University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Bone fracture identification using mask former segmentation and VIT model

Student Name and SRN:

Bhargavi Sapati and 21058787

Supervisor: Calum Morris

Date Submitted: 29/08/2024`

Word Count: 7399

Declaration statement

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used ChatGPT, or any other generative AI tool, to write the report or code (other than were declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Bhargavi Sapati

Student Name signature: Bhargavi

Student SRN number: 21058787

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Acknowledgement

I wish to convey my profound appreciation to all individuals who have provided assistance and guidance during the duration of this project. Above all, I express my gratitude to [Advisor/Supervisor's Name], whose perceptive feedback, motivation, and steadfast support played a crucial role in the accomplished culmination of this project. Their profound knowledge in the domain of computer vision and deep learning greatly enhanced the calibre of this endeavour.

The [Institution/Organization Name] deserves recognition for supplying the essential resources and creating a favourable setting for carrying out this research. Ultimately, I am profoundly grateful for the unwavering support and comprehension of my family and friends, whose constant encouragement and understanding were crucial in helping me endure and succeed in every phase of this undertaking.

I express my gratitude to each and every one of you for contributing to the success of our voyage, which has been both enriching and satisfying.

Abstract

This study explores the use of sophisticated machine learning methods to identify bone fractures in medical images. Specifically, it examines the effectiveness of combining MaskFormer segmentation with Vision Transformer (ViT) models. The main study inquiry is whether this amalgamation may surpass conventional Convolutional Neural Networks (CNNs) in precisely detecting and segmenting bone fractures. The goals include the creation of a sturdy and effective method for identifying fractures, boosting the precision of the model, and expanding the dependability of medical diagnostics.

The process entails gathering a large dataset of X-ray pictures from Kaggle and GTS AI, including different bone areas that are susceptible to fractures. The data was subjected to preprocessing and exploratory data analysis (EDA) to guarantee consistency and high quality. The MaskFormer model was used for accurate delineation of fracture locations, then followed by the implementation of the ViT model for categorization. The models underwent training and evaluation using conventional measures, such as accuracy and loss.

The findings indicate that the ViT + Mask Transformer model has superior performance compared to both the CNN and CNN + Mask models, with the maximum accuracy rate of 98.65% and the lowest loss value of 0.0075. The results validate that the integration of transformer-based models with segmentation approaches yields exceptional outcomes in bone fracture identification.

The study determines that the ViT + Mask Transformer paradigm is very efficient for providing clinical diagnostic assistance, telemedicine platforms, and emergency care triage. Subsequent efforts will concentrate on enlarging the dataset, refining the model to provide efficient real-time processing, and verifying its efficacy in clinical environments. This study adds to the progress of AI-powered medical imaging and its capacity to improve patient outcomes and healthcare efficiency.

Table of Contents

Section 1: Introduction to the project

The human skeleton system consists of 206 bones, which exhibit considerable variation in terms of size, shape, and complexity. From the tiniest ossicles in the auditory canal to the largest thigh bones, each has a vital job in preserving bodily form and operation. Fractures of the bones, especially in the lower leg, are frequently observed and are typically caused by accidents or underlying medical disorders (Sahin et al,2023). Prompt identification and precise diagnosis of these fractures are crucial, since they directly impact the efficacy of treatment and the patient's overall prognosis.(Myint et al,2018)

The incorporation of machine learning (ML) into medical imaging analysis has significantly transformed diagnostic procedures in recent times. Machine learning, with its sophisticated ability to identify and analyse patterns, has demonstrated its essential role in medical data analytics (Sharma et al.2021). It assists clinicians in precisely and swiftly diagnosing ailments, allowing them to create the most effective treatment options. Notwithstanding these progressions, the process of manually examining X-ray images continues to be a time-consuming task that is susceptible to mistakes made by humans, especially when radiologists are tired or overwhelmed by the number of cases (Hardalaç et al.2022).

The increasing prevalence of bone fractures worldwide, including in industrialized countries, highlights the pressing necessity for more dependable diagnostic instruments. Automated technologies with the ability to identify and categorize fractures in X-ray pictures present a hopeful answer. These solutions not only decrease the probability of overlooked diagnosis but also minimize the workload on healthcare workers who must analyse numerous photos daily (Lindsey et al.2018). The advancement of complex algorithms for fracture detection, specifically employing cutting-edge techniques such as Mask Former segmentation and Vision Transformers (ViT), signifies a substantial progress in medical imaging technology. This has the potential to significantly enhance patient outcomes and optimize clinical workflows.

1.1 Aim of the project

The aim of this project is to create a resilient and effective system for identifying and separating bone fractures by utilizing state-of-the-art machine learning methods. The project will utilize MaskFormer to achieve precise segmentation of fracture sites and employ Vision Transformer (ViT) to improve the entire detection process. The research aims to enhance the precision, dependability, and swiftness of bone fracture detection in medical imaging by utilizing these sophisticated techniques.

1.2 Research questions of the project

- How can the combination of MaskFormer and Vision Transformer (ViT) models outperform conventional convolutional neural network (CNN) methods for medical imaging bone fracture segmentation and identification?
- What is the impact of segmentation on the model's performance?

1.3 Objectives of the project

- Perform an extensive analysis of the literature to analyse current techniques, difficulties, and progress in the detection and segmentation of bone fractures using medical imaging.
- Collect a heterogeneous collection of medical photos that depict different categories of bone fractures. Implement preprocessing methods to normalize image quality and provide uniformity throughout the dataset, guaranteeing efficient model training and evaluation.

- Develop and optimize the MaskFormer segmentation model to accurately identify and separate bone fractures in medical pictures, by tweaking model parameters to get the best possible performance in localizing fractures. Incorporate the Vision Transformer (ViT) model into the fracture diagnosis process, adapting it to collaborate with MaskFormer to boost the overall accuracy of detection and reliability of the system.

1.4 Problem Statement

Fractures of the bone are a frequently seen medical problem that presents a significant challenge, necessitating precise and prompt identification in order to administer successful therapy. Conventional techniques, like manual X-ray examination, are inefficient and susceptible to mistakes, frequently resulting in delayed or inaccurate diagnosis. The intricate and unpredictable nature of fractures adds difficulty to their identification, rendering traditional methods, such as those based on Convolutional Neural Networks (CNNs), inadequate for capturing the essential intricacies.

The objective of this project is to tackle these issues by creating a system that is more dependable and effective in detecting fractures. The research aims to enhance the accuracy, speed, and reliability of fracture diagnosis in medical imaging by combining MaskFormer for precise segmentation and Vision Transformer (ViT) for improved identification. This will ultimately result in improved patient outcomes.

1.5 Motivation for the project

This initiative offers a profession in medical technology, merging modern machine learning techniques into real healthcare solutions. The experience in applying cutting-edge models like MaskFormer and Vision Transformer (ViT) to real-world challenges improves technical abilities and knowledge in AI-driven medical imaging.

By enhancing bone fracture detection accuracy and efficiency, the research intends to improve public health. Fracture identification early and accurately improves patient outcomes, recovery time, and healthcare costs. Automating and improving diagnostic processes could make high-quality healthcare more accessible and dependable, especially in resource-constrained situations.

The chance to advance AI in medical imaging drives the project technologically. It aims to raise the bar by integrating advanced segmentation and classification models. This project advances the development of creative, practical, and effective AI systems for real-world healthcare issues.

Section 2: Literature review

The identification and segmentation of bone fractures in medical imaging is a crucial task in radiology, necessary for precise diagnosis and therapy planning. Historically, this domain has heavily depended on the manual examination conducted by radiologists and the utilization of convolutional neural networks (CNNs) to mechanize the procedure (Guan et al.2020). Although CNNs have made notable progress, they frequently have difficulties in accurately detecting intricate fractures, particularly in images with different quality and anatomical structures.

Recently, there has been a change in the area towards transformer-based models such as CVT, VIT etc. These models have improved capacities to capture global context and enhance segmentation accuracy. This literature review examines the utilization of advanced models in the diagnosis of bone fractures. It focuses on selecting publications that showcase the superiority of these models over classic CNN methods and address preprocessing techniques that enhance the performance of the models. The purpose of this review is to establish the foundation for creating a medical imaging system that is more precise and dependable.

