The Academic College of Tel Aviv-Yaffo

THE SCHOOL OF COMPUTER SCIENCE

Lumbar Vertebrae Classification

Thesis submitted in partial fulfillment of the requirements for the M.Sc. degree in the School of Computer Science of the Academic College of Tel Aviv-Yaffo

By

Yamit Shefler

The research work for the thesis has been carried out under the supervision of

Professor Gideon Dror

June 2024

Acknowledgements

I would like to thank Professor Adi Shribman who supported and guided me on the topic of the thesis research, introduced me to Professor Dror and made sure that the work progressed as expected.
Also, I would like to thank Doctor Sarel Cohen for his help with the GAN.
Finally, I would like to deeply thank Professor Gideon Dror for a long and significant period of guidance in the thesis. Professor Dror was a source of knowledge and inspiration for me. Thank you very much for the cooperation, the responsiveness, the willingness to help at every stage, and the respectful conversation between us. I had the honor of working with you.

Contents

1 Abstract **3**

2 Introduction **4**

3 Data Prepossessing **5**

3.1 Orientation 6

3.2 Center 7

3.3 Resize 7

3.4 Image Type 7

4 The Classification Model **9**

4.1 CNN 9

4.2 Data augmentation 9

4.3 Transfer learning 10

4.4 InceptionV3 10

4.5 Google Colaboratory 11

4.6 Keras 11

4.7 Model stages 11

5 GAN **15**

5.1 Model stages 15

5.2 Generate Vertebrae Samples 17

6 Results **18**

7 Analysis **20**

7.1 Evaluate a single cell 20

7.2 Evaluate symmetric cells 20

7.3 Results 20

8 Summary **23**

9 Additional Sources **25**

1 Abstract

Classifying human lumbar vertebrae (L1-L5) in images traditionally relies on expert analysis, achieving accuracy between 68% and 84%[1]. This work explores a Convolutional Neural Network (CNN) based approach to automate the process. The study investigates the impact of image preprocessing (brightness adjustment, alignment, and standardization) and data augmentation with Generative Adversarial Networks (GANs) on classification accuracy. A pre-trained InceptionV3 architecture with transfer learning is employed within the CNN model. The model is trained and tested on a dataset of images of L1-L5 vertebrae, and experimented with several scenarios, using the original images, the preprocessed images, and the preprocessed images augmented with GAN-generated data. The results demonstrate that the model achieves the highest accuracy when classifying vertebrae in the preprocessed data augmented with GAN-generated samples. The Occlusion Sensitivity Analysis technique is used to identify image regions crucial for classification. This research suggests that image preprocessing and data augmentation with GANs can significantly improve CNN performance for human lumbar vertebrae classification.

2 Introduction

The main purpose of this thesis is to classify human vertebrae using machine learning. More specifically, the classification of photographs of human lumbar vertebrae (L1-L5) using neural networks.

Human vertebrae classification is an open problem. Currently[1], in order to classify the lumbar vertebrae, experts are required to have at least some simple measures of the vertebrae, such as the linear distances or landmarks on the vertebrae images. Correct classification of human lumbar vertebrae ranges from 68% to about 84%, although current methods are not very consistent[2]. Experts often use manual or semi-automated computer-aided tools to extract measurements from vertebrae images. They must manually analyze images to identify and measure relevant features. This process is somewhat subjective, as different experts may interpret the images differently. These methods might be affected by human error or by the quality of the images.

Machine learning algorithms offer a potential solution to these challenges. Machine learning algorithms can be trained to automatically identify and measure features in images. This can make the assessment process more objective and efficient.

Pattern recognition is the ability of machines to identify patterns in data, and then use those patterns to make decisions or predictions using computer algorithms. It is a vital component of modern artificial intelligence (AI) systems. Humans are natural pattern recognizers. Even young children can recognize digits and letters, regardless of their size, style, or orientation. However, we do not fully understand how humans recognize patterns. Current classification methods might be inconsistent and sometimes inaccurate, making it difficult for humans to interpret. This is driving the need for machines that can identify patterns quickly and accurately, in our case, human vertebrae classification.

This study proposes an automated vertebrae classification using CNN. Different data sources were used in the study. As part of the effort to correctly classify the vertebrae, the classification model was built according to this problem, changes were made to the original images, new samples were generated, and various experiments were performed on the base model.

