Fire Risk Profiles

Image Recognition on Floor Plans as Input for a Fire Risk Model

University of Amsterdam - Data Systems Project

Will Chien, Jelle van Elburg, Anton Kozačkov, Elisabeth Kräman, Marta Turek
Group E4

Outline

- The Problem
- The Model
- The Prototype
- Prototype Evaluation
- Conclusions & Future Work
- Questions

The Problem

- Problem definition
- Limitations / challenges
- Solution definition

Problem Ideation Insight

The goal of this project is presently more about ***gathering inputs*** *than it is about creating new models, given the number of missing inputs for any potential risk model.*

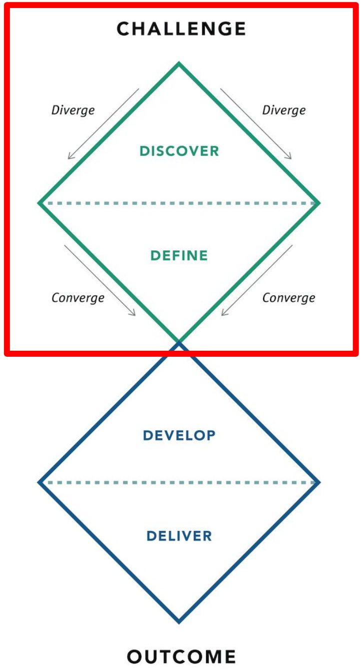
Problem Discovery & Definition

Discover: Explore the Problem

- Plans must be submitted for permits
- Plans contain key information on possible fire risk and impact of fire

Define: Decide What to Fix

- Computer vision model to parse floor plan images
- Extracting key information : compartments, doors, windows



Challenges

Model:

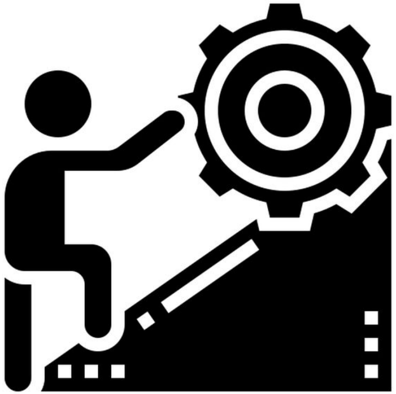
- Working with varied data
- Computationally heavy & expensive to train model (GPU cost)

Data:

- Target data not available
- Annotated data even rarer

Prototype:

- Visually appealing way to show model and results
- Interface

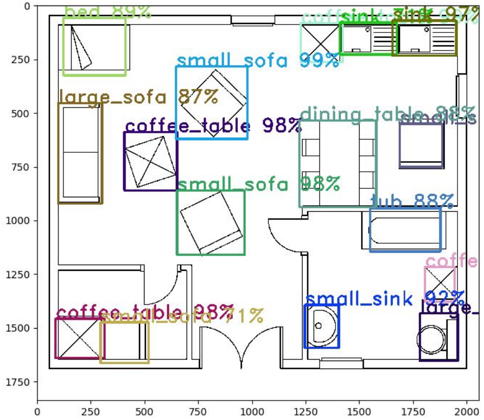


The Model

- Model selection
- Processing pipeline
- Data annotations
- Model architecture
- Hyperparameter tuning
- Output post-processing
- Model Evaluation

Object detection/ Instance segmentation

- Recognise floor plan elements (door, windows , walls)
- Common problem in computer vision
- Previous work on similar problems available
  - Algorithmic
  - Machine learning



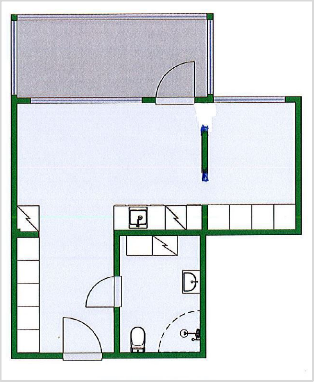
Algorithmic/ML

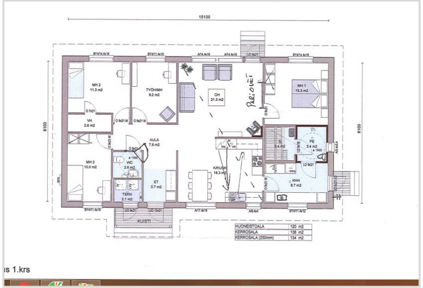
Algorithmic

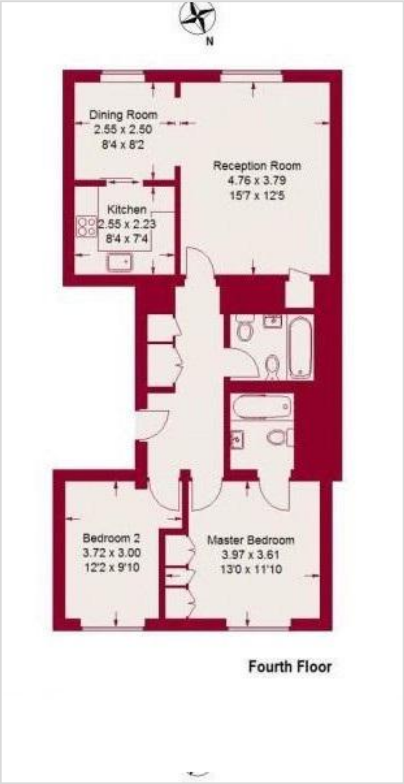
- Relies on relation between elements
- Lacks generality in terms of noise and datasets

Machine learning

- Robust to variation
- State-of-the-art
- However, annotated data required
    - Only recently used, little annotated data







