T8

AI In Earthquake Damage Assessment

Arda Erdal
Talha Demirbilek
Ensar Demirbilek

CHALLENGE

Community Mapping

A geographic information system (GIS) can create, manage, analyze, and map many types of data. With GIS and other mapping technologies, you can create a map of an area and layer open data over it spatially to reveal new, enriching insights.

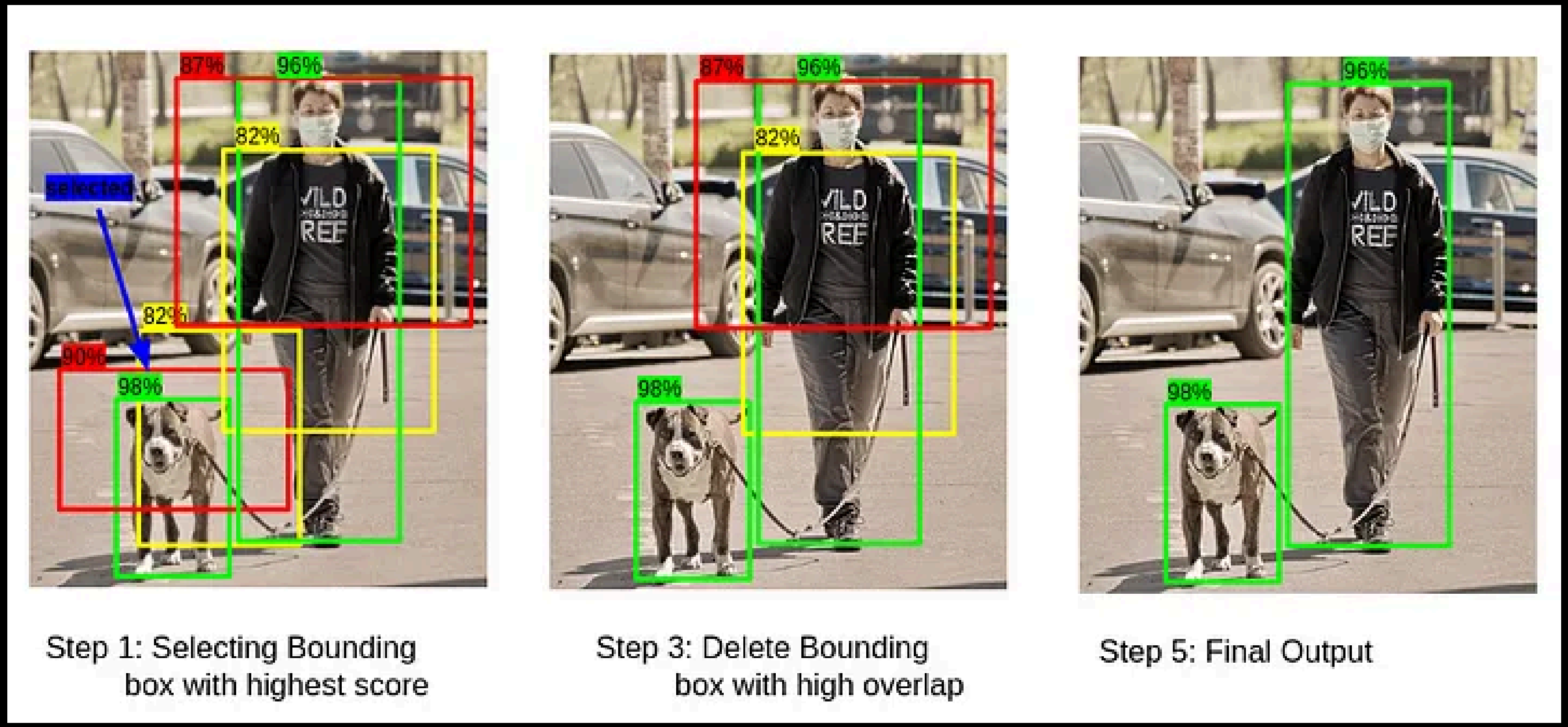
OUR GOAL

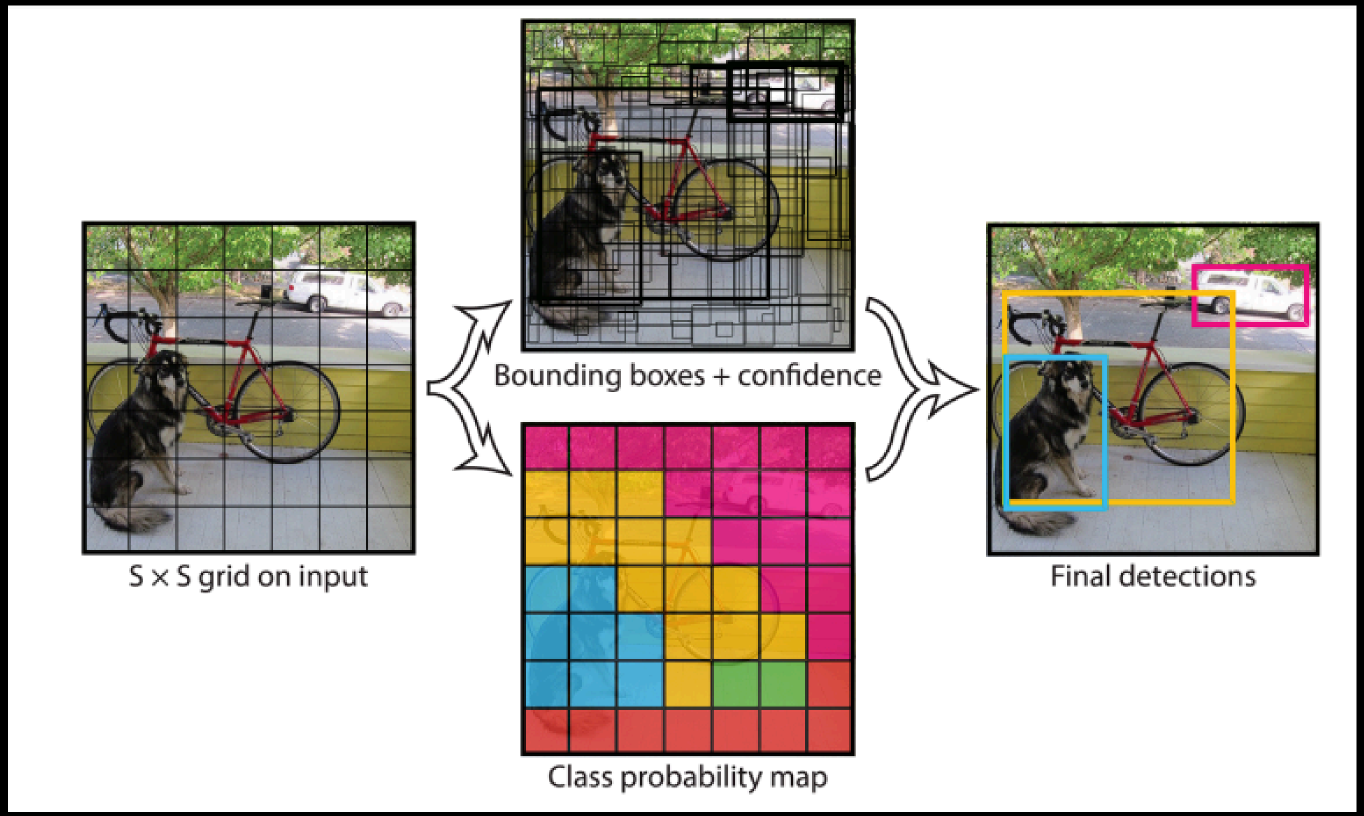
By leveraging the power of AI to create an AI image detection model that analyzes, and classifies buildings of a area that was hit by a earthquake. This will allow to know how severe the earthquake was in real time so authorities can analyze and take action accordingly. This will help save thousands of lives and will have massive economic benefits.

WHAT MAKES US DIFFERENT

While doing this project, we used the latest version of algorithms to maximize the accuracy rate and minimize possible errors.
The dataset we used in the project is a mixed data set. This data includes mostly of data that we created ourselves.
The model weights will be open source so anybody who wants use and make improvements on the model can use the previous checkpoint.

