**CLASSIFICATION OF LEUKOCYTES USING DEEP LEARNING**

**Alternative title:**

Neural Network Classification of White Blood Cell using Transfer Learning Techniques

**Aim:**

The aim of this paper is to develop an automatic white blood cell classification system using deep learning techniques. Specifically, the objective is to create a model capable of accurately classifying white blood cells (leucocytes) to aid in the diagnosis of various hematologic diseases, including anaemia and leukaemia. Unlike previous approaches that relied on transfer learning with models trained on the ImageNet dataset, this study proposes a novel deep learning model that is built from scratch, tailored to handle the unique characteristics of the white blood cell dataset used in this application.

**Abstract:**

The classification of white blood cells or leucocytes has become highly crucial for the diagnosis of anaemia, leukaemia and many other hematologic diseases. The density of WBCs in our blood stream provides a glimpse into the state of our immune system and any potential risks we might be facing. The main objective of this paper is to develop an automatic white blood cell classification system using deep learning. Most of the models proposed for this application so far has used transfer learning by fine tuning the ‘State of the art” models like ResNet50. But all these models were trained on ImageNet dataset which is completely different from the dataset used in this application. So in this study, we have proposed a deep learning model for the white blood cells classification.

**Existing System:**

In the existing system, the classification of white blood cells (leucocytes) for the diagnosis of hematologic diseases, such as anaemia and leukaemia, is achieved using Convolutional Neural Networks (CNNs).While this approach has shown promising results in various computer vision tasks, it comes with certain disadvantages when applied to the specific domain of white blood cell classification.

**Problem Definition:**

Limited Adaptation to Domain-Specific Features: CNN models dataset may not adequately adapt to the unique features and patterns present in white blood cell images. Fine-tuning a complex CNN architecture on a relatively small and domain-specific dataset can lead to overfitting. The limited size of the white blood cell dataset might not be sufficient to generalize well to diverse and unseen samples, affecting the model's accuracy and robustness.

**Proposed System:**

In this paper, we propose a novel white blood cell classification system using the ResNet50 architecture, which is a deep Convolutional Neural Network known for its state-of-the-art performance in various computer vision tasks. However, instead of employing transfer learning from pre-trained models on unrelated datasets like ImageNet, we fine-tune the ResNet50 specifically on our white blood cell dataset. This approach aims to leverage the advantages of the ResNet50 architecture while addressing the limitations observed in the existing system.

**Advantage:**

Fine-tuning ResNet50 on the relatively small white blood cell dataset can help mitigate over fitting issues compared to training from scratch. The pre-trained weights of ResNet50 serve as a useful starting point, capturing general features and allowing the model to focus on learning domain-specific features without losing generalization capability.

**Algorithm:**

**Transfer Learning (ResNet50):**

ResNet50 is a deep Convolutional Neural Network (CNN) architecture developed by Microsoft Research in 2015. It is part of the ResNet family (Residual Networks) and represents a significant breakthrough in image recognition tasks. Key Innovation: Skip Connections (Residual Blocks): ResNet50 introduced the concept of skip connections or shortcuts, which allow the network to learn residual mappings. These shortcuts enable the model to skip one or more layers, preventing the degradation of accuracy with increasing depth. Architecture Overview: ResNet50 consists of 50 layers, organized into blocks. Each block has several convolutional layers and identity shortcuts. The model architecture is divided into five stages, each containing a different number of blocks with varying numbers of filters. Deep Identity Mapping: ResNet50 uses the idea of identity mappings, where the output of a layer is added to the input, allowing the network to focus on learning the residual (difference) between the two. This enables the model to efficiently learn and propagate gradients through very deep networks, making training more stable. Pre-Trained Weights: ResNet50 is commonly pre-trained on large-scale image datasets like ImageNet, capturing generic features from a wide range of images. Fine-tuning the pre-trained ResNet50 on specific datasets, such as our white blood cell dataset, leverages the learned knowledge for improved performance.

**Modules:**

* Dataset Collection
* Data collection and preprocessing.
* Model Selection and Setup.
* Splitting the dataset.
* Fine tuning the ResNet50.
* Hyperparameter tuning.
* Evaluation.

**Dataset Collection:**

Dataset collection from Kaggle https://www.kaggle.com/datasets/brikwerk/bccd-white-blood-cell

**Data Collection and Preprocessing:**

Gather a large dataset of white blood cell images, annotated with their respective classes (e.g., neutrophils, lymphocytes, monocytes, eosinophils). Perform data preprocessing, including resizing images to a consistent resolution, normalization, and data augmentation to increase the dataset's diversity.

**Model Selection and Setup:**

Choose the ResNet50 architecture as the base model for white blood cell classification. Load the pre-trained ResNet50 weights (trained on ImageNet) to initialize the model.

**Splitting the Dataset:**

Divide the dataset into training, validation, and testing sets. The training set will be used for training the model, the validation set for hyper parameter tuning, and the testing set to evaluate the final model's performance.

**Fine-tuning ResNet50:**

Remove the top layers of ResNet50, which are designed for ImageNet classification, and replace them with new layers for white blood cell classification.

Freeze the weights of the early layers to retain the general feature learning and fine-tune the deeper layers to adapt to white blood cell characteristics.

Train the model on the training set using appropriate loss functions (e.g., categorical cross-entropy) and optimizers (e.g., Adam).

**Hyper parameter Tuning:**

Conduct hyper parameter tuning on the validation set to find the best learning rate, batch size, and other relevant hyper parameters to optimize the model's performance.

**Evaluation:**

Evaluate the fine-tuned model on the testing set to assess its accuracy, precision, recall, and other performance metrics.

Analyze the confusion matrix and visualize any misclassifications to identify potential areas for improvement.

**Hardware Requirements:**

* Hard Disk : 500GB and Above
* RAM : 4GB and Above
* Processor : I3 and Above

**Software Requirements:**

* Operating System : Windows 10 (64 bit)
* Software : Python
* Tools : Anaconda or Google colab

**Conclusion:**

The proposed model can be further improved by training on more data and using large models. Model suffered bias when trying to regularize overfitting .This can be avoided if we train very deep models. But large volume of data is required to develop such models. Thus in future, more accurate model can be proposed to avoid labor intensive manual white blood cell classification.

**Architecture Diagram:**

Data

Collection

Data pre-process

Transfer Learning (ResNet50)

Prediction

 Result