

딥러닝 작업을 위한 오픈소스 기반 이미지 어노테이션 시스템 구축 및 공유

Implementation of Open Source Image Annotation System for Deep Learning Task

요 약

최근 4차 산업시대에 접어들면서 대용량 데이터 처리와 패턴 인식을 위한 기술이 발전하고 있다. 특히 대량의 이미지에서 여러 가지 특징을 추출하고 이를 딥러닝 작업에 활용하는 기술을 많이 사용하고 있다. 하지만 기존의 이미지 특성을 추출하는 도구들은 사용자가 직접 임의의 크기를 결정하고, 태그 이름을 입력해야하는 불편함을 가지고 있으며, 대량의 데이터를 다루고 처리해야하는 시스템 구현과 개발 기간의 부담이 있다. 본 논문에서는 서버 구현과 운영, 관리의 부담을 줄이는 서버리스 시스템을 구축하고, 오픈소스를 활용하여 학습 이미지를 편리하게 추출할 수 있는 시스템 개발과 플랫폼 공유를 제안한다.

1. 서 론

최근 IT 기술의 발전으로 4차 산업시대에 접어들면서, 그 동안 정보 시스템에 축적된 대량의 데이터 활용에 큰 이목이 집중되고 있다. 특히 딥러닝을 적용한 이미지 분류(Classification) 시스템이 크게 성장하고 있다[1].

이미지 분류를 위한 딥러닝 작업은 대량의 트레이닝 데이터가 필요하며, 대부분의 트레이닝 데이터는 이미지 샘플 범위 내에서 동일한 크기(e.g., 64x64, etc)로 추출된 이미지와 각 특징을 분류할 수 있는 태그(Tag) 이름이 필요하다.

이미지 어노테이션 시스템은 이미지 샘플에서 사용자들이 특징을 직접 추출할 수 있도록 도와주는 시스템이다. 그러나 모든 작업을 사용자가 직접 담당하기 때문에 매번 동일한 크기의 이미지를 추출하기란 쉽지 않으며, 항상 태그(Tag) 이름을 입력해야하는 불편함이 있다.



(그림 1) 비효율적인 어노테이션

이를 단순화된 시스템으로 구축하고 공유하는 과제가 필요하며, 오픈소스 활용으로 과제 해결에 소모되는 개발 시간을 단축할 수 있었다. 대표적인 어노테이션 오픈소스는 LabelMe[2], Image Annotation Programme[3], 그리고 Annotorious[4]가 있으며, 모두 Web 기반 오픈소스다.

LabelMe와 Image Annotation Programme은 과제 해결을 위해 기능을 단순화하려면 전체 또는 일부 소스코드 수정이 필요하다. 반면에 Annotorious는 모든 기능이 모듈화로 구현되어 있어 소스코드를 직접 수정할 필요 없이 과제 해결을 위한 기능을 모듈로 만들어 제공할 수 있다.

안내되는 주소를 통해 소스코드를 확인할 수 있다. <https://github.com/annotorious/annotorious/releases/tag/v0.6.4>

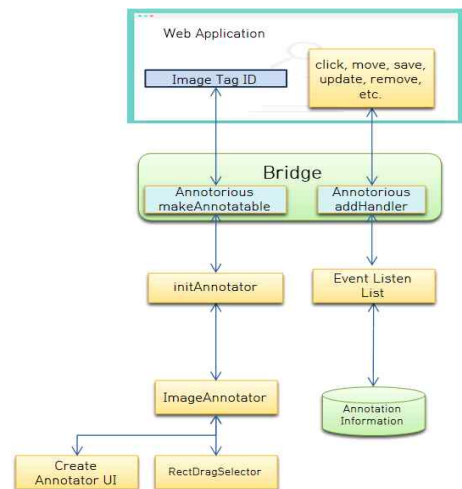
본 논문에서는 딥러닝 작업을 위한 이미지 트레이닝 데이터를 편리하게 추출할 수 있도록 단순화된 시스템을 구축하고 오픈 플랫폼으로 공유한다.

2. 이미지 어노테이션 시스템 구축

Annotorious는 웹 페이지(HTML) 구성 요소 중 이미지를 보여주기 위한 요소 ''의 태그 ID를 makeAnnotatable 메서드(method)가 참조한다.

참조된 이미지 태그 ID는 ImageAnnotator 메서드가 접근하여 (그림 2)와 같은 순서도를 따라 어노테이션 환경을 구축한다.

또한 addHandler 메서드를 사용하여 어노테이션 정보를 저장하거나 수정하는 기능을 구현할 수 있다.



(그림 2) Annotorious Architecture

2.1 기존 Annotorious 시스템의 고찰



(그림 3) Annotorious 사용 예

대부분의 딥러닝을 적용한 이미지 분류 작업에서 트레이닝 데이터들은 동일한 크기의 이미지 데이터들이 사용된다.