HOW DOES IT WORK





HOW DOES IT WORK

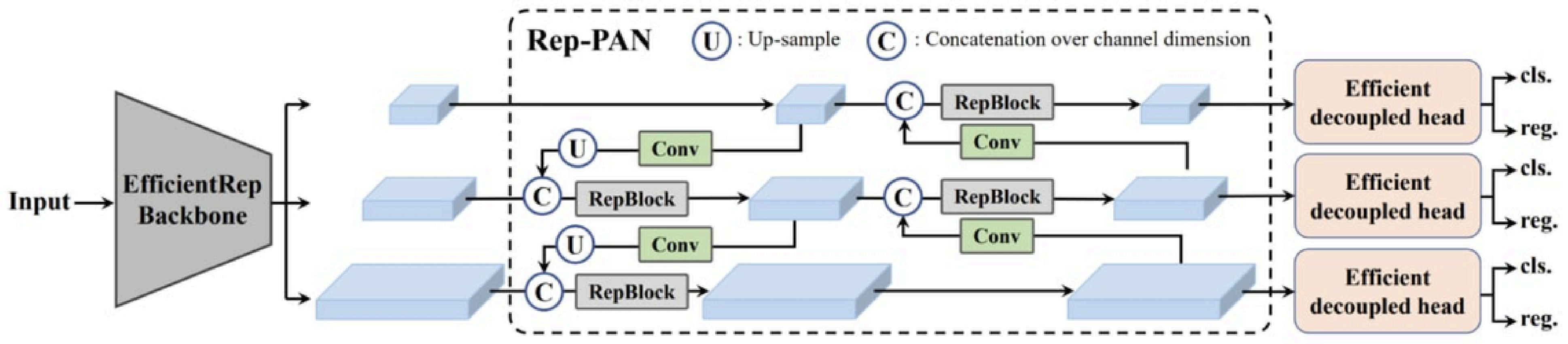
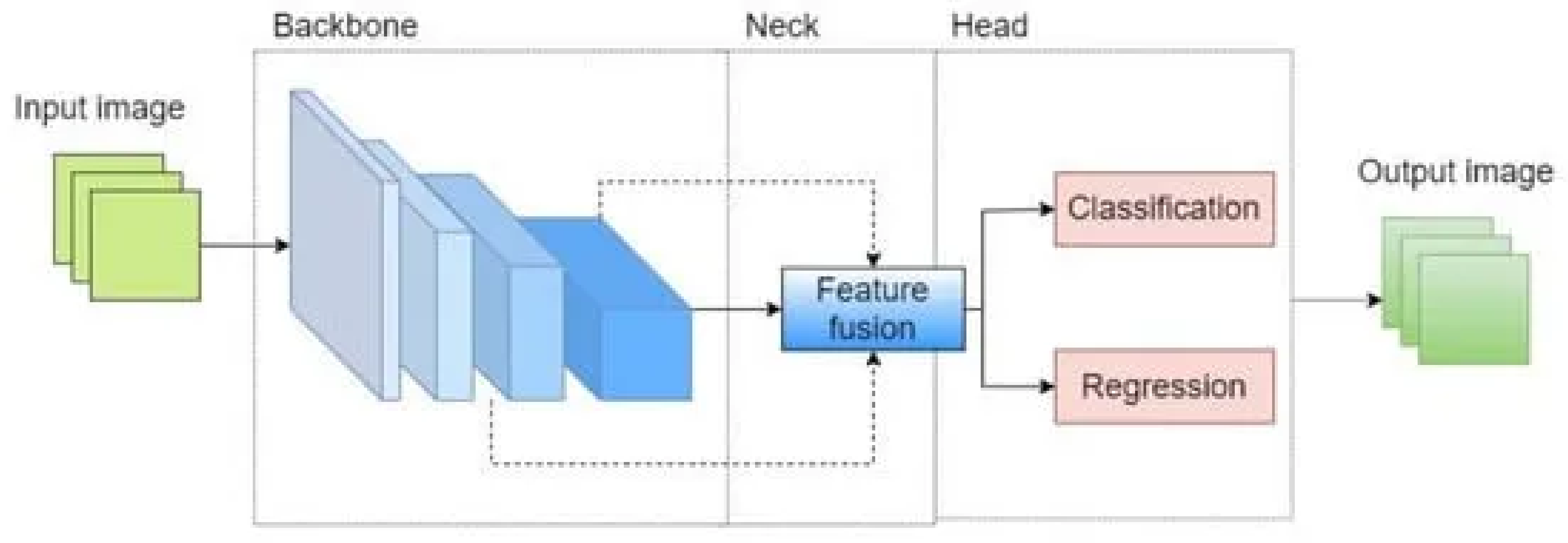
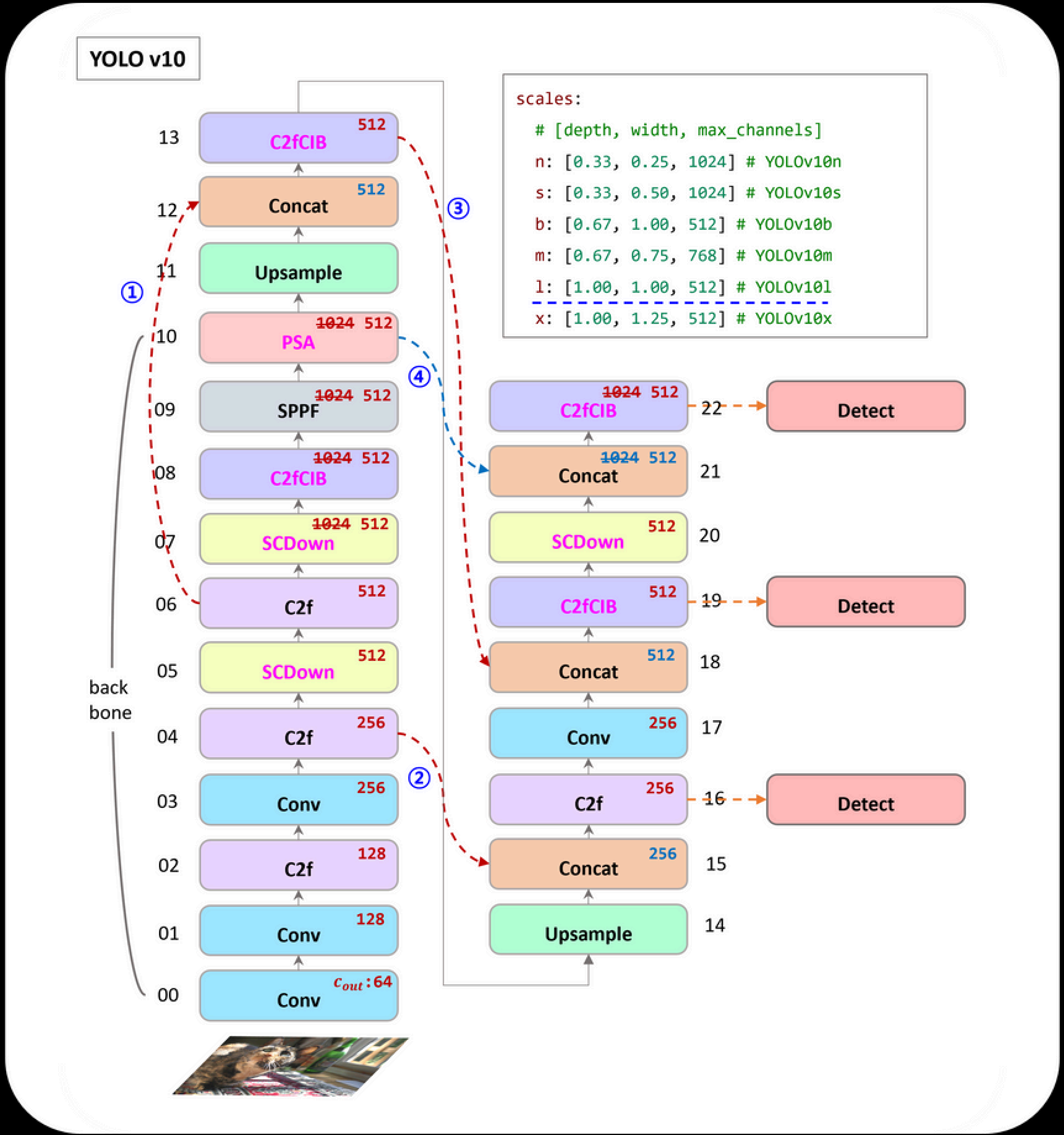