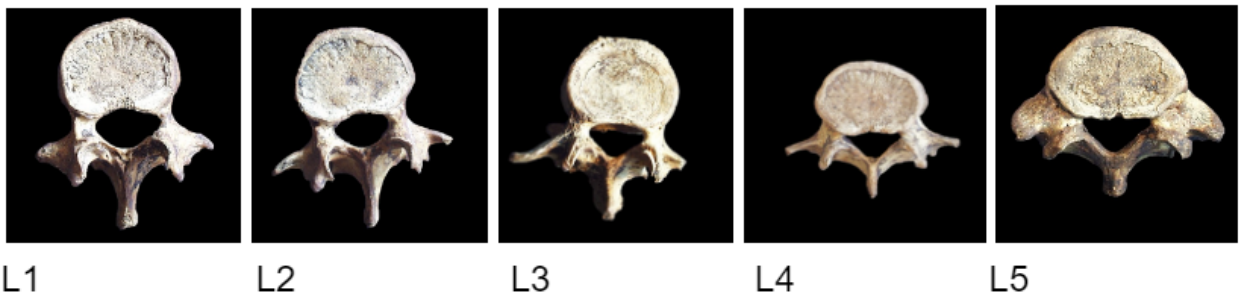

Figure 1: Sample of examples from the original dataset

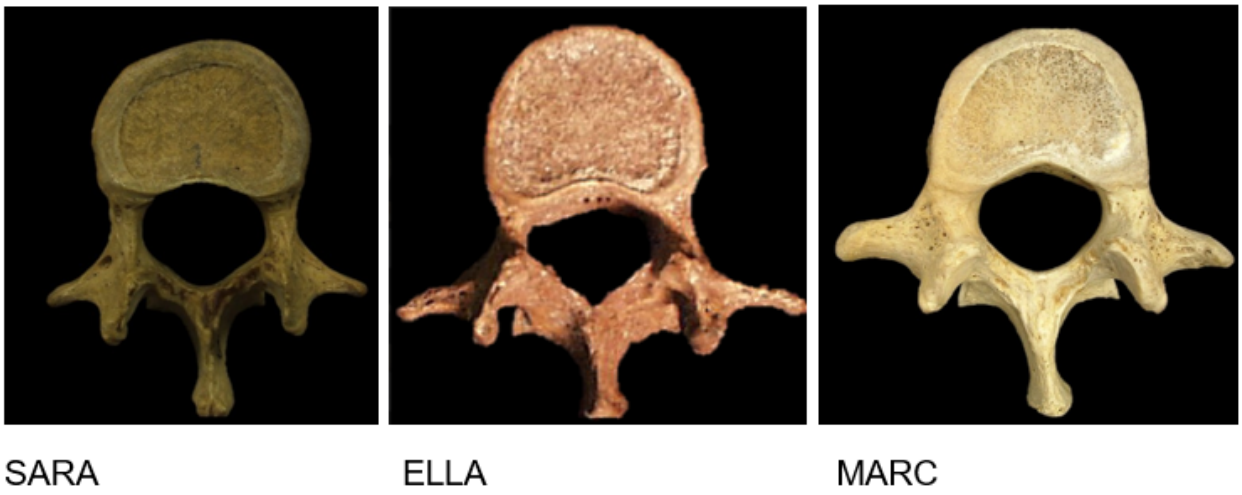

Figure 2: Colors Differences

3 Data Prepossessing

The dataset used for the research contains 765 photographs of L1-L5 vertebrae with similar proportions of L1, L2, L3, L4, and L5. The photographs have been segmented and the background has been rendered black. See Fig. 1 for an example of 5 dataset samples.

The dataset contains 3 data sources: SARA, ELLA, MARC, contributed by three researchers. The photographs have been divided into classes (L1-L5), and split into train set and test set. The test set includes about 10% of the data.

There were inconsistencies in the format of the images across the three data sources. The size and type of the item images varied: SARA images were the largest at 920 x 922 pixels and 589KB in PNG format. ELLA images were smaller at 452 x 449 pixels and came in TIF format, while MARC images were the smallest at 720 x 720 pixels and used JPG format. All three have color differences (see Fig. 2).

To ensure consistency between all images, the following modifications were done:

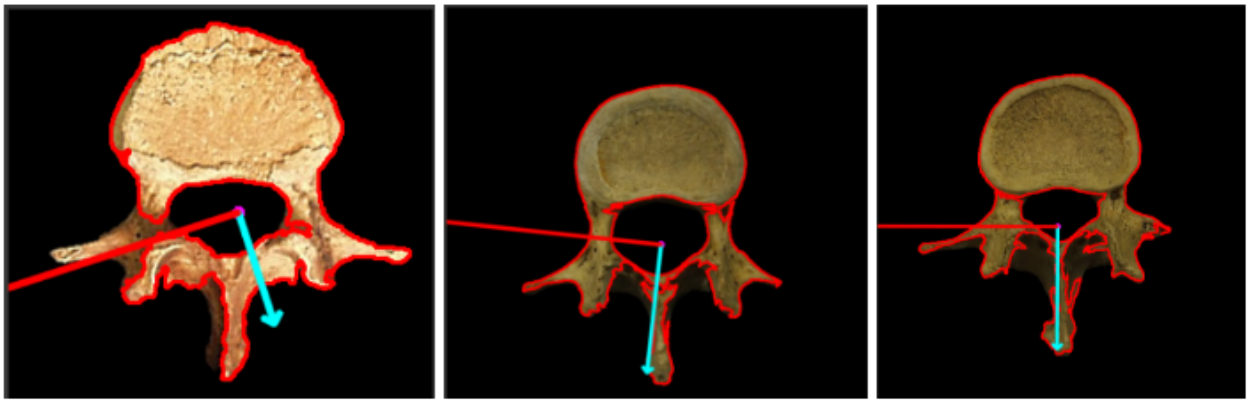

Figure 3: PCA before Histogram Equalization

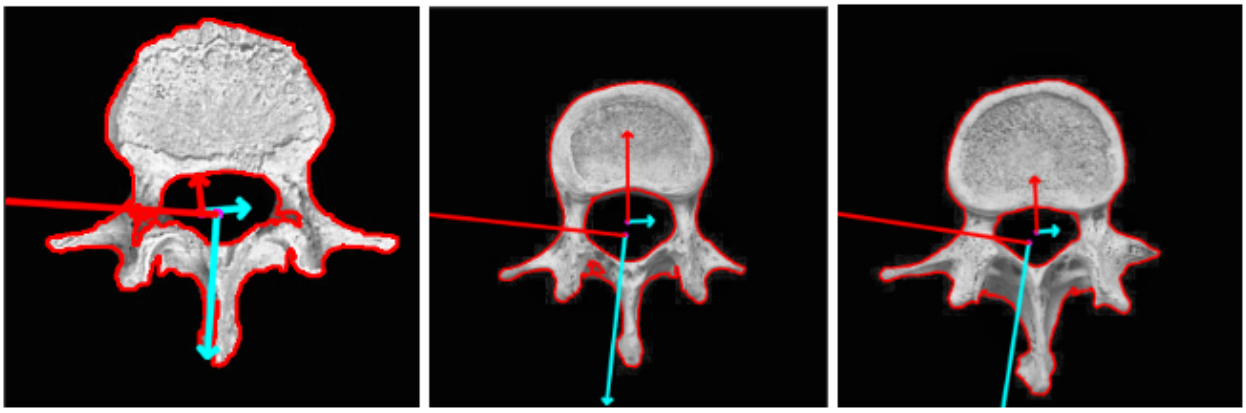

Figure 4: PCA after Histogram Equalization

3.1 Orientation

The vertebrae were not vertical: some vertebrae had to be rotated in order to be perfectly vertical. First, the algorithm extracts two contours for each vertebra: one for the bone itself and another for the hole in the center. Then, it creates a buffer for a technique called Principal Component Analysis (PCA). PCA is a common tool in data analysis that simplifies complex information by reducing its dimensionality while preserving key details. Using this analysis, the algorithm rotates each vertebra along its main axis. Initially, 17% of these rotations were incorrect (see Fig. 3). The issue stemmed from poorly lit images where the algorithm could not properly extract the contours of the vertebrae. To address this, the images required adjustments to become brighter and have more even light distribution. This is achieved through a technique called histogram equalization.

Histogram equalization enhances image contrast by manipulating the distribution of pixel intensities. An advanced version of this technique, called Contrast Limited Adaptive Histogram Equalization (CLAHE), was applied. After applying CLAHE, the rotation algorithm's accuracy improved by 40% (see Fig. 4).

Finally, in a small number of cases (around 4%), the algorithm still did not perform perfectly, and manual rotation of the vertebrae in the image was necessary (see Fig. 5).