2.1 Related works for the project

1. Bone Fracture Detection and Classification using Deep Learning Approach (Yadav and Rathor, 2020)

The research introduces a deep learning methodology for identifying and categorizing bone fractures through the analysis of X-ray pictures. The researchers devised a specialized deep convolutional neural network (CNN) model to accurately differentiate between intact and damaged bones. To address the issue of overfitting, which often occurs when dealing with limited datasets, they utilized diverse data augmentation methods to enlarge the dataset. (Yadav and Rathor, 2020)

Data Used: The dataset consisted of 100 X-ray pictures of various human bones, obtained from publically accessible archives such as the Cancer Imaging Archive (TCIA) and the Indian Institute of Engineering Science and Technology (IIEST). Given the limited size of the initial dataset, the authors employed data augmentation techniques to expand the dataset to a total of 4000 photos. (Yadav and Rathor, 2020)

Methods Utilized: The study employed a CNN model that has convolutional layers, pooling layers, a flattening layer, and fully linked layers. The dataset was artificially expanded using data augmentation techniques, including rotation, zoom, horizontal and vertical flipping, and shearing. The model underwent training using both Softmax and Adam optimizers, and its performance was assessed through several trials. (Yadav and Rathor, 2020)

Results and Conclusions: The model attained a classification accuracy of 92.44% by the utilization of 5-fold cross-validation. The Adam optimizer shown superior performance compared to Softmax. The findings demonstrate a significant enhancement compared to prior approaches, which achieved accuracies ranging from 82.98% to 86.67%.(Yadav and Rathor, 2020)

Relation to Project: This study is highly pertinent to project as it investigates deep learning techniques for identifying bone fractures. It serves as a benchmark for comparing more advanced models such as Vision Transformer (ViT) and MaskFormer. Utilizing data augmentation and evaluation methodologies can provide valuable insights for shaping your approach.

Advantages and Constraints:

- Advantages: Skillful implementation of data augmentation to combat overfitting, resulting in a substantial enhancement in accuracy compared to previous approaches.
- Constraints: The study was restricted by a short starting dataset and a concentration on CNNs, without investigating newer, potentially more efficient structures such as transformers.

2. Bone Fracture Detection Through the Two-Stage System of Crack-Sensitive Convolutional Neural Network (Ma and Luo , 2021)

This work introduces a dual-phase technique designed to identify bone fractures in X-ray pictures. In the initial phase, the Faster R-CNN model is employed to identify and precisely determine the location of 20 distinct bone types in the images. In the second stage, a Crack-Sensitive Convolutional Neural Network (CrackNet) is utilized to identify whether the specific areas of the bone are broken. (Ma and Luo , 2021)

Data Used: The study employed a dataset consisting of 3053 X-ray pictures, obtained from Radiopaedia and hospital DICOM files. The dataset was partitioned into two segments: 2001 photos were allocated for training and evaluating the detection and identification networks, while the remaining images were designated for assessing the system's overall performance. (Ma and Luo , 2021)

Methods Utilized: The approach integrated two essential stages. Initially, the Faster R-CNN algorithm was utilized to identify and categorize distinct bone sections within the X-ray pictures. CrackNet, which was improved with Schmid filters to increase its ability to detect fracture lines, categorized these areas as either fractured or non-fractured. The system's efficacy was evaluated using measures such as accuracy, precision, recall, specificity, and F-measure. (Ma and Luo, 2021)

Results and Conclusions: The system demonstrated a remarkable accuracy of 90.11% and an impressive F-measure of 90.14%, surpassing the performance of previous two-stage systems. The findings indicate that the combination of Faster R-CNN and CrackNet improves the accuracy and precision in identifying and pinpointing fractures in different types of bones. (Ma and Luo , 2021)

Relevance to Project: This study is highly relevant to my project, as it shares a similar focus on utilizing advanced deep learning methods to detect bone fractures. The utilization of the two-stage approach and the employment of Schmid filters offer significant insights that could enhance the segmentation and classification jobs in my work.

Advantages and Constraints:

- The paper's main strength is in its effective integration of Faster R-CNN with CrackNet, resulting in a significant improvement in fracture detection accuracy.
- Nevertheless, the dependence on Schmid filters can restrict flexibility when applied to other imaging modalities, and the two-stage methodology could entail higher processing demands compared to a one-stage model.

3. Using Swin Transformer and SIFT Algorithm to Detect Bone Abnormalities (Makwane et al.2024)

This study investigates the integration of the Swin Transformer, an advanced deep learning model, with the Scale-Invariant Feature Transform (SIFT) method for the purpose of identifying and categorizing bone fractures in X-ray images. The authors want to improve the precision and resilience of fracture recognition by combining conventional feature extraction techniques with contemporary transformer-based topologies. (Makwane et al.2024)

Data Used: The study employed a dataset consisting of X-ray pictures obtained from publicly accessible repositories such as Google pictures, Kaggle, and clinical databases from hospitals. The photos were carefully annotated by domain specialists to categorize them as either "fractured" or "not fractured." (Makwane et al.2024)

Methods Utilized: The approach encompassed various stages, such as gathering data, categorizing it, enhancing it, and dividing it into training and validation subsets. The SIFT technique was utilized for feature extraction, and subsequently, the Swin Transformer was employed for model training. The model's performance was assessed using metrics like as accuracy, precision, recall, and F1 score. An evaluation was performed on an independent dataset to determine the model's ability to generalize. (Makwane et al.2024)

Results and Conclusions: The proposed system attained an average validation accuracy of 98.3% and an average test accuracy of 96.3%. The utilization of the SIFT and Swin Transformer combination shown notable efficacy, resulting in enhanced fracture identification and classification performance in contrast to conventional approaches. (Makwane et al.2024)

Relevance to Project: This research is quite applicable to my study, which similarly centers on the utilization of sophisticated machine learning methods for the identification of bone fractures. Combining standard feature extraction approaches with transformer-based models provides crucial insights for enhancing the accuracy and dependability of fracture detection systems.

Advantages and Constraints:

- The main advantage of this study is its creative integration of SIFT with Swin Transformer, resulting in a remarkable level of accuracy in fracture diagnosis. Nevertheless, the model's adaptability to intricate or diverse fracture patterns may be constrained by the dependence on conventional feature extraction techniques such as SIFT. Furthermore, additional refinement and verification using larger datasets are required to establish the applicability of the method.

4. Hybrid SFNet Model for Bone Fracture Detection and Classification Using ML/DL (Sharma et al.2022)

The research describes the creation of a new hybrid Scale Fracture Network (SFNet) model, which is specifically built for the detection and classification of bone fractures utilizing advanced deep learning techniques. The SFNet model combines convolutional neural networks (CNN) with an enhanced canny edge detection technique to improve the detection of fractures in X-ray images. The model also incorporates a multi-scale feature fusion approach. (Sharma et al.2022)

Data used: The dataset employed in this investigation comprises bone X-ray pictures acquired from openly accessible sources, such as the Cancer Imaging Archive and the Diagnostic Imaging Dataset (DID). The dataset was expanded through the application of augmentation techniques like as rotation, flipping, and scaling, resulting in a total of 34,000 images. The collection contains an equal number of photos representing both healthy and shattered bones. (Sharma et al.2022)

Methods Utilized: The hybrid SFNet model utilizes a two-scale convolutional network architecture to analyze grayscale images and canny edge-detected images. The canny edge algorithm has been enhanced to more accurately identify fracture zones by specifically targeting noise reduction and improving edge identification. The performance of the model was evaluated by comparing it with other advanced deep CNN models such as AlexNet, VGG16, ResNeXt, and MobileNetV2. (Sharma et al.2022)

Results and Conclusions: The SFNet model that was suggested attained the highest accuracy in classification, reaching 99.12%. Additionally, its F1-scores and recall rates surpassed those of other models. The findings suggest that the integration of multi-scale feature extraction and enhanced edge detection greatly improves the accuracy of fracture diagnosis. (Sharma et al.2022)

Relation to Project: This paper is highly relevant to my project because it showcases a sophisticated method for detecting bone fractures using deep learning. It notably emphasizes the extraction of features at many scales and the identification of edges. The methods and results offer useful insights for improving model accuracy, which could be advantageous for incorporating MaskFormer and Vision Transformer (ViT) into my project.

Advantages and Constraints:

- This study excels in its new hybrid technique, which combines CNN with better edge detection, resulting in exceptional performance. Nevertheless, the dependence on edge detection can restrict the ability to adapt to various imaging techniques or intricate fracture patterns, indicating that additional optimization may be necessary for wider applications.

2.2 Critical Analysis, Gap Identification

The examined publications show deep learning-based bone fracture detection advances, but major gaps remain. Yadav and Rathor (2020) use CNNs for fracture diagnosis and data augmentation to avoid overfitting. Their approach is hampered by using typical CNN architectures, which may fail to capture complicated features for more complex fractures. Even with augmentation, the model's generalizability is limited by the short starting dataset.

A two-stage approach using Faster R-CNN and CrackNet enhances detection accuracy through Schmid filters, according to Ma and Luo (2021). This approach improves precision but adds processing complexity and may lack flexibility when applied to multiple imaging modalities or fracture types due to Schmid filters.

