

Recognition of Leukemia Sub-types Using Transfer Learning and Extraction of Distinguishable Features Using an Effective Machine Learning Approach

Tahsen Islam Sajon
Department of CSE, RUET
Rajshahi, Bangladesh
sajon.tahsen@gmail.com

Maria Chowdhury
Department of CSE, RUET
Rajshahi, Bangladesh
mariachowdhury831@gmail.com

Azmain Yakin Srizon
Department of CSE, RUET
Rajshahi, Bangladesh
azmainsrizon@gmail.com

Md. Farukuzzaman Faruk
Department of CSE, RUET
Rajshahi, Bangladesh
faarukuzzaman@gmail.com

S. M. Mahedy Hasan
Department of CSE, RUET
Rajshahi, Bangladesh
mahedy@cse.ruet.ac.bd

Abu Sayeed
Department of CSE, RUET
Rajshahi, Bangladesh
abusayeed.cse@gmail.com

A. F. M. Minhazur Rahman
Department of CSE, RUET
Rajshahi, Bangladesh
m.r.surov@gmail.com

Abstract—Although extensive research has been conducted on leukemia, the disease still accounts for more than 350,000 fatalities annually. Automated Leukemia diagnosis may alter the situation because actions can be taken immediately; as a result, accurate detection of Leukemia has been a subject of interest for researchers. As statistics grow and expand, the need for precise leukemia identification continues to increase. In this study, we investigated a dataset of leukemia that used the WHO classification scheme. We developed a modified DenseNet201 design that achieved an overall accuracy of 99.69% without relying on data augmentation. Additionally, we identified and validated key features for leukemia classification by utilizing three feature extraction approaches (i.e., hu moments, haralick texture and parameter-free threshold adjacency statistics) and several machine learning classifiers (i.e., Gaussian Process, Support Vector Machine, K-Nearest Neighbor or KNN, Extra Trees Classifier, and Logistic regression) that outperformed earlier feature extraction-based techniques.

Index Terms—Leukemia Sub-types Classification, Transfer Learning, Hu Moments, Haralick Texture, Threshold Adjacency Statistics, Gaussian Process Classifier

I. INTRODUCTION

Leukemia is a fatal blood malignancy characterized by abnormal and uncontrolled leukocyte proliferation. The generation of such a vast number of faulty leukemia cells affects the blood and restricts the ability of the bone marrow to manufacture platelets and erythrocytes [1], [2]. Inadequate levels of healthy blood cells released into the bloodstream have negative effects on blood coagulation, the ability of the immune system to fight infections, and the amount of oxygen delivered to the organs [2]. Additionally, these cancerous cells can harm other organs, including the liver, kidney, spleen, brain, and so forth, through the bloodstream, causing the emergence of other fatal cancers. [1]. The Surveillance, Epidemiology, and End Results (SEER) Program predicts 60,650 new cases of leukemia in the United States in 2022 [3]. Acute Lymphoblastic Leukemia, or ALL, accounts for 75 percent of

all leukemia cases in children, making it the most widespread type of cancer in this age group [4]. It is found in adults as well. In general, adults account for 4 out of every 10 instances of ALL [5]. The beginning of the treatment process and, consequently, the patient's survival depend on an early and prompt diagnosis of ALL [1]. Currently, microscopic analysis of the peripheral blood smear (PBS) is the most significant diagnostic technique for initial ALL screening. However, it might be difficult to distinguish between normal cells and young leukemic blasts under the microscope as the two cells are identical morphologically. As a result, findings from the manual examination of the slides are frequently erratic and subjective. Therefore, we require automated diagnosis systems for ALL screenings that are fast, reliable, and affordable [5]. Additionally, if we're able to identify and distinguish the fundamental aspects of this condition, a more successful course of treatment may be developed. By harnessing the features of blood smear images, we can design more effective drugs. Once diagnosed, the disease can be cured with the right medication.