Figure 5: Manual Rotation

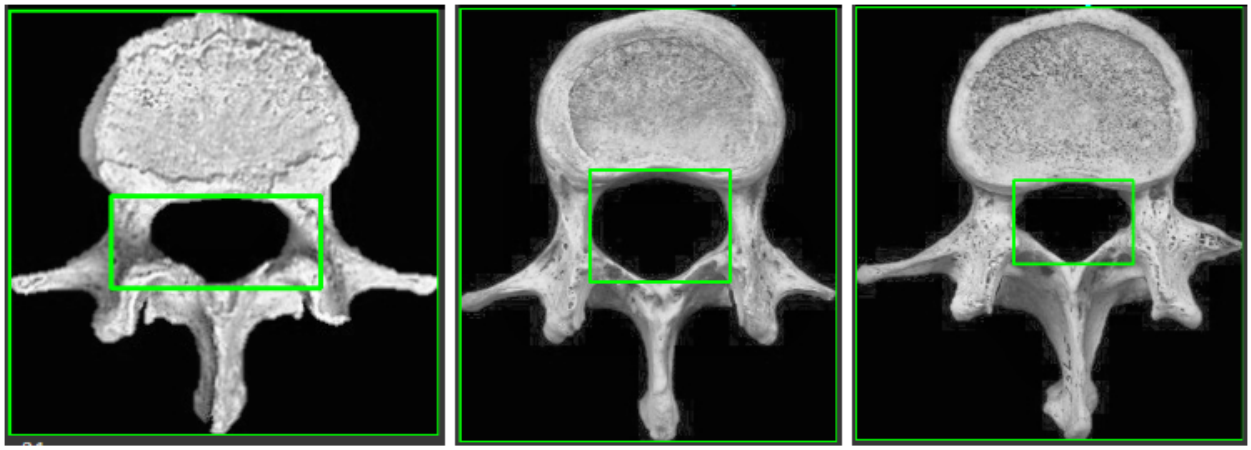

Figure 6: Centered

3.2 Center

Some vertebrae were not perfectly centered, and some images had different relationships between the background and the vertebra.

First, it was necessary to get the contours of the vertebra. Then, find the minimum bounding rectangle that encloses the contour, measure the width and height of the rectangle, and calculate a square according to the maximum value. Finally, crop a square around the vertebra (see Fig. 6).

3.3 Resize

All images have the same proportions (240 X 240).

3.4 Image Type

Save all images as JPG type.

After data preprocessing, all the vertebrae have uniform brightness, are vertical, centered, and have the same proportions and image type.

4 The Classification Model

As part of this work, various tools and techniques were used:

4.1 CNN

CNN, or Convolutional Neural Network, is a specific type of deep learning neural network architecture excelling at analyzing grid-like data, primarily images and videos.

CNN layers:

1. Input Layer: The input to a CNN is a 3D array representing an image (width, height, and channels for color images).

2. Convolutional Layers: These layers apply filters to the input, extracting features like edges, lines, and shapes. Each filter learns to detect a specific feature in the image. The output of a convolutional layer is called a feature map.

3. Pooling Layers: Pooling layers down-sample the feature maps, reducing their dimensionality. This helps to control overfitting and computational costs.

4. Activation Layers: These layers introduce non-linearity into the network, allowing it to learn more complex relationships between features. A common activation function used in CNNs is the ReLU (Rectified Linear Unit) function.

5. Fully Connected Layers: After the convolutional and pooling layers, the network typically has one or more fully connected layers similar to those found in standard neural networks. These layers combine the extracted features to make classifications or predictions.

6. Output Layer: For image classification, the final layer might have multiple neurons corresponding to different classes, with the highest output neuron indicating the predicted class.

4.2 Data augmentation

Data augmentation is a technique used in machine learning, particularly computer vision, to artificially expand the training dataset. It involves creating new training examples from existing data through various transformations. This is especially beneficial when dealing with limited datasets, as it helps the model generalize better and avoid overfitting.

The importance of Data Augmentation:

- Limited data: Acquiring large, labeled datasets can be expensive and time-consuming. Data augmentation helps overcome this limitation by creating more training data from existing samples.

- Improved generalization: By exposing the model to variations of the same data (e.g., rotations, flips, brightness changes), data augmentation helps it learn features that are robust to these variations. This leads to a model that performs well on unseen data, not just the training data it was shown.
- Reduced overfitting: With a larger and more diverse dataset, the model is less likely to overfit to the specific details of the training data and can learn more generalizable patterns.

4.3 Transfer learning

Transfer learning is a powerful technique in machine learning that leverages knowledge gained from a pre-trained model on one task (source task) to improve performance on a new, related task (target task). It is particularly beneficial when dealing with limited data for the target task [3].

Transfer learning with pre-trained image networks has been successfully applied to many applications, for example, object detection in self-driving cars [4]. Pre-trained models are used to detect objects like pedestrians, vehicles, and traffic lights on the road. This is crucial for autonomous vehicles to navigate their environment safely.

Benefits of Transfer Learning:

1. Reduced Training Time: By leveraging pre-trained knowledge, transfer learning can significantly reduce training time for the target task, especially when dealing with limited datasets.
2. Improved Performance: Transfer learning often leads to better performance on the target task compared to training a model from scratch, particularly when the source and target tasks are related.
3. Efficient Use of Data: It allows effective use of limited data for the target task, as the pre-trained model already has a strong foundation of knowledge.

4.4 InceptionV3

InceptionV3 is a convolutional neural network (CNN) architecture specifically designed for image recognition and analysis[5].

Key Features:

- Convolutional Architecture: Like most CNNs, InceptionV3 processes images through a series of convolutional layers that extract features from the data. These features become progressively more complex as the network progresses through deeper layers.
- Inception Modules: InceptionV3 utilizes a specific building block called the "Inception module." This module incorporates multiple convolutional and pooling operations within itself, allowing the network to learn features at different scales and resolutions simultaneously. This approach improves the model's ability to capture diverse information within an image.

- Pre-trained Model: InceptionV3 is widely available as a pre-trained model, meaning it has already been trained on a massive dataset. This pre-trained model can be a valuable starting point for various image recognition tasks through transfer learning.

4.5 Google Colaboratory

Google Colaboratory, often shortened to Colab, is a cloud-based platform for data analysis and machine learning. It essentially provides a virtual workspace to run code without preliminary work.

4.6 Keras

Keras is a deep learning library used on top of other frameworks like TensorFlow. It essentially provides a simpler and more user-friendly interface for building and experimenting with deep learning models. Keras is known for its straightforward API (Application Programming Interface) which makes it easier to learn and use. Keras models can be deployed across various platforms and devices, making them adaptable to different project needs. It also provides access to a collection of pre-trained models that can be used as a starting point for new projects. This can save developers significant time and effort compared to building models from scratch.

4.7 Model stages

The process begins by loading the training and test images. Next, a pre-trained InceptionV3 model is loaded and its convolutional blocks are frozen, essentially turning it into a feature extractor. This ensures the model focuses on learning from the newly added layers instead of retraining the entire InceptionV3 architecture. The final activation feature map from InceptionV3 provides bottleneck features, which are then flattened and fed to a fully connected deep neural network classifier.

