CPCS432-AI II

Team members

- Hanoof Mohammed –1909632
- Arwa Kulib – 2019574
- Ghaida Aleryani- 1906952
- Arwa Alamoudi – 2005539
- Rahaf Alghamdi – 2006609
- Najd Alotaibi - 2006156
- Roaa Alzahrani - 2005863
- Rahaf Jahlan - 19058545

Project Idea

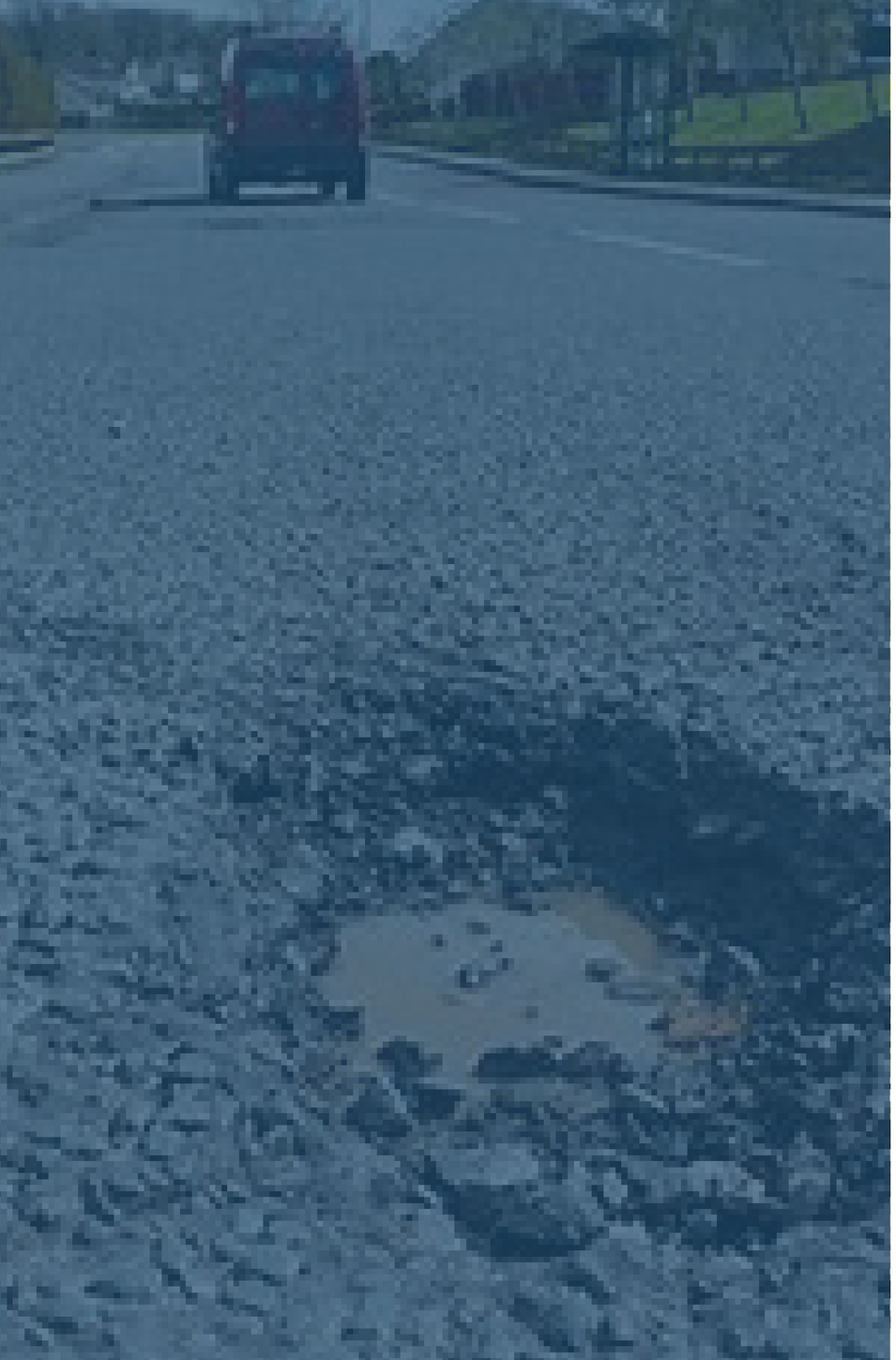
Use computer vision techniques, particularly advanced object detection models, to address the widespread problem of road potholes. By revolutionizing the process of pothole detection, we aim to enhance road safety and make it more effective and efficient compared to traditional methods such as manual inspection and visual surveys.

Potholes account for nearly
1.4% road accident deaths

CV model and methodology

We chose these four models:

1. SSD
2. YOLO
3. RetinaNet
4. EfficientDet



Dataset

- Acquired from Roboflow
- Consist of 2742 images
- Consist of one class (Potholes)
- Divided into three sets: the training set, the validation set, and the test set.
- Resizing preprocessing technique implemented in different format code for each model.

SSD

The Single Shot Multibox Detector (SSD) is a popular real-time object detection model in computer vision. Comprising a backbone (often VGG-16) and an SSD head, it extracts features and predicts bounding boxes and class scores. SSD efficiently detects objects of different sizes by dividing the image into grids, assigning scores, and reducing redundancy through overlapping bounding box filtering.