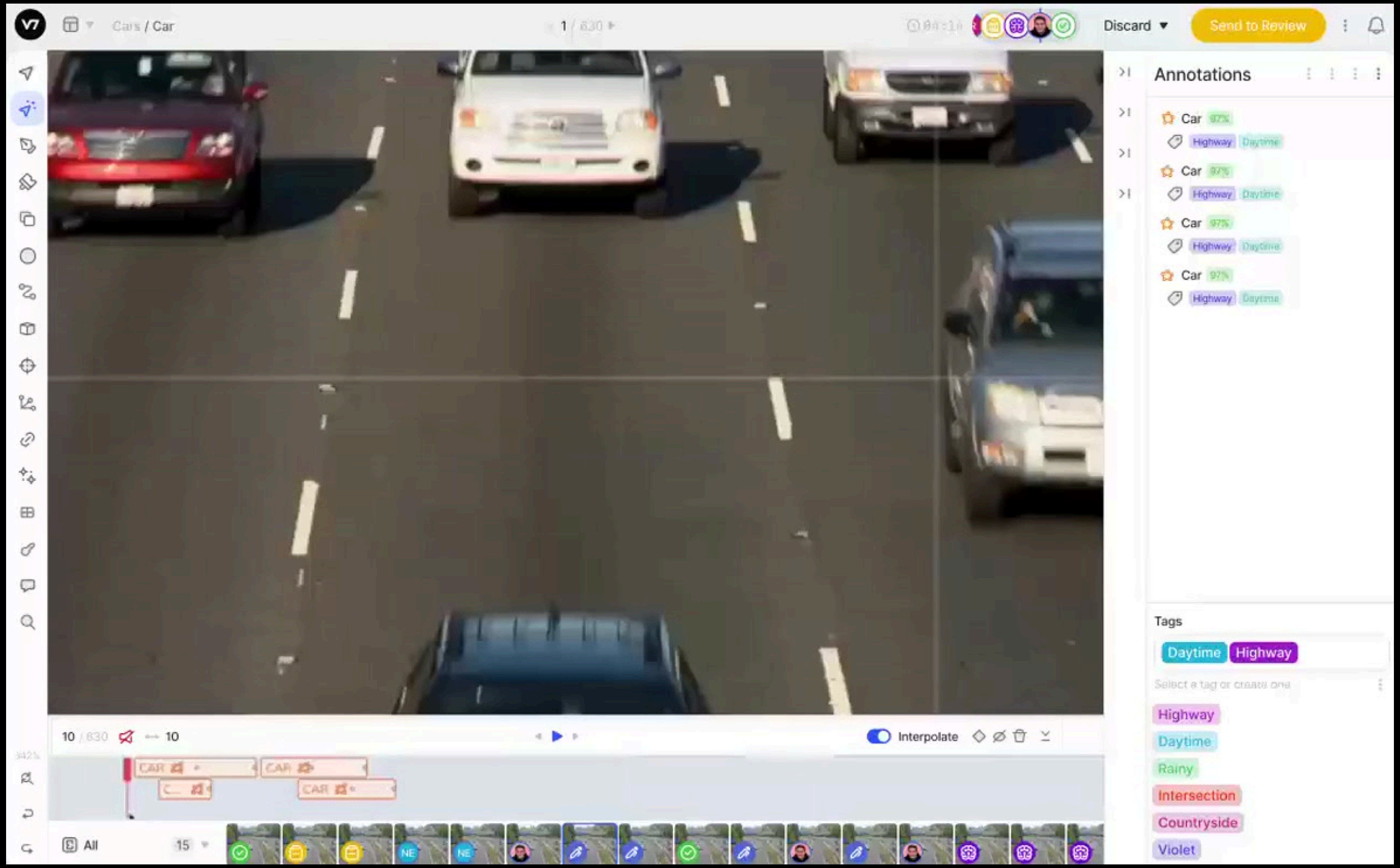
Figure 2: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep.