Room Function Extraction

- According to Red Cross, most house fires start in kitchen
- Function encoded in symbols/text
- Room function may predict escape route

Only model with room function prediction:
DeepFloorplan (Zeng et al., 2019)

- Published 2019
- Semantic Segmentation (pixel wise labeling)
- Convolutional Neural Network with TensorFlow

Data Processing Pipeline

Data Preparation

- Web-scraping
- Floor plans annotations

Model Implementation

- Train/test split
- Model training
- Hyperparameter fine-tuning

Model Evaluation

- Check accuracy on test sets

Training data preparation

Public training dataset:

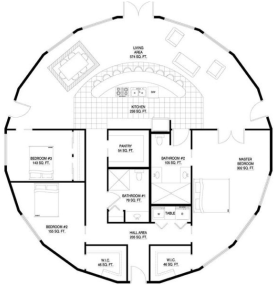
1. R2V - 815 images - Japan
2. R3D - 214 images - New York

Adding new data can enrich the model from two aspects:

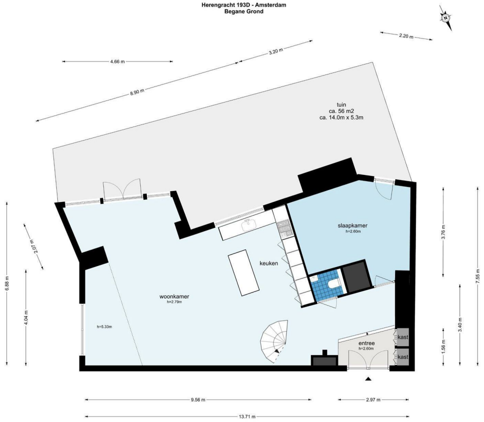
1. Dutch floor plan style
2. In Dutch language

A. Japan

B. New York

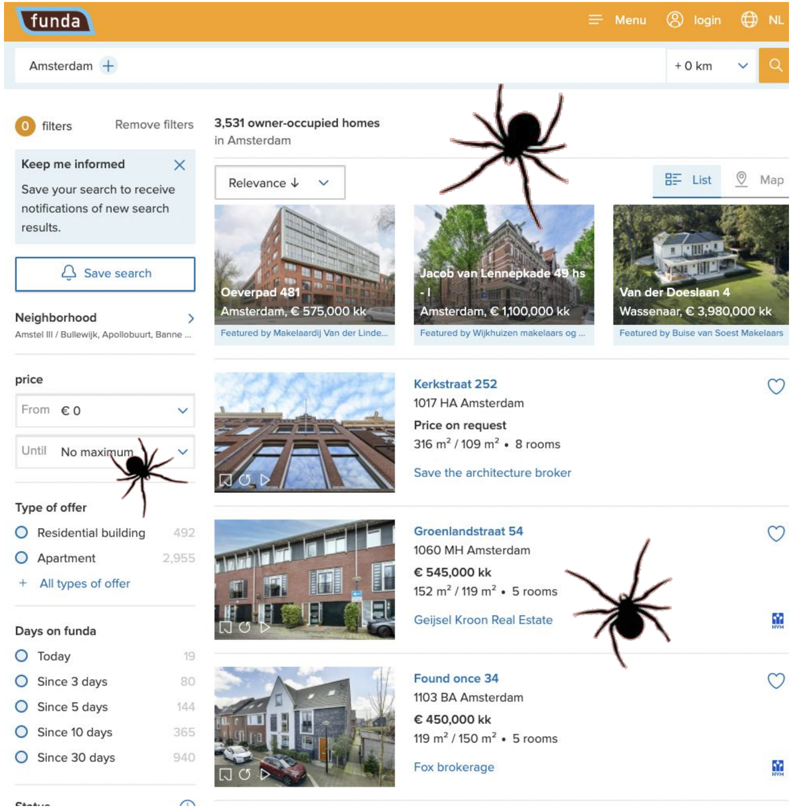


C. Amsterdam



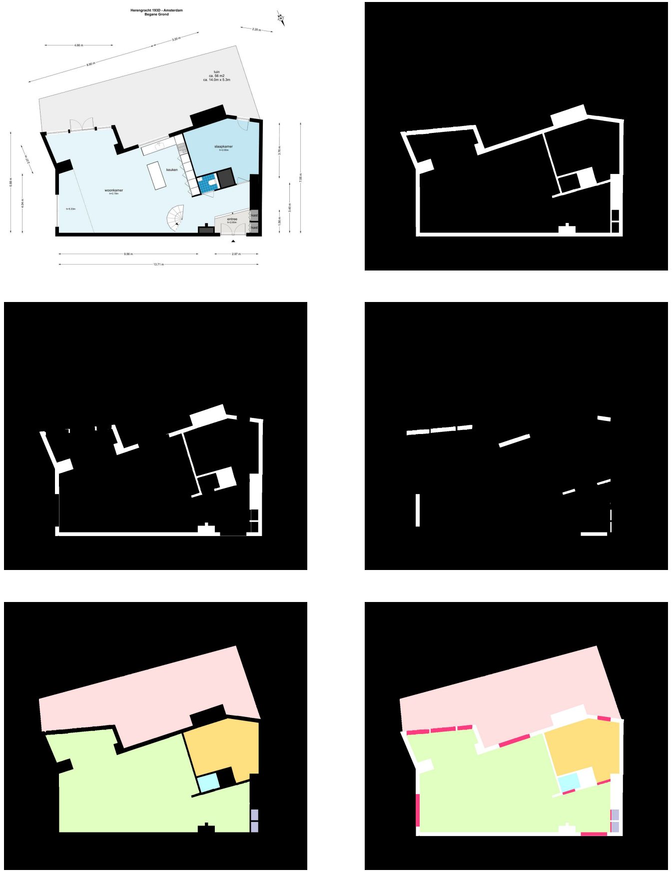
Web-scraping

- No access to the target floor plans due to privacy issues
- Alternatively, we scraped floor plan images from Funda automatically as supplementary training data



