# 

# Student Project Proposal

| Project Title | Annotheta, a Smart Annotation Platform |
| --- | --- |
| Industry Sponsorship (if Any) | Samsung SDS Research America |

## Project Description

**Problem definition**

*[50-100 word description of the problem which you will solve]*

| Image data annotation is a challenging task since it requires a rigorous manual training process that is time-consuming and cost-intensive to ensure standardized outcomes. Yet we can see the need out there by the language people use when describing this activity---hard, painful, horrible---and the dollar amount businesses are willing to pay for it. it is extremely important in machine learning tasks yet one of the major pain points associated with using unrefined data. In fact, according to a 2021 report from [Grand View Research](https://www.grandviewresearch.com/industry-analysis/data-collection-labeling-market), the estimated value for the global data collection and labelling market was USD 1.3B in 2020, with a projected compound growth rate of more than 25% from 2021 to 2028.  Our aim is to target the most expensive and time consuming aspect of this challenge, manual labelling effort, by automating object detection and initial segmentation to ease this process. |
| --- |

**Key Research Questions/ Technological constraints that the Project will Answer**

| 1. Is it possible to standardize and minimize manual annotation errors for outdoor (and indoor) images? 2. How much improvement in time can an automated detection and segmentation system provide? 3. What quantitative and qualitative metrics should be considered when designing an automated image labelling system? 4. How do we balance the trade-off between speed and accuracy in a labelling task? In what cases will such an automated annotation system necessarily fail on the one hand (speed) or the other (accuracy)? |
| --- |

**Final deliverables at the end of the project**

*[Please list the desired technical deliverables from the project team in as much detail as possible]*

| 1. A simple interactive web front end for receiving user input. This page will not be fancy because our focus will be in the detection/segmentation models. 2. A deployed ML back end with reasonable latency. This will be our focus. 3. Basic REST API endpoints to send/receive data exchanges between front and back ends, in standardized formats. 4. Written technical report about the Data, Process and Outcomes. |
| --- |

**Key activities/ technologies the project team may be expected to undertake/ work with**

*[E.g. What kind of technology stack will you work with, the datasets you may need to work on, what kind of analysis you may be expected to undertake, etc.]*

| 1. For training, we will use MS COCO dataset <https://cocodataset.org/> or ADE20K data set with indoors and outdoors images (<https://groups.csail.mit.edu/vision/datasets/ADE20K/>). Our request for access is currently pending approval. If we’re unable to get approval, we will use an older version from Kaggle (<https://www.kaggle.com/datasets/soumikrakshit/ade20k>). Another option is to use the CityScapes dataset (<https://www.tensorflow.org/datasets/catalog/cityscapes>) 2. We hope to improve upon CurveGCN (<https://github.com/fidler-lab/curve-gcn>), or PolyRNN++ (<https://github.com/fidler-lab/polyrnn-pp>). But the code is currently closed-source for CurveGCN. If we have problems with getting the code, our fallback is to use Mask RCNN (<https://github.com/matterport/Mask_RCNN>). Other possibilities as pointed out by Bruno are PSPNet (<https://hszhao.github.io/projects/pspnet/>) or DEXTR (<https://cvlsegmentation.github.io/dextr/>) 3. Implement REST API using Flask or FastAPI. 4. The final deployable ML system will contain an annotation front end to receive user-inputs and a backend that will detect the exact object edges. The backend can be deployed on AWS or GCP. |
| --- |

**Expected learning outcomes**

*[What do you expect to learn from the project? Please mention the technical skills you will imbibe over the project.]*

| 1. Hands-on experience with replicating an annotation platform and deploying it on cloud services. 2. Understanding the concept of object detection coupled with semantic segmentation for fast annotation. 3. Hands-on experience with ML Pipelines and Reporting outcomes. |
| --- |

| Team Size: | 3 |
| --- | --- |
| Member names: | Gonzalo Tixilima, Juswaldy Jusman, Zan Butt |

### 

## Tentative Time plan

Week 8 – 27 Mar to 2 Apr

Research and Submit Proposal

Week 9 – 3 Apr to 9 Apr

Start with Mask RCNN Backend

Research Segmentation Alternatives

Research Annotation Formats

Start Jupyter Notebook Frontend

Week 10 – 10 Apr to 16 Apr

Finish Mask RCNN Backend

Implement First Segmentation Alternative

Decide Annotation Format

Finish Jupyter Notebook Frontend

Start HTML Frontend

Week 11 – 17 Apr to 23 Apr

Finish Annotation Format Development

Start API Development

Second Segmentation Alternative

Week 12 – 24 Apr to 30 Apr

Finish API Development

Refine Frontend

Decide Segmentation Strategy

Week 13 – 1 May to 7 May

Deploy to Cloud and Test End-to-end

Fine-tune Model

Refine Process

Refine Documentation

Week 14 – 8 May to 14 May

Test End-to-end

Fine-tune Model

Focus on Outcomes

Start Final Report & Presentation

Week 15 – 15 May to 21 May

Finish Final Report

Final Presentation & Demo

## System Design

For the data we might use the COCO data format or another standard.

Planned iterations:

1. The first iteration will be mostly experimental. We will aim for a baseline with Jupyter Notebook frontend.
2. The second iteration will be more set on the backend, and we will aim at having a minimal pipeline and a frontend.
3. The final iteration will be a working end-to-end annotation system.

## Ethical Considerations

If possible we would like to avoid using images containing children, or military equipment. We don’t want this to be used for by the military and or on images that may contain or give away personal information.