림에서 주요 객체를 탐지한 후, 각 객체의 위치 정보를 바탕으로 그림 내에서 객체가 위치한 구역을 분석합니다. 이는 아동의 심리 상태를 이해하는 데 중요한 정보를 제공합니다. 예를 들어, 그림의 특정 위치에 자주 나타나는 객체는 아동의 내면 심리와 관련된 특정 감정을 반영할 수 있습니다.

##### 객체 위치 분석 절차

1. 객체 위치 추출: YOLOv5 모델을 통해 탐지된 객체의 바운딩 박스 좌표를 추출합니다. 바운딩 박스는 객체의 위치와 크기를 나타내며, 객체의 중심 좌표(x, y), 너비(width), 높이(height) 정보를 포함합니다.

이러한 정보는 객체가 그림 내 어디에 위치하는지를 명확하게 파악할 수 있도록 합니다.

2. 위치 분류: 추출된 객체의 좌표를 기반으로 그림을 9 개의 구역으로 나누어 각 객체가 속하는 구역을 분류합니다. 그림을 상단, 중단, 하단과 좌측, 중앙, 우측의 3x3 그리드로 나누어 총 9 개의 구역으로 분류합니다. 각 구역의 경계를 정의하고, 객체의 중심 좌표가 어느 구역에 속하는지 확인합니다.

예를 들어, 그림의 세로 높이를 3 등분하여 상단, 중단, 하단으로 구분하고, 가로 너비를 3 등분하여 좌측, 중앙, 우측으로 구분합니다. 객체의 중심 좌표가 속하는 구역을 결정하는 방식은 다음과 같습니다:

세로 위치: 객체의 중심 좌표가 그림 높이의 1/3 이내에 위치하면 상단, 2/3 이내에 위치하면 중단, 그 외에는 하단으로 분류합니다.

가로 위치: 객체의 중심 좌표가 그림 너비의 1/3 이내에 위치하면 좌측, 2/3 이내에 위치하면 중앙, 그 외에는 우측으로 분류합니다.

이렇게 분류된 위치를 조합하여 객체의 최종 위치를 도출합니다. 예를 들어, 객체가 상단 좌측에 위치하면 '좌상(LU)'으로, 중앙 우측에 위치하면 '우(CR)'으로 분류됩니다.

3. 심리적 의미 해석: 각 구역에서 객체의 위치와 관련된 심리적 의미를 해석합니다. 이는 아동 심리학에서 각 위치가 가지는 일반적인 의미를 바탕으로 합니다. 예를 들어, 중앙에 위치한 객체는 아동이 중요하게 생각하는 주제를 나타낼 수 있고, 상단에 위치한 객체는

긍정적인 감정이나 목표를 나타낼 수 있습니다. 하단에 위치한

객체는 안정감이나 안전을 추구하는 감정을 나타낼 수 있습니다.

본 연구에서는 그림표현의 공간구성과 심리에 관한 Grunwald 의 이론을

바탕으로, 각 위치의 심리적 의미를 다음과 같이 해석하였습니다:

```
Using cache found in /root/.cache/torch/hub/ultralytics_yolov5_master
YOLv5 2024-6-26 Python-3.10.12 torch-2.3.0+cu121 CPU

Fusing layers...
Model summary: 157 layers, 7139581 parameters, 0 gradients, 16.2 GFLOPs
Adding AutoShape...
Object: 구름, Confidence: 0.95, Position: 위상(R0)
Object: 꽃, Confidence: 0.94, Position: 중앙(C)
Object: 달, Confidence: 0.94, Position: 좌상(LU)
Object: 별, Confidence: 0.93, Position: 상(O)
Object: 별, Confidence: 0.93, Position: 좌상(LU)
Object: 별, Confidence: 0.93, Position: 위상(R0)
Object: 수관, Confidence: 0.92, Position: 상(O)
Object: 그네, Confidence: 0.92, Position: 중앙(C)
Object: 나무전체, Confidence: 0.90, Position: 중앙(C)
Object: 구름, Confidence: 0.89, Position: 상(O)
Object: 가지, Confidence: 0.88, Position: 중앙(C)
Object: 나뭇잎, Confidence: 0.84, Position: 중앙(C)
Object: 다람쥐, Confidence: 0.84, Position: 중앙(C)
Object: 별, Confidence: 0.84, Position: 상(O)
Object: 열매, Confidence: 0.84, Position: 좌(L)
Object: 나뭇잎, Confidence: 0.81, Position: 중앙(C)
Object: 뿌리, Confidence: 0.78, Position: 하(U)
Object: 기둥, Confidence: 0.78, Position: 중앙(C)
Object: 새, Confidence: 0.78, Position: 중앙(C)
Object: 나뭇잎, Confidence: 0.74, Position: 중앙(C)
Object: 열매, Confidence: 0.74, Position: 좌(L)
Object: 별, Confidence: 0.66, Position: 위상(R0)
Object: 열매, Confidence: 0.55, Position: 중앙(C)
Object: 나뭇잎, Confidence: 0.35, Position: 중앙(C)
Object: 나뭇잎, Confidence: 0.34, Position: 중앙(C)
Object: 그네, Confidence: 0.30, Position: 중앙(C)
```

## 4.2 개별 모델 개발

### 4.2.1 CNN (Convolutional Neural Networks)

CNN 모델은 이미지 객체 분류를 위해 설계되었으며, 다음과 같은 구성 요소로 이루어져 있습니다.