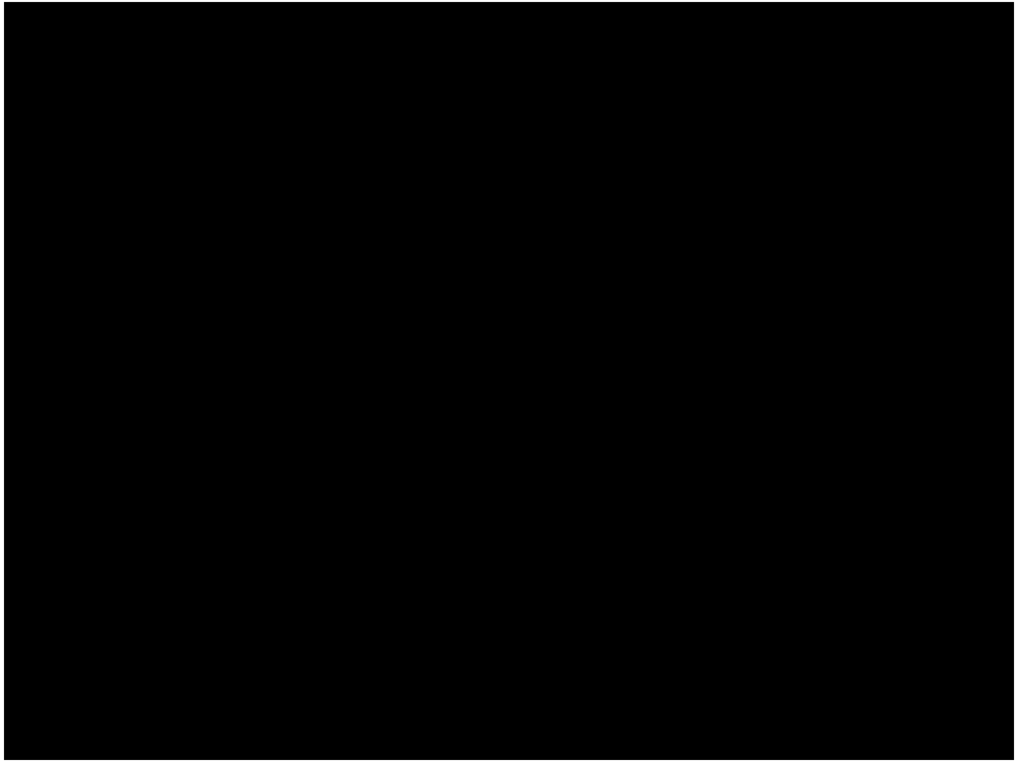
Data annotation

- Annotate each pixel in Photoshop/ Photopnea
- Time to annotate one floor plan: 45 - 60 minutes



Data annotation

Make sure every single pixel is labeled.

