DeepIndel: A ResNet-Based Method for Accurate Insertion and Deletion Detection from Long-Read Sequencing

Md. Shadmim Hasan Sifat[1], Khandokar Md. Rahat Hossain[2]
Dept. of Computer Science & Engineering, Shahjalal University of Science & Technology, Sylhet, Bangladesh[1]
Dept. of Computer Science & Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh[2]

***Abstract*—Structural variations (SVs) play a pivotal role in human genetics, influencing gene expression, disease mechanisms, and phenotypic diversity. Despite the advancements in short-read sequencing technologies, long-read sequencing offers superior resolution for detecting SVs, particularly in complex genomic regions. In this study, DeepIndel, a novel computational framework, is presented that leverages long-read sequencing data combined with a deep learning model to identify SV breakpoints accurately. This approach captures complex breakpoint patterns by aligning long reads to a reference genome and extracting 23 key features at each genomic location, including read support, candidate length, and strand-specific information. DeepIndel has been evaluated on the HG002 dataset, achieving exceptional performance with high precision and reliability in detecting insertions and deletions, with F1 scores (94.27% for insertions, 91.09% for deletions) and thereby demonstrating significant improvements over existing state-of-the-art tools, offering a more precise and robust approach to SV detection. This work advances structural variant analysis, with promising implications for genomic research, disease understanding, and personalized medicine.**

***Keywords*—Structural variations (SVs); indels; long-read sequencing; breakpoints; genomic features; diseases; deep learning; ResNet; HG002 dataset; precision medicine; gene expression; phenotypic diversity**

I. Introduction

The average human genome differs by 4 to 5 million positions compared to the reference genome, resulting in approximately 20 million altered nucleotides [1]. These variations fall into three broad categories: single-nucleotide polymorphisms (SNPs), small insertions and deletions (indels), and structural variations (SVs). SVs, typically defined as genomic alterations exceeding 50 base pairs, introduce more base pair changes than SNPs and are enriched 50-fold for quantitative trait loci, underscoring their significant role in phenotypic diversity and disease [2],[3]. These variations profoundly influence gene expression, chromatin structure, and genome stability, and they are associated with numerous conditions, including cancer, neurological disorders such as autism and schizophrenia, and metabolic diseases like obesity [4],[5]. Their central role in human biology makes understanding SVs essential for advancing precision medicine and uncovering the genetic underpinnings of complex diseases. Recent large-scale analysis, such as those examining structural variation across 1,019 diverse human genomes using long-read sequencing, further underscore the need for precise SV detection methods to capture the full spectrum of genomic diversity [6].

Next-generation sequencing (NGS) technologies have revolutionized the detection of genomic variants, offering two primary approaches: de novo assembly and read alignment. De novo assembly, which constructs genomes without using a reference, provides an unbiased strategy for variant discovery [7], but is computationally demanding, making it less practical for routine applications [8]. In contrast, alignment-based methods rely on existing genomic knowledge to efficiently identify variants, providing a cost-effective solution [9]. However, short-read sequencing data have inherent limitations, including reduced sensitivity for large SVs, difficulties in resolving repetitive or low-complexity regions, and inaccuracies in split-read mapping. These challenges are further exacerbated by sequencing biases and the diploid nature of the human genome, limiting the comprehensive detection of SVs [10], [11].

Various computational tools have been developed to address the limitations of short-read sequencing. Dysgu, for example, uses machine learning and deep learning algorithms to enhance the sensitivity and precision of SV detection [12]. Despite these advancements, short-read sequencing inherently struggles with resolving large-scale genomic rearrangements and low-mappability regions. This limitation has driven the adoption of long-read sequencing technologies, such as Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT). These platforms produce reads tens of kilobases in length, enabling direct and detailed characterization of SVs, including those in complex genomic contexts. Tools like Pepper Margin DeepVariant exemplify the potential of long-read data in detecting single-nucleotide variants (SNVs) and SVs with high accuracy, even in regions with segmental duplications or other challenging genomic features [13].