Data augmentation is then applied to the training data. This technique involves creating variations of the training data through random modifications. By ensuring the model never sees the exact same image twice, data augmentation helps prevent overfitting and improves the model's ability to generalize to unseen data. Keras' "ImageDataGenerator" class is a useful tool for achieving this.

Following data preparation, the model architecture is built. The model architecture is defined as a sequential model, where layers are added sequentially. The pre-trained model is incorporated as the very first layer. Following this, a dense layer with 256 neurons and a ReLU activation function is added to introduce non-linearity. To prevent overfitting, a dropout layer with a 20% dropout rate is introduced. This is followed by another dense layer with 64 neurons and a ReLU activation, along with another dropout layer with a 20% dropout rate. Finally, the output layer is added, containing a number of neurons equal to the number of classes the model is predicting for.

The model is then trained using the prepared data. Model performance is evaluated by plotting the accuracy and loss curves after each training epoch. Ideally, the training and validation accuracy should be close, indicating a good fit (see Fig. 7).

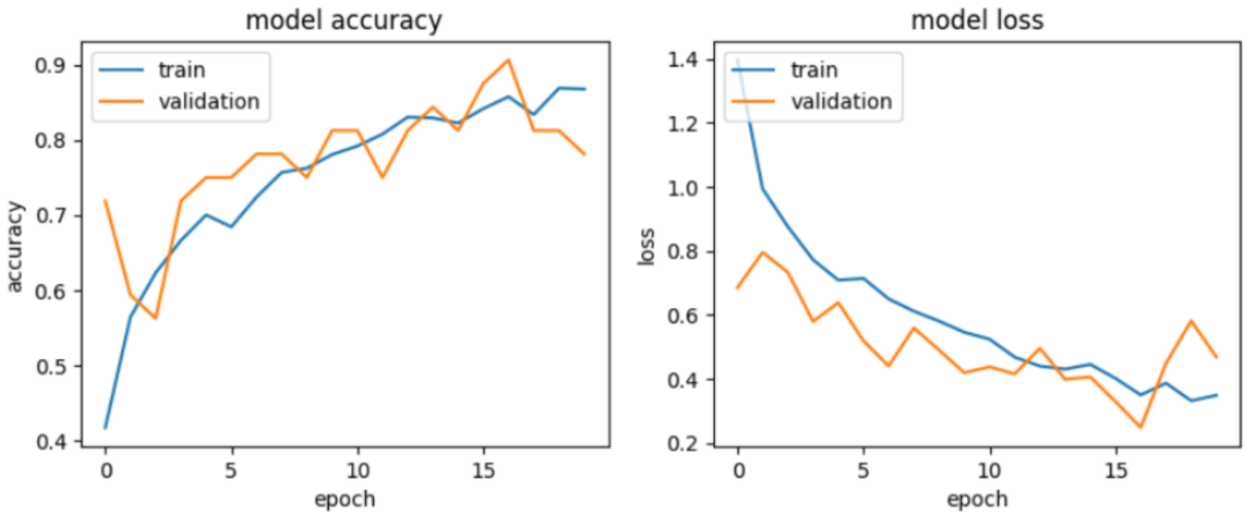

Figure 7: Model Loss and Accuracy

Once trained, the model can be used to make predictions on unseen test data. The model outputs probabilities for each class, and the class with the highest probability is chosen. Finally, these class numbers are converted back to class names.

To further analyze the model's performance, we compute the confusion matrix, that provides a clear breakdown of how the model classified the data (see Fig. 8). We also compute a classification report that provides various metrics to assess how well the model classified each category (see Fig. 9). The classification report includes:

- Precision: the proportion of correct positive predictions.
- Recall: the proportion of actual positive cases that the model identified correctly.
- F1-score: combines precision and recall, giving a balanced view of the model's performance.