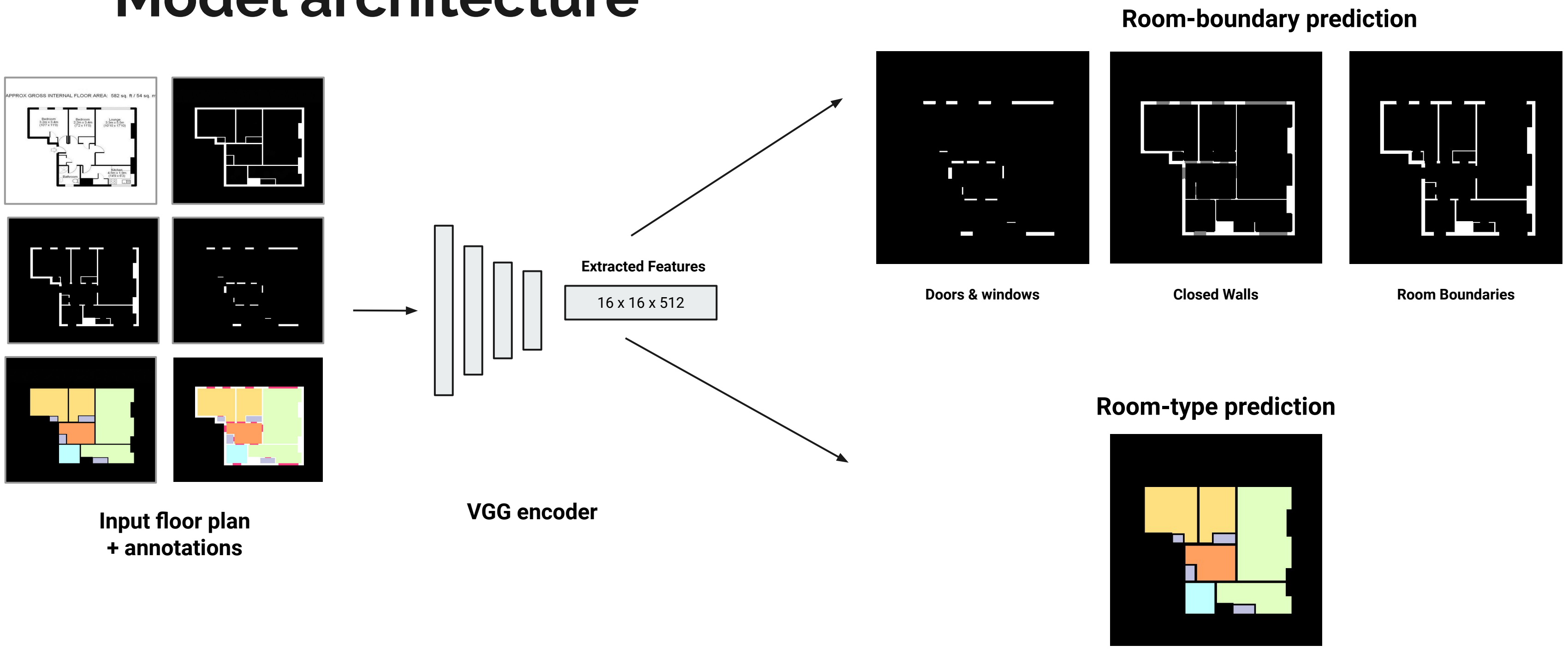


200 images * 45-60 minutes/image =

150 - 200 hours

Floor plan annotations are **labour intensive.**

Model architecture



Model Training

Datasets

New York Dataset
214 rectangular images
18 circular images

Train
179 images

Test
53 images

Network Training

Rented Paperspace machine
NVIDIA P5000
16 GB GDDR5X

Optimizer
Employed Adam optimizer
to update parameters

Hyperparameter Settings

Batch size: 1
Batch size > 1, model
struggles with prediction

Epochs: variable
Model runs until it
converges

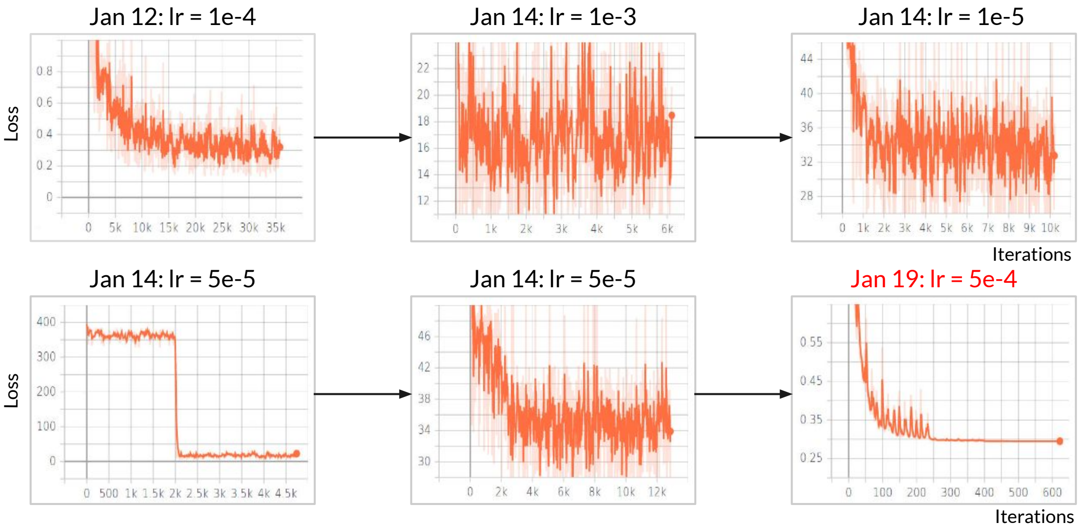
Step Size: 25
Epochs to allow before
decreasing learning rate

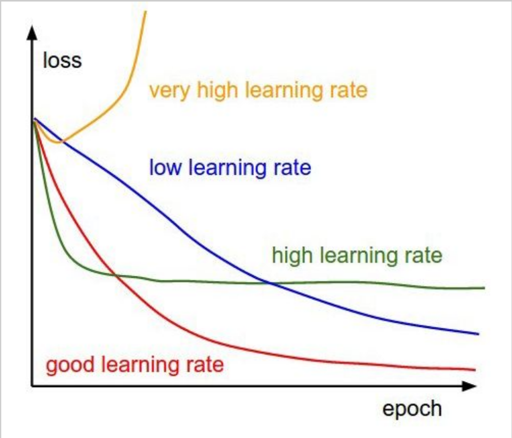
Gamma: 0.3
By how much to decrease
learning rate every step size

Learning Rate: 5e-4
How fast or slow we move
to optimal weights

Hyperparameter tuning - a focus on learning rate

Learning rate (lr) is a key hyperparameter for tuning neural networks





Effect of various learning rates on convergence (Img Credit: cs231n)

Model Evaluation Metrics

Overall pixel accuracy

$$overall\ accu = \frac{\sum_i N_i}{\sum_i \widehat{N}_i}$$

Per-class pixel accuracy

$$class\ accu(i) = \frac{N_i}{\widehat{N}_i}$$

$\widehat{N}_i$ is total number of the ground-truth pixels

N_i is correctly-predicted pixels for the *i-th* floor plan element

Model Evaluation

	Model_20_lr5-e4	Model_AMS3_lr5-e4
Overall accuracy	*0.7759*	*0.6328*
Room type (Mean accuracy)	*0.4267*	*0.1425*
Room type + boundary (Mean accuracy)	*0.5324*	*0.3075*