Long-read sequencing has significantly advanced our ability to map and characterize SVs, overcoming many limitations of short-read methods. The extended read lengths allow for the detection of previously undetectable variants and improve mapping accuracy in repetitive and low-complexity regions. Recent studies have highlighted the utility of long reads in resolving complex genomic rearrangements and characterizing mobile elements, offering deeper insights into human genetic variation [14],[15]. Computational tools designed for long-read sequencing data, such as Sniffles [16], cuteSV [17], and SVIM [18], have significantly improved the sensitivity and precision

of SV detection. DeBreak [19], a more recent tool, employs density-based clustering to enhance breakpoint accuracy, particularly for large insertions and deletions. Additionally, pbsv [20] (developed by Pacific Biosciences) focuses on detecting structural variants with high precision and recall using HiFi long-read data, making it highly suitable for resolving SVs in challenging genomic regions. Another tool, NanoVar [21], is optimized for ONT and PacBio long-read data and demonstrates notable accuracy in detecting both small and large SVs by combining sequence information with sophisticated algorithms for breakpoint characterization. Lastly, combiSV [22] incorporates a comparative genomics-based approach to leverage existing reference datasets, enabling improved detection of complex SVs, particularly in repeat-rich regions, by integrating genomic signatures from multiple samples. Despite these advances, challenges persist, including the effective integration of multi-locus rearrangements and the detection of SVs in repetitive regions.

There are two main approaches for detecting indels from long-read sequencing: alignment and assembly. Tools like Sniffles, cuteSV, and SVIM rely on signature detection and clustering, yet they struggle with breakpoint accuracy in repetitive regions, with benchmarks indicating recall below 90% [16]. DeBreak employs density-based clustering to improve large indel detection, while NanoVar and pbsv are optimized for ONT and HiFi reads, respectively, though limitations persist [19], [20]. Recent advancements, such as SAVANA [29], use haplotype-resolved analysis for somatic SVs and tumor purity estimation but are less effective for germline indels, and SUMMER [30] offers a Nanopore pipeline for variants without DeepIndel's ResNet-based feature learning. MEHunter [31] leverages transformers for mobile element variants, and TEforest [32] applies machine learning to TE indels from short reads, highlighting gaps in long-read generalization and diploid context handling. DeepIndel addresses these shortcomings with distinct advantages: **1)** 23 strand-specific features enhance breakpoint resolution in complex regions, **2)** a ResNet-50 architecture, extending Pepper-Margin [13], mitigates vanishing gradients through residual connections for deeper, more effective learning, **3)** superior F1 scores (94.27% insertions, 91.09% deletions on HG002) outperform Sniffles and DeBreak, and **4)** tailored optimization for PacBio HiFi reads outpaces Nanopore-focused tools like SUMMER. Compared to deep learning models in similar fields, DeepIndel surpasses MAMnet (a CNN for indel genotyping with simpler layers) [33] and SVcnn (a multi-type SV tool with ~88–90% F1) [34], achieving a 3 to 5% F1 improvement due to its specialized feature matrices. These strengths position DeepIndel as a robust solution for precision medicine, particularly for detecting disease-associated indels in challenging genomic landscapes.

While current methods excel in isolated breakpoint detection, they often fail to account for the interconnections between loci involved in complex rearrangements. Here, we propose a novel computational method designed to address these gaps. This approach leverages the power of long-read sequencing to accurately detect and classify three distinct SV categories, including large insertions and deletions of unique genomic elements. By incorporating advanced computational techniques, it demonstrates superior precision and recall compared to existing tools when applied to real-world datasets. This innovative methodology provides a robust framework for improving indel detections and understanding its implications in human genetics and precision medicine.

II. Materials and Methods

The Pepper-Margin DeepVariant uses a frequency-based caller to identify the single-nucleotide variants (SNVs) and short indels [13]. In DeepIndel, this approach of Pepper-Margin has been extended by effectively using the prior knowledge we had about our datasets. Our method (Section II.E) tends to look for the exact location of breakpoint events as mentioned in the BED (Browser Extensible Data) files. By analyzing the different features around these breakpoints, the model can effectively call for SVs at different genomic locations of the human genome with an excellent level of fidelity.