Numerous studies have employed various machine and deep learning-based methods to address the issue of automated leukemia identification using blood smear images. Hegde et al. [6] combined the Support Vector Machine (SVM) classifier with an Artificial Neural Network (ANN) to classify normal and leukemia cells, obtaining overall 98.8% classification accuracy. Abdeldaim et al. [7] extracted and normalized shape, texture, and color features. After that, the KNN classifier produced the best result, with an accuracy of 96.01%. Das et al. used a modified ResNet architecture to extract features. Finally, they used SVM, Logistic Regression, and Random Forest for ALL classification, where they gained 96.15% accuracy for all three approaches[8]. Clinton Jr et al. proposed a neural network with depthwise separable convolutions (Xception) to detect ALL and achieved 99% and 91% accuracy

on the training and testing sets based on the images from Cancer Imaging Archive [5]. Khandekar et al. implemented the Object Detection algorithm You Only Look Once (YOLOv4) for ALL identification and classification attaining an 95.57% mAP (Mean Average Precision) for the ALL-IDB1 dataset and 98.57% mAP for the C-NMC-2019 dataset [9]. Ghaderzadeh et al. presented a Convolutional Neural Network (CNN)-based model to distinguish between cases of ALL and hematogones and to ultimately identify the ALL subtypes. They tested 10 well-known CNN architectures for feature extraction and out of these, DenseNet201 obtained the best accuracy of 99.85%. However, their research performed poorly on EfficientNet, achieving 28.22% accuracy [1].

Upon reviewing the related literature, we found that the vast majority of them used the French–American–British (FAB) classification system to categorize ALL cancer cases. However, hematologists and oncologists have recently argued that the World Health Organization (WHO) methodology is preferable to FAB, as the WHO classification has more precise definitions for the subgroups and can be used to more accurately identify the various forms of leukemia. We also observed that the dataset used in this study has been the subject of very little research [1]. Furthermore, the majority of the aforementioned research employed data augmentation in the preprocessing stages, which can have adverse effects on the classifier.

This paper proposes two classification approaches that are based on machine learning and deep learning algorithms. Using publicly available ALL datasets we developed and optimized three transfer learning-based CNN models that achieved a high level of accuracy compared to previous works. Moreover, to find and evaluate the important features of PBS images in ALL classification, we extracted various features and employed several ML classifiers, which achieved similar results, proving the efficacy of the extracted features in leukemia classification.

II. MATERIALS AND METHODS

A. Dataset Description

We analyzed a publicly available ALL dataset that is categorized according to WHO [10]. The bone marrow laboratory at Taleqani Hospital, Tehran created the dataset images. The precise classification of these cells was carried out by an expert. It contained 3256 PBS images from 89 patients suspected of having ALL, having 25 healthy people with a benign assessment (hematogones) and 64 patients who had a confirmed diagnosis of one of the subtypes of ALL. The dataset consisted of four classes: Benign, Early Pre-B, Pre-B, and Pro-B. Non-leukemic cells made up the first category, whereas ALL cases made up the remaining ones [1].

B. Transfer Learning

When performing classification tasks with a small dataset, transfer learning is an effective strategy for overcoming overfitting complications and delivering notable results. Transfer learning is the practice of applying weights that have already

been learnt to resolve problems that are entirely different yet nevertheless related. For instance, there are 1,000 distinct groups in the ImageNet dataset. Some of these groupings are quite similar to one another, while others are wholly distinct. The ImageNet challenge weights will therefore be a good place to start for any recognition that involves similar or dissimilar groupings.

In this paper, we have used three transfer learning-based CNN architectures- DenseNet201, EfficientNetB6, and Xception. Huang et al. presented DenseNet, which is renowned for doing exceptionally well with object recognition benchmark datasets like CIFAR-100 and ImageNet. The DenseNet architecture employs a straightforward connectivity pattern that links all layers directly with each other in a sequential manner, enhancing the flow of information across the network’s layers [11]. The acronym for the EfficientNet family of CNNs is EfficientNetB0-EfficientNetBX, where X represents the network’s parameter count; the higher the number, the larger the network’s parameter count. Compared to earlier systems of comparable complexity, smaller EfficientNet conducts categorization more accurately [12]. The Xception architecture is a straightforward arrangement of residually connected depthwise separable convolution layers. Because of this, the architecture is incredibly simple to define and change [13].