Experiments &

- Utilized pycocotools and torchmetrics Python libraries in the experiment.
- Downloaded the dataset in Pascal VOC format.
- Separated XML annotations from images into two folders.
- Executed pre-processing steps (resizing) using the code: A.Resize(height=416,width=416).
- Employed a batch size of 4 for model training.
- Generated bounding boxes and applied data augmentation.
- Trained the ssd300_vgg16 model for 15 epochs with default weights.
- Evaluated model results using the Mean Average Precision function.

Results

the model achieved:

Mean Average Precision: 0.22

Mean Average Precision @ 0.5: 0.45

Mean Average Precision @ 0.75: 0.20

YOLO

YOLO (You Only Look Once) is a fast and accurate object detection algorithm that simplifies object detection by treating it as a regression problem .

Instead of sliding windows or region-based methods, YOLO divides the image into a grid, where each cell predicts bounding boxes, class probabilities, and objectness scores.

YOLO can struggle with small object detection ,which can be a challenge for detect small-sized potholes.

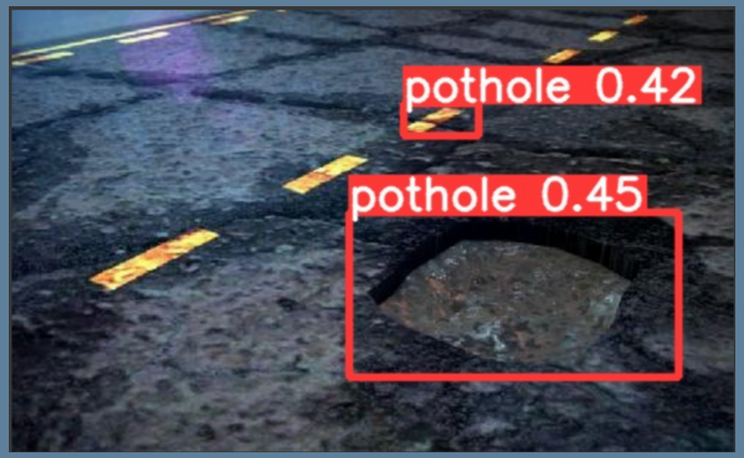
YOLO has evolved through various versions (YOLOv2 to YOLOv8),

Experiments &

- **Transfer Learning Approach:** fine-tuning the model on a custom dataset.
- **Model Architecture**: multiple layers, including the anchor layer responsible for the detection mechanism.
- **Hyperparameters:** Utilizes a learning rate of 0.01, stochastic gradient descent (SGD) optimizer with momentum 0.9, a batch size of 16, epoch 15 ,and an image size of 416 for efficient training.
- **preprocessing** : !python train.py --img 416 which is resized the image to 416
- **Output:** Predicts bounding boxes and class probabilities directly.

Results

- Achieved results comparable to research papers after training for 15 epochs.
- Precision (P): 56.8%
- Recall (R): 41%
- mAP: 43%



RetinaNet

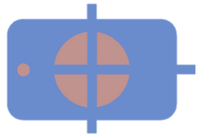
RetinaNet is one of the best one-stage object detection models that has proven to work well with dense and small scale objects. RetinaNet's architecture is based on the Feature Pyramid Network (FPN), which enables the model to efficiently detect objects of various sizes. This pyramid structure facilitates the detection of objects across a range of scales, enhancing the model's ability to handle small and large objects simultaneously.

Experiments &

- Trained from scratch using PyTorch on custom dataset
- Initialized with ResNet-50 weights
- Resized images to 416x416
- Batch size set to 2 for regularizing effect
- Initially trained for 10 epochs, then 5 more after validation results
- Performed testing and printed images with bounding boxes, most of which were accurate

Results

Testing	
Pothole AP	55%
Precision	24%
Recall	62%



RetinaNet Preprocessing

```
57          dataset_train = CSVDataset(train_file=parser.csv_train, class_list=parser.csv_classes,
58                                     transform=transforms.Compose([Resizer()]))

339     class Resizer(object):
340         """Convert ndarrays in sample to Tensors."""
341
342         def __call__(self, sample, min_side=416, max_side=416):
343             image, annots = sample['img'], sample['annot']
344
345             rows, cols, cns = image.shape
346
347             smallest_side = min(rows, cols)
```

EfficientDet

EfficientDet is the object detection version of EfficientNet, building on the success EfficientNet has seen in image classification tasks. EfficientNets come from a family of models that achieve a high performance on benchmark tasks while controlling for a number of efficiency parameters EfficientDet. EfficientDet achieves the best performance in the fewest training epochs among object detection model architectures.As we know, each model may face some challenges.

Experiments &

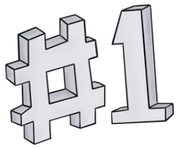
- trained by using a pytorch implementation on a custom dataset that has been token from RoboFlow.
- used image detection library from Tessellate-Imaging for object detection
- the base version of EfficientDet-d0
- the EfficientNet base backbone,

Results

Testing	
Precision	100%
Recall	0.086%

Validation	
Precision	100%
Recall	0.01%

Comparison



Model	Precision
EfficientDet	100%
YOLO	56%
RetinaNet	24%
SSD	22%

After getting all the results of the trained models,EfficientDet model gave the highest precision in detecting road potholes which is 100%. Compared to the related research, Yolo model achieved higher precision than yolo model of the related research that had precision of 50.96% .

Thank You For Your Time!