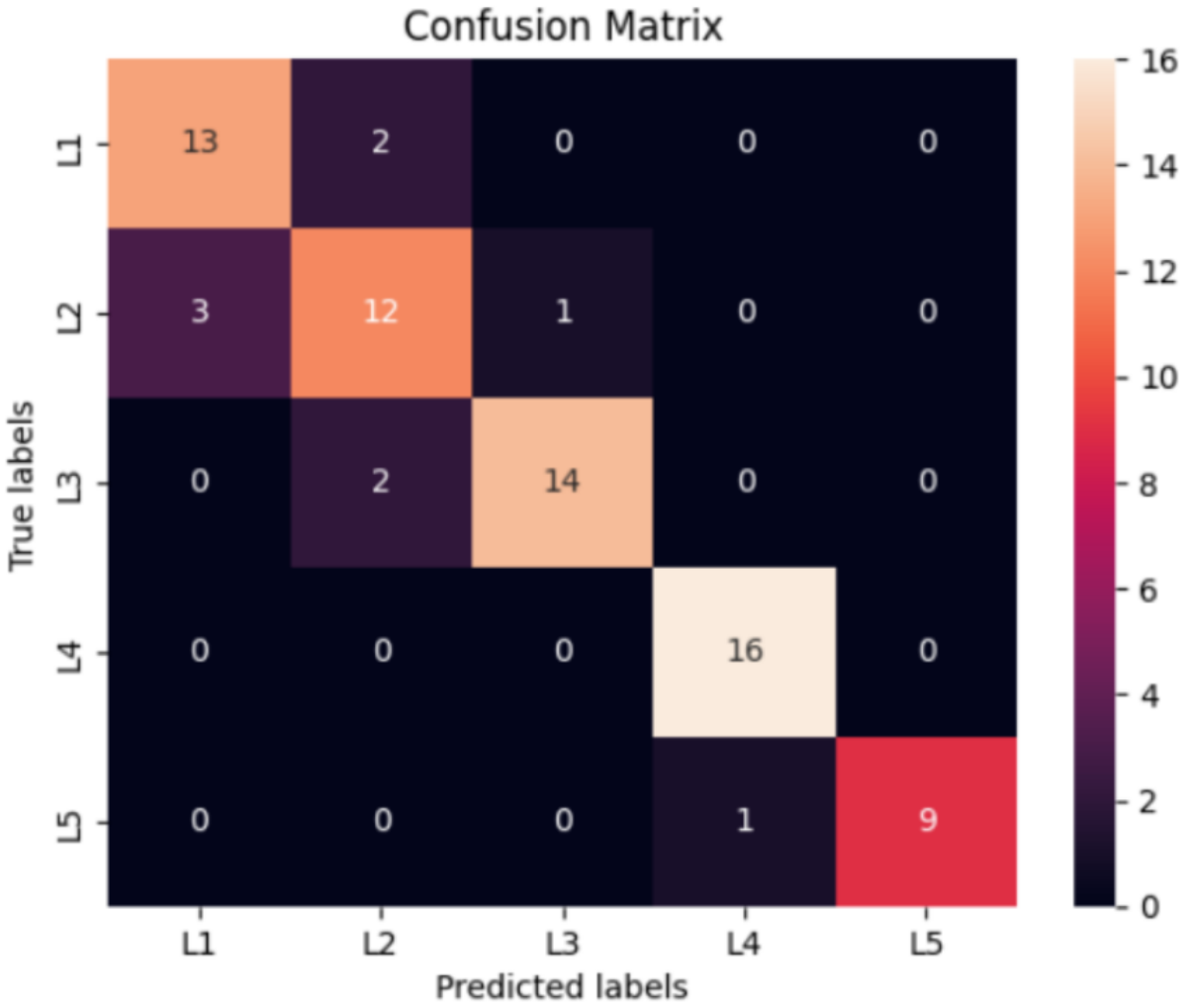

Figure 8: Confusion Matrix

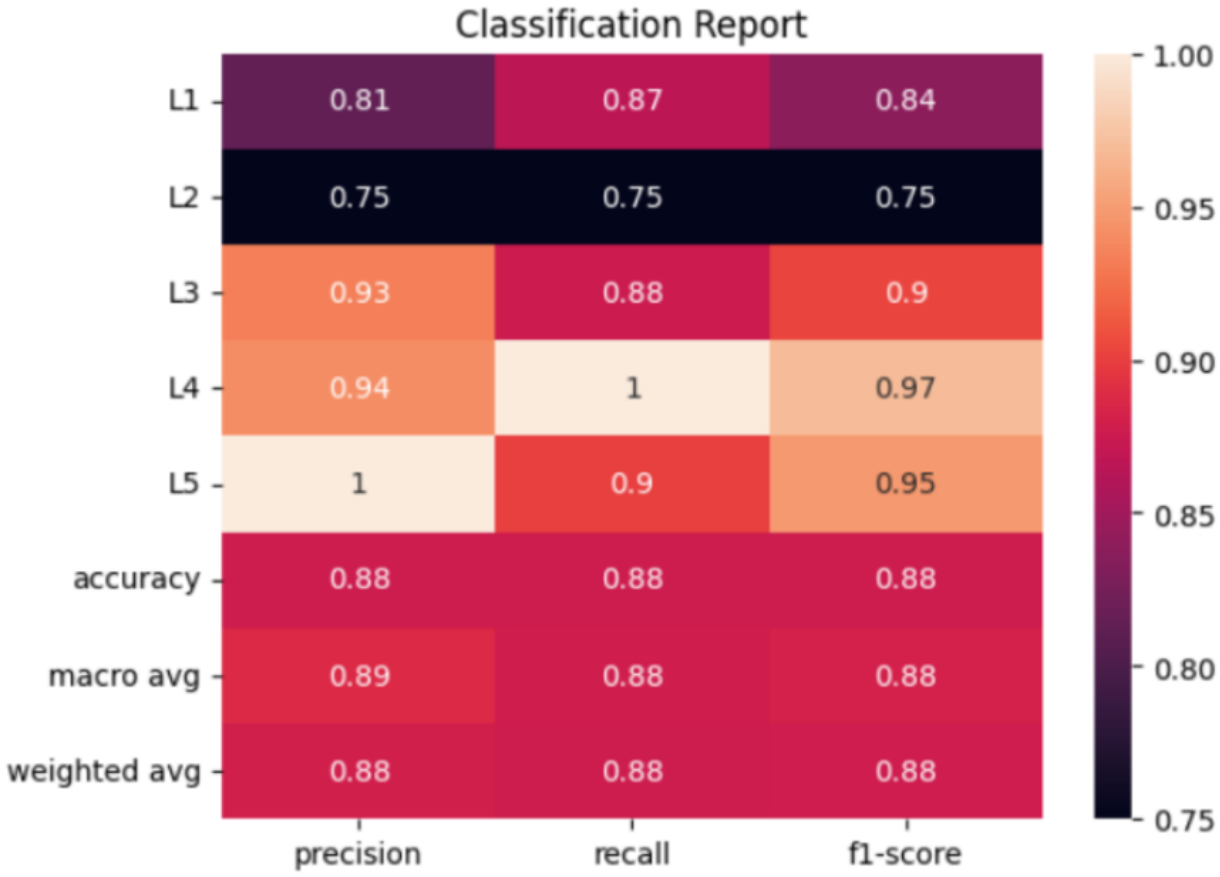

Figure 9: Classification Report

5 GAN

Conditional GAN

Generative artificial intelligence (generative AI) is an artificial intelligence capable of generating text, images, videos, or other data using generative models, often in response to prompts. Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.

A Generative Adversarial Network (GAN) is a class of machine learning frameworks and a prominent framework for approaching generative AI. A GAN consists of two neural networks competing against each other: a generator and a discriminator. The generator tries to create new data, while the discriminator attempts to determine if the data is real or generated. This competition helps the generator improve its ability to create realistic data[6].

DCGAN, which stands for Deep Convolutional Generative Adversarial Network, is specifically designed to generate new data that closely resembles real-world data like images. Unlike regular GANs, DCGAN uses convolutional layers in both the generator and discriminator. These layers are particularly useful for processing image data, allowing the DCGAN to capture the spatial features of images, like edges and shapes.

GANs generate novel image data, video data, or audio data from a random input. The random input is sampled from a normal distribution, before going through a series of transformations that turn it into something plausible (image, video, audio, etc.).

While a simple DCGAN offers generated samples, it lacks control over their specific characteristics (e.g., class). In this case, a GAN creating vertebrae images was necessary. A basic DCGAN wouldn't allow choosing the type of vertebra generated. To influence the output, the GAN needs additional information, like the image's class, to condition its generation process.

5.1 Model stages

The process begins by loading the training images for the Conditional Generative Adversarial Network (CGAN). Next, the dataset is preprocessed, which involves defining constants and hyperparameters like batch size, number of channels, number of classes, and image size.

A key difference in Conditional GANs compared to regular GANs is the incorporation of class labels. This is achieved by calculating the number of input channels for both the generator and discriminator to accommodate these labels. The number of classes is added to the standard noise input dimension for the generator and the generated image input dimension for the discriminator.

Following data preparation, the generator and discriminator architectures are defined:

The generator starts with random noise in the latent space, projects it to a higher dimension, and uses deconvolutional layers to progressively increase the resolution and detail of the generated image. The generator network utilizes a sequential model architecture, where layers are stacked linearly. The first layer is the input layer, which defines the dimensionality of the latent space. This essentially represents the

number of random numbers used as input to create the image. Following the input layer is a dense layer. This layer takes a vector of random numbers as input and uses a Leaky ReLU activation function with a small negative slope to prevent "dying neurons" (neurons that always output zero). The size of this layer's output is equal to the number of channels in the image. This dense layer transforms the random noise from a lower-dimensional latent space into a higher-dimensional space suitable for reshaping into an image-like representation. A reshape layer then takes the output from the dense layer and converts it into a 3D tensor with specified dimensions. This creates an initial low-resolution activation map. The core functionality of the generator comes from deconvolutional layers, also known as transposed convolution layers. These layers progressively upscale the feature maps, essentially learning to generate increasingly higher-resolution and detailed features. Each deconvolutional layer is followed by a Leaky ReLU activation with a small negative slope (0.2) to introduce non-linearity.