HOW DOES IT WORK

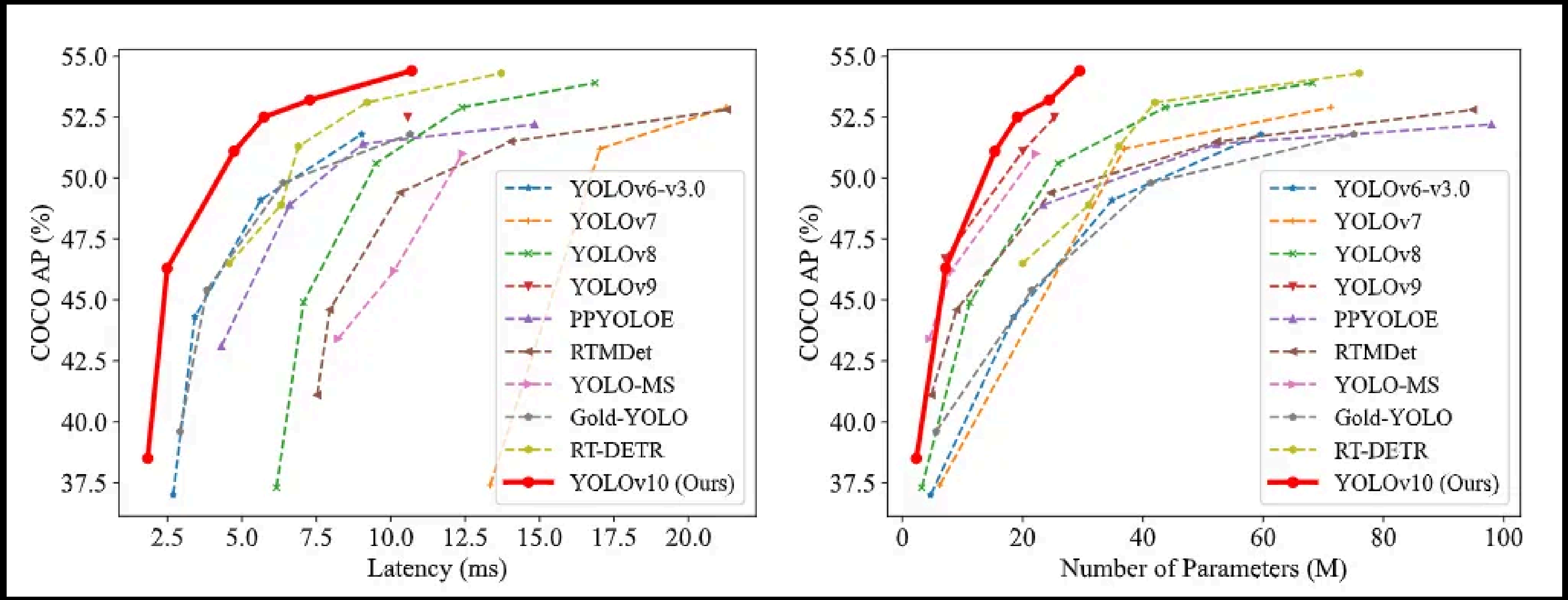


HOW DOES IT WORK



WHY WE CHOSE YOLOV10

- Speed and Efficiency
- Improved Accuracy
- Better Architecture
- Customization and Flexibility
- Real-Time Performance
- Wide Community Support
- Applications Across Domains



WHY WE CHOSE YOLOV10

YOLO v8

```
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]] # 5-P4/16
  - [-1, 6, C2f, [512, True]]
  - [-1, 1, Conv, [1024, 3, 2]] # 7-P5/32
  - [-1, 3, C2f, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]] # 9


# YOLOv8.0n head
head:
  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 6], 1, Concat, [1]] # cat backbone P4
  - [-1, 3, C2f, [512]] # 12

  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 4], 1, Concat, [1]] # cat backbone P3
  - [-1, 3, C2f, [256]] # 15 (P3/8-small)

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 12], 1, Concat, [1]] # cat head P4
  - [-1, 3, C2f, [512]] # 18 (P4/16-medium)

  - [-1, 1, Conv, [512, 3, 2]]
  - [[-1, 9], 1, Concat, [1]] # cat head P5