A. Model

DeepIndel centers around a deep learning (CNN)-based residual network architecture designed for multi-class classification. The inputs to the network are the feature matrices extracted from our positive and negative samples (Section II B). The matrices are then fed to a neural network. Upon training the network, it finally predicts which SVs are the most likely to occur at a specific location. A high-level overview of the model can be seen in Fig. 1.

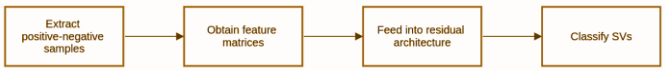


Fig. 1. High-level overview of the model.

B. Sample Extraction

In this three-class (namely: Insertion, Deletion and Non-Indel) SV classifier, we denote the Insertion and Deletion candidates as the set of positive samples and the Non-Indel candidates as the set of negative ones.

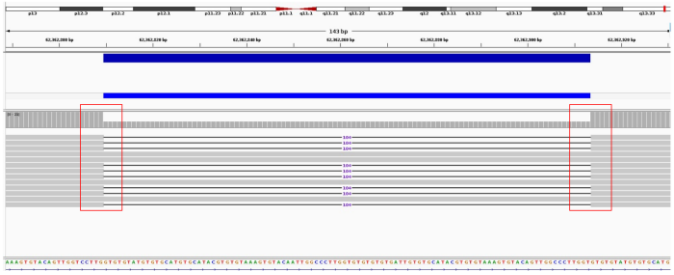


Fig. 2. Placing windows at deletion breakpoints.

1) Positive samples:

a) Insertion: At the positions where an insertion breakpoint is encountered (either the beginning or ending one), we place our windows at those locations to obtain the features (Section D) for the generation of our input image matrices.

b) Deletion: Likewise, at the deletion breakpoints, we place our windows at the beginning or ending locations, where the breakpoints are mentioned. Fig. 2 depicts a 104 bp (base pairs) long deletion event and an emplacement of our windows is shown in this figure.

2) Negative samples: In the case of non-indels, we choose random genomic locations in the sample other than any of the insertion or deletion events to place our windows. To successively obtain these, we choose the regions between the ending and starting breakpoints of two consecutive events, respectively. An example of emplacement is shown in Fig. 3.

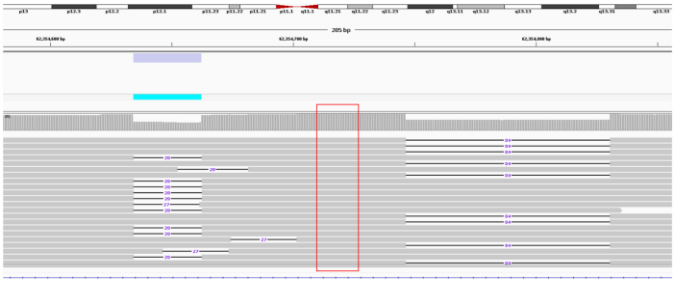


Fig. 3. Choosing window placement for non-indels.

C. Creating a Summary of Potential Variants

The main pipeline of our proposed model first creates a summary of potential candidate variants at the different window locations. To achieve this, we follow the steps outlined below:

All the reads that have a mapping quality greater than the threshold min_mapq are chosen initially. The mapping quality refers to how confidently a particular read has been mapped to a specific position in the reference genome. Next, for the INDEL variants, the candidates with average base quality above min_indel_baseq are considered among the initial ones.

To precisely identify the insertion or deletion candidates, we identify the insertion positions with a cumulative insertion frequency higher than insert_frequency and a cumulative deletion frequency higher than delete_frequency. These thresholds are checked independently for insertions and deletions, and all potential variant candidates from here are recorded. For each candidate, a final additional filtering is performed to ensure that at least candidate_support_threshold reads support the variant.

D. Obtaining Feature Matrix

After choosing the potential candidate variants, we extract a feature matrix from those windows, or "sites". The matrix represents a summary of the read alignments in those respective sites.

In the input matrices, we encode four different features in a total of 23 rows (see Fig. 4). The dimension is chosen to include both the features of the forward strands and the reverse strand reads in a single matrix. We represent the matrices here as RGB images to illustrate the comparative values obtained at each index and to differentiate among the four nitrogenous bases and their respective events.

