Convolutional neural network for fracture detection in x-ray images

**Juan José García Aguirre**

**Abstract** - This study introduces a convolutional neural network (CNN) tailored for detection of bond’s fractures based on X-ray images. Image classification is a crucial task in computer vision, finding applications across diverse fields. In the medical one, X-ray images serve as pivotal diagnostic tools, offering insights into a patient's health condition. This work focuses on automating fracture detection using only X-ray images, leveraging CNNs for image classification. The proposed methodology integrates various tools and techniques within a CNN framework, implemented using PyTorch and executed on the Google Colab environment with a Tesla T4 GPU.

**Introduction**

Image classification is a fundamental problem in computer vision within various application fields. One of these could be the processing of X-rays, which encompass different types of radiographies to obtain predictions about a person's health based only on the X-ray images. This serves as a diagnosis tool. In [1], this approach is presented for classifying X-ray images to detect pneumonia. It is important to note that the procedure is only used as a diagnostic support tool. Another example is presented in [2], where another specific type of X-ray is used to detect gastritis applying convolutional neural networks (CNNs), which are commonly employed in image processing.

The present work describes the design of a neural network for the classification of X-ray images, determining whether a fracture exists or not, using the labels "fractured" and "not fractured". For this purpose, we start from the CNN type of neural network, integrating it with various tools. The framework for the development of the network was conducted in PyTorch, and executed entirely in the Google Colab environment, leveraging the available GPU defined as Tesla T4.

**Methodology**

The methodology employed is illustrated in Figure 1, delineating two main processes: training and testing. Initially, the training dataset was augmented to enhance its robustness. This augmentation strategy primarily involved image flipping. Subsequently, the training dataset was partitioned into randomly selected subsets for training and validation purposes. The validation subset played a crucial role in evaluating the model's performance without bias towards the training data. The obtained model was then deployed for the testing phase. Prior conducting the prediction process, the test data underwent was preprocessed by image flipping.

**A. Dataset**

The dataset used for the work was the "Fracture detection using x-ray images" dataset [3] obtained from the Kaggle website. All images are in jpg format and have dimensions of 224 x 224 pixels. Images are further classified into two labels "Fractured" or "Not fractured". Table 1 shows the composition of the dataset, However, following the augmentation process, the data distribution changes.

Table 1 Dataset composition



Diagrama

Descripción generada automáticamente**B. Pre-processing**

Fig. 1 Metodology of CNN

The preprocessing process includes the initial transformations performed on the images, such as 'ToTensor' to convert the images into PyTorch tensors, and normalization based on the calculation of the mean and standard deviation of the image groups. This normalization is applied for a faster convergence of the model converge.

**C. Augmentation**

To improve the model's performance, augmentation strategies were chosen, using a simple transformation such as 'flip'. Also, before the training process begins, the data is normalized based on its initial mean and standard deviation. Then, after performing the transformation and splitting into different loaders for training, validation, and testing, the dataset's distribution obtained is presented in Table 2.

Table 2 Loaders composition



**D. Model architecture and training**

The architecture of the proposed model is based on a convolutional neural network with Inception blocks. These blocks are designed to extract features in parallel through different convolution and pooling operations. The Inception modules configuration with dimension reduction is used. After each Inception block, a pooling layer is added to reduce dimensions.

The model starts with an initial convolutional layer that processes input images of 3 channels (RGB) and produces 32 features using a 7x7 kernel. This is intended to provide a broad analysis of the image, aiming to find patterns that can be used for prediction. Additionally, a stride of 2 is used to counteract the high kernel, reducing the amount of data for memory saving. Then, a series of Inception blocks follow, each followed by a pooling layer to reduce dimensions. After the Inception blocks, the model has final layers, including fully connected linear layers and a *softmax* output layer to classify the images into the desired classes.

Regarding training, the cross-entropy loss function is used to calculate the error between the model's predictions and the actual labels. The Adam optimizer is employed to adjust the model's weights during training, the learning rate used was 0.001, as it is a common value for Gráfico, Gráfico de líneas

Descripción generada automáticamenteneural networks of this type. Regularization is incorporated through dropout after some convolutional and fully connected layers to prevent overfitting. The model is trained using training data, and its performance is validated using validation data.

Fig. 3 Train-loss vs steps

Tabla

Descripción generada automáticamente

Fig. 2 summary of proposed architecture

**Results**

**A. Training**

The training process was carried out with a total of 15,954 images as indicated in Table 2, over a total of 30 epochs. A validation accuracy of 85% was obtained for the model validation, which is a good performance compared to the neural networks developed in [2]. It should be noted that they are different types of X-rays, However, they can serve as a reference point to gauge what constitutes adequate values.

**B. Testing results**

Regarding the testing results, an accuracy of 81% was achieved, indicating a reasonable predictive capability for detecting fractures in X-ray images. This result demonstrates the effectiveness of the trained model in fracture detection.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig. 2 Test-accuracy vs steps

**Conclusion**

This project proposes a CNN architecture with an augmentation strategy to detect fractures in X-ray images. The experiment results show that the architecture used produces optimal results. It seems that the model may be undergoing overfitting, but it could be improved by increasing the size of the data used. Compared to other models, this dataset can be considered small, so the performance achieved (81% test accuracy) demonstrates that the model can be used with small datasets or much larger ones.

**References**

[1] S. A. Khoiriyah, A. Basofi, and A. Fariza, “Convolutional Neural Network for Automatic Pneumonia Detection in Chest Radiography,” *IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort*, pp. 476–480, Sep. 2020, doi: 10.1109/IES50839.2020.9231540.

[2] R. Togo *et al.*, “Detection of gastritis by a deep convolutional neural network from double-contrast upper gastrointestinal barium X-ray radiography,” *J Gastroenterol*, vol. 54, doi: 10.1007/s00535-018-1514-7.

[3] “Fracture detection using x-ray images.” Accessed: May 24, 2024. [Online]. Available: https://www.kaggle.com/datasets/devbatrax/fracture-detection-using-x-ray-images/data