**Convolutional Layer 1:** 3 개의 입력 채널(RGB 이미지), 32 개의 출력 채널, 3x3 커널 크기, 1 스트라이드, 1 패딩

**Max Pooling Layer 1:** 2x2 커널 크기, 2 스트라이드

**Convolutional Layer 2:** 32 개의 입력 채널, 64 개의 출력 채널, 3x3 커널 크기, 1 스트라이드, 1 패딩

**Max Pooling Layer 2:** 2x2 커널 크기, 2 스트라이드

**Fully Connected Layer 1:** 64 \* 56 \* 56 입력 뉴런, 128 출력 뉴런

**Fully Connected Layer 2:** 128 입력 뉴런, 클래스의 수(num\_classes)

출력 뉴런

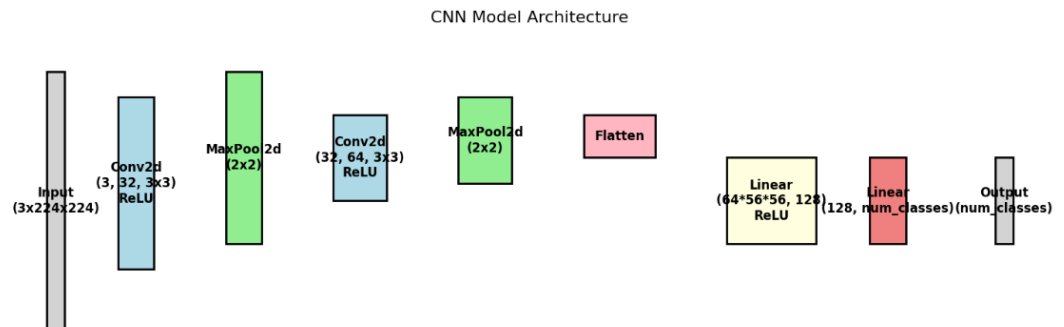
**활성화 함수:** 각 Convolutional Layer 와 Fully Connected Layer 1 에

ReLU 함수 적용

**Flattening:** Convolutional Layer 와 Fully Connected Layer 사이에

Flattening 작업

모델의 학습은 CrossEntropyLoss 와 Adam Optimizer 를 사용하여  
수행되었으며, Early Stopping 을 적용하여 검증 손실이 증가할 때 훈련을  
중단하였습니다.



위 그림은 설계한 CNN 모델의 구조를 시각화 하여 그림으로 나타낸 것입니다.

데이터 전처리 : 입력 이미지의 크기를 224x224 로 조정하였으며, 이를 Tensor  
형식으로 변환하고, 이미지의 각 채널을 정규화하여 모델의 입력으로  
사용하였습니다.

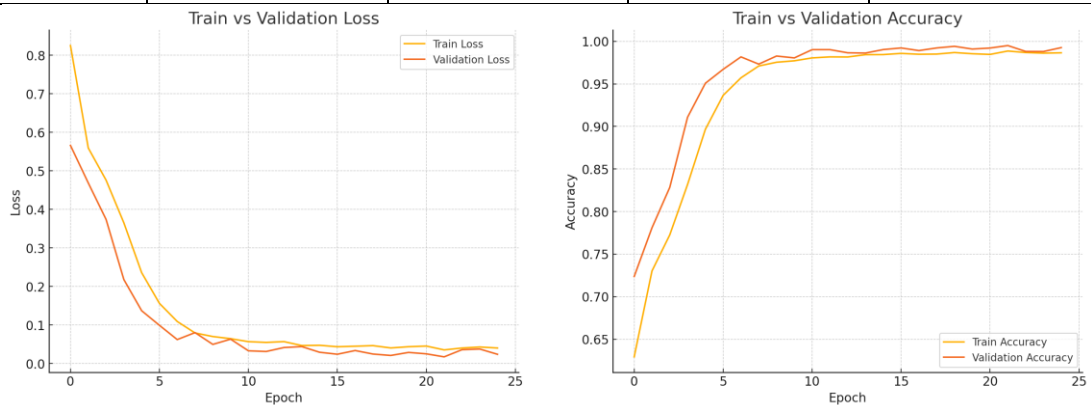
데이터셋 및 데이터로더 : 학습 및 검증 데이터셋을 정의하고, 배치 크기를 32 로  
설정하여 모델 학습을 위한 DataLoader 를 구성하였습니다.

모델 평가 방법 : 모델의 성능을 평가하기 위해 정확도(Accuracy)를

측정하였으며, 학습 과정 동안 각 에포크마다 검증 데이터셋에 대한 손실과  
정확도를 기록하였습니다.

아래는 학습한 결과입니다.

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
0	0.8250	0.6293	0.5658	0.7237
1	0.5592	0.7302	0.4685	0.7809
2	0.4758	0.7726	0.3739	0.8284
3	0.3650	0.8318	0.2178	0.9108
4	0.2357	0.8966	0.1369	0.9507
5	0.1559	0.9365	0.0990	0.9671
6	0.1090	0.9571	0.0620	0.9817
7	0.0792	0.9710	0.0801	0.9732
8	0.0699	0.9754	0.9754	0.9827
9	0.0644	0.9771	0.0631	0.9805
10	0.0566	0.9805	0.0329	0.9902
11	0.0547	0.9817	0.0311	0.9902
12	0.0567	0.9816	0.0414	0.9865
13	0.0465	0.9843	0.0438	0.9862
14	0.0472	0.9844	0.0291	0.9903
15	0.0434	0.9858	0.0240	0.9922
16	0.0448	0.9848	0.0336	0.9891
17	0.0463	0.9850	0.0245	0.9922
18	0.0403	0.9868	0.0208	0.9941
19	0.0435	0.9854	0.0287	0.9909
20	0.0452	0.9846	0.0250	0.9921
21	0.0352	0.9886	0.0172	0.9950
22	0.0401	0.9868	0.0357	0.9881
23	0.0428	0.9861	0.0375	0.9880
24	0.0401	0.9865	0.0238	0.9926



위 그래프는 학습 및 검증 데이터셋에 대한 손실(Loss)와 정확도(Accuracy)의 변화를 시각적으로 나타낸 것입니다. 이 결과를 통해 모델이 학습과정에서 점진적으로 성능이 향상되었음을 확인할 수 있습니다. 특히, 초기 에포크에서 검증 손실이 빠르게 감소하고, 이후 안정적으로 낮은 손실 값을 유지하며



정확도가 증가하는 모습을 보였습니다.