Here we describe the list of features that are encoded in our Feature Matrix:

- REF: Encoding of the Reference Base. The four different bases are converted to four colors. We used the encoding mapping as A: "Blue", G: "Green", T: "Yellow", and C: "Red".
- **I_L, D_L:** Encoding of the candidate length of the INDELs. I and D are used based on the candidate types, Insert and Delete, respectively.
- **R_F:** Count of the forward strand reads that support the reference allele.
- **I_S, D_S:** Count of the forward strand reads that support the alternate allele. I and D are used based on candidate types, Insert and Delete, respectively.
- **A_F, G_F, T_F, C_F:** Total read count in forward strands expressing each base. The opacity of the colors is scaled in the range [0, 255]. The opacity is increased if its base has a higher frequency.
- **I_F:** Total number of insertions based on forward strand reads.
- **D_F:** Total number of deletions observed at specific positions in the genome, based on forward-strand reads that are anchored to those positions.
- **$*_F$:** Total number of deletions observed from forward strand reads, without anchoring them to particular positions in the genome.

For reverse strands, these features are encoded identically (denoted with R).

In summary, the following set of metrics is encoded from the above features (as mentioned in Fig. 4):

1) Candidate length: The length of candidates at a certain position or over a span of region, denoted by I_L *and* D_L.

2) Read support: The number of reads supporting the reference allele. Since multiple candidates can be reported by different reads over the same specific region, the read support is calculated as the percentage of reads that support the same signature there. It is denoted by R_F (for the reference allele) and I_S, D_S (for the alternate allele).

3) Base counts: The number of bases that are aligned with the reads on a specific column, denoted by A_F, G_F, T_F and C_F.

4) Candidate counts: The number of candidates being reported over a specific region. It is counted as the actual counts of candidates that are reported by several reads over a span of region, denoted by I_F, D_F and $*_F$.

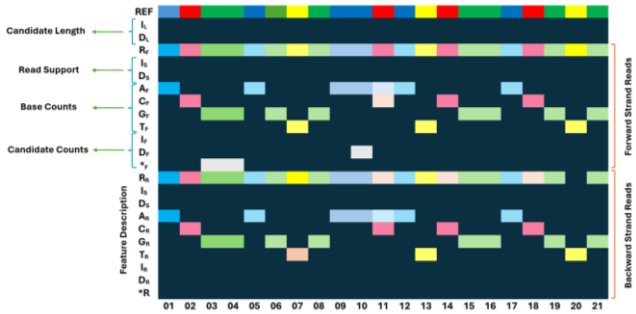


Fig. 4. Encoding of features.

E. The Network Architecture

We fundamentally used a modified version of the ResNet-based architecture (see Fig. 5) to train our dataset. Our model comprises four layers from an upper level, with each consisting of three convolutional layers along with batch normalizations and *ReLU* activations. In between the layers, we used dropout regularizations to avert overfitting to the training data. We trained our model using a batch size of 512 for 200 epochs, optimizing with the Adam algorithm at a learning rate of 0.1. The loss function employed was Cross-Entropy Loss. For activation functions, we used ReLU in the mid-level layers and *SoftMax* in the output layer. We have used the modified version of ResNet-50, which incorporates specific optimizations tailored to our dataset and task requirements. ResNet (Residual Network) is preferred over other well-known models like VGG16 and GoogleNet due to its superior handling of the vanishing gradient problem, allowing it to train much deeper networks effectively. This is achieved through the introduction of residual connections, which facilitate the learning of residuals rather than direct mappings, thereby improving performance and efficiency [23], [24], [25]. Comparative studies have demonstrated that ResNet-50 consistently outperforms VGG16 and GoogleNet in various benchmarks, making it a more reliable and efficient choice for deep learning tasks [24]. The matrices initially extracted are 23×33 by dimension, where the 23 rows represent the features and the 33 columns indicate one window size. We resized these into 30×30 square images to be fed into the network. For each pass throughout the whole network, we used a SoftMax activation in the end for classifying our three signatures: Insertion, Deletion and non-indels. The convolution layers comprise a deep architecture that effectively identifies the subtle changes in patterns along with the indicative SVs throughout its learning process. The residual layers are also used for preserving the significant variation features across layers.

