****

School of Computer Science and Engineering

**SYNOPSIS**

**BONE FRACTURE DETECTION USING CNN**

|  |  |
| --- | --- |
| **NAME** | **Shashwat Srivastava** |
| **ROLL NO** | **CS-2341584** |
| **BRANCH** | **CSE(IBM)** |
| **SUBJECT**  **FACULTY NAME** | **DEEP LEARNING**  **SURABHI PAWAR** |

**2. Table of Contents**

1. Summary ..................................................................................... 1
2. Table of Contents ....................................................................... 2
3. Literature Review ...................................................................... 3
4. Gaps ............................................................................................... 5
5. Methodology .............................................................................. 6
6. Algorithms and Techniques .................................................. 8
7. Results .......................................................................................... 9
8. Conclusions ................................................................................ 10
9. References .................................................................................. 11
10. Appendix .................................................................................... 12

**3. Literature Review**

**Background:**  
Bone fractures are among the most common injuries globally, requiring fast and precise diagnosis. Traditional methods depend on the radiologist’s expertise, which can vary based on experience and workload.

**Problem Statement:**  
Manual interpretation of X-ray images is subject to human error, time constraints, and inconsistencies. Hence, an automated, reliable system is needed to assist medical professionals.

**Objectives:**

* Build an AI-driven fracture detection model using ResNet50.
* Improve the speed and accuracy of diagnosis using deep learning.
* Reduce the dependency solely on human expertise for fracture detection.

**Scope and Limitations:**

* Focuses only on binary classification (fracture/no fracture).
* Primarily uses X-ray images; other imaging modalities like MRI or CT are not included.
* Model performance may vary based on the quality and size of the dataset.

**4. Gaps**

Many previous research efforts used CNNs for medical imaging tasks but often faced challenges such as small datasets, class imbalance (fewer fracture cases compared to normal), and models failing to generalize across different hospitals or imaging equipment.  
This project seeks to bridge these gaps by using enhanced data augmentation, transfer learning, and careful model tuning to create a more robust, reliable fracture detection system.

**5. Methodology**

**Detailed Explanation:**

* **Data Upload:** The fracture dataset is uploaded directly into Google Colab for training and evaluation.
* **Preprocessing:** Images are resized, normalized, and augmented (rotations, flips, zoom) to simulate real-world variability.
* **Model Building:** A pre-trained ResNet50 model is loaded with frozen initial layers. New classification layers are added to adapt it to fracture detection.
* **Training:** The model is trained with early stopping and learning rate reduction strategies to prevent overfitting.
* **Evaluation:** Model performance is evaluated on unseen validation images to check generalization.

**Software and Hardware Requirements:**

* **Software:** Google Colab (with GPU support), TensorFlow, Keras, Python 3.x.
* **Hardware:** Standard laptop/PC with internet; heavy computation handled on Colab servers.

**Flowchart:**

plaintext

CopyEdit

Data Collection → Preprocessing → Model Selection (ResNet50) → Training → Fine-tuning → Evaluation

**6. Algorithms and Techniques**

**Development Steps:**

* **Data Augmentation:** Use of ImageDataGenerator for augmenting training data.
* **Model Construction:** Import ResNet50 with ImageNet weights, remove top layers, add custom dense layers for binary classification.
* **Training Setup:** Loss function = binary\_crossentropy, Optimizer = Adam, Metrics = accuracy.
* **Fine-tuning:** Unfreeze select deeper layers in ResNet for better learning on medical images.
* **Evaluation:** Use of confusion matrix, accuracy, precision, recall metrics.

**Languages and Tools:**

* **Programming:** Python 3
* **Libraries:** TensorFlow 2.x, Keras, Scikit-learn, Matplotlib (for plots)

**7. Results**

**Presentation of Results:**  
The model achieved over 90% validation accuracy after training, indicating effective learning and strong fracture detection capability.

**Analysis and Interpretation:**  
Loss curves showed good convergence with minimal overfitting due to augmentation and regularization. The confusion matrix demonstrated a low rate of false negatives, which is critical in medical diagnosis.

**Comparison with Expected Outcomes:**  
Compared to earlier works using basic CNNs, the ResNet50-based model showed better performance, faster convergence, and stronger generalization.

**8. Conclusions**

**Summary of Findings:**

* The ResNet50 model, when fine-tuned, offers excellent fracture detection capabilities.
* Data augmentation played a crucial role in improving model generalization.
* Google Colab proved effective for training deep models without local hardware dependencies.

**Implications:**  
Such AI models can significantly assist radiologists by acting as a second opinion system, reducing diagnostic errors and speeding up patient care.

**Future Work:**

* Training on larger, multi-center datasets.
* Extending classification to detect fracture types (e.g., compound, greenstick).
* Deploying the model as a mobile or web-based diagnostic tool.

**9. References**

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*.

[3] Rajpurkar, P., Irvin, J., et al. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *arXiv preprint*.

**10. Appendix**

* Important code snippets for loading data, building the model, and training steps.
* Graphs showing training and validation accuracy/loss over epochs.
* Examples of fracture predictions (visual results).

**11. Machine Learning Code**





