The discriminator aims to distinguish between real images and images generated by the GAN's generator network. The discriminator network is built using a sequential model architecture. The first layer is the input layer, defined with parameters for image size (height, width) and channels. Following the input layer are convolutional layers. Each convolutional layer uses 64 filters. A ReLU activation function is applied after each convolutional layer to introduce non-linearity. Pooling layers are then introduced, each with a stride of 2 to reduce the spatial dimensions of the data. This helps capture higher-level features and reduces model complexity. A dropout layer with a 50% dropout rate is placed after the third convolutional layer to prevent overfitting. The convolutional layers are flattened into a one-dimensional vector to prepare the data for fully-connected layers. Two dense layers with 800 neurons each are added, with ReLU activation following each layer. Dropout with a rate of 50% is used again after each dense layer for regularization. Finally, the model ends with a single neuron output layer. No activation function is applied here because the discriminator typically outputs a value between 0 and 1, representing the likelihood of the input being real data.

These components are then combined to create the complete CGAN model. The CGAN model effectively combines pre-built discriminator and generator architectures with conditional training using class labels. It facilitates training a model that can generate new data conditioned on specific class information.

Once the model is built, the training process begins. After training, the CGAN's capabilities are explored by interpolating between classes with the trained generator. This involves sampling noise from a normal distribution, reshaping it appropriately, and then uniformly distributing it along with the desired class identities.

- Conditional GAN Aspect - Conditioning on Labels: The "one hot labels" are used to condition the generation process. These labels are first expanded with dummy dimensions and then repeated to match the image dimensions. This allows them to be concatenated with the images as extra channels. When training the generator, the random noise from the latent space is concatenated with the class labels before being fed to the generator. This injects class information into the generation process, allowing the generator to produce images specific to the provided class labels.

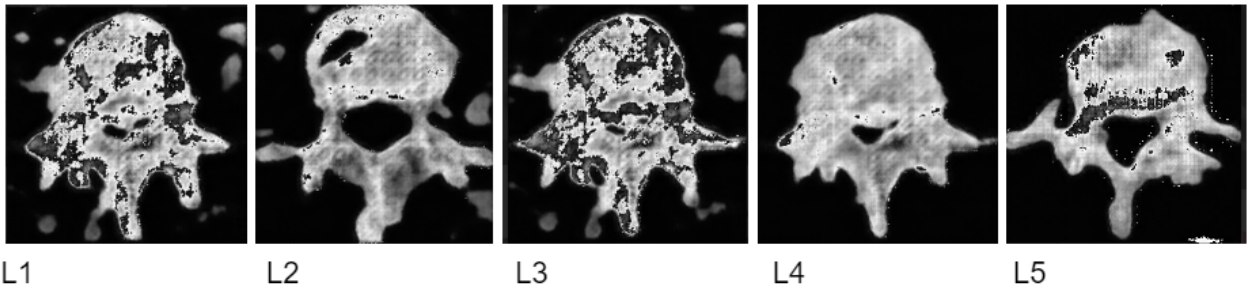

Figure 10: removed images

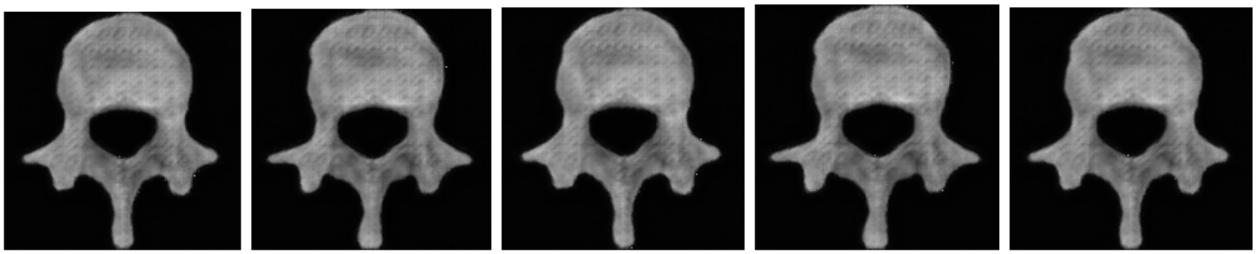

Figure 11: similar images

- Training Loop: The training loop has two main parts: training the discriminator and training the generator.

  Discriminator Training: combines real and fake images (generated by the generator) with their corresponding class labels. Then, trains the discriminator to distinguish between real and fake images.

  Generator Training: samples new random noise from the latent space, concatenates the noise with class labels to provide conditional information to the generator, generates fake images using the conditioned noise, and trains the generator to fool the discriminator by making the generated images appear realistic for the given class labels.

- Monitoring Losses: after each training step, the losses for both the generator and discriminator are calculated and stored in the internal metrics for monitoring purposes.

5.2 Generate Vertebrae Samples

Creating more samples using Conditional GAN allows to control the amount of samples that will be created for each class. The goal was to produce a maximum of effective samples for each class. The new samples were added only to the training set, so the test set includes only real vertebrae images.

100 images were generated for each label, some had to be removed due to poor quality (see Fig. 10).

For more than 100 images for each label, the Conditional GAN generated almost the same images, with only slight differences. The similar images had to be removed too (see Fig. 11).

New samples were generated for each class using Conditional GAN. After adding the new samples, the train set increased by 40% (see Table 1).

	Train	GAN	Total
L1	152	57	209
L2	153	47	200
L3	126	65	191
L4	121	42	163
L5	125	58	183

Table 1: Conditional GAN - Training Data

6 Results

The model's performance was compared using three scenarios:

1. **Original Images**

   Evaluation with the unprocessed dataset.

2. **Preprocessed Data**

   Evaluation with the preprocessed dataset.

3. **Preprocessed Data with GAN generated examples**

   Evaluation with the preprocessed dataset supplemented by synthetic samples generated by the CGAN.

The results demonstrated that the model achieved the highest test-accuracy when classifying vertebrae in the preprocessed data with GAN-generated samples (see Table 2).

The results likely showed the highest accuracy for that data for a few key reasons:

1. **Improved Image Quality**

   Preprocessing addresses issues like inconsistent lighting and positioning. This standardization allows the model to focus on the relevant features of the vertebrae, reducing noise and improving its ability to learn accurate patterns.