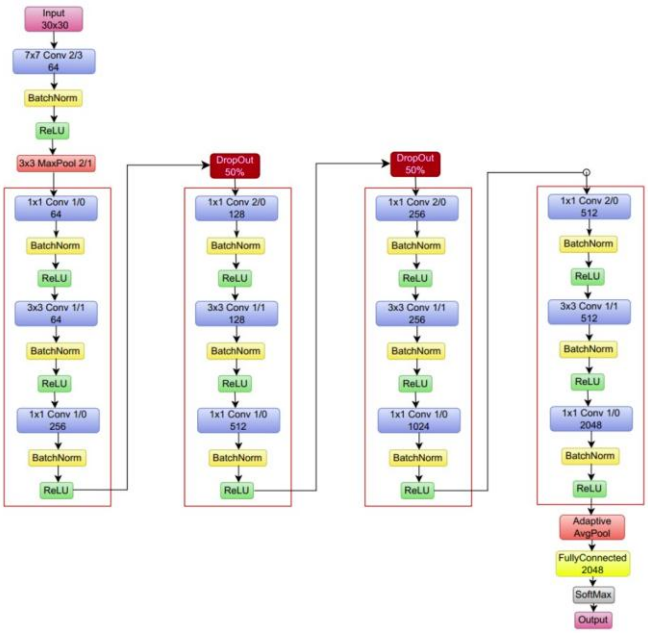


Fig. 5. The overall network architecture for DeepIndel.

F. Validation Metrics

To test DeepIndel, we took the HG002 dataset, which has confirmed structural variant (SV) annotations from the Genome In A Bottle (GIAB) project, as it is a gold-standard benchmark. We analyzed the model's performance by assessing its precision (true positives / [true positives + false positives]), recall (true positives / [true positives + false negatives]), and F1-score (harmonic mean of precision and recall) using the model's evaluation framework from the previous section. All of these are important for assessing its performance. Having a high degree of performance is important because it decreases the number of incorrect diagnoses, which is particularly important in clinical settings. Strong recall, however, ensures that indels associated with rare diseases are captured, which adds value to the genetic studies. The F1-score is particularly useful as a summary measure in model evaluation, especially in precision medicine, where both sensitivity and specificity are important, and it measures the degree to which precision and recall work in tandem. Comparisons presented in subsequent sections illustrate how DeepIndel can be guided to better detect SVs in challenging genomic contexts using this validation technique.

III. Dataset Preparation

A. Benchmark Datasets

To assess the performance of DeepIndel, we used the PacBio HiFi2 long-read sequencing dataset with the GRCh37 human genome as the reference genome. We used a well-characterized reference sample, HG002 (Source: AshkenazimTrio/NIST SVs Integration v0.6), from the GIAB (Genome In A Bottle) projects [28].

B. Preprocessing

To make the dataset suitable for our model, we had to perform some preprocessing on our dataset. Most of the processing was done on the Ubuntu 20.04 operating system.

1) Read alignment: DeepIndel accepts sorted BAM files as input and utilizes state-of-the-art long-read aligners to construct SV detection pipelines. Aligners that demonstrate robust performance in handling large insertions and deletions or that are capable of generating accurate split alignments are particularly preferred. For our framework, we aligned the simulated reads to our reference genome GRCh37 using *Minimap2* (v2.17-r941) [26]. Minimap2 initially generates a file in SAM format, which is a text-based generic alignment format used for storing read alignments against a reference sequence. Next, to get the sorted BAM, we used Samtools [27] to convert the SAM format to its respective BAM.

2) Indexing of VCF file: The BED files were first converted into VCF (Variant Call Format) files to enable the representation of genomic variants. This conversion was done using *bed2vcf*. Subsequently, the VCF index files were generated to facilitate the matching of records at each breakpoint position. For this purpose, we utilized *tabix*. The VCF files being indexed with *tabix* enabled us to query for specific genomic regions, allowing for the retrieval of variants located within those regions.