C. Utilized Feature Extraction Techniques

Machine learning classifiers rely on selective image feature extraction. The feature selection techniques considered in this research are Hu Moments, Haralick Texture and Parameter-Free Threshold Adjacency Statistics. These specific feature extractors were considered as they have the capacity to gather pertinent information about the shape, texture, and structure of objects in an image.

1) *Hu Moments*: A key component in image classification is the image shape feature. Effective and efficient shape descriptors are the core elements of the image representation. Moments are shape descriptors commonly used to characterize image shapes. For a 2D image I of size $M \times N$, where $f(x, y)$ denotes the gray level at pixel (x, y) , the raw moment of order $(p + q)$ is:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (1)$$

In practical applications, the raw moment is typically replaced by the central moments. It is defined as:

$$\delta_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (2)$$

$$\bar{x} = m_{10}/m_{00}, \bar{y} = m_{01}/m_{00} \quad (3)$$

Hu derived the Hu Moment Invariants (HMI) by applying the algebraic invariant theory to the normalized central moment, and generated 7 rotation and translation invariant moments that are used to describe the shape information [14].

TABLE I: Haralick features used for classification

Haralick Features	
Angular Second Moment	Info. Measures of Correlation 2
Contrast	Inverse Difference Moment
Correlation	Sum Average
Difference Entropy	Sum Entropy
Difference Variance	Sum of Squares: Variance
Entropy	Sum Variance
Info. Measures of Correlation 1	-

2) *Haralick Texture*: Haralick is a texture feature descriptor that is used extensively in image classification due to its simplicity and straightforward interpretations. Haralick texture features are statistical constructs designed to highlight particular texture characteristics. These features include 14 statistics obtained from the Grey Level Co-occurrence Matrix (GLCM) [15]. In our research, only the first 13 features were utilized. The last (F14) feature was not considered due to computational instability. Table I states the proposed Haralick features for describing texture in PBS images. The computation can be divided into two steps: building the co-occurrence matrices and then calculating the 13 texture features based on them.

3) *Parameter-Free Threshold Adjacency Statistics*: Threshold adjacency statistics is a straightforward and quick morphological metric that was introduced [16] for classifying cell phenotype images. We chose the Parameter-Free Threshold Adjacency Statistics (PFTAS), the parameter-free variant of TAS, as PBS images and these images have certain similarities. In multiple-threshold binarized images, PFTAS works on the premise of accumulating pixels in the histogram bins based on the quantity of white neighbors. Three distinct threshold ranges are used to binarize the original image: $[\mu + \sigma, \mu - \sigma]$, $[\mu - \sigma, 255]$, and $[\mu, 255]$, with μ being an Otsu-defined threshold and σ being the standard deviation of the pixels above the threshold. A normalized histogram of pixels with i (i ranging from 0 to 8) white pixels as neighbors is calculated for each binarized image. For each of the three RGB channels, the three histograms are combined to create a 27-dimensional feature vector, resulting in an 81-dimensional feature vector. Finally, a 162-dimensional feature vector is created by concatenating this vector with its bitwise negated counterpart. Threshold adjacency statistics do not require individual cell cropping of images. In comparison to other frequently used statistics, they can be computed an order of magnitude faster and still produce classification accuracy that is equivalent or superior.

D. Utilized Machine Learning Classifiers

In this section, the five classifiers—Gaussian Process, Support Vector Machine, K-Nearest Neighbor, Extra Trees Classifier and Logistic Regression—have been discussed. The five classifiers were selected due to their wide use and established status in machine learning. Logistic Regression is simple, flexible and computationally efficient. SVM is a powerful algorithm that can handle both linear and non-linear data.

KNN is simple to implement and offers high generalization performance, while Gaussian Process is a probabilistic model that is useful for modeling complex and uncertain relationships between variables. Finally, Extra Trees Classifier is robust, combining multiple decision trees. These classifiers provide a diverse set of methods for solving a variety of classification problems, making them a good choice for comprehensive comparison and evaluation.