  - [-1, 3, C2f, [1024]] # 21 (P5/32-large)

  - [[15, 18, 21], 1, Detect, [nc]] # Detect(P3, P4, P5)
```

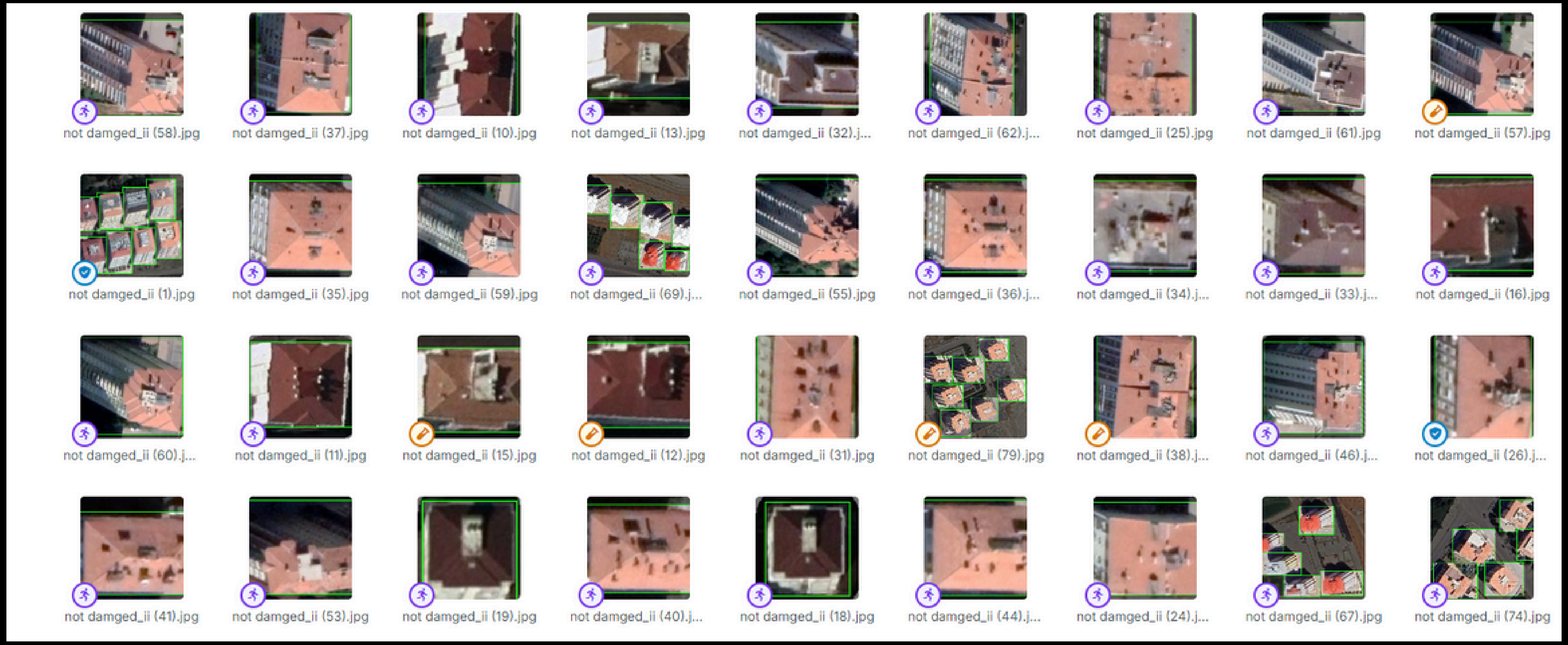
YOLO v10

```
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, SCDown, [512, 3, 2]] # 5-P4/16
  - [-1, 6, C2fCIB, [512, True]]
  - [-1, 1, SCDown, [1024, 3, 2]] # 7-P5/32
  - [-1, 3, C2fCIB, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]] # 9
  - [-1, 1, PSA, [1024]] # 10

# YOLOv8.0n head
head:
  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 6], 1, Concat, [1]] # cat backbone P4
  - [-1, 3, C2fCIB, [512, True]] # 13

  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 4], 1, Concat, [1]] # cat backbone P3
  - [-1, 3, C2f, [256]] # 16 (P3/8-small)

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 13], 1, Concat, [1]] # cat head P4
  - [-1, 3, C2fCIB, [512, True]] # 19 (P4/16-medium)

  - [-1, 1, SCDown, [512, 3, 2]]
  - [[-1, 10], 1, Concat, [1]] # cat head P5
  - [-1, 3, C2fCIB, [1024, True]] # 22 (P5/32-large)

  - [[16, 19, 22], 1, v10Detect, [nc]] # Detect(P3, P4, P5)
```