IV. RESULTS

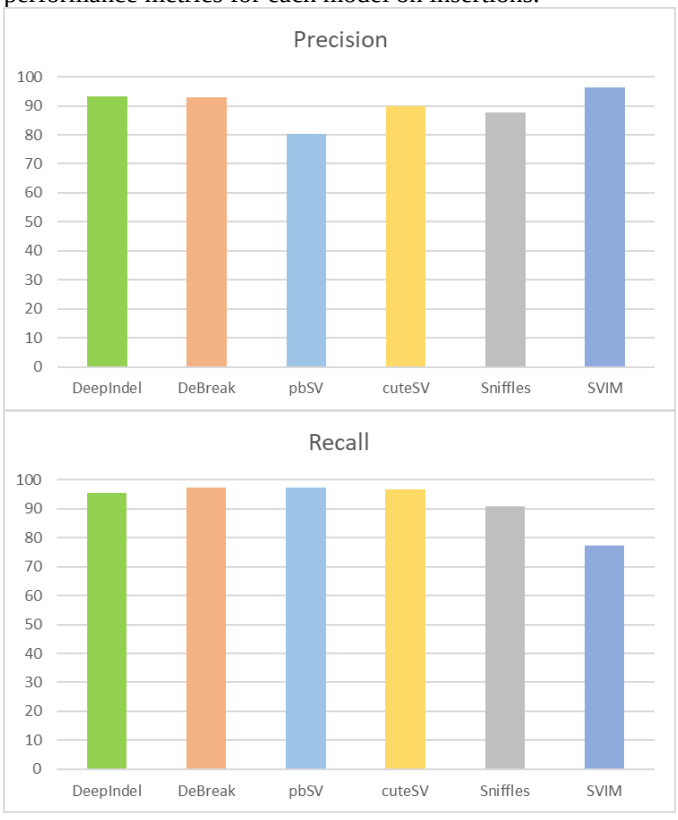
A. Evaluation Criteria

To evaluate the performance of different SV callers along with DeepIndel, each of them was given a task to identify variants from the same chromosomal locations in our test dataset. Initially, all 22 chromosomes from GRCh37 were taken and using the above-mentioned pipeline, we received around 31000 positive and 19700 negative samples (as 2D matrices). Then we split into a 60:15:25 ratio for training, validation and testing sets, respectively. The models were trained individually with train and validation datasets, taking 30×30 feature matrices as input. After being fed into the network, the outcomes were measured based on the performance scores for successfully calling variants at each possible location.

B. Performance in Long Reads

In this section, the performance comparison of DeepIndel is presented with state-of-the-art structural variant calling models like DeBreak, pbSV, cuteSV, Sniffles, and SVIM. The comparison was made across their precision, recall and F1-score.

1) Insertion: DeepIndel demonstrates superior insertion precision, outperforming all other models except SVIM with an impressive precision rate of 93.19%. For recall, DeepIndel demonstrates a strong performance too, with a recall rate of 95.38%. DeepIndel achieves an F1 score of 94.27%, ranking highly among all models and presenting an excellent balance between precision and recall. Fig. 6 illustrates these performance metrics for each model on insertions.



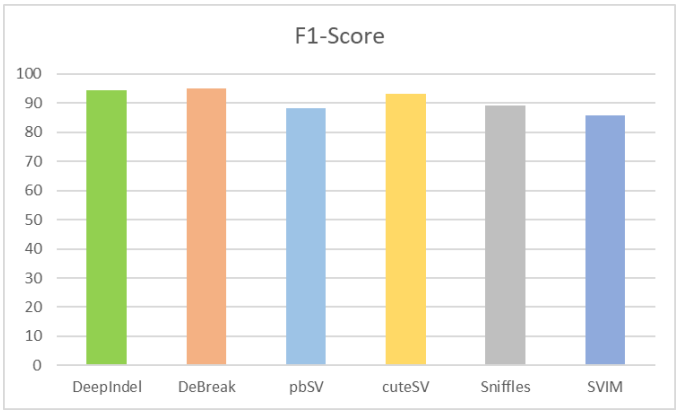


Fig. 6. Performance comparison in insertion: precision, recall, f1-score.