1) *Logistic Regression*: The classification function is generated from a class, and the model is a single multinomial logistic regression model with a single estimator. In a particular manner, logistic regression typically identifies the location of the boundaries between the classes as well as how far away from the boundaries the class probabilities are. Based on the values of the input variables, it provides the binomial result, which indicates the likelihood that an event will occur or not (in terms of 0 and 1). It generates more accurate, precise predictions [17].

2) *Support Vector Machine*: SVMs in machine learning look at the data used for regression and classification analyses using supervised learning models and associated learning methods. By implicitly transforming their inputs into high-dimensional feature spaces, SVMs can successfully do non-linear classification in addition to linear classification. The kernel trick is the name of this tactic. It essentially draws lines between the classes. In order to reduce classification error, the margins are designed to have the minimum possible distance between them and the classes [18]. If the proper kernel function can be identified, it can manage complex functions in addition to structured and semi-structured data. The generalization approach used by SVM lowers the risk of overfitting. Furthermore, it is scalable with high-dimensional data [17].

3) *K-Nearest Neighbor*: The sample data point that is provided to the KNN Algorithm is classified using a database that comprises data points sorted into various classes. The algorithm locates the k closest neighbors in the training dataset from the query instance for categorizing a new dataset. The algorithm will then forecast the query instance using the consensus of the k nearest neighbors [19]. KNN is referred to as non-parametric since it makes no assumptions about the distribution of the underlying data [17].

4) *Gaussian Process*: A posterior predictive distribution is computed for fresh data using the Gaussian process, a kernel-based fully Bayesian regression procedure. The Gaussian process is a multivariate Gaussian extension that, thanks to its analytical features, may be used to model groups of real-valued variables. A multivariate Gaussian distribution-based sub-collection of random variables is the basis of a stochastic process known as a Gaussian process. The Gaussian process makes use of the kernel as a covariance function to model distinct distributions in terms of mean square derivatives for a range of different types of data. Since it enables evaluating prediction uncertainty owing to errors in parameter estimates as well as intrinsic noise in the problem, the Gaussian process is an appealing model for using in regression problems [20].

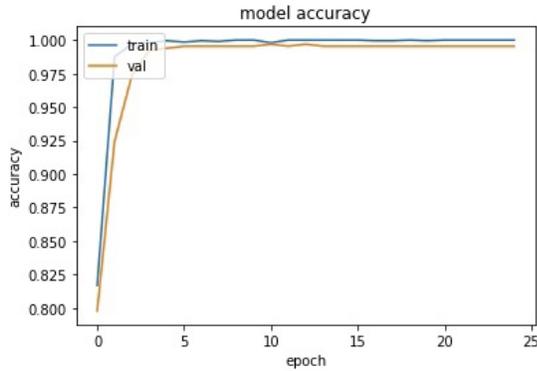


Fig. 1: Training and validation accuracy of proposed DenseNet201 architecture

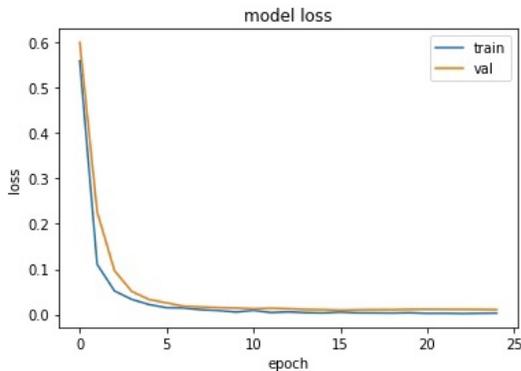


Fig. 2: Training and validation loss of proposed DenseNet201 architecture

5) *Extra Trees Classifier*: The Extra Trees classifier is a method that creates a collection of unpruned decision trees using the conventional top-down approach. The process includes heavily randomizing the selection of both the attribute and the cut-point during node splitting. In the worst case, it generates completely random trees that are unrelated to the output values in the training sample. This is different from other tree-based ensemble algorithms in two ways: the trees are constructed using the whole training sample and nodes are split by randomly choosing cut points. The projections from all the trees are combined using a majority vote to get the final projection [21].