The Prototype

- Requirements
- Development process
- Prototype demo

MVP Requirements

Functionality

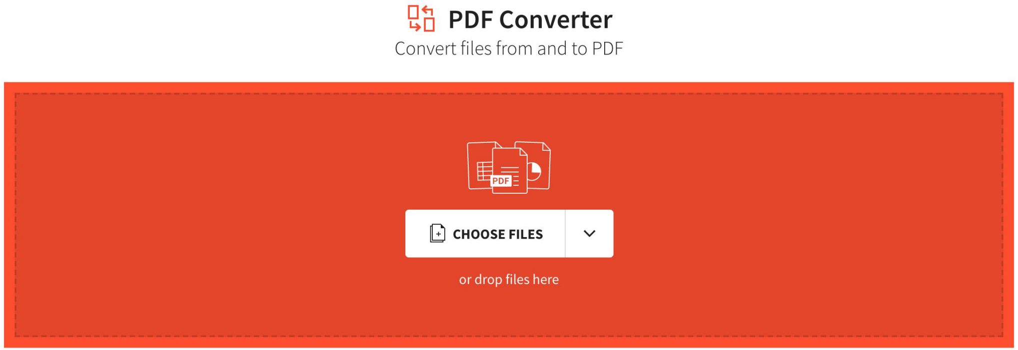
- Floor plans -> extracted data CSV
- Batch processing
- Extracted data visualization

Environment

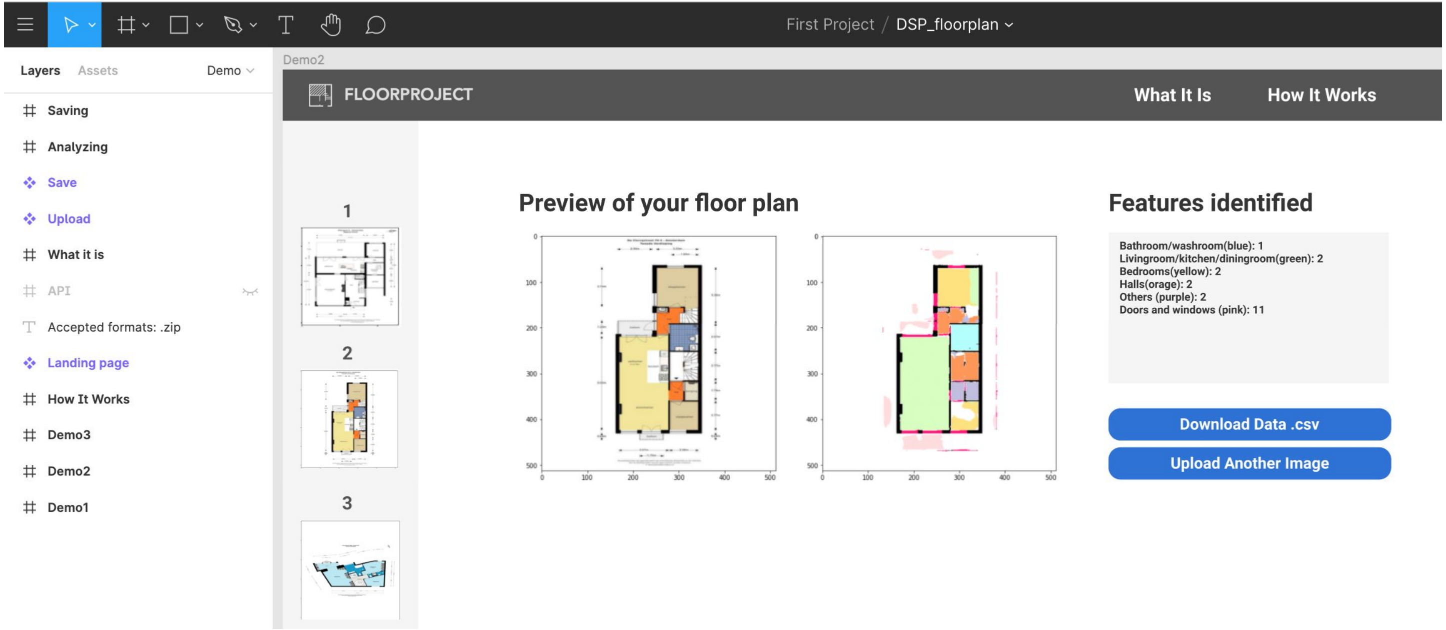
- Data security
- Hardware constraint GPU
- Web application

Competitive usability analysis

- Google Vision AI (similar intended function)
- Smallpdf (similar intended structure)



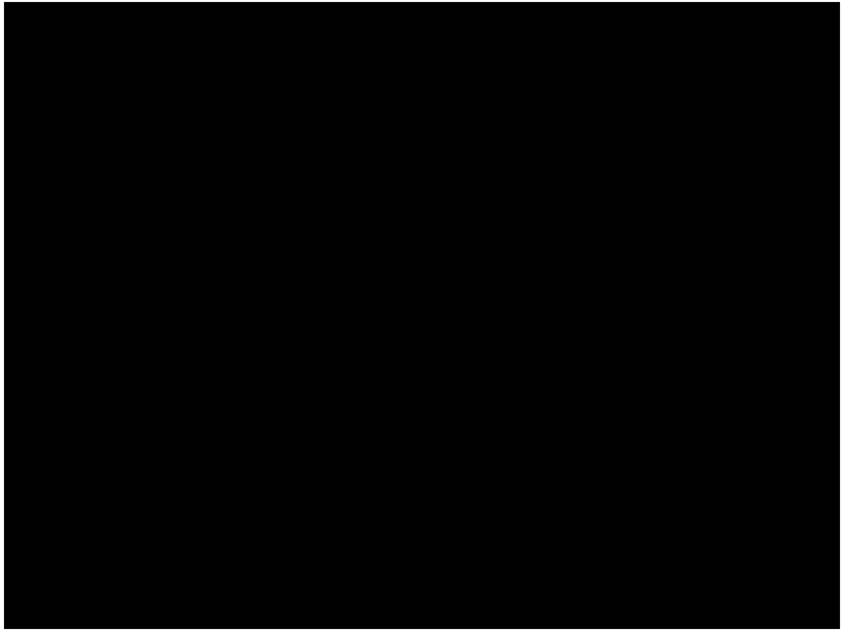
First Iteration: Figma click dummy prototype