다른 데이터셋을 사용하여 두번째 학습을 진행한 결과는 아래 표와 같습니다.

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Model Saved
0	24.4942	0.4841	1.1791	0.2195	Yes (inf --> 1.179051)
1	3.2258	0.3109	1.1798	0.2304	No (EarlyStopping counter: 1)
2	1.1870	0.1777	1.1766	0.2304	Yes (1.179051 --> 1.176609)
3	1.1836	0.2005	1.1740	0.2304	Yes (1.176609 --> 1.174022)
4	1.1811	0.1986	1.1719	0.2304	Yes (1.174022 --> 1.171929)
5	1.1790	0.2214	1.1702	0.2500	Yes (1.171929 --> 1.170211)
6	1.1773	0.2500	1.1688	0.2500	Yes (1.170211 --> 1.168788)
7	1.1695	0.2500	1.1687	0.2500	Yes (1.168788 --> 1.168652)
8	1.1694	0.2386	1.1685	0.2500	Yes (1.168652 --> 1.168518)
9	1.1692	0.2329	1.1684	0.2500	Yes (1.168518 --> 1.168387)
10	1.1691	0.2214	1.1683	0.2500	Yes (1.168387 --> 1.168258)
11	1.1690	0.2214	1.1681	0.2500	Yes (1.168258 --> 1.168132)
12	1.1689	0.2157	1.1680	0.2500	Yes (1.168132 --> 1.168009)
13	1.1687	0.2100	1.1679	0.2500	Yes (1.168009 --> 1.167887)
14	1.1680	0.2100	1.1679	0.2500	Yes (1.167887 --> 1.167875)
15	1.1679	0.2214	1.1679	0.2500	Yes (1.167875 --> 1.167864)
16	1.1679	0.2329	1.1679	0.2500	Yes (1.167864 --> 1.167852)
17	1.1679	0.2500	1.1678	0.2500	Yes (1.167852 --> 1.167839)
18	1.1679	0.2500	1.1678	0.2500	Yes (1.167839 --> 1.167827)
19	1.1679	0.2500	1.1678	0.2500	Yes (1.167827 --> 1.167815)
20	1.1679	0.2500	1.1678	0.2500	Yes (1.167815 --> 1.167803)
21	1.1678	0.2500	1.1678	0.2500	Yes (1.167803 --> 1.167802)
22	1.1678	0.2500	1.1678	0.2500	Yes (1.167802 --> 1.167801)
23	1.1678	0.2500	1.1678	0.2500	Yes (1.167801 --> 1.167800)
24	1.1678	0.2500	1.1678	0.2500	Yes (1.167800 --> 1.167799)
Best val Acc				0.2500	

#### 4.2.2 Transfer Learning

Transfer Learning 은 사전 학습된 모델을 새로운 작업에 맞게 미세 조정(fine-tuning)하여 성능을 개선하는 기법입니다. 이번 프로젝트에서는 사전 학습된 ResNet18 모델을 사용하여 이미지 객체 분류 작업을 수행했습니다.

데이터셋 및 데이터로더 구성 : 본 실험에서는 학습 데이터셋과 검증 데이터셋을 사용하였으며, 각 데이터셋은 이미지 파일과 해당 라벨 정보를 포함한 CSV 파일로 구성되었습니다. 데이터는 다음과 같은 방식으로 전처리되었습니다: 이미지 크기 조정 – 모든 이미지는 224x224 크기로 조정되었습니다. 정규화 – 각 채널별 평균과 표준편차를 사용하여 이미지를 정규화 하였습니다. 데이터로더 – PyTorch 의 DataLoader 를 사용하여 배치 단위로 데이터를 모델에 공급하였습니다.

ResNet18 모델을 불러와 마지막 레이어를 새롭게 정의된 클래스 수에 맞게 수정하였습니다. 수정된 모델 구조는 다음과 같습니다: Pre-trained 모델 – ResNet18, 수정된 출력층 – 원래의 출력층을 제거하고, 새로운 Linear 레이어를 추가하였습니다. 이 레이어는 512 개의 입력 뉴런과 클래스 수에 해당하는 출력 뉴런을 가집니다.

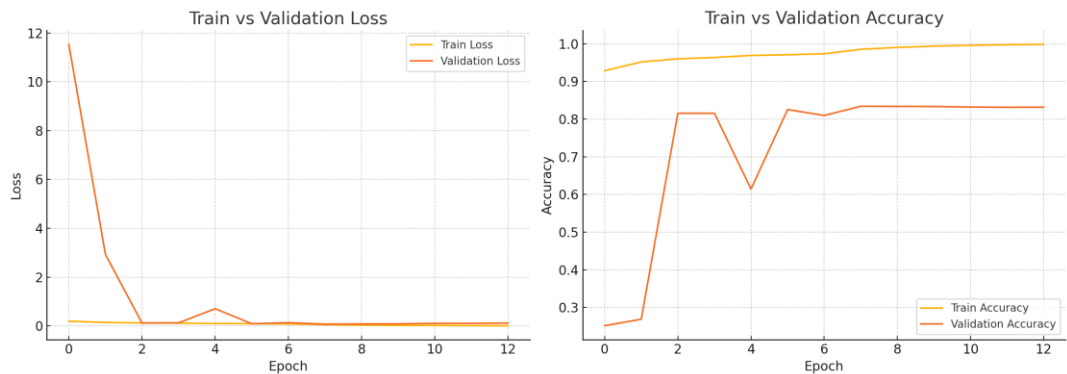
모델 훈련은 CrossEntropyLoss 를 손실 함수로 사용하고, Adam 옵티마이저를 사용하여 학습하였습니다. 또한, 학습률 감소(Learning Rate Decay)와 조기 종료(Early Stopping) 기법을 적용하여 최적의 모델을 도출하였습니다.