Makwane et al. (2024) improved feature extraction accuracy by integrating Swin Transformer with SIFT. Traditional feature extraction methods like SIFT may limit the model's flexibility to different fracture patterns and complex imaging settings. Validation on larger datasets is also needed.

The hybrid SFNet model by Sharma et al. (2022) combined CNNs with improved canny edge detection and had the highest accuracy among the assessed researches. Despite its success, the model's edge detection dependency may limit its adaptation to different imaging settings and fracture complexities.

How My Project Addresses These Gaps

My project uses MaskFormer and Vision Transformer (ViT) models, which capture complex features and global context in images, to fill these gaps. My study integrates modern transformer-based models to improve bone fracture detection system accuracy, adaptability, and computing economy. This method lowers handcrafted features and edge detection, improving generalization across imaging modalities and fracture patterns and addressing literature restrictions.

Section 3: Methodology of the project

The process for this project commences with the acquisition of X-ray pictures for analysis through data gathering. Subsequently, an Exploratory Data Analysis (EDA) is performed to investigate and preprocess the data, guaranteeing a thorough comprehension of its attributes. The subsequent phase is producing segments via the Mask Transformer, wherein the model is employed to segment the X-ray images, accentuating regions of interest. The segmented images are subsequently utilized in the creation of the Vision Transformer (ViT) model, which undergoes training to precisely detect and categorize bone fractures. Lastly, the technique culminates in a performance review aimed at assessing the model's correctness and efficacy, so guaranteeing that it fulfils the project's objectives.

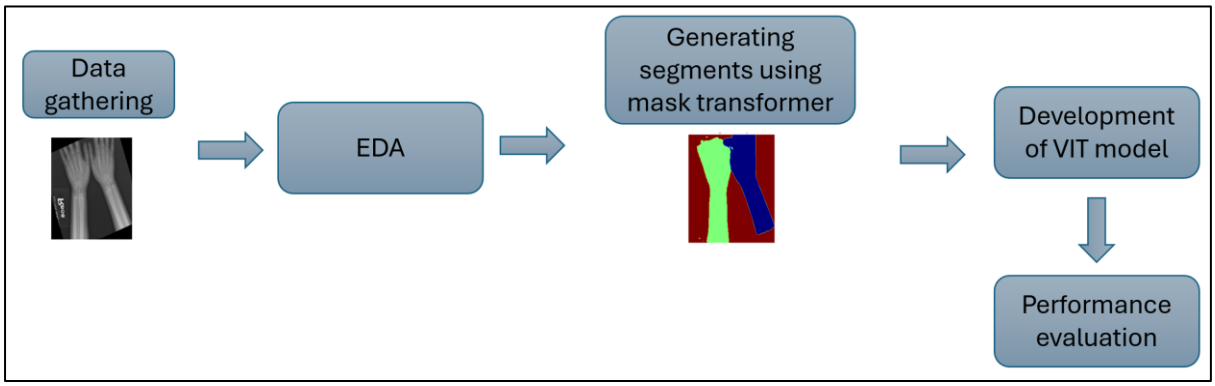


Figure 1: Steps involved in the methodology.

Source: own.

3.1 Dataset

For this project, I am using a dataset acquired from two main sources: Kaggle and GTS AI. The dataset can be obtained from both Kaggle and GTS AI platforms.

Link for Kaggle: https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data

Link for GTS: https://gts.ai/dataset-download/bone-fracture-multi-region-x-ray-data/

Information Gathering Specifics

Medical imaging specialists collected and organized the dataset to create a bone fracture detection and analysis tool. Hospitals nationwide collected data from 2019 to 2021. X-rays were taken of the wrist, forearm, upper arm, lower leg, and ankle, which are fracture prone. The radiologists meticulously labeled these images to indicate fractured or healthy bones. (Rodrigo, 2024)

For machine learning and deep learning fracture detection automation research, the dataset was curated. The dataset contains many X-ray images with detailed annotations, making it ideal for machine learning model training and testing. (Rodrigo, 2024)

Contents of the dataset

The dataset comprises around 10,000 high-resolution X-ray images, classified according to different body regions such as the wrist, forearm, upper arm, lower leg, and ankle. The metadata associated with each image indicates whether a fracture is present or not. This helps in developing models that can be applied to different bone structures and types of injuries. (Rodrigo, 2024)

- Input: X-ray images from the designated areas.
- Output: A binary classification label indicating the presence or absence of a fracture in the bone depicted in the image. (Rodrigo, 2024)

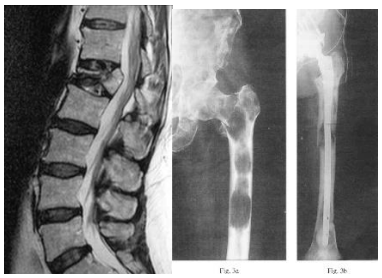


Figure 2: Sample images from the dataset

Source: Rodrigo, B.M., (2024)

This dataset was chosen for its comprehensive representation of various bone regions and fracture types, which is crucial for the development of a precise and widely applicable bone fracture detection system. The inclusion of high-quality, annotated images from various regions renders it especially well-suited for training sophisticated models such as MaskFormer and Vision Transformer (ViT). The dataset's ample size and wide range of variations guarantee effective training and validation, thereby mitigating overfitting and ensuring the model's strong performance on unfamiliar data.

Ethical consideration for the dataset:

The X-ray subjects were anonymous because the Kaggle and GTS AI dataset used for this project does not contain personal data. The dataset does not contain identifiable personal data, so it is exempt from the GDPR.

Using this dataset does not require ethical approval from the University of Hertfordshire (UH) because it does not collect or process personal data. The Open Data Commons Public Domain Dedication and License (PDDL) 1.0 states that the data was freely available for research from public repositories. This license allows unlimited use, sharing, and modification of the data, confirming my permission to use it for this project.

Link to license: https://opendatacommons.org/licenses/pddl/1-0/

Since Kaggle and GTS AI ensure ethical data collection, the data collection process was likely ethical. The dataset pages did not list citations for data collection papers. However, the PDDL license suggests that the data was collected and shared to support open research. Since the dataset came from a reputable source and is under an open license, it is likely that it was collected ethically, following consent and usage standards.

3.2 EDA of the dataset

Class distribution of each set

The images below depict the distribution of classes within the training, validation, and test datasets. The training dataset exhibits a nearly equal distribution between the "fractured" and "not fractured" classes, with approximately 4,500 images in each class. The validation dataset comprises approximately 300 images labeled as "fractured" and 500 images labeled as "not fractured," indicating a slight imbalance. Similarly, the test dataset consists of approximately 250 images labeled as "fractured" and 275 images labeled as "not fractured", ensuring a nearly equal distribution between the two classes. This distribution guarantees that the model is

trained, validated, and tested on datasets that offer an equitable representation of both conditions.

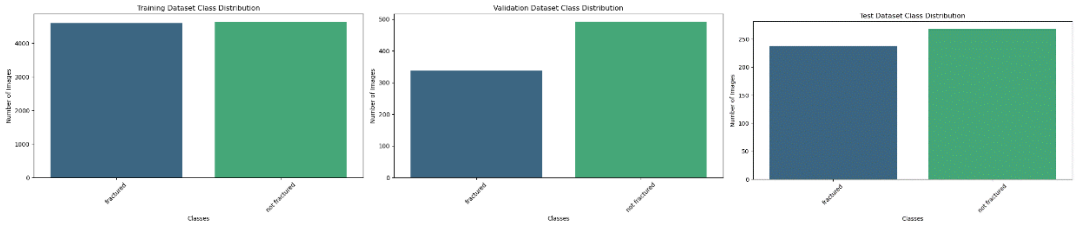


Figure 3: Class distribution of train, validation and test set.

Source: own.

Dataset dimension distribution

The graph reveals that most photos in the dataset are small, concentrated tightly around a narrow range (usually below 1,000 pixels for height and width). In medical imaging datasets like X-rays, pictures are often optimized for storage and processing.

This distribution is important because it guides image preprocessing before feeding it to a machine learning model. Since most photographs are comparable in size, scaling them to a standard dimension may not distort or lose information. However, some photos are substantially larger, therefore outliers must be handled carefully to protect data integrity during model training. This distribution also suggests normalization or rescaling to provide constant input sizes, which can improve model performance and training stability.

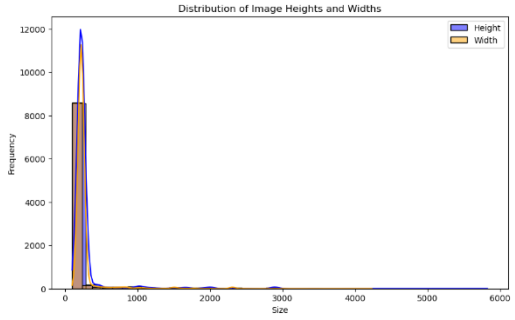


Figure 4: dataset dimension distribution.