Technology used

- Backend - Django REST API
- Frontend - React Javascript framework
- Deployed to remote cloud infrastructure

Demo



Prototype Evaluation

- Usability testing

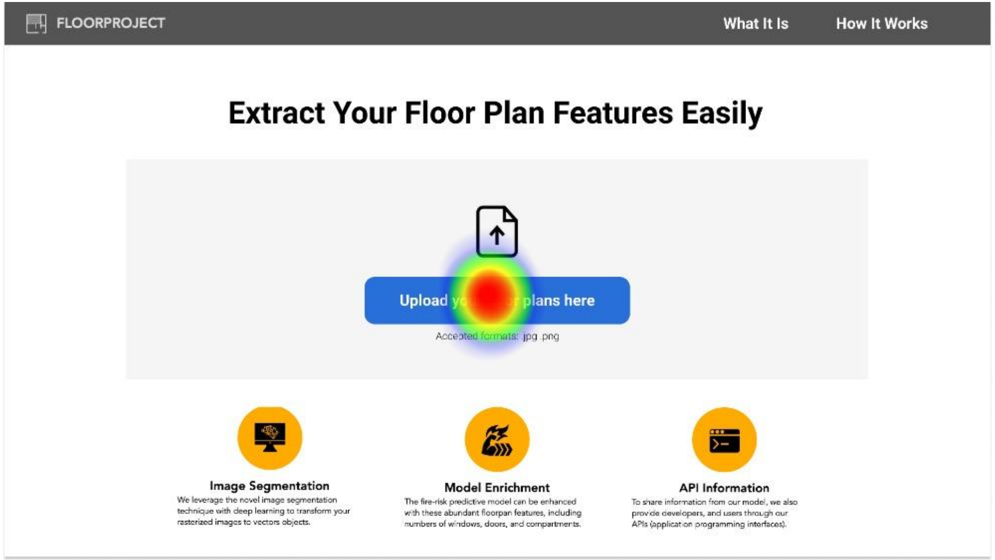
Usability Testing - UX test 1

- Test on Figma prototype using Maze and Zoom
- Explorative usability test
- 5 participants
  - Information Studies
  - High digital literacy
- Different tasks and questions

Usability Testing - Example Maze

Usability Testing - Results 1

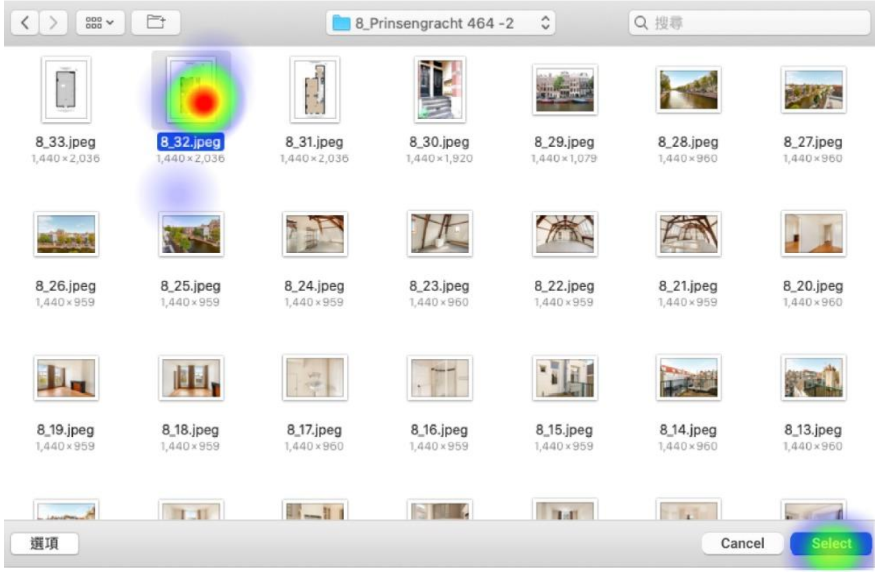
- **Function was clear**
- **Intuitive and easy to use**
- Output page and data presentation was not clear
  - Extracted features table unclear
  - Adding a legend would help
- Reliability somewhat questionable



Figma landing page heatmap

Usability Testing - Results 1

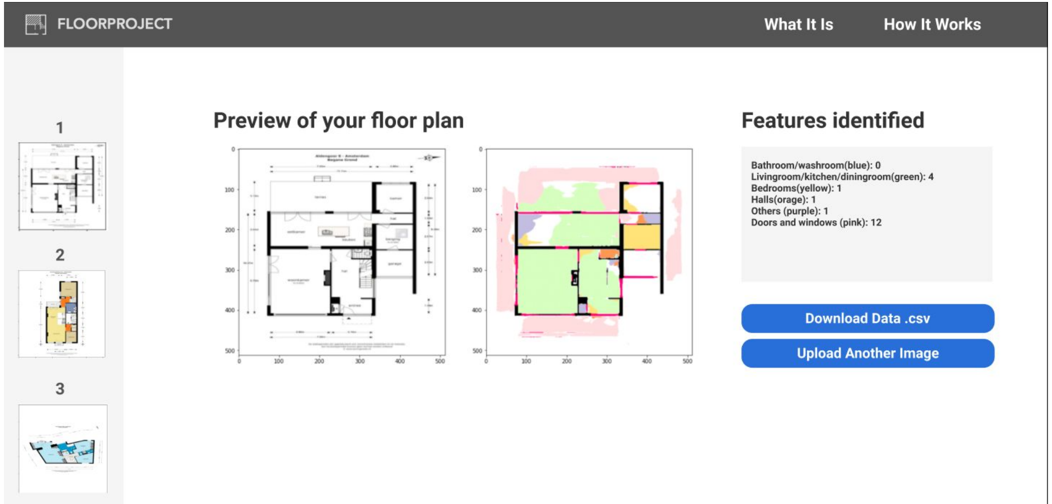
- **Function was clear**
- **Intuitive and easy to use**
- Output page and data presentation was not clear
    - Extracted features table unclear
    - Adding a legend would help
- Reliability somewhat questionable