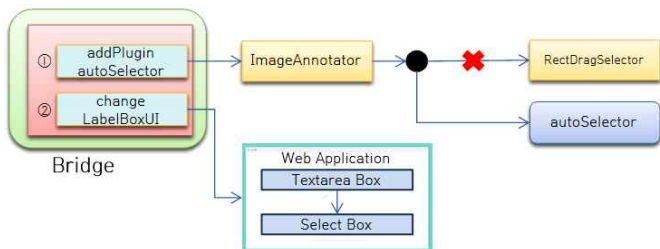
그러나 기존의 Annotorious 시스템의 추출 단계는 각 특징을 동일한 크기의 이미지로 추출하는 것과, 태그(Tag) 이름을 매번 입력하는 것은 쉽지 않으며, 다른 어노테이션 시스템들도 유사한 문제점을 가지고 있다.

하지만 Annotorious는 다른 시스템과 비교했을 때, 모듈 구현이 용이한 점이 있기 때문에 과제 해결을 위한 트레이닝 데이터 추출의 단순화 구현에 집중할 수 있었다.

2.2 기능 단순화 모듈 구현

본 논문에서 제안한 어노테이션 시스템 구축은 두 가지 모듈을 구현하고, 이를 적용하여 오픈 플랫폼으로 공유하는 것이다.

구현된 시스템 소스코드는 안내되는 주소로 접속하여 확인할 수 있다.
<https://github.com/kmu-bigdata/annotorious/tree/master/autoselectorPackAndExam>



(그림 4) 모듈 구현과 적용 순서도

첫 번째로 구현할 기능은 마우스 클릭 지점을 중심으로 사각형(e.g., 64 x 64, etc)의 이미지 특징을 추출하는 플러그인 모듈 구현이며, 두 번째는 추출하려는 특징을 표현할 수 있는 태그(Tag) 이름을 미리 만들어 둔 List에서 선택할 수 있도록 구현하는 것이다.

```
annotorious.plugin.autoSelector.Selector.prototype.drawRect =
function (click_x, click_y) {
  if (this._drawLocking == false) {
    this._g2d.strokeStyle = "#35E5F1"; // 청색
    this._g2d.lineWidth = 2;

    var x = click_x - 32;
    var y = click_y - 32;
    var w = 64;
    var h = 64;

    if (x < 0) {
      x = 0;
    }
    if (y < 0) {
      y = 0;
    }
    this._g2d.strokeRect(x, y, w, h);
  }
}
```

(그림 5) 64 x 64 정사각형 소스코드

addPlugin으로 모듈을 추가할 경우 (그림 4)의 ①처럼 기존의 사용하고 있던 모듈을 새로 구현한 autoSelector 모듈로 대체한다.

(그림 5)는 추출하려는 이미지의 특징을 클릭 할 때, 정사각형 테두리를 그려주는 소스코드다.

```
function annotorious_changeToTagList() {
  $('annotorious-editor-text.goog-textarea')
  .replaceWith("<select id='selectTagName'>" + $(this).text() + "</select>");

  var tagNames = ["frill", "red", "goldEmbroidery", "check", "line", "black"];

  tagNames.forEach(tag => {
    var typeOption = document.createElement('option');
    var tagOption = document.createTextNode(tag);
    typeOption.appendChild(tagOption);

    document.getElementById("selectTagName").appendChild(typeOption);
  });
}
```

(그림 6) 태그 입력 UI 변경

②는 추출하려는 특징의 태그(Tag) 이름을 직접 입력하는 방식에서 List에서 선택할 수 있도록 UI를 변경하는 기능으로 (그림 6)에서 소스코드를 확인할 수 있다.

```
function bridge() {
  anno.makeAnnotatable(document.getElementById('preview'));
  anno.addHandler('onAnnotationCreated', function (annotation) {
    getData(annotation);
  });
  adding_customModule();
};

function adding_customModule(){
  annotorious_LabelBoxUI_custom(); // for Customizing
  anno.addPlugin('autoSelector', { activate: true });
};

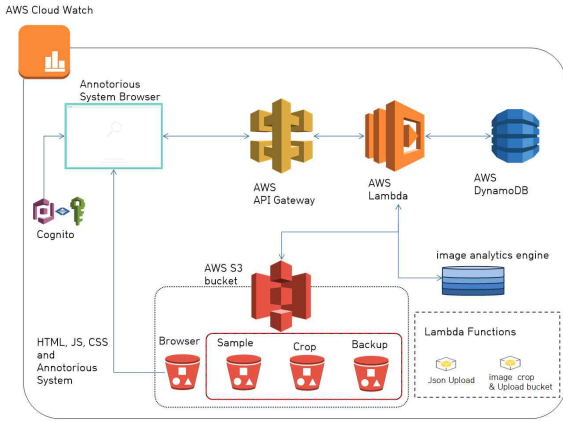
function getData(annotation) {
  text = $("#selectTagName option:checked").text();
  annotation.text = text;
  sourceImage = annotation.src;
  geometry = annotation.shapes[0].geometry; array form.
  changeJson(text, sourceImage, geometry);
};
```

(그림 7) 시스템 구축을 위한 JS 예제 소스코드

구현한 모듈을 적용한 어노테이션 시스템 구축의 Javascript 예제 소스코드를 (그림 7)에서 확인할 수 있다.