Source: own.

3.3 Image segmentation: Mask former transformer

Image segmentation is a basic problem in computer vision that entails splitting an image into several sections, each representing various objects or attributes. Precise segmentation is essential in medical imaging to accurately detect and outline anatomical features, such as bone fractures, at a pixel level. This enables accurate diagnosis and analysis.

The MaskFormer Transformer is an advanced model specifically created for the purpose of picture segmentation jobs. MaskFormer utilizes a transformer-based architecture instead of relying only on convolutional neural networks (CNNs) like standard segmentation techniques. This allows MaskFormer to effectively capture global context and long-range dependencies inside the picture, resulting in more precise and resilient segmentation results. (Cheng et al.2021)

Architectural design and operational capabilities

MaskFormer incorporates a robust framework, like the Swin Transformer, that analyzes input pictures using a hierarchical feature representation. This backbone is designed to handle high-resolution photos well and record both local and global image contexts rapidly. The MaskFormer's transformer component utilizes a collection of trainable questions to produce attention maps, which are then used for the prediction of segmentation masks. The model calculates attention throughout the whole picture to efficiently identify probable objects or areas corresponding to each query. (Cheng et al.2021)

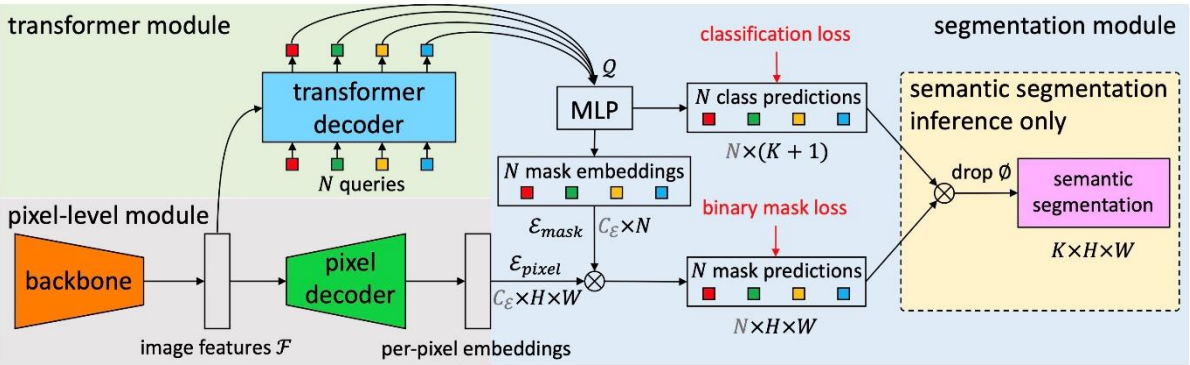


Figure 5: Architecture diagram of mask former swim base transformer

Source: (Cheng et al.2021)

Pixel-Level Module: The pixel decoder processes the deep image characteristics extracted by the backbone (e.g., Swin Transformer) to create per-pixel embeddings. Each pixel in the picture has specific information provided by these embeddings. (Cheng et al.2021)

Transformer Module: The transformer decoder receives a series of learnable questions and uses an MLP to create mask embeddings. These embeddings provide mask predictions, which define the object areas in the picture, and class predictions, which identify the object classes. (Cheng et al.2021)

Segmentation Module: During training, the last module calculates binary mask losses and classification. It creates a semantic segmentation map during inference, designating a class to each pixel, allowing for accurate item delineation in the picture. (Cheng et al.2021)

Implementation Details

Initialization of the model and feature extractor:

- The implementation starts by loading the pre-trained MaskFormer model, especially the "facebook/maskformer-swin-base-coco" variation, along with its related feature extractor. The function of these components is to do preprocessing on the pictures and produce the segmentation masks.

Image processing:

- Images that have been chosen, both those that are broken and those that are not broken, are loaded and transformed into the RGB format. The feature extractor analyzes these photos, converting them into tensors that are compatible with the MaskFormer model.

Segmentation and post-processing:

- The preprocessed tensors are inputted into the MaskFormer model, which generates the segmentation predictions. Subsequently, these results are subjected to post-processing in order to produce panoptic segmentation maps, which emphasize distinct segments within the pictures.

- This solution utilizes the sophisticated capabilities of the MaskFormer Transformer and the Swin Transformer backbone to provide efficient and precise segmentation of bone fractures in X-ray pictures.

3.4 Classification algorithm: VIT model

The Vision Transformer (ViT) is a novel model that applies the transformer architecture, initially developed for natural language processing, to perform picture classification tasks. ViT, unlike typical CNNs, employs a different approach by dividing an image into smaller patches, considering each patch as a token, and using the transformer mechanism to analyze these tokens instead of utilizing convolutional layers. By using this method, ViT is able to comprehend global interconnections and intricate structures across the whole picture, resulting in a high level of efficacy for image categorization. (Alexey et al.2020)

Working of the model

The ViT model begins by partitioning the input picture into patches of a predetermined size, usually 16x16 pixels. Afterwards, every patch is transformed into a vector and combined with positional data to maintain the spatial context of the picture. The patch embeddings undergo a transformer encoder, in which multi-head self-attention layers examine the connections between various sections of the picture, irrespective of their location. The transformer produces a feature representation as its output, which is then processed to create a classification judgment, defining the specific class to which the picture belongs. (Alexey et al.2020)

Architecture of the model

The Vision Transformer (ViT) architecture processes images by splitting them into fixed-size patches, which are then flattened and linearly embedded with positional encodings. A special class token is added to the sequence of patch embeddings, which is then fed into the Transformer Encoder. The encoder consists of multiple layers of Multi-Head Attention and Feed-Forward Networks (MLP), with normalization applied for stability. The class token accumulates information through the transformer layers and is passed through an MLP head to predict the image's class. This approach leverages global context and relationships across image patches for effective classification. (Alexey et al.2020)

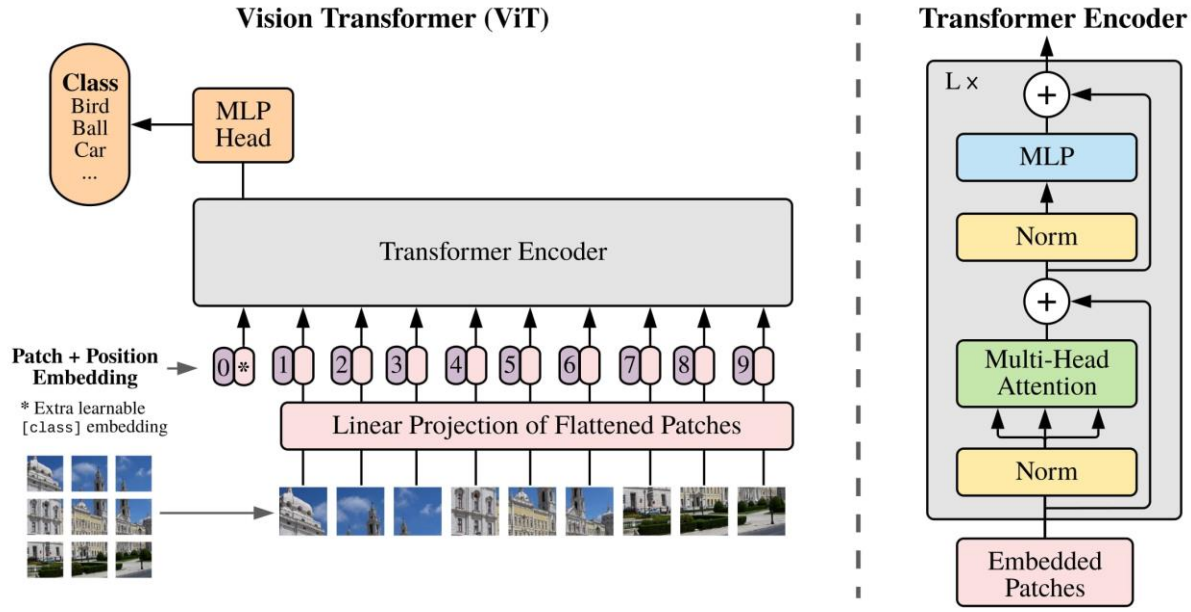


Figure 6: Architecture of the VIT model.