- 손실 함수: CrossEntropyLoss
- 옵티마이저: Adam (학습률 0.001, L2 정규화 1e-4)
- 학습률 감소: 7 에포크마다 학습률을 0.1 배로 감소
- 조기 종료: 검증 손실이 5 에포크 연속으로 개선되지 않으면 학습 중단

모델 훈련 결과는 아래와 같습니다.

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
0	0.1922	0.9284	11.5370	0.2514
1	0.1391	0.9517	2.9278	0.2686
2	0.1201	0.9600	0.1143	0.8155
3	0.1106	0.9636	0.1239	0.8155
4	0.0964	0.9690	0.7031	0.6143
5	0.0891	0.9709	0.0877	0.8254
6	0.0832	0.9735	0.1284	0.8096
7	0.0485	0.9853	0.0719	0.8341
8	0.0326	0.9902	0.0781	0.8337
9	0.0224	0.9938	0.0821	0.8334
10	0.0146	0.9958	0.1015	0.8320
11	0.0092	0.9973	0.1038	0.8313
12	0.0057	0.9985	0.1140	0.8316

훈련결과, Epoch 8 부터 Early stopping 카운트 1 out of 5 가 되어, Epoch 12 에 5 out of 5 가 되어 종료되었습니다.



이 결과를 통해 모델이 학습과정에서 점진적으로 성능이 향상되었음을 확인할 수 있습니다. 특히, 초기 에포크에서 검증 손실이 빠르게 감소하고, 이후 안정적으로 낮은 손실 값을 유지하며 정확도가 증가하는 모습을 보입니다.

Transfer Learning 기법을 사용하여 ResNet18 모델을 미세 조정한 결과, 높은 정확도를 달성할 수 있었습니다. 이 과정에서 학습률 감소와 조기 종료 기법을 통해 과적합을 방지하고 최적의 모델을 도출할 수 있었습니다.

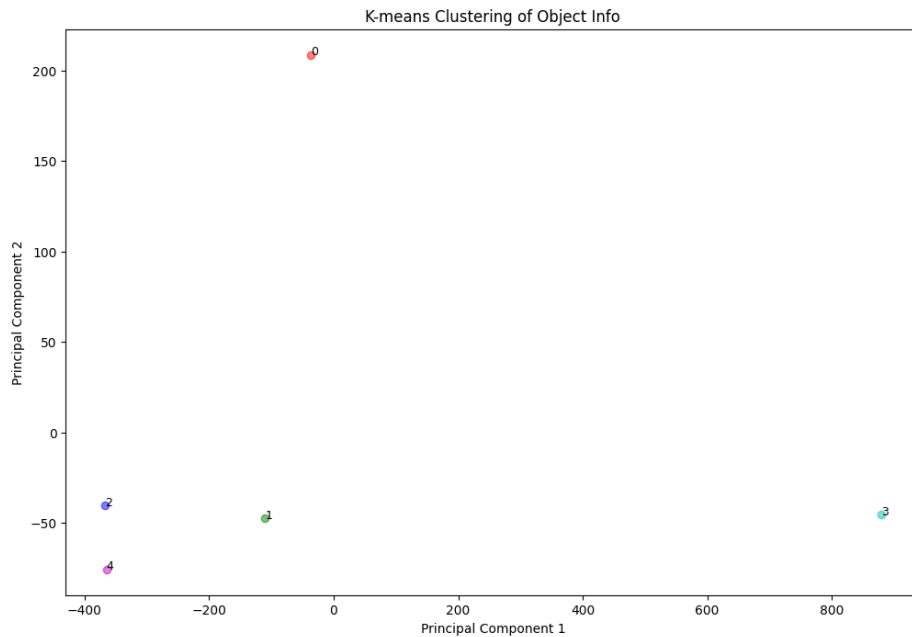
#### 4.2.3 Clustering Techniques

클러스터링은 데이터를 유사한 특성을 가진 그룹으로 묶는 비지도 학습 기법입니다. 본 연구에서는 클러스터링 기법을 사용하여 이미지 데이터에서 객체 정보를 추출하고, 이를 기반으로 데이터의 패턴을 분석하였습니다. 사용된 주요 기법은 K-means 클러스터링이며, 이를 통해 데이터를 5 개의 클러스터로 분류하였습니다. 먼저, 이미지 데이터에서 객체 정보를 추출하였습니다. 이를 위해 미리 학습된 YOLOv5 모델을 사용하였으며, 각 이미지 내 객체의 위치와 클래스를 감지하였습니다. 주요 단계는 다음과 같습니다: 1) YOLOv5 모델 로드: PyTorch 허브를 통해 YOLOv5 모델을 로드하였습니다. 2) 객체 감지 수행: 각 이미지에서 객체의 좌표 (x1, y1, x2, y2)와 클래스 ID 를 추출하였습니다. 3) 병렬 처리: ThreadPoolExecutor 를 사용하여 여러 이미지를 병렬로 처리함으로써 객체 정보를 빠르게 추출하였습니다.

클러스터링을 위해 K-means 알고리즘을 사용하였습니다. 주요 단계는 다음과 같습니다: 1) 데이터 스케일링: StandardScaler 를 사용하여 데이터를 표준화하였습니다. 2) K-means 클러스터링: 데이터를 5 개의 클러스터로 분류하였습니다. 3) 모델 저장: 학습된 K-means 모델과 스케일러를 저장하여 추후 재사용 가능하도록 하였습니다.