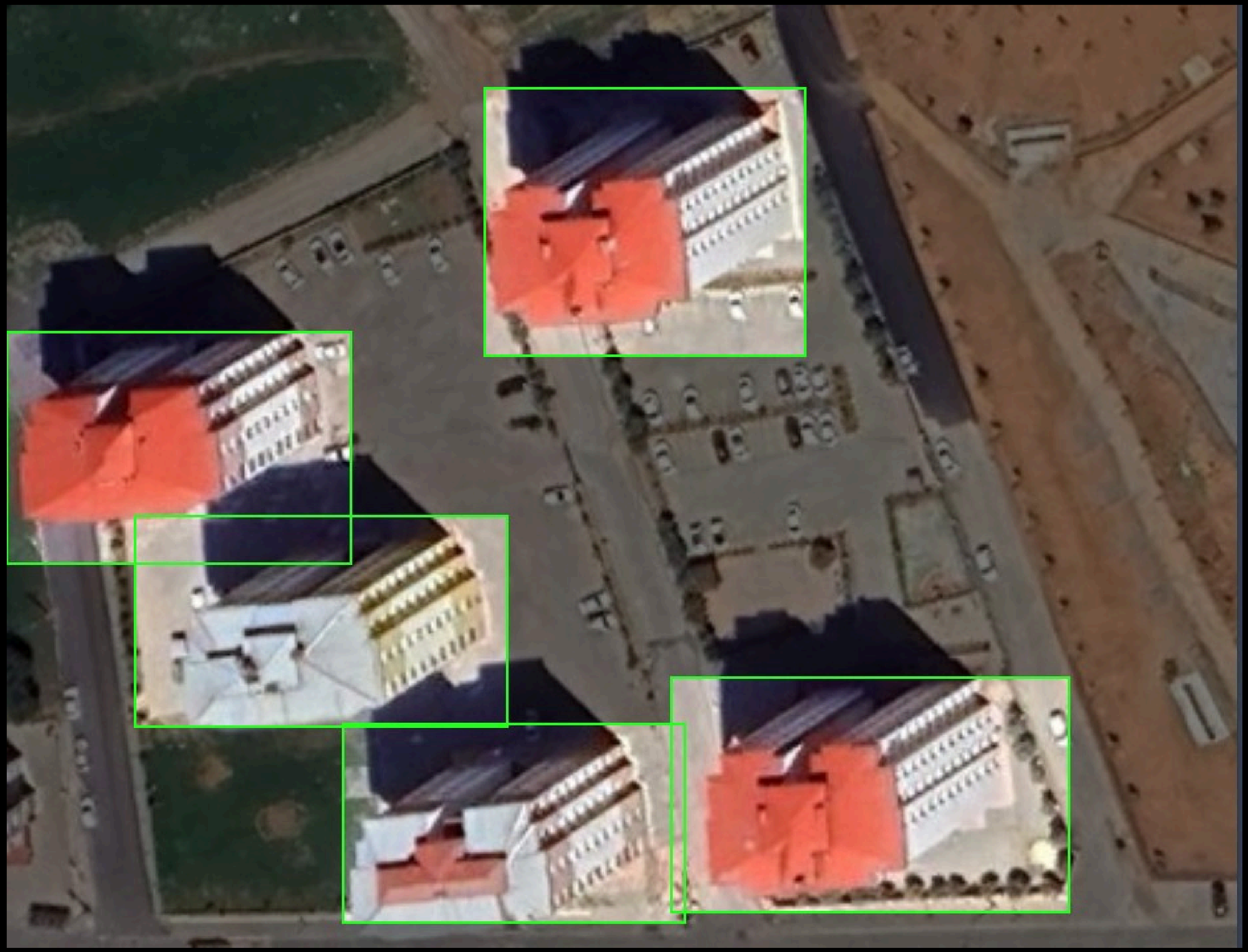
DATASET

The model was trained on a dataset consisting of 842 images.