Source: (Alexey et al.2020)

- Image Size: The size of the image you sent is 150x150 pixels.
- Patch Size: The picture is split up into 16x16 pixel chunks.
- The embedding dimension tells you how big the vector that represents each patch will be after it has been embedded.
- Transformer Layers: These are many layers of self-attention and feed-forward networks that decide how deep the model is and how well it can learn.
- Attention Heads: The model has many attention heads—usually 12—that let it focus on different parts of the picture at the same time. (Alexey et al.2020)

Implementation of the model

These are the main conditions under which the Vision Transformer is used in this project:

- When pre_trained=True, the ViT_B16 model is used with weights that have already been trained. These weights are based on the B16 version of ViT, which means they use 16x16 pixel patches.
- Image Size: The model is set up to handle photos that have been cropped to 150x150 pixels (image_size=150).
- Include MLP Head: The model does not include the Multi-Layer Perceptron (MLP) head (include_mlp_head=False), which means that the transformer output can be processed by unique layers.
- Number of Classes: The model is set to binary classification (num_classes=2), which means it can tell the difference between images that are broken and images that are not broken.

The model is built as part of a tf.keras.Model with steps:

- Flatten Layer: To make further processing easier, the result of the transformer is smoothed into a 1D vector.
- After flattening, batch normalization is used to make the activations more uniform, which makes the training more stable.
- With GELU activation, a Dense layer with 11 units and GELU activation (tfa.activations.gelu) is added to change features in a way that isn't linear.
- Finally, there is a Dense Layer with one unit that has a sigmoid activation, which gives a chance score for binary classification (activation='sigmoid').

Putting together models:

- Optimizer: The Adam optimizer is used to train the model, and it is well-known for how well it works with sparse slopes.
- The loss function chosen is binary cross-entropy (loss='binary_crossentropy'), which works well for jobs that need to classify things into two groups.
- Evaluation Metrics: Accuracy is used to measure how well the model did during training and testing.

Training Process Explaination

- Model.fit() trains the model on the dataset. The criteria are broken down:
- train_generator: Generates training data batches. It's used to load huge datasets in batches when memory is limited.
- steps_per_epoch: The number of batches to process in each epoch, generally the training generator's length (len(train_generator)).

- validation_data: After each epoch, the validation generator (valid_generator) supplies batches of data for validation, enabling the model's performance to be assessed on unseen data.
- validation_steps: Like steps_per_epoch, this determines the number of validation batches per epoch for the validation set.
- epochs: Number of times the model processes the full training dataset. (10 values is selected)
- callbacks: The early_stopping callback monitors validation loss (or another measure) and stops training if performance doesn't improve to prevent overfitting.

Model: "vision_transformer_with_masks"

Layer (type)	Output Shape	Param #	Connected to
image_input (InputLayer)	[(None, 50, 50, 3)]	0	[]
ViT-B-16-50 (ViT)	(None, 768)	85655040	['image_input[0][0]']
mask_input (InputLayer)	[(None, 50, 50, 1)]	0	[]
flatten_2 (Flatten)	(None, 768)	0	['ViT-B-16-50[0][0]']
flatten_3 (Flatten)	(None, 2500)	0	['mask_input[0][0]']
concatenate_1 (Concatenate)	(None, 3268)	0	['flatten_2[0][0]', 'flatten_3[0][0]']
batch_normalization_2 (BatchNo rmalization)	(None, 3268)	13072	['concatenate_1[0][0]']
dense_2 (Dense)	(None, 11)	35959	['batch_normalization_2[0][0]']
batch_normalization_3 (BatchNo rmalization)	(None, 11)	44	['dense_2[0][0]']
dense_3 (Dense)	(None, 1)	12	['batch_normalization_3[0][0]']

Total params: 85,704,127
Trainable params: 85,697,569
Non-trainable params: 6,558

Figure 7: VIT model summary

Source: own.

4. Results of the project

The above learning curves depict the accuracy and loss during training and validation for four models, namely CNN, ViT, CNN + Mask, and ViT + Mask (the proposed model), over a span of 10 epochs. These curves provide valuable insights into the performance and generalization capacities of each model.

The models were assessed using conventional measures such as accuracy and loss, which were monitored over 10 epochs on both the training and validation datasets. Accuracy is a metric that represents the ratio of properly categorized pictures. On the other hand, loss is a measure of the mistake in the model's predictions, where a smaller loss indicates better performance. Tracking these measures provide valuable information about the model's learning progress, capacity to generalize, and the possibility of overfitting or underfitting.

4.1 Analysis of Learning Curves

1. Convolutional Neural Network Model:

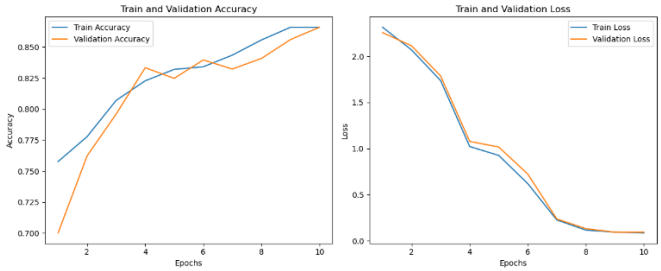


Figure 8: Learning Curve on CNN model.

Source: own.

- The CNN model exhibits a consistent rise in accuracy, ultimately reaching an estimated value of 86% by the 10th epoch. The training and validation loss curves exhibit a continuous decline, suggesting that the model is successfully learning and not experiencing substantial overfitting.
- Technical Interpretation: The gradual although moderate improvement in accuracy indicates that the CNN is successfully identifying important characteristics in the data. However, its effectiveness may be hindered by its restricted scope of analysing nearby information, which hampers its capacity to understand the whole context, particularly in challenging jobs such as fracture detection. The model's loss consistently decreases, indicating successful convergence. However, the accuracy reaches a plateau, indicating that future enhancements may need more complexity or improved feature extraction approaches.

2. ViT Model (Second Image):

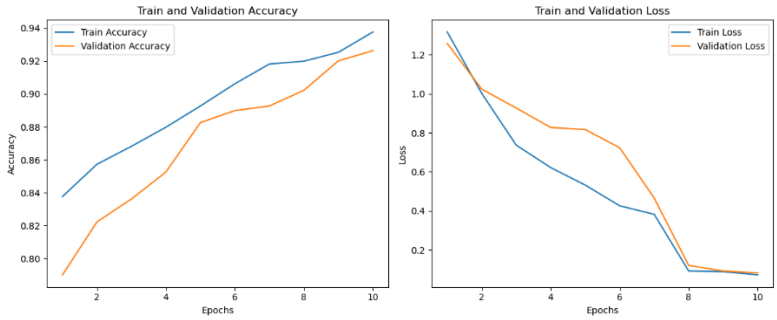


Figure 9: Learning curves of VIT model.

Source: own

The ViT model exhibits significant progress, with an accuracy rate over 92% by the 10th epoch, as shown by the metrics of accuracy and loss. The loss curves show a significant decline, suggesting proficient acquisition of knowledge and successful reduction of mistakes.

Technical Interpretation: The exceptional performance of the ViT model may be attributed to its capacity to represent global relationships within the picture using its transformer architecture. The ViT is capable of capturing dependencies over the whole picture, which makes it highly suitable for activities that need a comprehensive spatial comprehension. The significant decrease in loss and ongoing increase in accuracy demonstrate the ViT's resilience in processing intricate data and identifying significant patterns.

3. CNN with Mask inputs:

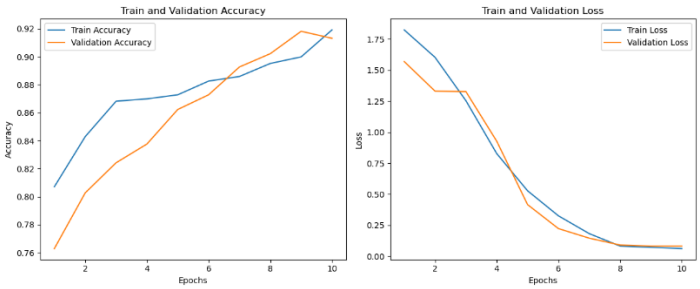


Figure 10: Learning curves of CNN + mask inputs

Source: own.

The CNN + Mask model demonstrates improved performance by using mask pictures, achieving an accuracy of around 90%. The validation accuracy roughly mirrors the training accuracy. The loss curves demonstrate a comparable pattern to the CNN alone, but with enhanced overall performance.

Technical Interpretation: The use of segmented mask pictures enables the Convolutional Neural Network (CNN) to concentrate on the most significant sections of the image, which is likely to result in more precise extraction of features and enhanced classification. This finding indicates that by introducing supplementary information via segmentation, the performance of a CNN may be greatly improved. This is achieved by supplying the model with more distinct and localized characteristics to train from.