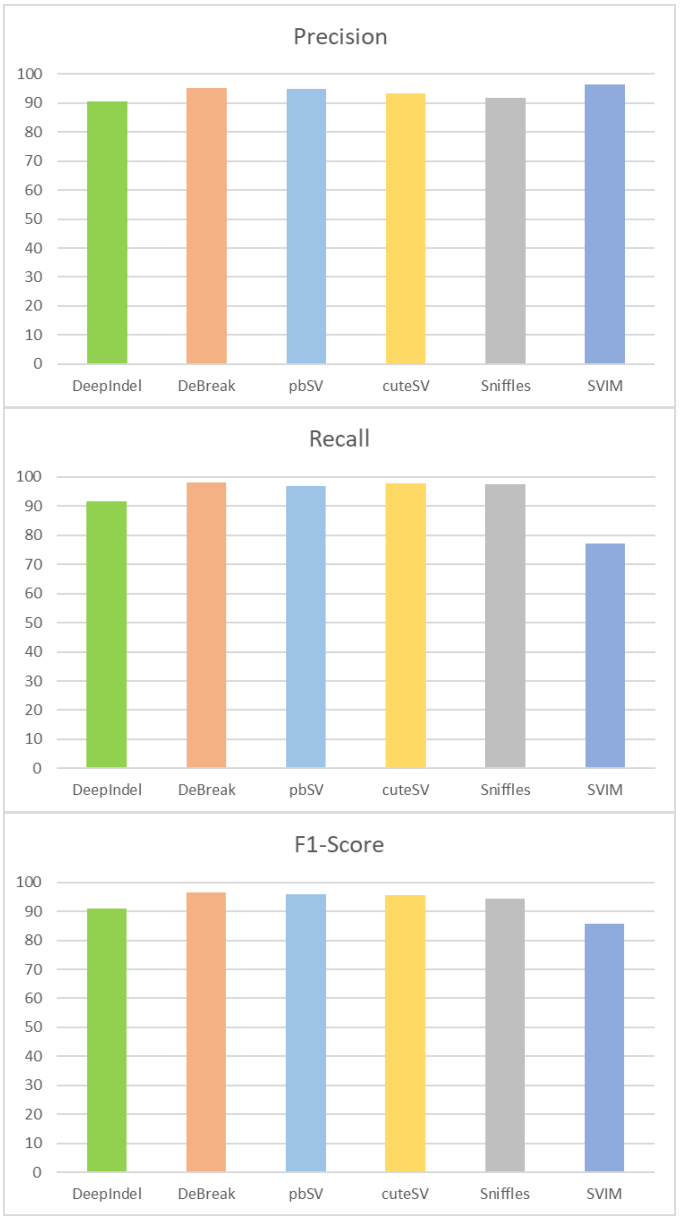


Fig. 7. Performance comparison in deletion: precision, recall, f1-score.

2) Deletion: DeepIndel exhibits strong performance in precision for deletions, achieving a precision rate of 90.42%, which is competitive with other models. For recall, DeepIndel achieves 91.76%, positioning it favorably compared to the other models. Regarding the F1-score, DeepIndel ranks well with a value of 91.09%, highlighting a good overall balance between precision and recall compared to the other models. Fig. 7 illustrates these comparisons on deletions.

V. Discussion

DeepIndel achieved outstanding accuracy in detecting SVs for the HG002 dataset, demonstrating the effectiveness and robustness of our approach. The high precision and recall rates, along with competitive F1 scores for both insertions and deletions, underscore the reliability of DeepIndel in variant calling tasks. The significance of these validation measures lies in their direct relevance to clinical utility: high precision reduces the risk of erroneous variant calls that could mislead therapeutic decisions, while robust recall ensures no critical SVs are missed in disease studies. The F1-score, achieving 94.27% for insertions and 91.09% for deletions, reflects a balanced performance that is essential for trustworthy genomic analysis. Thus, it's highlighting its strong performance overall. However, the model's performance was slightly better for insertions than for deletions. This variation in performance can be attributed, in part, to the characteristics of the benchmark dataset. The model was evaluated on the latest, most conserved BED file for HG002 (*HG002 SVs Tier1 v0.6.2.bed*), which contains 5260 insertions and 4138 deletions [22]. The higher representation of insertions in the dataset provides a richer training and evaluation context for the model, potentially contributing to its superior precision and recall for insertions. In contrast, the slightly lower performance for deletions may reflect the relatively smaller number of examples available, as well as inherent challenges in accurately identifying deletions within the genomic sequence.

