Undergraduate Research Opportunity Programme (UROP) Project Report

Project Title

By

John Doe

Department of Computer Science

School of Computing

National University of Singapore

2024/2025

Undergraduate Research Opportunity Programme (UROP) Project Report

Project Title

By

John Doe

Department of Computer Science

School of Computing

National University of Singapore

2024/2025

Project ID: UXXXXX

Advisor: Professor John Doe

Deliverables:

Report: 1 Volume

Abstract

Your abstract goes here, maximum 200 words.

Subject Descriptors:

D.2.4 Software/Program Verification

D.2.5 Testing and Debugging

Keywords:

Foo, Bar, Baz

Implementation Software and Hardware:

Hardware

Software

Acknowledgements

Your acknowledgements go here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Example citation (Lamport, 1978).

1.1 Motivation

Chapter 2

Related Work

2.1 Design Overview

```java
public class HelloWorld {
    public static void main(String[] args) {
        // Prints "Hello, World!" to the terminal window.
        System.out.println("Hello,␣World!");
    }
}
```

Figure 2.1: Hello World in Java

2.2 Testing

2.3 Problems Encountered

2.4 Limitations

Chapter 3

Materials and Methods

3.1 Dataset

The main dataset used in this project was collected from XXX hospital and consists of 4369 elbow radiographs in either the lateral or anterior posterior view. Each radiograph was named according to the type of injury present (supracondylar fracture, dislocation, contusion etc). Using this information, the data was labelled as either injury and normal. The injury set includes radiographs with fractures and/or dislocations while the normal set consists of all other radiographs. The data was further divided into training, validation, and test sets as shown in figure 3.1. (figure)

In addition, the MURA dataset from Stanford Machine Learning Group was used for pretraining the CNN models. This dataset consists of 40561 radiographs of 7 different bodyparts: elbow, finger, forearm, hand, humerus, shoulder and wrist. Each image is labelled as either positive (abnormality) or negative (no abnormality). Notably, the criteria for abnormality used in MURA differs from the criteria for injury in the main dataset. For instance, a contusion would be classified as normal in the main dataset but would be classified as abnormal in the MURA dataset. The division of the data into training and validation sets is shown in figure 3.2. (figure)

3.2 Data Processing

3.3 CNN Models

3.3.1 Architecture

Convolutional Neural Networks (CNNs) are a type of neural network which use convolutional filters or kernels to learn features from data. For this project, 3 CNN models were used: ConvNeXt, EfficientNetV2 and MobileNetV3. These models were chosen due to their excellent performance on ImageNet and low low computational cost compared to similarly sized CNN models. After training the individual models, an ensemble model was created using feature fusion for the final classification. The framework used for the CNN models is outlined below.

- 1. Transfer Learning
  Transfer learning is a machine learning technique where a machine learning model is trained on one task and is then reused for another related task instead of starting again from scratch. This technique not only saves on training time but can also improve performance on the second task, especially if the amount of available training data is limited.

  In this case, the CNN models were initialised with weights which had been pretrained on ImageNet 1k. ImageNet 1k is a large dataset with over a million RGB images divided into 1000 categories such as plants, animals and furniture and is commonly used as a source of transfer learning for image classification tasks. The weights for the models were obtained from Pytorch Hub which is a publicly available model repository.

- 2. Greyscale vs RGB
  ImageNet is a colour dataset with 3 channel (RGB) images. Hence, the models cannot directly accept greyscale radiographs which only have 1 channel. This issue can be solved by duplicating the single channel twice to create a 3 channel image. However, this process is inefficient as it increases the memory cost of storing the image as well as the computational cost of processing the image through the model. To solve this, the first convolutional layer of each model was modified by summing the per channel weights of each kernel. This allows the models to directly accept greyscale images without affecting the output of the first layer.

- 3. Convolutional Block Attention Module
  Convolutional Block Attention Module (CBAM) is an attention mechanism for CNN models which can be incorporated into an existing architecture and trained together with the rest of the model. CBAM takes in an intermediate feature map and infers a channel attention map and a spatial attention map. The intermediate feature map is then refined using element wise multiplication with the 2 attention maps. CBAM has been shown to improve the performance of CNNs on different computer vision tasks while incurring

little additional computational cost.

ConvNeXt was modified with 4 CBAM modules placed after each of the 4 ConvNeXt blocks. EfficientNet was modified with a single CBAM module between the FusedMBConv modules and the MBConv modules, as well as a second CBAM module after the final convolutional layer. MobileNet was modified with a single CBAM module after the final convolutional layer.

- 4. Ensembling
  linebreak Ensembling is a commonly used technique in machine learning which combines the output of several different base models to get a single model which outperforms the individual ones. This can be done through several methods such as a simple majority vote or training a model on the outputs of the individual models.

  In our case, feature fusion was used to combine the outputs of the 3 individual CNN models. This is done by concatenating the flattened feature vectors of each model before they are sent to the fully connected layer. By default, Convnext has 768 output channels while EfficientNet has 1080 output channels and MobileNet has 960 output channels. To ensure each model has equal input to the ensemble, the final convolutional layer of EfficientNet was modified to have 768 output channels and MobileNet had an additional convolutional layer added to reduce the channels from 960 to 768.

  The concatenated feature vector with 2304 features is then normalised and passed to a dropout layer which randomly sets 30% of the features to 0 to mitigate overfitting. The feature vector is then used as the input for a small neural network to generate the final prediction. (figure)

3.3.2 Training

As mentioned previously, ImageNet is a dataset of natural objects and is very different from the greyscale radiographs in our target dataset. Hence, it is not an ideal source of transfer learning as the features learnt from imagenet may not be useful. To address this problem, the MURA dataset was used for a pretraining step before finetuning further on the target dataset.

3.4 Vision Language Models

Chapter 4

Evaluation

4.1 Hardware

4.2 Experimental Setup

4.3 Results

ID	Name	Score
1	Alice	88
2	Bob	92
3	Carol	85

Table 4.1: Example of a simple table.

Chapter 5

Challenges

Chapter 6

Conclusion

6.1 Summary

6.2 Future Work

References

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, *21*(7), 558–565. https://doi.org/10.1145/359545.359563

Appendix A - Something

Appendix B - Another Thing