3. Serverless 웹 어플리케이션 구축 및 시스템 연동

딥러닝 작업을 위한 시스템 구축은 대량의 데이터를 저장하고 처리할 수 있는 고성능 서버가 필요하며, 데이터 처리를 위한 다양한 어플리케이션을 구축하거나 개발해야한다.



(그림 8) Serverless Web Application Architecture

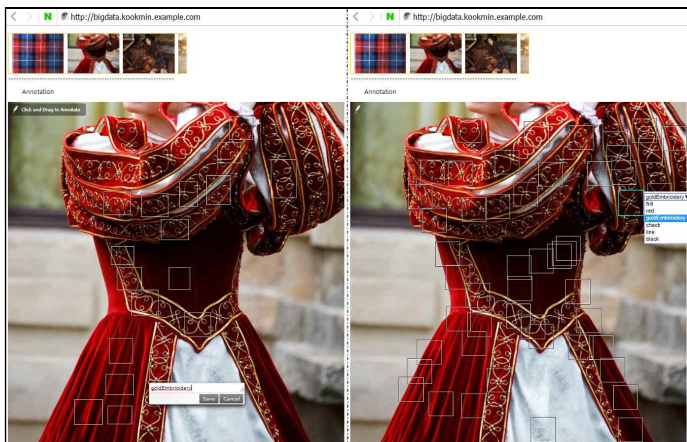
반면에 AWS를 활용한 서버리스(Serverless) 웹 어플리케이션을 구축할 경우 별도의 서버를 구매하거나 관리할 필요가 없었으며, 데이터 처리를 위한 다양한 어플리케이션이 제공되어 시스템 구축과 모듈 구현에 집중할 수 있었다. 어노테이션 시스템과 서버리스 웹 어플리케이션 연동 구조는 (그림 8)과 같다.

서버리스 웹 어플리케이션 구조에서 S3 버킷(bucket)은 저장소 역할을 담당하며, Browser 버킷은 웹 페이지 구성요소와 어노테이션 시스템 구축에 필요한 소스코드가 저장되고, 작업에 필요한 이미지와 결과물들은 각각 Sample, Crop, Backup 버킷에 저장된다.

Dynamo 데이터베이스는 어노테이션 작업 환경을 효율적으로 관리하기 위해 이미지의 저장위치, 작업 상태와 같은 필요최소의 정보를 저장한다.

그리고 람다(Lambda)는 어노테이션 시스템의 사용자 요청과 작업 결과를 처리하여 S3 버킷이나 Dynamo 데이터베이스로 요청 및 작업 결과를 반영한다.

4. 구현 모듈을 활용한 이미지 추출 결과



(그림 9) 결과 비교

Annotorious를 포함한 기존 어노테이션 시스템들은 이 이미지에서 추출하려는 특징을 동일한 크기로 추출하기 어렵거나, 태그(Tag) 이름을 직접 입력해야하는 번거로움이 있었으며, 딥러닝 작업에 필요한 대량의 트레이닝 데이터 추출이 편리하지 못 했다.

img20_frill_8.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 1.59KB
img20_frill_9.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 1.38KB
img20_frill_10.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 1.20KB
img20_frill_11.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 1.06KB
img20_goldEmbroidery_0.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 2.12KB
img20_goldEmbroidery_1.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 2.17KB
img20_goldEmbroidery_2.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 1.89KB
img20_goldEmbroidery_3.jpg	유형: JPG 파일 사진 크기: 64 x 64	크기: 2.27KB

(그림 10) 추출 이미지

반면에 본 논문에서 구현한 플러그인 모듈을 활용하면 원본 이미지에서 특징을 단순히 클릭하는 것으로 편리하게 추출할 수 있음을 Python으로 개발된 이미지 추출 프로그램을 연동하여 (그림 10)의 결과를 확인할 수 있었다.

5. 결론 및 향후 계획

최근 이미지를 활용한 딥러닝 적용 기술의 관심이 증가하고 연구가 활발히 진행되고 있는 가운데, 기존의 어노테이션 도구들을 사용한 트레이닝 데이터 추출 작업은 비효율적이다.

본 논문에서 제안한 시스템은 기존의 어노테이션 작업을 보다 효율적이고 편리하게 트레이닝 데이터로 추출할 수 있음을 결과를 통해 확인할 수 있었으며, 향후 이번에 구현한 모듈을 활용하여 이미지 분석을 위한 딥러닝 시스템 연구를 계획할 수 있을 것으로 기대하고 있다.

참 고 문 헌

- [1]:김지원, 표현아, 하정우, 이찬규, 김정희 (2015). 다양한 딥러닝 알고리즘과 활용. 정보과학회지, 33(8), 25-31.
- [2]:B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman. LabelMe: a Database and Web-based Tool for Image Annotation. International Journal of Computer Vision, 77(1-3):157-173, 2008
- [3]:https://github.com/frederictost/images_annotation_programme
- [4]:<https://github.com/annotorious/annotorious>