EdgeAI for Defect Detection using Transfer Learning Techniques in the Context of Smart Manufacturing

Atah Nuh Mih, Hung Cao

Analytics Everywhere Lab, University of New Brunswick

UNB EST. 1785 | Analytics Everywhere Lab

Eigen Innovations NBIF

MOTIVATION

❖EdgeAI has become a crucial factor for the automation of processes in Smart Manufacturing. Production processes such as defect detection in quality assurance continue to benefit from the recent advancements in machine learning techniques on edge devices.

❖Machine learning models trained to identify imperfections on one object suffer from performance loss when applied to a different object. We therefore use transfer learning techniques to solve this challenge.

❖We propose a pre-processing technique that aims to minimize less important features of the object and build a robust model on an edge device to identify the defects.

❖We train the model on a dataset of a source object and use the model to detect defects on different target objects.

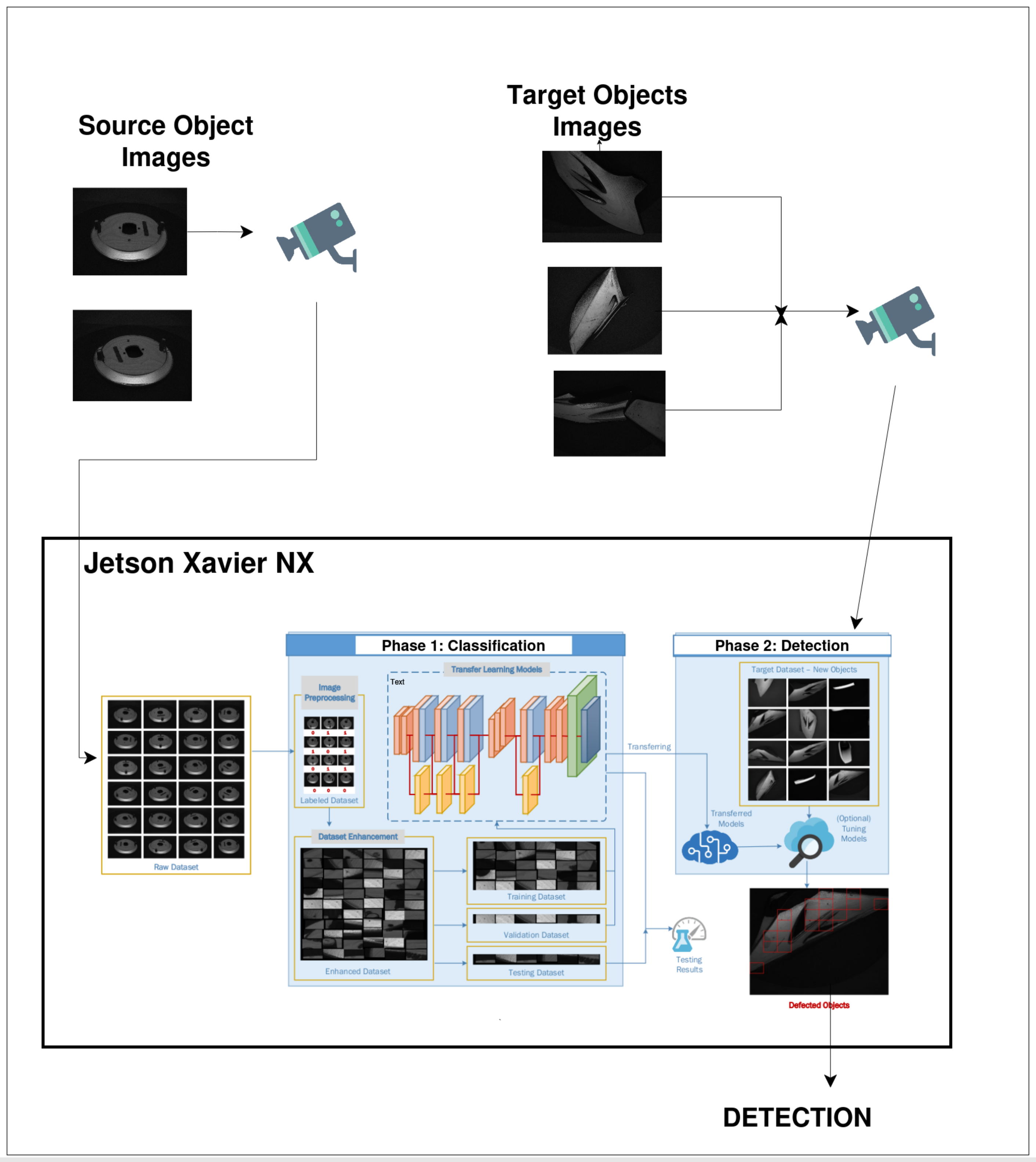
PROPOSED METHOD

Phase 1: Classification

- Image Pre-processing crops the image and maps the defect labels on the image
- Dataset Enhancement splits the images into tiles and labels as defective or non-defective. A new dataset is generated from the labelled image tiles.
- Classifier built over the Xception architecture and trained on the new dataset using transfer learning.

Phase 2: Detection

- Target image split into tiles as in the preprocessing
- Each tile passed to the classifier for inference
- Defective tiles identified and bounded



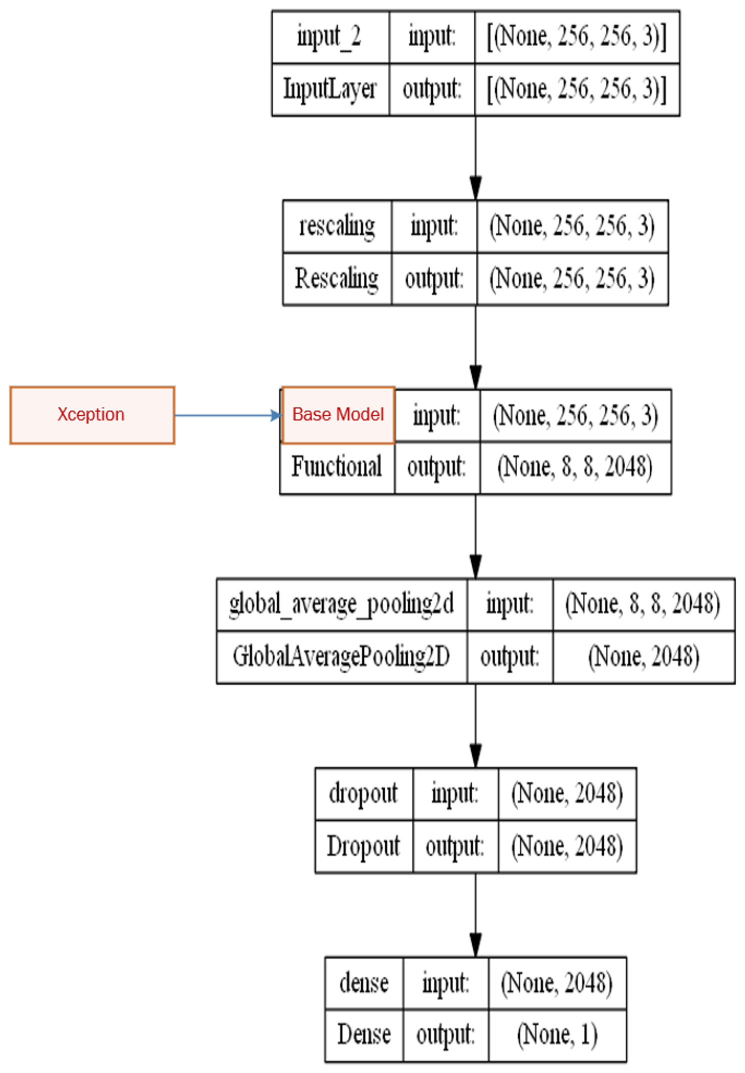
IMPLEMENTATION DETAILS

System Specifications

- 6-core 64-bit CPU, NVIDIA Carmel ARMv8.2
- 384-core NVIDIA Volta GPU
- 8 GB 128-bit LPDDR4x 59.7GB/s
- 21 TOPS
- Ubuntu 20.04 LTS