- 774 training images
- 50 validation images
- 18 testing images



LABELING



DATA PREPROCESSING

Preprocessing data helps to improve the accuracy, reduce the time and resources required to train the model, prevent overfitting, and improve the interpretability of the model

Used preprocessing methods:

- Auto-Orient
- Resize (640x640)

DATA AUGMENTATION

Data augmentation is a technique used to artificially increase the size and diversity of a training dataset and helps prevent overfitting when you're training Machine Learning models.

Used data augmentation methods:

- 90 degree rotation
- Rotation between -15 and +15 degrees
- Exposure between -15% and +15%
- Noise up to 1.01% of pixels

TRAINING

The model was trained in GoogleColab with T4 GPU

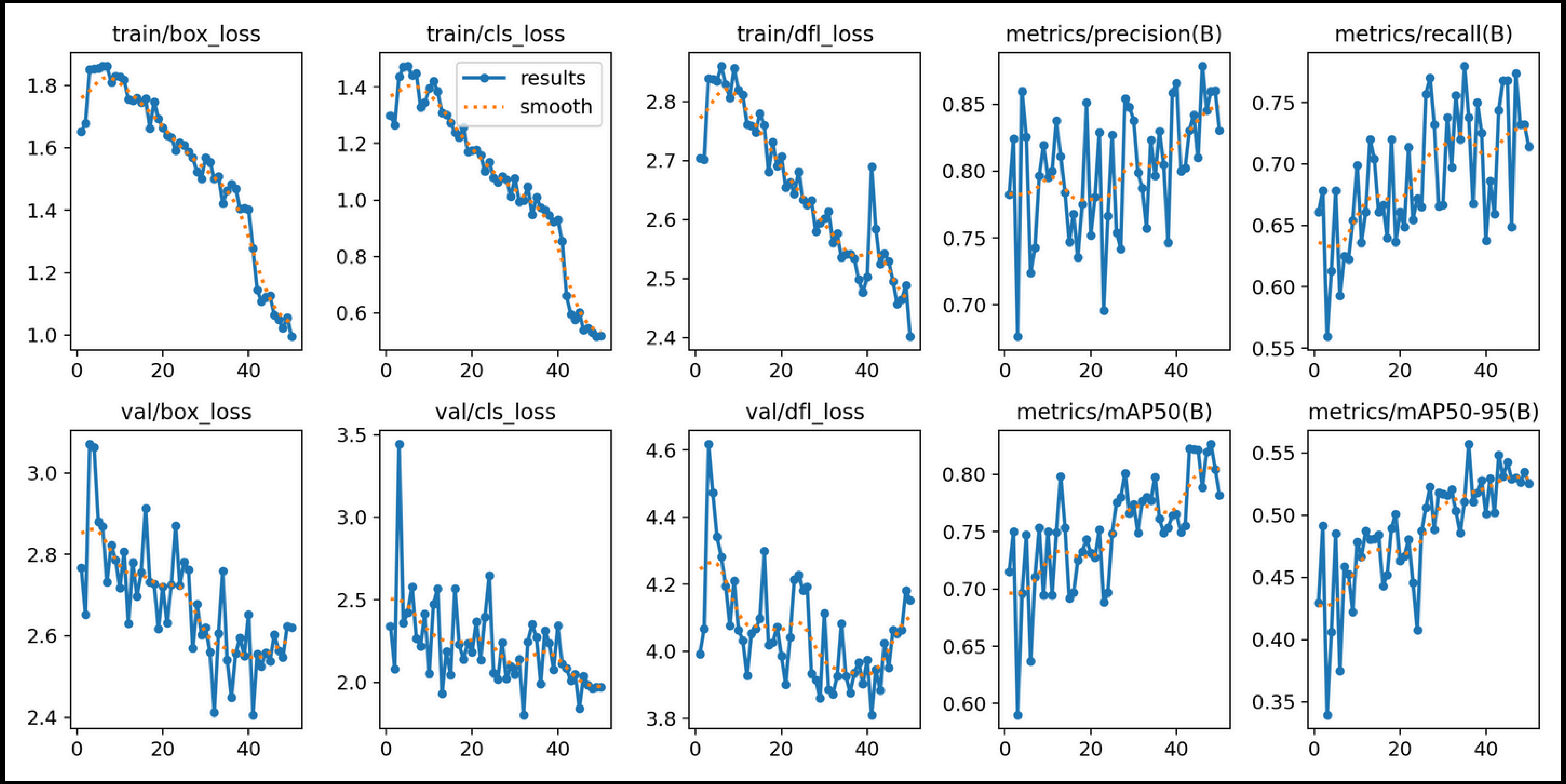
- 50 epochs
- Batch size of 16
- Image size: 640
- Patience: 100 (to stop overfitting)
- Optimizer: AdamW
- Learning rate: 0.001667
- Momentum: 0.9

```
model.train(data="/content/Nasa-SpaceApps-2/data.yaml", epochs=50, imgsz=640, batch=16)
```