Figma upload page heatmap

Usability Testing - Results 1

- Function was clear
- Intuitive and easy to use
- **Output page and data presentation was not clear**
    - Extracted features table unclear
    - Adding a legend would help
- Reliability somewhat questionable



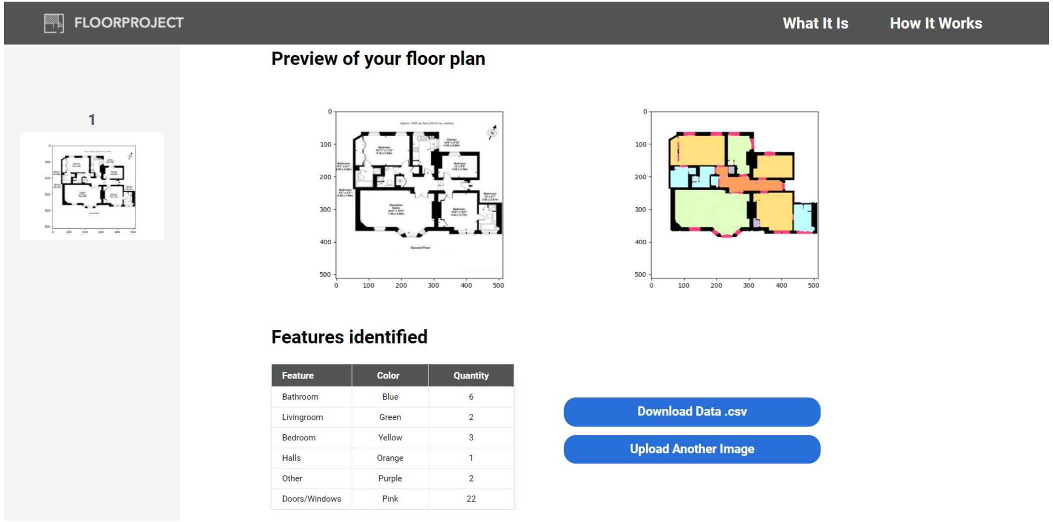
Figma output page

Usability Testing - UX test 2

- Test on real prototype using Zoom
- Classic usability test
- 5 participants
  - Information Studies
  - High digital literacy
- Tasks, reflection, and open ended questions

Usability Testing - Results 2

- Still clear, intuitive, and easy to use
- Questionable layout scalability
- **Output page and data presentation could still be improved according to most users**
    - Adding a legend would help
    - Labeling images



Prototype output page

Conclusion & Future Work

Conclusion

System transforms unstructured floor plans to structured data, which can be a supplementary input to the fire risk model

Room boundary detection, counting compartments: **accuracy of 77%**

Future work

1. Annotate and train with Dutch floor plans
    a. Room function inference harder: more data might help
    b. Training data should be representative of target data: Dutch features (language, styling) on image important

