DEEP LEARNING PROJECT

CRACK DETECTION USING DEEP LEARNING ALGORITHMS

Kasyap Velampalli – 21011101141
S.M. Noor – 21011101114

AIDS – ‘B’

SHIV NADAR
UNIVERSITY
CHENNAI

INTRODUCTION

- This paper introduces a supervised deep learning approach for automated pavement crack detection, using a convolutional neural network (CNN) to analyze raw images of various pavement conditions. It creates a large training database from small patches of crack images and addresses imbalanced data issues.
- The method is evaluated on two public databases and compared to five existing methods, demonstrating its effectiveness in crack detection
- Dataset link: crack_segmentation_dataset.zip - Google Drive

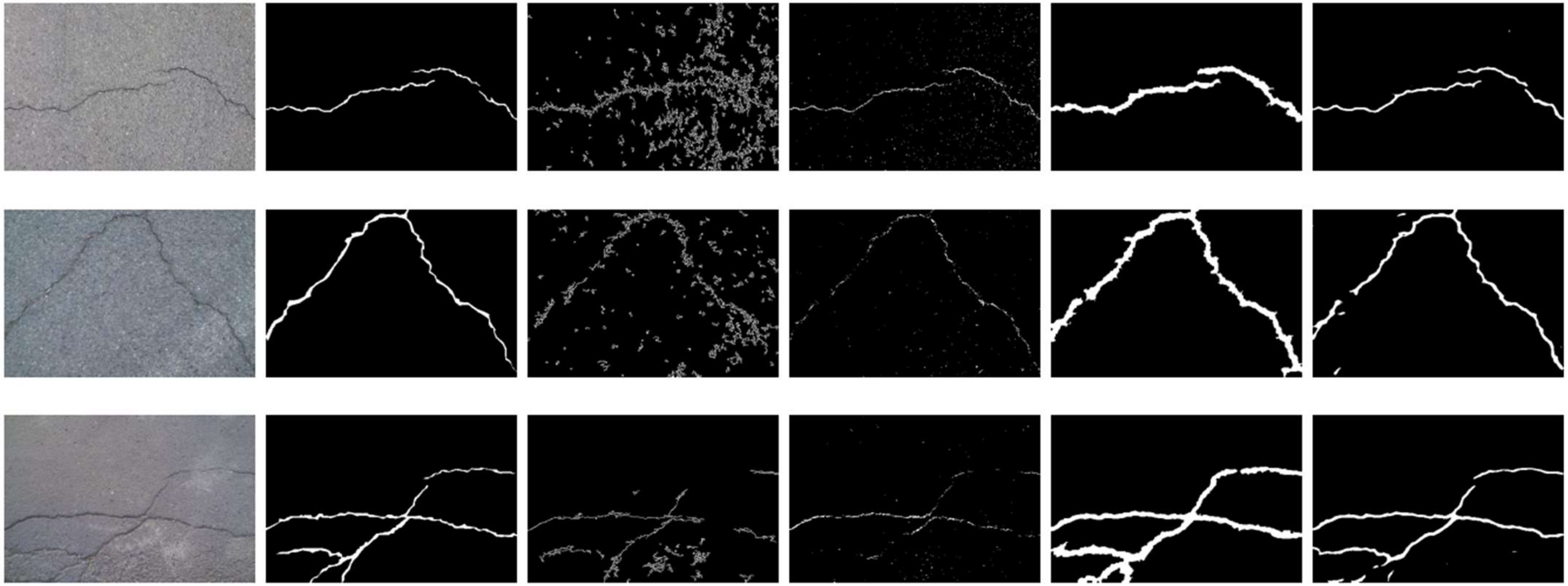
OBJECTIVE

- To present a supervised deep learning methodology utilizing a convolutional neural network (CNN) for automated pavement crack detection, aimed at effectively addressing the challenge of analyzing varied pavement conditions and imbalanced data. The study intends to demonstrate the effectiveness of this method by comparing it to existing approaches using two public databases.