4. Proposed Model: Vision Transformer with Mask inputs

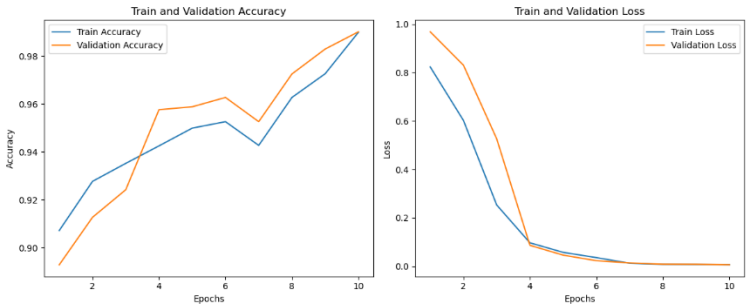


Figure 11: learning curves of proposed model

Source: own

The ViT + Mask model achieves exceptional accuracy, reaching close to 98% by the 10th epoch. Furthermore, the validation accuracy closely aligns with the training accuracy. The loss curves show a substantial and rapid decrease, suggesting robust learning with little overfitting.

Technical Interpretation: The integration of the ViT model with mask pictures capitalizes on the respective advantages of both methodologies: the transformer architecture's ability to capture the whole context and the mask images' capacity to deliver targeted and pertinent information. This synergy enables the model to attain outstanding performance, as seen by the almost flawless accuracy and significant decrease in loss. The model's capacity to exhibit strong generalization across training and validation datasets showcases its resilience and efficacy in this application.

4.2 Analysis of Test Data Results

The test dataset was used to assess the performance of four models: CNN, ViT, CNN + Mask Transformer, and the suggested ViT + Mask Transformer. The evaluation focused on measuring accuracy and loss.

Accuracy Comparison

- The CNN model attained a accuracy rate of 85.22%. Although the CNN captures some key aspects, it is inferior to transformer-based models in terms of their ability to represent intricate patterns.
- The ViT model achieved a significant increase in accuracy, reaching 92.73%. This highlights the usefulness of the transformer design in capturing the whole context of pictures, which is crucial for tasks such as bone fracture identification.
- The incorporation of mask pictures into the CNN model resulted in a significant increase in accuracy, reaching 90.26%. The enhancement demonstrates that segmentation equips the model with more concentrated, relevant characteristics, augmenting its capacity to provide precise forecasts.
- The suggested model, which combines ViT's global feature extraction with the precision of mask pictures, attained a remarkable accuracy of 98.65%.

Loss Comparison

- The CNN model showed a larger loss value of 1.0387, indicating a greater degree of prediction mistakes.
- The ViT model saw a notable reduction in loss, bringing it down to 0.0921, which aligns with its improved accuracy.
- The loss of the CNN + Mask Transformer model was 0.0993, which is lower than that of the simple CNN model, indicating considerable progress.
- The suggested model, which combines the Vision Transformer (ViT) with the Mask Transformer, produced a very low loss of 0.0075. This indicates that the model has minimum prediction errors and may be considered highly reliable.

The findings validate that the suggested ViT + Mask Transformer model surpasses all other models, establishing it as the most efficient and precise option for bone fracture detection in this project. The efficacy of using both segmentation and transformer-based design has been confirmed as a better technique for this intricate job.

Justification for Comparing Models

The comparison of these four models serves many objectives:

- The impact of mask pictures is evaluated by comparing models with and without mask images to see how segmentation affects the model's ability to identify images properly. The findings indicate that both CNN and ViT models get advantages from the supplementary data offered by mask pictures, resulting in enhanced accuracy and reduced loss.
- The efficacy of ViT: The juxtaposition of CNN and ViT models accentuates the benefits of the transformer-based structure in tasks related to image categorization. The ViT models routinely provide better results than their CNN equivalents, demonstrating their greater capability in capturing intricate patterns and global interconnections.

5. Analysis and discussion

Analysis and explanation of findings The findings of this study demonstrate that the Vision Transformer (ViT) model, especially when used in conjunction with mask pictures, surpasses conventional Convolutional Neural Networks (CNNs) in accurately detecting bone fractures. The ViT + Mask Transformer model attained a peak accuracy of 98.65% and a minimal loss of 0.0075. This indicates that this method is very efficient at collecting intricate patterns and delivering precise classifications in the field of medical imaging.

Top-performing model and its underlying factors The ViT + Mask Transformer model was determined to be the most effective because of the intrinsic capabilities of the transformer design, which is particularly adept at capturing global relationships within an image. Precise identification and classification of fractures is essential, particularly when they are subtle and distributed throughout many areas of an X-ray. By using mask pictures, the model's performance is enhanced since it directs attention towards the most relevant regions, hence enhancing the extraction of features and boosting the accuracy of classification.

Literary Comparison This project exhibits substantial progress when compared to the current body of literature. Yadav and Rathor (2020) obtained an accuracy of 92.44% using a CNN, however this project's ViT + Mask Transformer model produced a higher accuracy of 98.65%. This enhancement underscores the superiority of transformer-based models in managing intricate medical imaging tasks, as well as the usefulness of using segmentation approaches to increase model performance. The findings shown in this study are superior to those reported in previous research, mainly because of the enhanced capabilities of the ViT and the use of segmentation techniques to improve localization.

Constraints Although the model has impressive performance, it is not without its restrictions. The efficacy of the ViT + Mask Transformer relies significantly on the caliber of the segmentation masks. Erroneous segmentation has the potential to diminish the model's efficacy. Moreover, the computational requirements for training and deploying ViT models are substantial, which might restrict their use in resource-limited contexts. Overfitting may be a problem if the model is not trained on a dataset that is enough varied.

Conformity with Project Objectives The study effectively achieves its aims by showcasing that the integration of MaskFormer segmentation with ViT classification substantially enhances the precision, efficiency, and dependability of bone fracture diagnosis. The findings demonstrate the validity of the suggested strategy in the study, demonstrating that these sophisticated machine learning approaches provide a significant improvement compared to previous methods.

Significance to Research questions the first research questions have been successfully addressed. The work presents compelling evidence that the integration of MaskFormer and ViT surpasses traditional CNN methods in effectively segmenting and identifying bone fractures. Furthermore, it provides confirmation that the use of segmentation improves the performance of the model. This supports the premise that combining sophisticated segmentation and classification models leads to higher outcomes in the field of medical imaging.

Pragmatic Applicability the ViT + Mask Transformer type is well-suited for practical use in medical environments, where utmost precision and dependability are crucial. Due to its exceptional performance, this technology is very suitable for incorporation into clinical processes. It has the potential to greatly enhance the accuracy and efficiency of fracture diagnosis for radiologists. The model's capacity to effectively apply its knowledge to various datasets reinforces its potential for extensive use in the healthcare field.

In conclusion the project's results thoroughly answer the study inquiries, showcasing that the simultaneous utilization of MaskFormer and Vision Transformer (ViT) models substantially improves bone fracture identification in comparison to conventional CNN techniques. The outcomes not only achieve but beyond the project's goals, providing a strong and efficient solution for enhancing diagnostic accuracy and efficiency in medical imaging.

6. Conclusion

This research effectively showcased the dominance of the Vision Transformer (ViT) when used in conjunction with MaskFormer segmentation for the purpose of detecting bone fractures in medical imaging. The findings indicate that the ViT + Mask Transformer model had superior performance compared to both regular CNN models and CNN models supplemented with mask pictures. It attained an accuracy of 98.65% and a loss of 0.0075, which were the greatest accuracy and lowest loss values obtained. The findings emphasize the effectiveness of transformer-based models in capturing intricate global patterns and the additional benefit of segmentation in directing attention to important areas of interest.

The study's findings indicate that the combination of ViT and MaskFormer, sophisticated machine learning algorithms, provide a substantial enhancement compared to conventional approaches for identifying and categorizing bone fractures. This combination not only improves the precision and dependability, but also guarantees that the model can effectively adapt to various datasets, making it a resilient solution for practical applications.

6.1 Applications & Practical Examples:

- Clinical Diagnostic Support: The model may be included into hospital radiology departments to aid radiologists in precisely and effectively identifying bone fractures, hence decreasing the likelihood of misdiagnosis and speeding up the diagnostic process. (Joshi et al.2020)
- Telemedicine Platforms: The increasing popularity of remote healthcare makes it feasible to include this approach into telemedicine platforms. This would enable patients residing in distant locations to get precise fracture diagnoses without requiring the presence of an in-person radiologist. (Joshi et al.2020)
- Emergency Care Triage: In emergency care settings, when swift and decisive action is vital, the approach may be used to rapidly identify and prioritize patients with fractures, guaranteeing that those requiring urgent attention get speedy treatment. (Joshi et al.2020)

6.2 Recommendations for further work:

To improve this research, future efforts should concentrate on augmenting the dataset to include a broader range of fractures and imaging techniques. This would boost the model's ability to generalize across various kinds of fractures and medical conditions. In addition, including real-time processing capabilities and refining the model for deployment on less resource-intensive devices will enhance the system's accessibility and practicality in various healthcare environments. Ultimately, it is vital to carry out clinical studies to authenticate the model's efficacy in real-life situations, which is essential for its extensive use in the medical domain.