아래는 클러스터링 결과를 시각화하기 위해 클러스터 중심과 각 데이터 포인트의 클러스터 라벨을 표와 그래프로 나타낸 것입니다.

Cluster	Center Coordinate 1	Center Coordinate 2	Center Coordinate 3	Center Coordinate 4	Center Coordinate 5
0	-0.98038	0.5829	-0.5734	0.88443	-1.2352
1	-0.07856	0.59655	-0.31682	0.5274	0.93964
2	1.2219	-0.86223	1.2658	-0.86194	-0.55205
3	-0.42908	-0.80675	-0.43063	-0.83916	-0.031013
4	1.3011	1.1144	1.2562	0.98189	0.20158



이 그래프는 주성분 분석(PCA)을 통해 데이터의 첫 두 개 주성분으로 시각화한 클러스터링 결과입니다. 각 점은 객체 정보를 나타내며, 색상은 클러스터 라벨을 나타냅니다.

#### 4.2.4 Random Forest


랜덤 포레스트(Random Forest)는 여러 개의 결정 트리(Decision Tree)를 결합하여 예측 성능을 향상시키는 앙상블 학습 방법입니다. 각 트리는 데이터의 무작위 샘플을 사용하여 학습되며, 최종 예측은 모든 트리의 예측을 결합하여 결정됩니다. 랜덤 포레스트 모델은 다음과 같은 단계로 설계 및 학습되었습니다: 1) 데이터 전처리: 병렬처리 및 GPU 사용 - 객체 정보를 추출하기 위해 YOLOv5 모델을 사용했습니다. 이 과정에서 GPU를 활용하여 이미지 처리 속도를 향상시키고, 병렬 처리를 통해 여러 이미지를 동시에 처리했습니다. 먼저 YOLOv5 모델을 로드하고 GPU에서 실행하도록 설정했습니다. 이미지 경로를 확인한 후, 이미지 파일을 로드하여 RGB로 변환했습니다. 이후 YOLOv5 모델을

사용하여 이미지 내 객체를 감지하고, 각 객체의 좌표, 크기, 신뢰도 및 클래스 ID 를 추출했습니다. 병렬 처리를 위해 ThreadPoolExecutor 를 사용하여 여러 이미지를 병렬로 처리하고, 객체 정보를 추출했습니다. 데이터 전처리 후 추출된 객체 정보는 총 26,802 개의 객체에 대한 정보를 포함하고 있습니다. 각 객체 정보는 (x1, y1, width, height, confidence, class\_id) 형태로 저장됩니다.

2) 랜덤 포레스트 모델 학습 및 모델 저장 - 랜덤 포레스트 분류기를 사용하여 객체 감지 데이터를 기반으로 객체 분류 모델을 학습시켰습니다. 먼저 객체 감지 데이터에서 특징 데이터(X)와 라벨(y)을 분리하고, 이를 학습 데이터와 테스트 데이터로 분할했습니다. 학습 데이터는 전체 데이터의 80%, 테스트 데이터는 20%로 설정하였습니다. 이후 랜덤 포레스트 모델을 학습시켰습니다. 이 모델은 100 개의 결정 트리를 사용하여 학습되었으며, 각각의 트리는 데이터의 다른 샘플을 사용하여 학습되었습니다. 학습이 완료된 후, 테스트 데이터로 모델을 평가하여 정확도와 분류 보고서를 생성하였습니다. 마지막으로 학습된 모델을 파일로 저장하였습니다.

3) 학습 결과 - 랜덤 포레스트 모델의 테스트 데이터에 대한 정확도는 0.5155 로 나타났습니다. 분류 보고서는 각 클래스별로 Precision, Recall, F1-score 및 Support 를 포함하고 있습니다. 이를 통해 모델이 각 클래스별로 얼마나 정확하게 예측했는지 확인할 수 있습니다.