2. **Data Augmentation Diversity**

   The data augmentation technique introduced variations in the generated images, such as slight rotations, scaling, or noise. This exposes the model to a wider range of possible vertebral appearances, making it more robust to real-world variations in the data.

3. **Transfer Learning with InceptionV3**

   By leveraging the pre-trained InceptionV3 architecture, the model benefits from its ability to recognize general image features. Fine-tuning InceptionV3 for the specific task of vertebrae classification further improves its ability to identify the key features that differentiate the vertebrae.

	Original Data	After Data Preprocessing	Add GAN Samples
Train Accuracy	93%	95%	96%
Validation Accuracy	75%	78%	84%
Test Accuracy	78%	84%	88%

Table 2: Results

4. **Increased Training Data**

   The CGAN generated new, realistic images of vertebrae. This effectively expands the training dataset, providing the model with more examples to learn from. A larger and more diverse dataset helps the model generalize better to unseen data during evaluation.

In essence, the combination of preprocessing, GAN-generated data augmentation, and transfer learning provides the model with a cleaner, more diverse, and informative dataset. This allows the model to learn more robust and accurate representations of the vertebrae, leading to superior classification performance.

7 Analysis

Occlusion Sensitivity

This method involves systematically occluding (blocking out) different regions of the image and observing the change in the model's classification output. Regions that cause a significant drop in accuracy or a shift in the predicted class are likely to be important for the model's decision.

One way to do so is to divide the image into (N x N) cells and measure the performance of the trained model if a single cell is "deleted" (mask with a black square). Then, plot the accuracy of the model for each deleted cell. The expectation is that "deleting" cells outside the vertebrae boundary will have no effect on performance, and cells that reside on the boundary of the vertebrae should have a more significant effect.

In this analysis, the changes were made on the test set, while the training set remained as it was. Then, measure the test accuracy for each deleted cell, to see which part of the image is most important for the classification.

7.1 Evaluate a single cell

The goal is to examine which cell has the most influence on the classification. 3 cell sizes were evaluated: (64, 64), (32, 32), and (16, 16). For each cell size, the image is divided into (N x N) cells. Then, iterate for each cell location, "delete" a single cell on each image in the test set, and measure the test accuracy of the trained model. Finally, display the accuracy for all cells (see fig. 12).

7.2 Evaluate symmetric cells

The assumption is that a vertebra is somewhat symmetric, so it must be examined whether deleting symmetric cells will have a different effect on the classification. Same as above, 3 cell sizes were evaluated: (64, 64), (32, 32), and (16, 16). For each cell size, the image is divided into (N x N) cells. Then, iterate for each 2 symmetric cells location, "delete" 2 symmetric cells on each image in the test set, and measure the test accuracy of the trained model. Finally, display the accuracy for all cells (see fig. 12).

7.3 Results

Analysis results in two scenarios. First scenario is deleting a single cell for each image, and the second scenario is deleting 2 symmetric cells for each image.

For a single cell evaluation:

1. for mask size (64, 64) the lowest accuracy is 0.58 in coordinates (96, 96)
2. for mask size (32, 32) the lowest accuracy is 0.75 in coordinates: (48, 80), (94, 128), (128, 128)
3. for mask size (16, 16) the lowest accuracy is 0.77 in coordinates (112, 176)

For 2 symmetric cells evaluation

1. for mask size (64, 64) the lowest accuracy is 0.57 in coordinates (0, 96), (160, 96)
2. for mask size (32, 32) the lowest accuracy is 0.69 in coordinates (80, 80), (128, 80)
3. for mask size (16, 16) the lowest accuracy is 0.74 in coordinates (48, 96), (176, 96)

As expected, deleting cells on the vertebrae boundary has less effect than residing on the boundary of the vertebrae. In addition, the larger the mask, meaning the cell size, the lower the test accuracy, as it covers a larger area on the vertebra.

Reference[7].

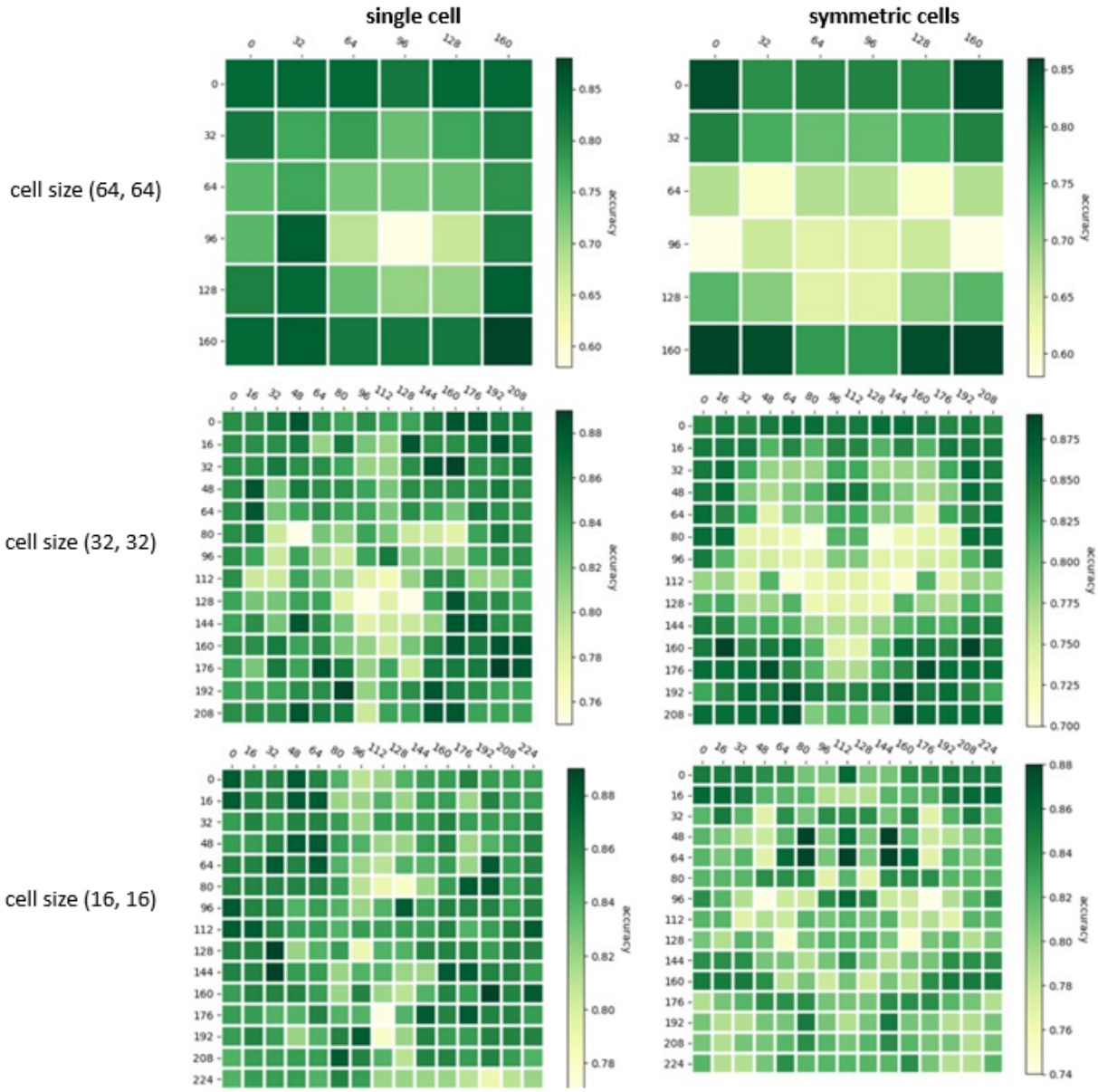

Figure 12: Analysis

8 Summary

Improving Vertebrae Classification with Preprocessing and Generative Adversarial Networks (GANs)