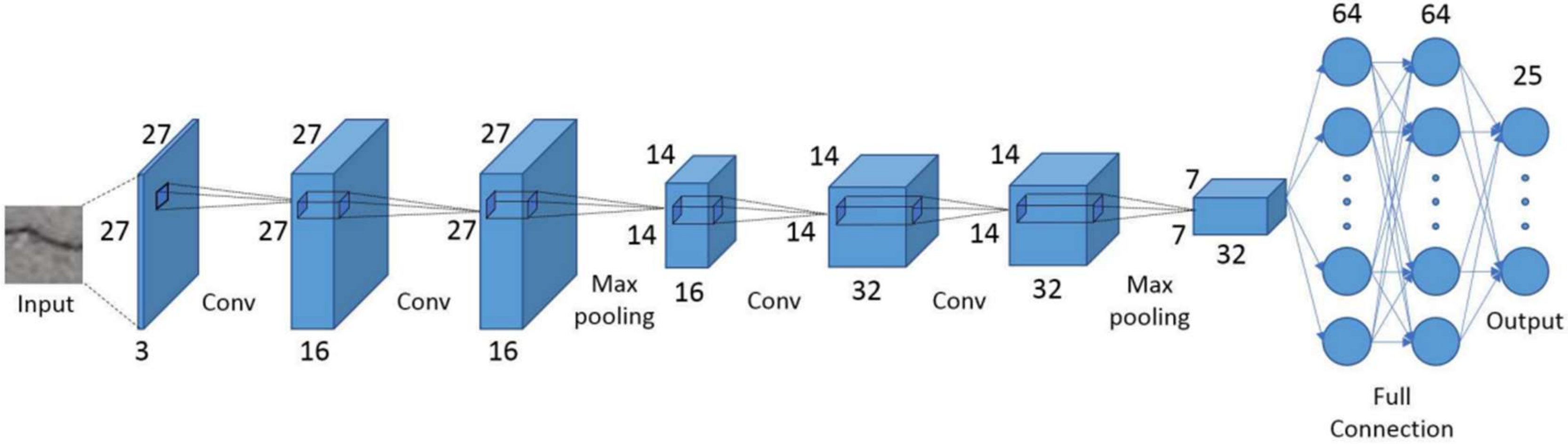
MOTIVATION

- Enhances road safety by identifying and repairing cracks before they lead to accidents.
- Extends pavement lifespan and reduces maintenance costs.
- Ensures efficient transportation infrastructure by preventing road damage.
- Contributes to overall urban aesthetics and quality of life.
- Supports eco-friendly practices by minimizing resource wastage during repair work.

ORIGINAL IMAGE, GROUND TRUTH, CANNY, LOCAL THRESHOLDING, CRACKFOREST, THE PROPOSED METHOD



BASE ARCHITECTURE – STANDARD MODIFICATIONS :



BASE ARCHITECTURE – DESCRIPTION

- CNN Architecture
- The proposed neural network architecture for crack detection consists of four convolutional layers, two max-pooling layers, and three fully-connected layers. It takes input patches with three channels (CFD) or one channel (AigleRN) and employs 3x3 kernels with a stride of 1 for convolution, preserving spatial resolution with zero-padding. Max pooling is done with a 2x2 window and stride 2. The network's output predicts the structure within the input patch, represented as s x s, and is treated as a multi-label problem. The s x s window is flattened into s^2 neurons connected to the second last layer. The network uses sigmoid activation for multi-label classification, except for output units, which use Rectified Linear Units (ReLU) for nonlinearity.

CONTD.

- In training, cross entropy is used as the loss function, considering that multiple positive outputs can exist simultaneously. Weight decay with L2 penalty is applied to prevent overfitting. Dropout is employed in the first two fully-connected layers with a 0.5 dropout ratio. The weights are initialized using the Xavier method, and the Adam optimizer with a learning rate of 0.001 is used. Batch size is set to 256, and a specific number of iterations are conducted for each dataset. During testing, each pixel generates an input patch, leading to overlapping output windows. The results from all pixels are summed and normalized to produce a probability map for the entire image.

PRIMARY SOFTWARE REQUIREMENTS

- Anaconda - Jupyter Notebook
- NumPy
- Pandas
- Keras
- Scikit-learn
- TensorFlow

REFINEMENTS

- Explore transfer learning by initializing the network with pre-trained weights from a relevant architecture to potentially improve performance.
- Experiment with different activation functions and dropout ratios to fine-tune the network for better generalization and prevention of overfitting.
- Tuning the model