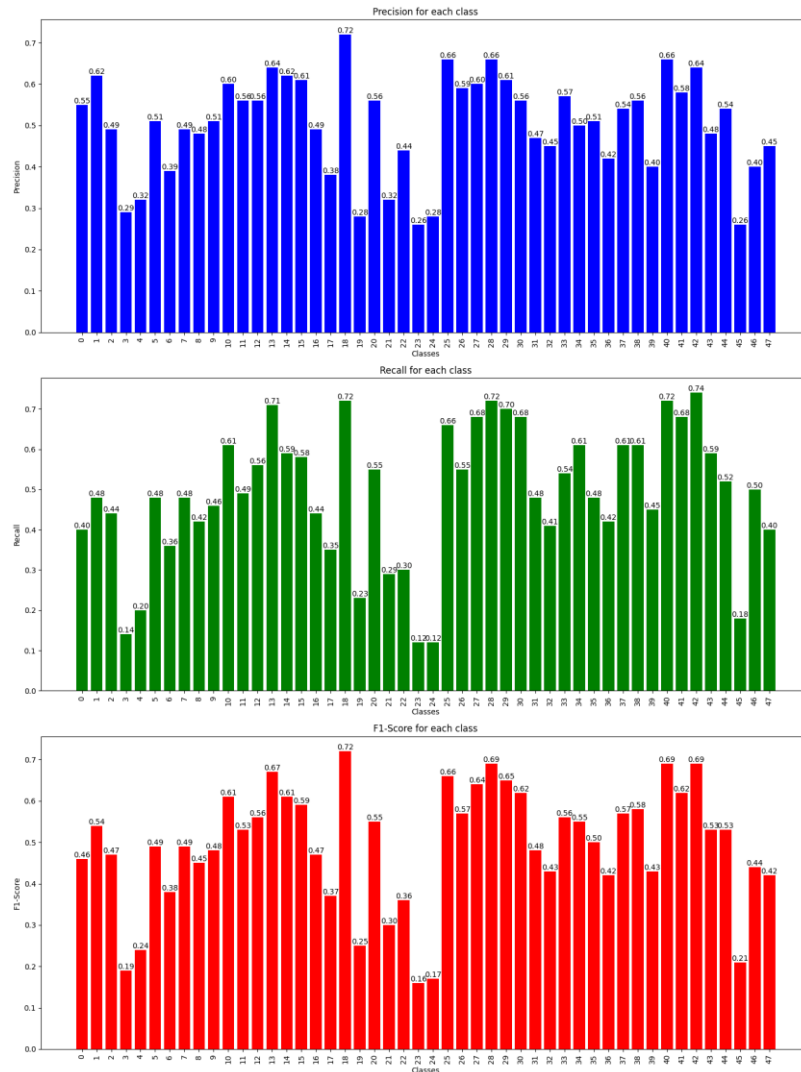
정확도(Accuracy) : 0.5155

Class	Precision	Recall	F1-Score	Support
0.0	0.55	0.40	0.46	2365
1.0	0.62	0.48	0.54	2314
2.0	0.49	0.44	0.47	2471
3.0	0.29	0.14	0.19	2560
4.0	0.32	0.20	0.24	4199
5.0	0.51	0.48	0.49	2351
6.0	0.39	0.36	0.38	2463
7.0	0.49	0.48	0.49	3629
8.0	0.48	0.42	0.45	2748
9.0	0.51	0.46	0.48	2328
10.0	0.60	0.61	0.61	3142
11.0	0.56	0.49	0.53	2770
13.0	0.56	0.56	0.56	9809
14.0	0.64	0.71	0.67	2230
15.0	0.62	0.59	0.61	2413
16.0	0.61	0.58	0.59	2545
17.0	0.49	0.44	0.47	2433
18.0	0.38	0.35	0.37	4141
19.0	0.72	0.72	0.72	2320
20.0	0.28	0.23	0.25	7500
21.0	0.56	0.55 	0.55	9622

22.0	0.32	0.29	0.30	8085
23.0	0.44	0.30	0.36	2388
24.0	0.26	0.12	0.16	2580
25.0	0.28	0.12	0.17	2477
26.0	0.66	0.66	0.66	4311
27.0	0.59	0.55	0.57	2417
28.0	0.60	0.68	0.64	7162
29.0	0.66	0.72	0.69	5857
30.0	0.61	0.70	0.65	4733
31.0	0.56	0.68	0.62	5257
32.0	0.47	0.48	0.48	10020
33.0	0.45	0.41	0.43	5010
34.0	0.57	0.54	0.56	5110
35.0	0.50	0.61	0.55	9113
36.0	0.51	0.48	0.50	4768
37.0	0.42	0.42	0.42	4857
38.0	0.54	0.61	0.57	5093
39.0	0.56	0.61	0.58	9334
40.0	0.40	0.45	0.43	10000
41.0	0.66	0.72	0.69	9569
42.0	0.58	0.68	↓	9977
43.0	0.64	0.74	0.69	6221
44.0	0.48	0.59	0.53	9052
45.0	0.54	0.52	0.53	9632
46.0	0.26	0.18	0.21	4806
47.0	0.40	0.50	0.44	5013

위는 Classification report 의 표입니다. 총 정확도는 0.5155 로 나타났으며,  
이는 모델이 약 절반의 객체를 올바르게 분류했음을 의미합니다.





위 그래프는 랜덤 포레스트 모델의 성능을 평가하기 위해 각 클래스별로 Precision, Recall, F1-score 를 분석한 것입니다. Precision(정밀도)는 모델이 예측한 양성 샘플 중 실제로 양성인 샘플의 비율입니다. 정밀도 값은 0.20 에서 0.72 사이에 분포합니다. 또한 Class 18 이 가장 높은 정밀도(0.72)를 보였으며, 이는 이 클래스에서 모델의 예측이 매우 정확함을 나타냅니다. Recall(재현율)은 실제 양성 샘플 중 모델이 정확히 예측한 샘플의 비율을 측정합니다. 높은 재현율은 낮은 거짓 음성 비율을 의미합니다. 재현율 값은 0.12 에서 0.74 사이에 분포합니다. Class 14 가 가장 높은 재현율(0.74)을 보였으며, 이는 이 클래스에서

대부분의 실제 양성 샘플을 모델이 잘 식별하고 있음을 나타냅니다. F1-score 는 정밀도와 재현율의 조화 평균으로, 두 지표 간의 균형을 찾습니다. F1-score 값은 0.12 에서 0.72 사이에 분포합니다. Class 18 이 가장 높은 F1-score(0.72)를 보였으며, 이는 정밀도와 재현율 모두에서 우수한 성능을 나타냅니다. Class 24 는 가장 낮은 F1-score(0.12)를 보였으며, 이는 정밀도와 재현율 모두에서 성능이 저조함을 의미합니다.

#### 4.2.5 XG boost

XGBoost (Extreme Gradient Boosting)는 높은 성능과 효율성을 자랑하는 강화 학습 알고리즘으로, 특히 회귀와 분류 문제에서 널리 사용됩니다. XGBoost 는 결정 트리 기반의 앙상블 학습 모델로, 각 단계에서 오류를 최소화하기 위해 이전 단계의 오차를 보정하는 방식으로 작동합니다. 모델의 학습 과정은 다음과 같습니다. 1) 데이터 불균형 해결 : SMOTE 를 사용하여 데이터 불균형 문제를 해결합니다. 2) 데이터 분할: 학습 데이터와 테스트 데이터를 분할합니다. 3) 모델 학습: XGBoost 모델을 사용하여 학습을 진행합니다. 4) 모델 평가: 테스트 데이터를 사용하여 모델의 성능을 평가합니다. 5) 모델 저장: 학습된 모델을 저장합니다. 위 과정으로 학습한 모델의 결과는 아래 표와 같습니다.