This study addressed the challenge of classifying human lumbar vertebrae (L1-L5) in photographs using machine learning. Traditionally, this task relies on expert analysis of simple measurements, resulting in accuracy ranging from 68% to 84%. These methods are subjective, time-consuming, and prone to human error.

This study proposes a novel approach utilizing a Convolutional Neural Network (CNN) model for automated classification, and investigated the impact of image preprocessing and data augmentation on the accuracy of human vertebrae classification using this model.

The research began with a dataset of vertebrae images. These images underwent a preprocessing pipeline:

1. Brightness adjustment - images were normalized to improve clarity using CLAHE - Contrast Limited Adaptive Histogram Equalization.
2. Alignment - vertebrae were rotated and centered within the image frame for consistent positioning. For orientation, the algorithm used PCA (Principal Component Analysis), and for centering the algorithm used the minimum bounding rectangle.
3. Standardization - images were converted to grayscale format to avoid color differences, resized to a uniform size and modified the data type to avoid differences between image types.

The CNN model leverages transfer learning by employing the pre-trained InceptionV3 architecture. This approach utilizes the powerful feature extraction capabilities of InceptionV3, while fine-tuning the final layers for the specific task of human lumbar vertebrae classification. Additionally, the model incorporates data augmentation techniques to artificially expand the training dataset and improve model generalizability.

Following, the CNN model was trained and evaluated on the processed data. The model's performance was then compared using three scenarios:

- Original Images
- Preprocessed Data
- Preprocessed Data with additional GAN samples

The results demonstrated that the model achieved the highest test-accuracy when classifying vertebrae with the preprocessed data combined with GAN-generated samples.

The results likely showed the highest accuracy for the model using preprocessed data augmented with GAN-generated samples due to:

- Improved Image Quality
- Data Augmentation Diversity
- Transfer Learning with InceptionV3
- Increased Training Data

Finally, the study employed additional analyses to identify the image regions most crucial for accurate classification by the CNN model. We saw that deleting cells on the vertebral border has less effect than deleting cells near the vertebral border. Also, the larger the deleted cell, the lower the test accuracy.

This research suggests that image preprocessing and data augmentation with GANs can significantly improve the performance of CNNs in classifying human vertebrae from images.

9 Additional Sources

- Lumbar Vertebrae Classification
- Lumbar vertebrae
- Differentiation and classification of thoracolumbar transitional vertebrae
- Computational techniques to segment and classify lumbar compression fractures
- How to convert an image to Black and White in Python
- Histogram Equalization
- Carnivores Image Classification using Google Colab
- Conditional GAN
- Deep Convolutional Generative Adversarial Network
- Wikipedia
- Tensorflow
- Keras
- Gemini
- Google Colaboratory

References

[1] N. Frater, ``Back problems : functional, historical and paleo-anthropological aspects,'' Ph.D. dissertation, University of Zurich, Zürich, 2017. [Online]. Available: https://www.zora.uzh.ch/id/eprint/150644/

[2] R. F. Masood, I. A. Taj, M. B. Khan, M. A. Qureshi, and T. Hassan, ``Deep learning based vertebral body segmentation with extraction of spinal measurements and disorder disease classification,'' *Biomedical Signal Processing and Control*, vol. 71, p. 103230, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809421008272

[3] H. Bichri, A. Chergui, and M. Hain, ``Image classification with transfer learning using a custom dataset: Comparative study,'' *Procedia Computer Science*, vol. 220, pp. 48--54, 2023, the 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923005446

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ``You only look once: Unified, real-time object detection,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779--788. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ``Going deeper with convolutions,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1--9. [Online]. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf

[6] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, ``Generative adversarial networks: An overview,'' *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53--65, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8253599

[7] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, ``Sanity checks for saliency maps,'' *Advances in neural information processing systems*, vol. 31, 2018. [Online]. Available: https://arxiv.org/pdf/1810.03292

תקציר

סיווג חוליות מותניות אנושיות (L1-L5) בתמונות מסתמך בדרך כלל על ניתוח ראשוני של מומחה, ומשיג דיוק בין 68% ל- 84% [1]. עבודה זו בוחנת גישה מבוססת רשתות נוירונים לאוטומציה של התהליך. המחקר חוקר את ההשפעה של עיבוד מקדים של תמונה (התאמת בהירות, יישור וסטנדרטיזציה) והגדלת נתונים באמצעות רשתות יריבות גנרטיביות (GANs) על דיוק הסיווג. ארכיטקטורת InceptionV3 מאומנת מראש עם שימוש בטכניקת Transfer Learning במודל CNN. המודל מאומן ונבדק על מערך נתונים של תמונות של חוליות L1-L5, ונוסה במספר תרחישים, תוך שימוש בתמונות המקוריות, התמונות המעובדות מראש והתמונות המעובדות מראש בתוספת דגימות שנוצרו על ידי GAN. התוצאות מוכיחות שהמודל משיג את הדיוק הגבוה ביותר עבור תמונות מעובדות מראש בתוספת דגימות שנוצרו על ידי GAN. כמו כן, טכניקת אנליזה של הדאטה Sensitivity Occlusion משמשת לזיהוי אזורי תמונה החיוניים לסיווג. מחקר זה מצביע על כך שעיבוד מקדים של תמונה והגדלת נתונים עם GANs יכולים לשפר משמעותית את ביצועי רשת הנוירונים עבור סיווג חוליות מותניות אנושיות.

המכללה האקדמית תל אביב - יפו

בית הספר למדעי המחשב

קלסיפיקציית חוליות מותניות

חיבור זה הוגש כחלק מהדרישות לקבלת התואר "מוסמך" - M.Sc.
במכללה האקדמית תל אביב - יפו

על ידי

ימית שפלר

העבודה הוכנה בהדרכתו של

פרופסור גדעון דרור

יוני 2024