III. EXPERIMENTAL ANALYSIS

A. Preprocessing

Since the CNN is a competent tool for separating valuable information from raw photos, a rigorous preprocessing of the images was omitted when feeding them to the network. However, some preprocessing were required. The supplied images were in various aspects, therefore they were transformed to $224 \times 224 \times 3$. These reshaped images were also used for the

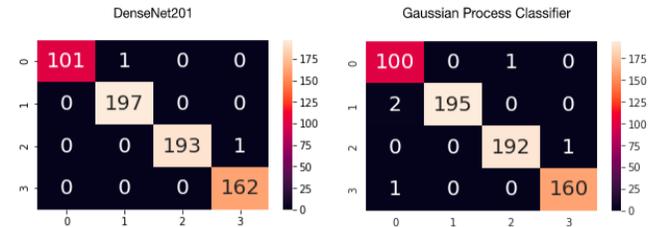


Fig. 3: Model evaluation confusion matrix for test data

ML approach. In this case, the colormap of the images was then converted from RGB to Grayscale. No data augmentation was performed in either approach.

B. Design of Experiment

1) *Transfer Learning*: With a batch size of 24, we trained the model for 25 epochs since after that, the validation loss stayed essentially constant for the remaining epochs. For optimizing the error function, we utilized Adam optimizer [22] using a 0.00001 learning rate. Next, for the error or loss function, a categorical cross-entropy method was used. The dropout technique was used to circumvent overfitting.

2) *Machine Learning*: Three feature extraction methods were used to retrieve a total of 113 features for each image. By eliminating the mean and scaling to unit variance, the features were normalized. These features and the corresponding labels were used to train the classifiers.

C. Result Analysis

After the preprocessing steps, the dataset [10] was partitioned into train, validation, and test sets for our transfer learning approach. 60% data was set aside for training, while 20% and rest 20% was used for validation and testing respectively. Later, the proposed architectures—DenseNet201, EfficientNetB6, and Xception were employed. Here, the transfer learning-based CNN models utilized the original images in the dataset. The training and validation accuracy of our design is shown in Figure-1, and the training and validation loss is shown in Figure-2. The performance of each subtype of leukemia is reported in Table II, including accuracy, precision, recall, f1-score, and support. For the Leukemia dataset[10], our suggested design generated a best overall accuracy of 99.69% using the DenseNet201 architecture. For the ML classifiers, 80% of the data was considered for training and 20% for testing purposes. The segmented images in the dataset [10] were used in this approach. After the preprocessing steps, the data was fed into the three feature extraction algorithms: Hu Moments, Haralick Texture, and PFTAS. For each image in the train and test datasets, these algorithms produced a total of 113 features, which were then applied to train and evaluate the classifiers. Here, we selected five classifiers: Support Vector Machine, Gaussian Process, Extra Trees Classifier, K-Nearest Neighbor and Logistic regression. Gaussian Process obtained a best overall accuracy of 99.23%. Table III summarizes the outcomes that were attained. In order to achieve the best

TABLE II: Class-wise evaluation metrics (accuracy, precision, recall, f1-score, and support) for each class using transfer learning with a scale of 1.00 for 100% and 0.00 for 0%

Classes	Class-wise Accuracy	Class-wise Precision	Class-wise Recall	Class-wise F1-Score	Class-wise Support
Proposed Modified DenseNet201					
Benign	0.990	1.00	0.99	1.00	102
Early	1.000	0.99	1.00	1.00	197
Pre-B	0.995	1.00	0.99	1.00	194
Pro-B	1.000	0.99	1.00	1.00	162
Proposed Modified EfficientNetB6					
Benign	0.990	0.98	0.99	0.99	102
Early	0.995	0.99	0.99	0.99	197
Pre-B	0.989	1.00	0.99	0.99	194
Pro-B	0.994	0.99	0.99	0.99	162
Proposed Modified Xception					
Benign	0.990	0.97	0.99	0.98	102
Early	0.995	0.99	0.99	0.99	197
Pre-B	0.985	1.00	0.99	0.99	194
Pro-B	0.994	0.99	0.99	0.99	162

TABLE III: Class-wise evaluation metrics (accuracy, precision, recall, f1-score, and support) for each class using ML classifiers with a scale of 1.00 for 100% and 0.00 for 0%