Libraries

- Python v3.10
- Tensorflow v2.11
- Nvidia Jetpack v5.02
- CUDA

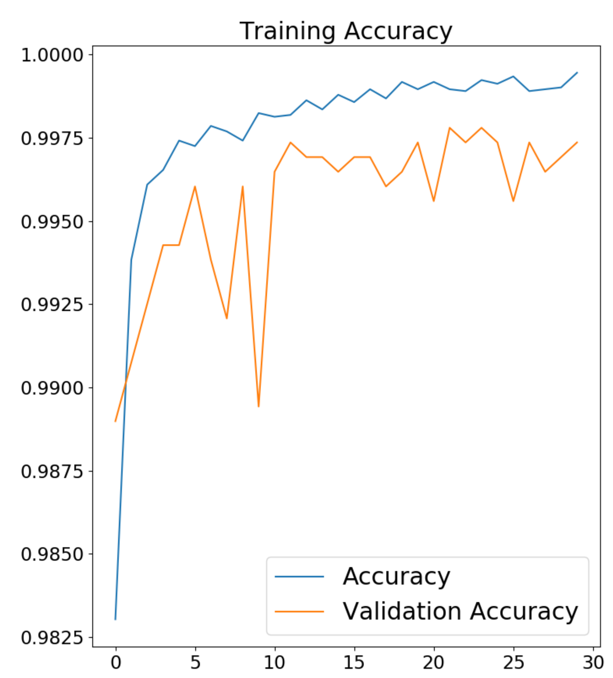


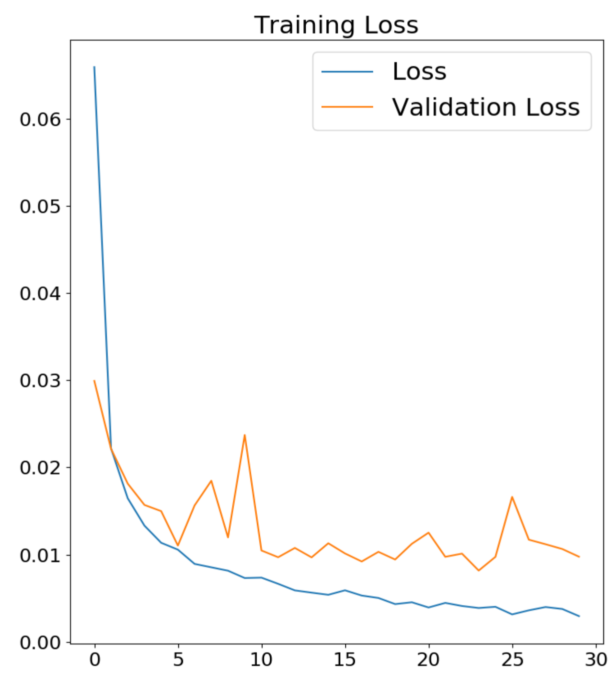
- Classifier built upon the Xception architecture
- Input layer takes in images of shape (256, 256, 3)
- deep convolutional neural network architecture with depth-wise separable convolutions
- consist of a depth-wise convolution (each filter operates on single input channel) followed by a pointwise convolution (1x1 filters are applied to the output of the depth-wise convolution).
- 36 convolutional layers: 14 depth-wise separable convolutional layers, with skip connections between some of the layers.
- global average pooling layer followed by a fully connected layer with sigmoid activation for classification.

PRELIMINARY RESULTS

➢ Training on the generated dataset of the source object produced an accuracy of 0.9994 and a loss 0.0029

➢ Inference on the target object did not yield the expected outcome of identifying defective tiles but this will be explored further





Conclusion

We have proven the success of our pre-processing approach by generating a sufficiently large dataset and using it to train a model. The model produced convincing results with a final accuracy of 0.9995 and a loss of 0.0029. We will further explore methods of improving the model's performance on the new and unseen target objects.

Emails: {atah.mih, hcao3}@unb.ca