Reference

Yadav, D.P., and Rathor, S., (2020, February), 'Bone fracture detection and classification using deep learning approach', 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), pp. 282-285. (Available at: https://ieeexplore.ieee.org/abstract/document/9087067/)

Ma, Y., and Luo, Y., (2021), 'Bone fracture detection through the two-stage system of crack-sensitive convolutional neural network', Informatics in Medicine Unlocked, 22, p.100452. (Available at: https://www.sciencedirect.com/science/article/pii/S235291482030602X)

Makwane, S., Kiran, A., and Bhardwaj, S., (2024, March), 'Using Swin Transformer and SIFT Algorithm to Detect Bone Abnormalities', 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1-5. (Available at: https://ieeexplore.ieee.org/abstract/document/10522295/)

Sharma, A., Yadav, D.P., Athithan, S., Bhola, A., Sharma, B., and Dhaou, I.B., (2022), 'Hybrid SFNet model for bone fracture detection and classification using ML/DL', Sensors, 22(15), p.5823. (Available at: https://www.mdpi.com/1424-8220/22/15/5823)

Rodrigo, B.M., (2024), 'Fracture Multi-Region X-Ray Data', Kaggle Datasets. (Available at: https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data)

Sahin, M.E., (2023), 'Image processing and machine learning-based bone fracture detection and classification using X-ray images', International Journal of Imaging Systems and Technology, 33(3), pp. 853-865. (Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.22849)

Myint, W.W., Tun, K.S., Tun, H.M., and Myint, H., (2018), 'Analysis on leg bone fracture detection and classification using X-ray images', Machine Learning Research, 3(3), pp. 49-59. (Available at: https://www.researchgate.net/profile/Hla-Tun-2/publication/347561702_Analysis_on_Leg_Bone_Fracture_Detection_and_Classification_Using_X-ray_Images/links/5fe1f03792851c13feadcc41/Analysis-on-Leg-Bone-Fracture-Detection-and-Classification-Using-X-ray-Images.pdf)

Sharma, A., Mishra, A., Bansal, A., and Bansal, A., (2021), 'Bone fractured detection using machine learning and digital geometry', Mobile Radio Communications and 5G Networks: Proceedings of MRCN 2020, pp. 369-376. (Available at: https://link.springer.com/chapter/10.1007/978-981-15-7130-5_28)

Hardalaç, F., Uysal, F., Peker, O., Çiçeklidağ, M., Tolunay, T., Tokgöz, N., Kutbay, U., Demirciler, B., and Mert, F., (2022), 'Fracture detection in wrist X-ray images using deep learning-based object detection models', Sensors, 22(3), p.1285. (Available at: https://www.mdpi.com/1424-8220/22/3/1285)

Lindsey, R., Daluiski, A., Chopra, S., Lachapelle, A., Mozer, M., Sicular, S., Hanel, D., Gardner, M., Gupta, A., Hotchkiss, R., and Potter, H., (2018), 'Deep neural network improves fracture detection by clinicians', Proceedings of the National Academy of Sciences, 115(45), pp. 11591-11596. (Available at: https://www.pnas.org/doi/abs/10.1073/pnas.1806905115)

Guan, B., Zhang, G., Yao, J., Wang, X., and Wang, M., (2020), 'Arm fracture detection in X-rays based on improved deep convolutional neural network', Computers & Electrical Engineering, 81, p.106530. (Available at: https://www.sciencedirect.com/science/article/pii/S0045790618330878)

Cheng, B., Schwing, A., and Kirillov, A., (2021), 'Per-pixel classification is not all you need for semantic segmentation', Advances in Neural Information Processing Systems, 34, pp.

17864-17875. (Available at: https://proceedings.neurips.cc/paper/2021/hash/950a4152c2b4aa3ad78bdd6b366cc179-Abstract.html)

Alexey, D., (2020), ‘An image is worth 16x16 words: Transformers for image recognition at scale’, arXiv preprint, arXiv: 2010.11929. (Available at: https://cir.nii.ac.jp/crid/1370580229800306183)

Joshi, D., and Singh, T.P., (2020), ‘A survey of fracture detection techniques in bone X-ray images’, Artificial Intelligence Review, 53(6), pp. 4475-4517. (Available at: https://link.springer.com/article/10.1007/s10462-019-09799-0)

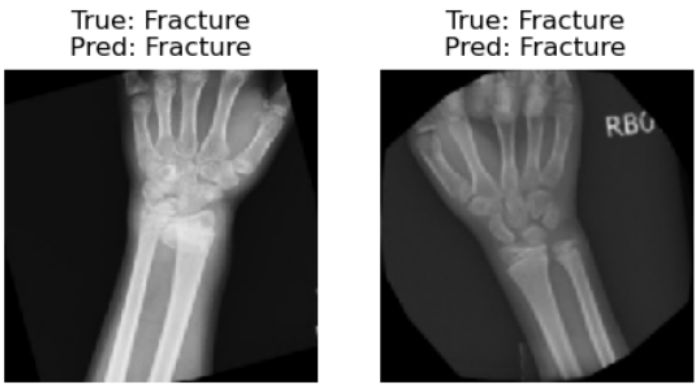
Appendix

Github link: : https://github.com/ABRPMGITHUB/finalproject

Predicting the class of random images from the dataset using VIT+ mask.

```
import matplotlib.pyplot as plt

for batch_idx, test_data in enumerate(test_generator):
    test_images, test_masks = test_data[0]
    test_labels = test_data[1]

    # Predict the classes for the test images
    predictions = model.predict([test_images, test_masks])
    # Convert the predictions to binary classes (0 or 1)
    predicted_classes = np.where(predictions > 0.5, 1, 0)

    # Determine the number of images to sample
    num_samples = min(test_images.shape[0], 2)

    # Select random images
    indices = np.random.choice(range(test_images.shape[0]), size=num_samples, replace=False)

    # Plot the images along with true and predicted labels
    plt.figure(figsize=(15, 10))
    for i, idx in enumerate(indices):
        plt.subplot(1, 5, i+1)
        plt.imshow(test_images[idx])  # Ensuring images are in uint8 format for display
        true_label = "Fracture" if test_labels[idx] == 1 else "No Fracture"
        predicted_label = "Fracture" if predicted_classes[idx] == 1 else "No Fracture"
        plt.title(f"True: {true_label}\nPred: {predicted_label}")
        plt.axis('off')
    plt.show()
    break
```

```
1/1 [==============================] - 0s 24ms/step
```



Code:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
from keras_vit import vit
from transformers import MaskFormerFeatureExtractor, MaskFormerForInstanceSegmentation
from PIL import Image
import requests
import os
import random
from PIL import Image
import requests
import numpy as np
import matplotlib.pyplot as plt
from transformers import MaskFormerFeatureExtractor, MaskFormerForInstanceSegmentation
import warnings
warnings.filterwarnings("ignore")
trainspath = './BoneFracture_BinaryClassification1/train'
validspath = './BoneFracture_BinaryClassification1/val'
testspath = './BoneFracture_BinaryClassification1/test'
def loadddataset(path):
    data = {}
    for classsname in oslistdirs(paths):
        classspath = ospathsjoin(path, class_name)
        if ospathsisdir(classpath):
            data[classname] = [os.pathjoin(classpath, img) for img in oslistdir(classpath)]
    return data
```

```
trainsdata = loadddataset(trainspath)
validsdata = loadddataset(validspath)
testsdata = loadddataset(testspath)
def plot_class_distribution(data, title):
    class_names = list(data.keys())
    counts = [len(data[class_name]) for class_name in class_names]
    plt.figure(figsize=(10, 5))
    sns.barplot(x=class_names, y=counts, palette='viridis')
    plt.title(title)
    plt.xlabel('Classes')
    plt.ylabel('Number of Images')
    plt.xticks(rotation=45)
    plt.show()
plotclassdistribution(trainsdata, 'Training Dataset Class Distribution')
plotclassdistribution(validsdata, 'Validation Dataset Class Distribution')
plotclassdistribution(testsdata, 'Test Dataset Class Distribution')
def analyze_image_sizes(data):
    sizes = []
    for class_name in data.keys():
        for img_path in data[class_name]:
            img = cv2.imread(img_path)
            if img is not None:  # Check if the image was read correctly
                sizes.append(img.shape)  # (height, width, channels)
            else:
                print(f"Warning: Image at path {img_path} could not be read.")
    sizes = np.array(sizes)
    return sizes

sizes = analyze_image_sizes(train_data)