Classes	Class-wise Accuracy	Class-wise Precision	Class-wise Recall	Class-wise F1-Score	Class-wise Support
Proposed Gaussian Process					
Benign	0.990	0.97	0.99	0.98	101
Early	0.989	1.00	0.99	0.99	197
Pre-B	0.995	0.99	0.99	0.99	193
Pro-B	0.994	0.99	0.99	0.99	161
Proposed Support Vector Machine					
Benign	1.000	0.95	1.00	0.98	101
Early	0.985	1.00	0.98	0.99	197
Pre-B	0.989	0.99	0.99	0.99	193
Pro-B	0.981	0.99	0.98	0.98	161
Proposed Extra Trees Classifier					
Benign	0.950	0.88	0.95	0.91	101
Early	0.944	0.97	0.94	0.96	197
Pre-B	0.979	0.99	0.98	0.98	193
Pro-B	0.994	0.99	0.99	0.99	161
Proposed K-Nearest Neighbor					
Benign	0.891	0.91	0.89	0.90	101
Early	0.959	0.95	0.96	0.95	197
Pre-B	0.979	0.98	0.98	0.98	193
Pro-B	0.994	0.99	0.99	0.99	161
Proposed Logistic regression					
Benign	0.911	0.88	0.91	0.90	101
Early	0.959	0.96	0.96	0.96	197
Pre-B	0.984	0.99	0.98	0.99	193
Pro-B	0.981	0.99	0.98	0.99	161

results, the parameter values for SVM were tuned using the well-known Grid Search method. Figure 3 represents the confusion matrices from the proposed DenseNet201 model and the Gaussian Process classifier. The comparison between our suggested techniques and earlier efforts on the same dataset is shown in Table IV in terms of overall accuracy. It can be

seen that even though [1] performed better with DenseNet201, their other models did not do so well. This may be accredited to a lack of optimization which we addressed in our work and were able to achieve a steady performance throughout both our methods. A comparison between previous Leukemia classification using ML classifiers and our research is shown

TABLE IV: Comparison of notable previous works with our proposal on the same dataset

Approaches	Augmentation?	Overall Accuracy
DenseNet201 [1]	Yes	99.85%
VGG-16 [1]	Yes	98.01%
Xception [1]	Yes	96.70%
MobileNetV3 [1]	Yes	50.15%
EfficientNet [1]	Yes	28.22%
Proposed Modified DenseNet201	No	99.69%
Proposed Gaussian Process Classifier	No	99.23%

TABLE V: Comparison between proposed classifier and existing work incorporating ML classifiers

Dataset Used	Classifiers	Overall Accuracy
Private Dataset	SVN with ANN [6]	98.80%
ALL-IDB2	KNN [7]	96.01%
ALL-IDB2	SVM, Random Forest, Logistic Regression [8]	96.15%
ALL dataset	Proposed Gaussian Process Classifier	99.23%

in Table V. Our suggested methods have, as can be observed, markedly surpassed the earlier efforts.

IV. CONCLUSION

In this study, we examined a dataset of leukemia subtypes. Among the most prevalent cancers in the globe, leukemia claims many lives every year. Here, we devised a method for precisely predicting Leukemia subtypes utilizing CNN architectures and the transfer learning principle. Through optimization, our models were able to consistently achieve a high level of accuracy surpassing earlier research. Furthermore, in contrast to the previous work on the dataset [1], these results were obtained without the aid of data augmentation. CNNs, however, are often perceived as a “black box” in that we are unable to evaluate a trained model to determine which input features were most significant. Consequently, we attempted to pinpoint the crucial elements in the classification of leukemia using feature extraction techniques. The effectiveness of our acquired features was demonstrated by ML classifiers that performed similar to the CNN models and outperformed earlier feature extraction-based methods in terms of accuracy. Our findings, we trust, can aid in the field of Leukemia diagnosis. Our goal in the future is to conduct research with bigger datasets, discover other feature extraction techniques and design efficient models for leukemia segmentation.

REFERENCES

[1] M. Ghaderzadeh, M. Aria, A. Hosseini, F. Asadi, D. Bashash, and H. Abolghasemi, “A fast and efficient cnn model for b-all diagnosis and its subtypes classification using peripheral blood smear images,” *International Journal of Intelligent Systems*, vol. 37, no. 8, pp. 5113–5133, 2022.