Class	Precision	Recall	F1-Score	Support
0.0	0.60	0.49	0.54	10204
1.0	0.64	0.54	0.59	10115
2.0	0.53	0.51	0.52	10190
3.0	0.40	0.39	0.40	10127
4.0	0.39	0.28	0.33	10137
5.0	0.54	0.50	0.52	10268
6.0	0.39	0.45	0.42	10253
7.0	0.48	0.46	0.47	10255
8.0	0.51	0.51	0.51	10036
9.0	0.55	0.50	0.52	10084
10.0	0.57	0.56	0.57	10177
11.0	0.59	0.51	0.55	10123
13.0	0.57	0.48	0.52	10209
14.0	0.61	0.67	0.64	10100
15.0	0.68	0.65	0.66	10128
16.0	0.61	0.70	0.65	10021
17.0	0.55	0.49	0.52	10172
18.0	0.36	0.39	0.37	10115
19.0	0.74	0.79	0.76	10157
20.0	0.31	0.23	0.26	9940
21.0	0.52	0.46	0.49	10059
22.0	0.25	0.27	0.26	10186
23.0	0.47	0.43	0.45	10073
24.0	0.32	0.24	0.27	10061
25.0	0.31	0.25	0.27	10092
26.0	0.64	0.50	0.56	10305
27.0	0.59	0.53	0.56	10087
28.0	0.55	0.61	0.58	10177
29.0	0.53	0.62	0.57	9901
30.0	0.56	0.65	0.60	10181
31.0	0.53	0.63	0.58	10139
32.0	0.43	0.35	0.38	10002
33.0	0.49	0.59	0.53	10072
34.0	0.60	0.65	0.63	10173
35.0	0.49	0.55	0.52	10174
36.0	0.45	0.40	0.42	9938
37.0	0.47	0.53	0.50	9959
38.0	0.49	0.61	0.54	10135
39.0	0.49	0.45	0.47	10237
40.0	0.37	0.36	0.36	10107

41.0	0.58	0.62	0.60	10075
42.0	0.51	0.64	0.56	9955
43.0	0.67	0.76	0.71	10053
44.0	0.45	0.58	0.50	10250
45.0	0.50	0.41	0.45	10194
46.0	0.33	0.33	0.33	10107
47.0	0.43	0.62	0.51	10213

### 4.3 색상 분석 모델

#### 4.4.1 색상 히스토그램 생성

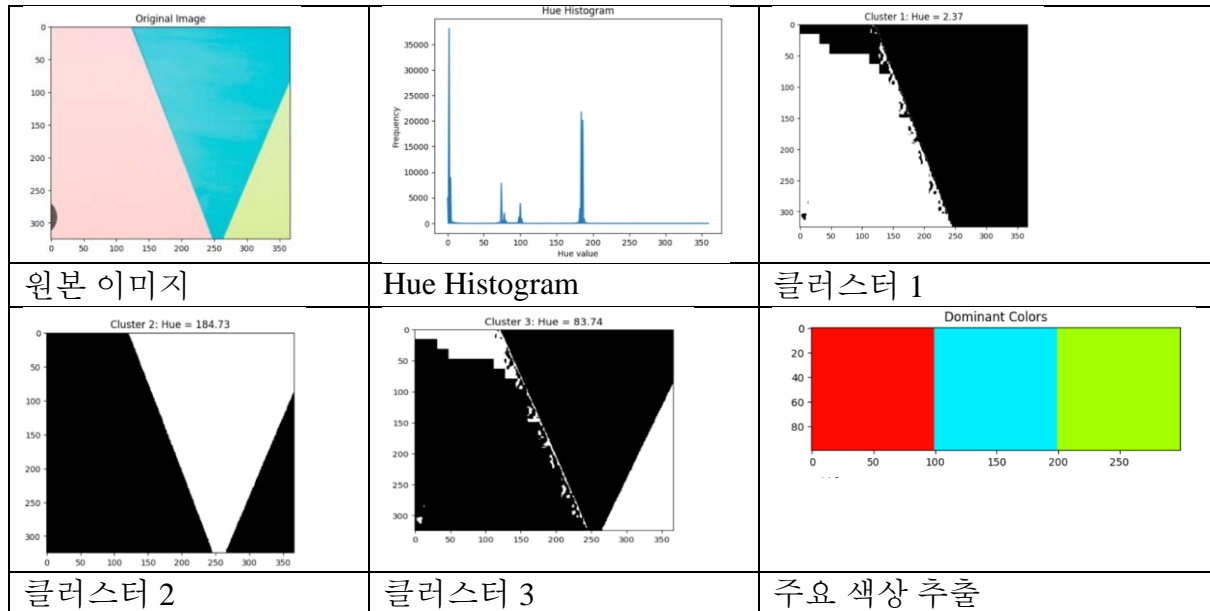
이미지를 불러와 RGB 에서 HSV(Hue, Saturation, Value) 색 공간으로 변환합니다. 이는 색상의 분포를 보다 쉽게 분석하기 위함 입니다. HSV 이미지의 Hue 채널을 사용하여 히스토그램을 계산합니다. Hue 값은 OpenCV 에서 0-180 범위로 나타나며, 이를 0-360 범위로 확장하여 색상 분포를 정확하게 분석합니다. 각 색상의 빈도수를 계산하여 히스토그램을 생성합니다. 히스토그램을 시각화하여 이미지에서 각 색상이 얼마나 많이 나타나는지 확인할 수 있습니다.

#### 4.4.2 히스토그램 클러스터링 기법

Hue 값을 기반으로 K-means 클러스터링을 수행하여 이미지에서 주요 색상(주요 클러스터)을 추출합니다. K-means 클러스터링은 다음과 같은 단계로 진행됩니다. 1) 이미지의 Hue 채널 값을 1 차원 배열로 변환합니다. 2) K-means 알고리즘을 사용하여 n 개의 클러스터로 분할합니다. 3) 각 클러스터의 중심값을 계산하여 주요 색상을 식별합니다. 여기서는 3 개의 클러스터를 사용하여 대표 색상을 찾습니다.