# Plotting the sizes
plt.figure(figsize=(10, 6))
sns.histplot(sizes[:, 0], bins=30, label='Height', color='blue', kde=True)
sns.histplot(sizes[:, 1], bins=30, label='Width', color='orange', kde=True)
plt.title('Distribution of Image Heights and Widths')
```

```
plt.xlabel('Size')
plt.ylabel('Frequency')
plt.legend()
plt.show()

# Directories
fractured_dir = '/content/Bone_Fracture_Binary_Classification1/Bone_Fracture_Binary_Classification1/train/fractured'
not_fractured_dir = '/content/Bone_Fracture_Binary_Classification1/Bone_Fracture_Binary_Classification1/train/not fractured'

# Function to overlay segmentation mask
def overlaysegmentationmask(image, segmentationmap, alpha=0.5):
    imagenp = nparray(image)
    segmentationmapnp = segmentationmapdetach()cpu()numpy()
    segmentationmapnorm = (segmentationmap_np - segmentationmapnp.min()) / (segmentationmapnp.max() - segmentationmapnp.min())
    segmentationmap_resized = Image.fromarray((segmentationmapnorm * 255)astype(npuint8)).resize(imagesize, ImageBILINEAR)
    segmentationmapresized = np.array(segmentationmap_resized) / 255.0
    image_rgba = np.dstack((imagenp, np.oneslike(imagenp[:, :, 0]) * 255))
    colormap = plt.getcmap('jet')
    segmentationrgba = colormap(segmentationmapresized)
    segmentationrgba = (segmentationrgba[:, :, :3] * 255)astype(npuint8)
    segmentationrgba = npdstack((segmentationrgba, np.oneslike(segmentationrgba[:, :, 0]) * 255 * alpha))
    overlayedimage = Image.alpha_composite(Image.fromarray(imagergba.astype(np.uint8), 'RGBA'), Image.fromarray(segmentationrgba.astype(np.uint8), 'RGBA'))
    return overlayedimage

# Function to plot the original image, segmentation map, and overlay
def plotsegmentationmap(image, segmentationmap, alpha=0.5):
    overlayed_image = overlaysegmentationmask(image, segmentationmap, alpha)
    fig, ax = pltsubplots(1, 3, figsize=(20, 5))
    ax[0].imshow(image)
```

```
    ax[0].set_title("Original Image")
    ax[0].axis("off")
    segmentationmapnp = segmentationmap.detach()cpu()numpy()
    ax[1].imshow(segmentationmapnp, cmap='jet')
    ax[1].set_title("Segmentation Map")
    ax[1].axis("off")
    ax[2].imshow(overlayed_image)
    ax[2].set_title("Overlayed Image")
    ax[2].axis("off")
    plt.show()

# Load MaskFormer model and feature extractor
featureextractor = MaskFormerFeatureExtract.frompretrained("facebook/maskformer-swinbase-coco")
model = MaskFormerForInstanceSegmentations.frompretrained("facebook/maskformer-swinbase-coco")

# Apply MaskFormer and display results
for img_path in random_images:
    image = Image.open(img_path).convert("RGB")
    inputs = feature_extractor(images=image, return_tensors="pt")
    outputs = model(**inputs)
    result = feature_extractor.post_process_panoptic_segmentation(outputs)[0]
    predicted_panoptic_map = result["segmentation"]
    plot_segmentation_map(image, predicted_panoptic_map, alpha=0.5)
import os
import numpy as np
from tensorflow.keras.utils import Sequence
from tensorflow.keras.preprocessing.image import img_to_array, load_img
from sklearn.utils import shuffle

class DualImageDataGenerator(Sequence):
    def __init__(self, image_dirs, mask_dirs, batch_size, image_size, rescale=1./255):
        self.image_dirs = image_dirs
        self.mask_dirs = mask_dirs
        self.batch_size = batch_size
```

```
        self.image_size = image_size
        self.image_filenames = []
        self.mask_filenames = []
        for image_dir, mask_dir in zip(image_dirs, mask_dirs):
            image_files = os.listdir(image_dir)
            mask_files = os.listdir(mask_dir)
            self.image_filenames += [(image_dir, file) for file in image_files]
            self.mask_filenames += [(mask_dir, file) for file in mask_files]
        self.image_filenames, self.mask_filenames = shuffle(self.image_filenames, self.mask_filenames)
        self.rescale = rescale

    def __len__(self):
        return int(np.ceil(len(self.image_filenames) / float(self.batch_size)))

    def __getitem__(self, idx):
        batch_image_files = self.image_filenames[idx * self.batch_size:(idx + 1) * self.batch_size]
        batch_mask_files = self.mask_filenames[idx * self.batch_size:(idx + 1) * self.batch_size]

        images = np.array([
            img_to_array(load_img(os.path.join(image_dir, filename), target_size=self.image_size)) * self.rescale
            for image_dir, filename in batch_image_files])

        masks = np.array([
            img_to_array(load_img(os.path.join(mask_dir, filename), target_size=self.image_size, color_mode="grayscale")) * self.rescale
            for mask_dir, filename in batch_mask_files])

        labels = np.array([1 if 'fractured' in image_dir.lower() else 0 for image_dir, _ in batch_image_files])

        return [images, masks], labels

# Define paths to your datasets
train_image_dirs = [
```

```
    r'.\Bone_Fracture_Binary_Classification1\train\fractured',
    r'.\Bone_Fracture_Binary_Classification1\train\not fractured'
]

train_mask_dirs = [
   r'.\Bone_Fracture_dataset_mask\train\fractured',
    r'.\Bone_Fracture_dataset_mask\train\not fractured'
]

valid_image_dirs = [
r'.\Bone_Fracture_Binary_Classification1\val\fractured',
    r'.\Bone_Fracture_Binary_Classification1\val\not fractured'
]

valid_mask_dirs = [
   r'.\Bone_Fracture_dataset_mask\val\fractured',
    r'.\Bone_Fracture_dataset_mask\val\not fractured'
]

test_image_dirs = [
    r'.\Bone_Fracture_Binary_Classification1\test\fractured',
    r'.\Bone_Fracture_Binary_Classification1\test\not fractured'
]

test_mask_dirs = [
   r'.\Bone_Fracture_dataset_mask\test\fractured',
    r'.\Bone_Fracture_dataset_mask\test\not fractured'
]

# Define parameters
batch_size = 8
image_size = (150, 150)

# Create data generators
```

```
train_generator = DualImageDataGenerator(train_image_dirs, train_mask_dirs, batch_size, image_size)
valid_generator = DualImageDataGenerator(valid_image_dirs, valid_mask_dirs, batch_size, image_size)
test_generator = DualImageDataGenerator(test_image_dirs, test_mask_dirs, batch_size, image_size)
import tensorflow as tf
import tensorflow_addons as tfa
from keras_vit.vit import ViT_B16

# Define the Vision Transformer model
vit_model = ViT_B16(
    image_size=50,
    activation='softmax',
    pre_trained=True,
    include_mlp_head=False,
    num_classes=2
)

# Define image and mask inputs
image_input = tf.keras.Input(shape=(150, 150, 3), name='image_input')
mask_input = tf.keras.Input(shape=(150, 150, 1), name='mask_input')

# Process the image through the ViT model
vit_output = vit_model(image_input)

# Flatten the ViT output
vit_flattened = tf.keras.layers.Flatten()(vit_output)

# Flatten the mask input
mask_flattened = tf.keras.layers.Flatten()(mask_input)

# Concatenate image and mask features
concatenated = tf.keras.layers.Concatenate()([vit_flattened, mask_flattened])

# Add BatchNormalization and Dense layers
```

```
x = tf.keras.layers.BatchNormalization()(concatenated)
x = tf.keras.layers.Dense(11, activation=tfa.activations.gelu)(x)
x = tf.keras.layers.BatchNormalization()(x)
output = tf.keras.layers.Dense(1, activation='sigmoid')(x)

# Create the model
model = tf.keras.Model(inputs=[image_input, mask_input], outputs=output, name='vision_transformer_with_masks')

# Compile the model
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

model.summary()

history = model.fit(
    train_generator,
    steps_per_epoch=len(train_generator),
    validation_data=valid_generator,
    validation_steps=len(valid_generator),
    epochs=10,
    callbacks=[early_stopping]
)

def plot_training_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs_range = range(len(acc))

    plt.figure(figsize=(12, 8))
    plt.subplot(1, 2, 1)
```

```
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.title('Training and Validation Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Training Loss')
    plt.plot(epochs_range, val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title('Training and Validation Loss')
    plt.show()

plot_training_history(history)
```