[2] G. Atteia, A. A. Alhussan, and N. A. Samee, “Bo-allcnn: Bayesian-based optimized cnn for acute lymphoblastic leukemia detection in microscopic blood smear images,” *Sensors*, vol. 22, no. 15, p. 5520, 2022.

[3] “Leukemia - cancer stat facts.” [Online]. Available: <https://seer.cancer.gov/statfacts/html/leuks.html>

[4] “Leukemia-patient version.” [Online]. Available: <https://www.cancer.gov/types/leukemia>

[5] L. P. Clinton Jr, K. M. Somes, Y. Chu, and F. Javed, “Acute lymphoblastic leukemia detection using depthwise separable convolutional neural networks,” *SMU Data Science Review*, vol. 3, no. 2, p. 4, 2020.

[6] R. B. Hegde, K. Prasad, H. Hebbar, B. M. K. Singh, and I. Sandhya, “Automated decision support system for detection of leukemia from peripheral blood smear images,” *Journal of digital imaging*, vol. 33, no. 2, pp. 361–374, 2020.

[7] A. M. Abdeldaim, A. T. Sahlol, M. Elhoseny, and A. E. Hassanien, “Computer-aided acute lymphoblastic leukemia diagnosis system based on image analysis,” in *Advances in Soft Computing and Machine Learning in Image Processing*. Springer, 2018, pp. 131–147.

[8] P. K. Das, A. Pradhan, and S. Meher, “Detection of acute lymphoblastic leukemia using machine learning techniques,” in *Machine learning, deep learning and computational intelligence for wireless communication*. Springer, 2021, pp. 425–437.

[9] R. Khandekar, P. Shastry, S. Jaishankar, O. Faust, and N. Sampathila, “Automated blast cell detection for acute lymphoblastic leukemia diagnosis,” *Biomedical Signal Processing and Control*, vol. 68, p. 102690, 2021.

[10] M. Aria, M. Ghaderzadeh, D. Bashash, H. Abolghasemi, F. Asadi, and A. Hosseini, “Acute lymphoblastic leukemia (all) image dataset,” *Kaggle*, 2021.

[11] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, “A generic intelligent tomato classification system for practical applications using densenet-201 with transfer learning,” *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.

[12] Y. Sun, F. A. B. Hamzah, and B. Mochizuki, “Optimized light-weight convolutional neural networks for histopathologic cancer detection,” in *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*. IEEE, 2020, pp. 11–14.

[13] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[14] G. Xie, B. Guo, Z. Huang, Y. Zheng, and Y. Yan, “Combination of dominant color descriptor and hu moments in consistent zone for content based image retrieval,” *IEEE Access*, vol. 8, pp. 146 284–146 299, 2020.

[15] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, “Textural features for image classification,” *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[16] N. A. Hamilton, R. S. Pantelic, K. Hanson, and R. D. Teasdale, “Fast automated cell phenotype image classification,” *BMC bioinformatics*, vol. 8, no. 1, pp. 1–8, 2007.

[17] S. Ray, “A quick review of machine learning algorithms,” in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[18] B. Mahesh, “Machine learning algorithms-a review,” *International Journal of Science and Research (IJSR)*. [Internet], vol. 9, pp. 381–386, 2020.

[19] N. Supardi, M. Mashor, N. Harun, F. Bakri, and R. Hassan, “Classification of blasts in acute leukemia blood samples using k-nearest neighbour,” in *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*. IEEE, 2012, pp. 461–465.

[20] H.-S. Jo, C. Park, E. Lee, H. K. Choi, and J. Park, “Path loss prediction based on machine learning techniques: principal component analysis, artificial neural network, and gaussian process,” *Sensors*, vol. 20, no. 7, p. 1927, 2020.

[21] E. K. Ampomah, Z. Qin, and G. Nyame, “Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement,” *Information*, vol. 11, no. 6, p. 332, 2020.

[22] Z. Zhang, “Improved adam optimizer for deep neural networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.