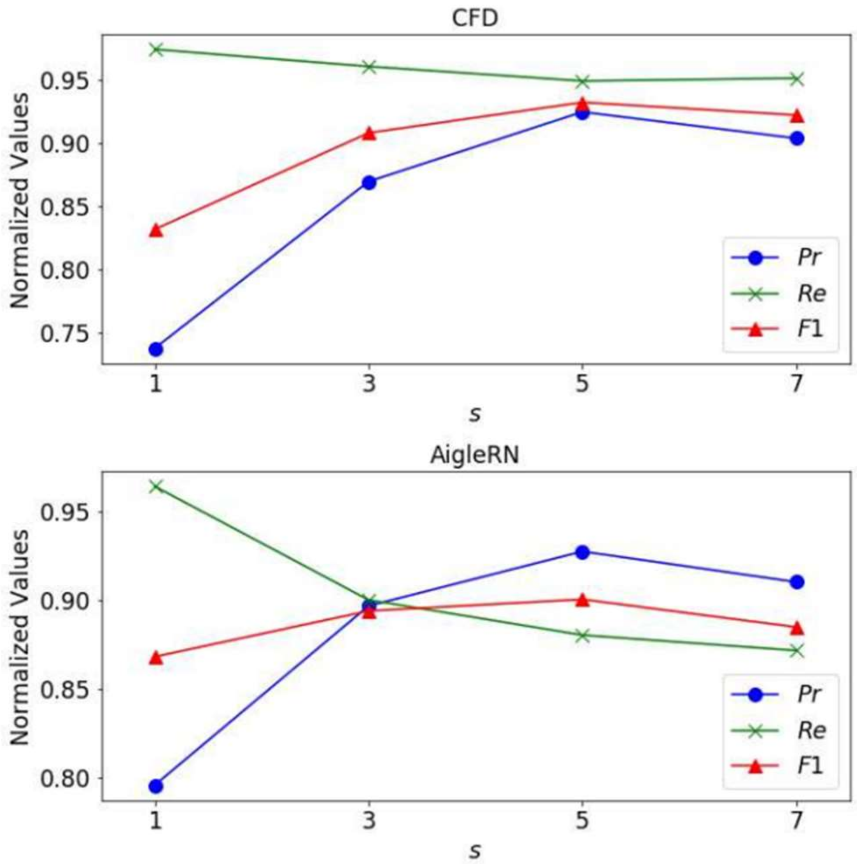
STUDIES

- Dataset description:

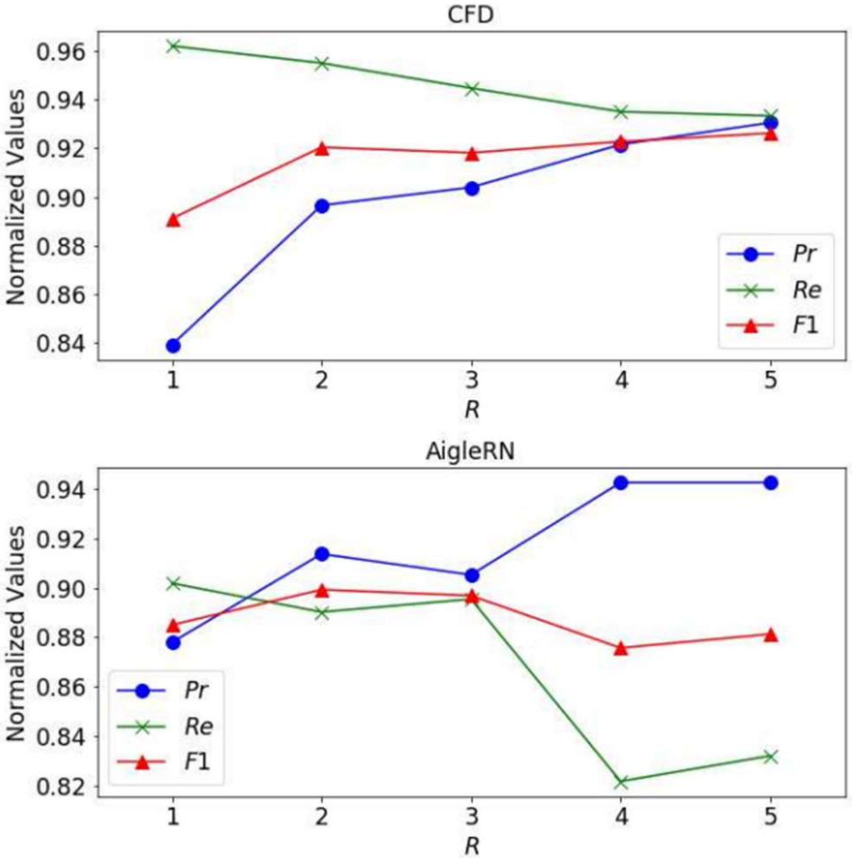
The dataset used in the project is the largest crack segmentation dataset to date, comprising approximately 11,200 images consolidated from 12 different crack segmentation datasets. Each image's name prefix corresponds to its source dataset, and there are images without crack pixels, identified by the "noncrack*" naming pattern. All images are resized to a uniform size of (448, 448). The dataset is organized into folders: "images" and "masks" for all images, and "train" and "test" for stratified training and testing data, ensuring a balanced representation of datasets in each split.

- Link: crack_segmentation_dataset.zip - Google Drive

METRIC VARIATIONS WITH DIFFERENT SIZES OF OUTPUT STRUCTURE. TESTING ON CFD AND AIGLERN RESPECTIVELY



METRIC VARIATIONS WITH DIFFERENT RATIOS OF POSITIVE TO NEGATIVE SAMPLES. TESTING ON CFD AND AIGLERN RESPECTIVELY.



RESULTS

CROSS DATABASE TESTING

Train \ Test	AigleRN	CFD
AigleRN	$Pr = 0.9178$ $Re = 0.8812$ $F1 = 0.8954$	$Pr = 0.9651$ $Re = 0.3832$ $F1 = 0.4812$
CFD	$Pr = 0.6488$ $Re = 0.8819$ $F1 = 0.7182$	$Pr = 0.9119$ $Re = 0.9481$ $F1 = 0.9244$
Hybrid	$Pr = 0.9042$ $Re = 0.8448$ $F1 = 0.8677$	$Pr = 0.9018$ $Re = 0.9494$ $F1 = 0.9210$

Thank you