TRAINING

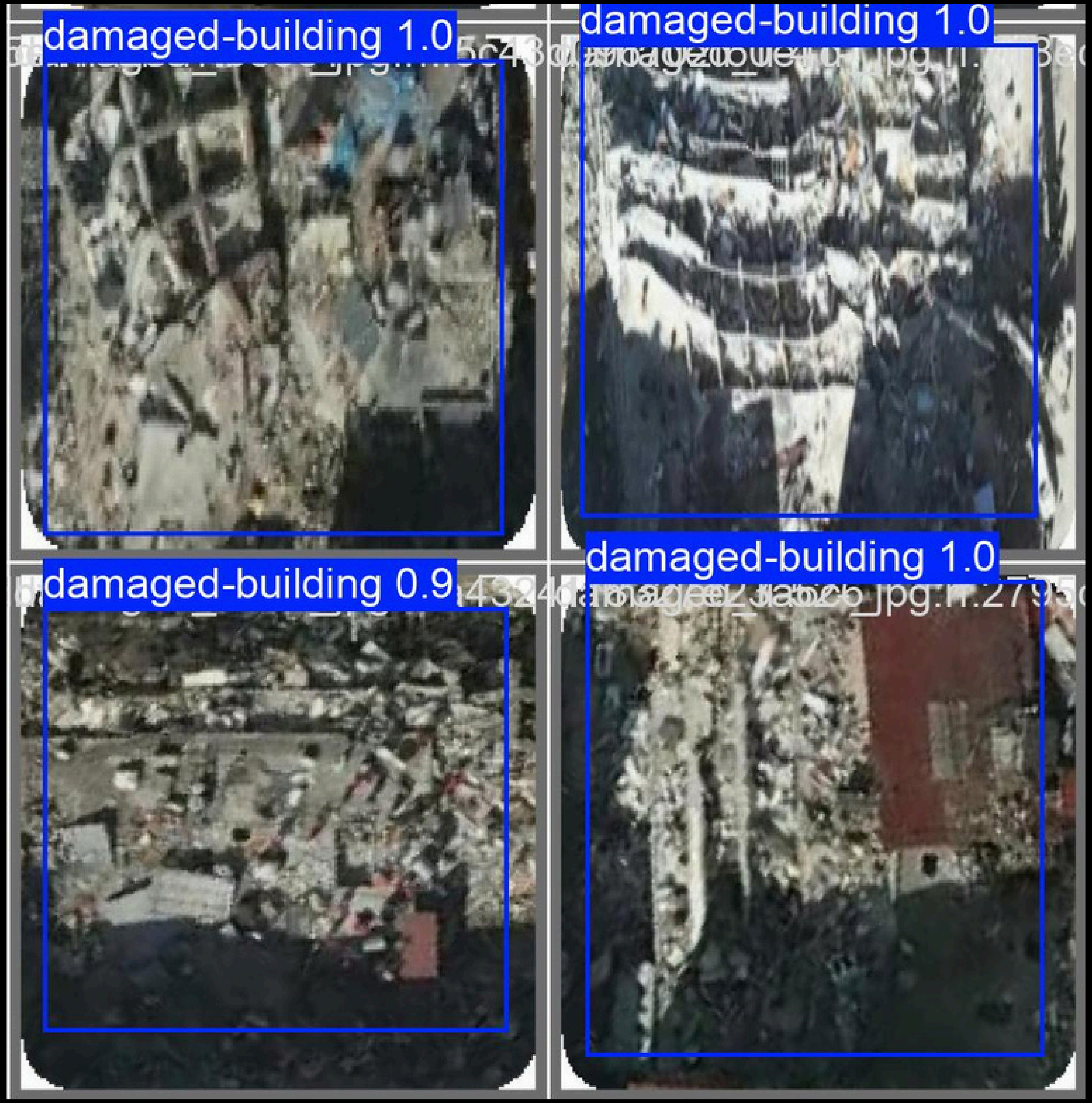
```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 1/50      1.66G      2.055      1.985      3.136         20        640: 100%|██████████| 97/97 [00:24<00:00,  4.02it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  5.75it/s]

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 2/50       1.6G      2.086      2.033      3.173         20        640: 100%|██████████| 97/97 [00:23<00:00,  4.07it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  8.43it/s]

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 3/50       1.6G      2.259      2.208      3.296         24        640: 100%|██████████| 97/97 [00:22<00:00,  4.32it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  7.49it/s]

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 4/50       1.6G      2.153      2.128      3.202         20        640: 100%|██████████| 97/97 [00:22<00:00,  4.38it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  5.94it/s]

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 5/50       1.6G      2.128      2.063      3.163         19        640: 100%|██████████| 97/97 [00:24<00:00,  3.98it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:00<00:00,  5.24it/s]
```

TRAINING


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

VALIDATION


not-damaged-building 0.8
not-damaged-building 0.9
not-damaged-building 0.9
not-damaged-building 1.0
not-damaged-building 0.7
not-damaged-building 0.4
not-damaged-building 0.8
not-damaged-building 0.3


damaged-building 1.0
damaged-building 1.0
damaged-building 0.9
damaged-building 1.0

TEST SAMPLE


not-damaged-building 0.9
not-damaged-building 0.9
not-damaged-building 0.4


damaged-building 0.9

T8

THANK YOU !

NASA

NASA
INTERNATIONAL
SPACE APPS
CHALLENGE