클러스터링 결과를 시각화하여 이미지에서 각 클러스터가 차지하는 영역을

확인합니다. 각 클러스터의 중심 색상도 함께 시각화하여 주요 색상을 명확하게 보여줍니다. 또한 가장 우세한 색상뿐만 아니라 두 번째로 우세한 색상을 추출하고, 이를 기반으로 추가적인 심리 상태를 분석합니다.



#### 4.4.3 색상 정보를 통한 심리 상태 분석 방법

주요 색상을 추출한 후, 해당 색상의 심리적 의미를 분석합니다.

심리 상태 분석 절차는 다음과 같습니다:

- 1)색상 추출: K-means 클러스터링을 통해 주요 색상을 추출합니다.
- 2)색상 분류: 추출된 색상을 특정 범위로 분류합니다.
- 3)심리적 의미 분석: 각 색상의 심리적 의미를 사전 정의된 규칙에 따라 해석합니다.

참고 문헌의 논문을 바탕으로 정의한 심리적 의미는 다음과 같습니다.

표 1. 색의 임상적의미  
Table 1. Clinical implication of color

색채명	임상적 의미(당위성확률, %)
회색	경계심(53.3) 열등감(45.3)
검은색	공포심(67.2), 감정억제(48.2)
빨간색	열등감(65.7), 공격성(55.5) 압박한감정(53.6), 불안(54.3)
주황색	동정과우애를구함(75.2), 성숙하고 싶은욕구(65.0), 행동보다 상상에 치중(43.8)
고동색	불안과 식욕(40.9%)
노란색	방어적 반응(32.9), 애정욕구(54.0), 퇴행(45.3)
녹색	허약, 피로(66.4), 자제력결핍(48.2), 복종(44.5)
파란색	깨끗이하려는욕구(81.8), 성장욕구(77.4), 억제하며추종(63.5%),
보라색	힘을 나타내고 싶은욕구(55.5) 우인관계나쁨, 질병(33.6)

표 2. 주조색의 조합 심리  
Table 2. Combination psychology of dominant color

색채명	주조색의 조합심리
빨강색 파랑색	부모의 높은 요구 수준을 압력으로 느끼는 표현.
빨강색 노란색	적개심, 공포감, 부모의 히스테리에 대한 변민, 애정불만.
빨강색 녹색	열등감이 강하고 어머니에 대한 불만.
주황색 노란색	동정을 구하거나 성숙하고자 하는 욕구가 강하거나 애정에 대한 갈망.
파란색 고동색	대소변 가리기에 문제가 있다.
주황색 파랑색	지능이 낮고 우둔한 어린이는 자기 몸의 불결함을 말함을 의미.

아래는 위와 같은 표 내용을 바탕으로 색상 정보를 추출한뒤 심리를 분석한  
내용입니다.

Primary Color: green  
Secondary Color: orange  
임상적 의미와 당위성 확률:  
Primary color (green): 희망, 피로(66.4), 자제력결핍(48.2), 불쾌(44.5)  
Secondary color (orange): 동정적, 애호구립(75.2), 성취하고 싶은 욕구(65.0), 행동보다 상상의 치중(43.8)  
단일 색채의 심리:  
green: 균형을 상징으로 감정을 강하게 표현하지 않고, 충동적인 감정을 통제하는 내성적인 유아들이 많습니다.

## 5. 결론 및 제언

### 5.1 연구 결과 요약

본 연구에서는 아동 그림에서의 객체 탐지와 색상 분석을 통해 아동의 심리  
상태를 예측하는 모델을 개발하였습니다. YOLOv5 모델을 사용하여 주요 객체를  
탐지하고, CNN 과 Transfer Learning 모델을 통해 객체를 분석하였으며, K-means  
클러스터링을 통해 객체의 위치와 색상 정보를 각각 분석하였습니다. 또한, Random  
Forest 와 LightGBM 을 사용하여 객체 분류 모델을 개발하였습니다. 색상 히스토그램을  
분석하여 주요 색조를 두개 뽑아내고 심리 분석에 활용하였습니다.

## 5.2 연구의 의의와 기여

본 연구는 아동의 심리 상태를 실시간으로 분석할 수 있는 도구를 제공하여 아동의 정신 건강 평가와 치료에 크게 기여할 수 있습니다. 특히, 색상 정보와 객체의 위치 정보를 결합한 분석을 통해 아동의 내면 세계를 보다 정밀하게 이해할 수 있습니다. 이는 아동의 심리적 문제를 조기에 발견하고 적절한 치료를 제공하는 데 중요한 역할을 할 수 있습니다.

## 5.3 향후 연구 방향 제안

향후 연구에서는 다음과 같은 방향을 제안합니다:

1. 더욱 다양한 데이터셋을 활용하여 모델의 일반화 성능을 향상시킬 필요가 있습니다.
2. 객체 탐지와 색상 분석을 결합한 앙상블 모델을 개발하여 예측 성능을 더욱 향상시킬 수 있습니다.
3. 실시간 분석을 위한 최적화 기법을 도입하여 모델의 효율성을 높일 수 있습니다.

이와 같은 연구의 구성을 통해 아동 그림을 분석하여 심리 상태를 자동으로 파악하는 알고리즘을 체계적으로 개발하고 그 성능을 평가할 수 있습니다.

## 6. 참고 문헌

AI 허브 데이터셋 링크:

<https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=71399>

색상 및 위치에 따른 심리 분석 논문 :

<https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=JAKO201322045164288&ocn=JAKO201322045164288&dbt=JAKO&journal=NJOU00304070>