2. Differentiate doors from windows

3. Web application improvements based on feedback

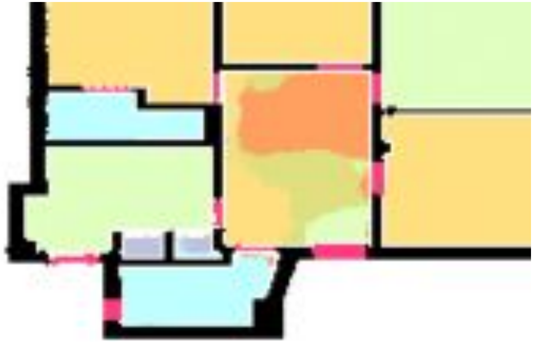
Thank you

Dank je wel

Appendix

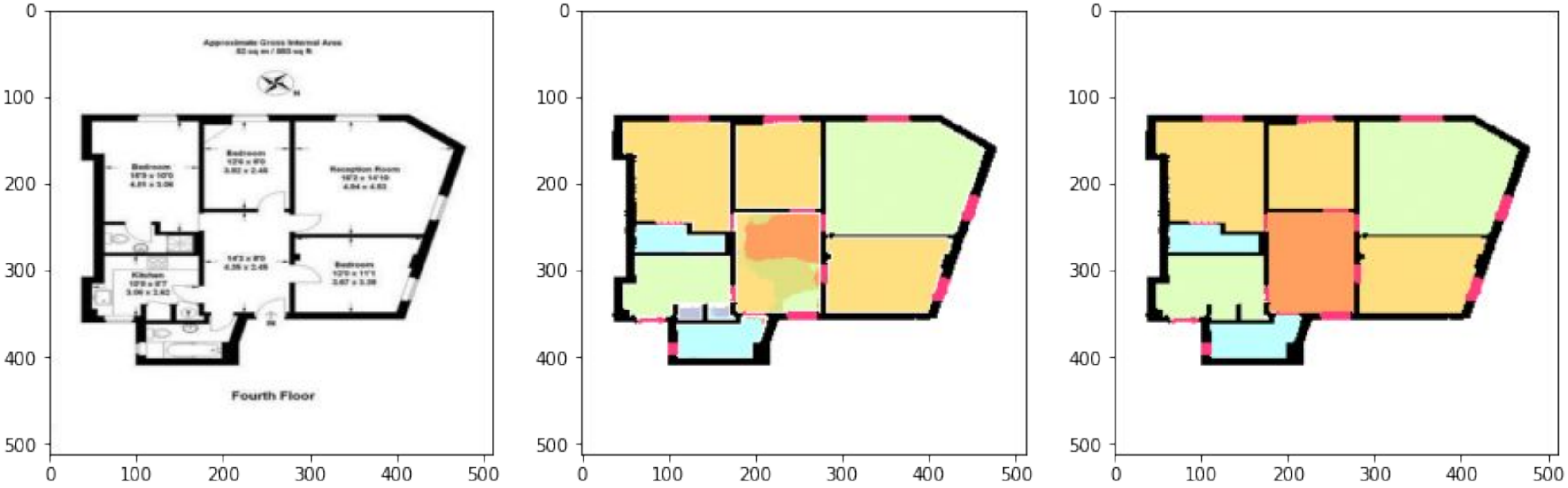
Output post-processing

- Due to the per-pixel prediction, the output is noisy
- Post-processing is performed:
    a. Locate room regions by the room-boundary pixels
    b. Count the number of pixels of each room type in the bounded region
    c. Set the overall predicted type by the largest frequency



Raw prediction output

After post-processing



Fourth Floor

Counting compartments

- Aggregate connected pixels to count different rooms
- Connected component labeling:
  - 4-connected neighborhood
  - 8-connected neighborhood

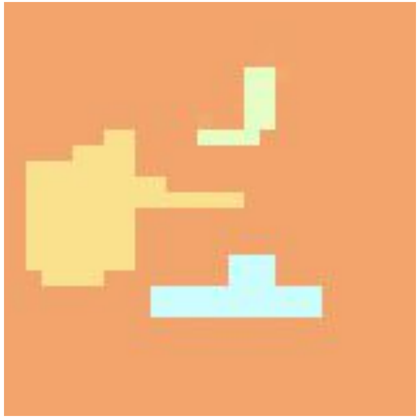
0	1	0
1	1	1
0	1	0

4-connected neighborhood

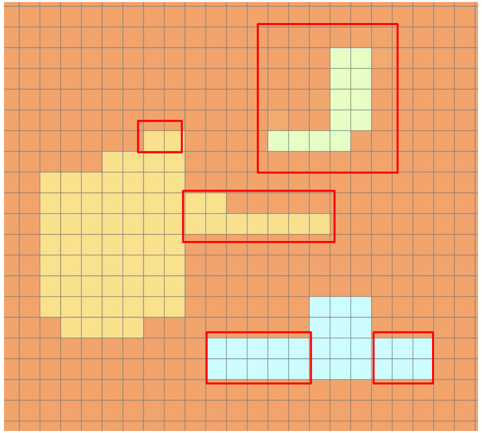
1	1	1
1	1	1
1	1	1

8-connected neighborhood

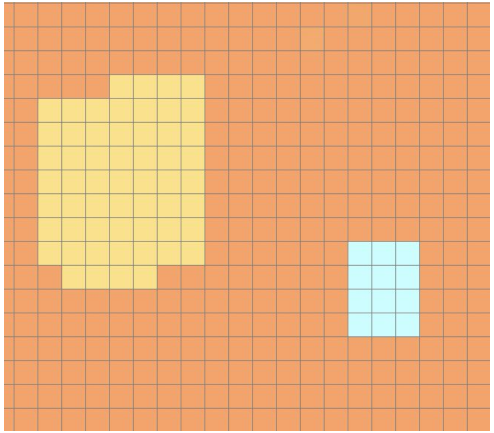
Counting compartments



Raw prediction



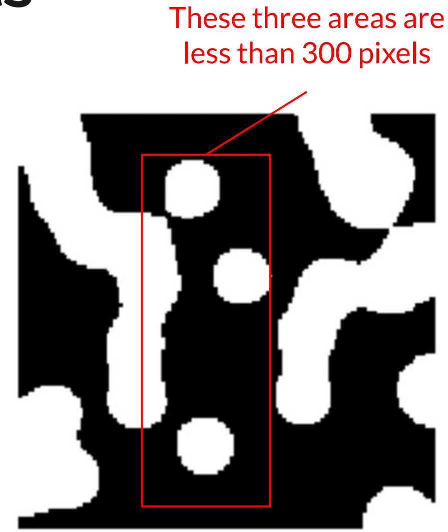
Pixels not having
8-connected neighbors



Remove isolated pixels

Counting compartments

- Only the area larger than certain pixels will be counted
- Different threshold for different types of items:
  - Door & windows: 80 pixels
  - Living room & Dining room: 120 pixels
  - Storage room/closet : 200 pixels



Original image

Removing areas under 300 pixels

Model Evaluation - Feature Elements

		Accuracy	
Element #	**Feature Element**	**Model_20_lr5-e4**	**Model_AMS3_lr5-e4**
0	background	0.9966	0.9972
1	closet	0.1656	0.0
2	bathroom/washroom	0.4469	0.0
3	livingroom/kitchen/dining room	0.4077	0.0
4	bedroom	0.4628	0.0
5	hall	0.3281	0.0
6	balcony	0.1792	0.0
9	door & window	1.0000	1.0000
10	wall	0.8046	0.7704