While DeepIndel's performance is commendable, there is room for improvement in refining feature extraction and window positioning near breakpoint candidates to address edge cases for deletions. Nevertheless, the consistent and competitive results across various performance metrics establish DeepIndel as a highly effective tool for SV detection. The findings from the HG002 dataset confirm the robustness of DeepIndel's methodologies and its potential for broader genomic applications. Addressing minor limitations in feature extraction and ensuring robust truth datasets will further enhance the reliability and accuracy of the model.

VI. Conclusion

In this study, a novel deep learning framework, DeepIndel, was designed to enhance the accurate detection of indels from long-read sequencing data. By leveraging a comprehensive set of 23 genomic features extracted from aligned reads, the model effectively captures complex breakpoint patterns in challenging genomic regions. Trained on the HG002 benchmark dataset using a modified ResNet-50 architecture with residual connections, dropout regularization, and cross-entropy loss optimization, DeepIndel processes significant feature matrices to classify genomic sites as insertions, deletions, or non-indels with high fidelity.

DeepIndel's advancements highlight the transformative potential of integrating long-read sequencing with deep learning for SV analysis, addressing limitations in traditional methods, such as difficulties in repetitive regions and interconnected loci. By improving precision and recall, this approach facilitates more reliable insights into genomic diversity, gene expression alterations, and disease associations, including cancer, neurological disorders, and metabolic conditions. This contributes to the broader goals of precision medicine, enabling better characterization of phenotypic diversity and personalized therapeutic strategies.

Despite these strengths, some limitations and future opportunities are acknowledged that include potential refinements needed in feature extraction and window positioning to handle edge cases. Future directions will focus on expanding evaluations to diverse datasets beyond HG002, incorporating additional SV types like inversions and duplications, and optimizing for real-time clinical applications. Overall, DeepIndel marks a significant advancement in SV-calling tools, paving the way for enhanced genomic research and its translation into impactful clinical outcomes.

Code Availability

The full source code is available at this link.

Acknowledgment

We thank Dr. Atif Hasan Rahman (Associate Professor, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh) for his guidance and Dr. Kishwar Shafin (Research Scientist at Google, Health AI) for providing useful comments on our research project.

References

[1] T. G. P. Consortium, "A global reference for human genetic variation," Nature, vol. 526, pp. 68–74, 2015.

[2] R. L. Collins et al., "A structural variation reference for medical and population genetics," Nature, vol. 581, pp. 444–451, 2020.

[3] J. R. Belyeu, M. Chowdhury, J. Brown, B. S. Pedersen, M. J. Cormier, A. R. Quinlan, and R. M. Layer, "Samplot: a platform for structural variant visual validation and automated filtering," Genome Biology, vol. 22, no. 1, pp. 1–9, 2021.

[4] P. A. Audano, A. Sulovari, T. A. Graves-Lindsay, S. Cantsilieris, M. Sorensen, A. M. Welch, M. L. Dougherty, B. J. Nelson, P. Shah, S. K. Dutcher, and E. E. Eichler, "Characterizing the major structural variant alleles of the human genome," Cell, vol. 176, no. 3, pp. 663–675, 2019.

[5] P. H. Sudmant, T. Rausch, E. J. Gardner, R. E. Handsaker, A. Abyzov, J. Huddleston, Y. Zhang, K. Ye, G. Jun, M. H.-Y. Fritz, et al., "An integrated map of structural variation in 2,504 human genomes," Nature, vol. 526, no. 7571, pp. 75–81, 2015.

[6] Schloissnig, S., Pani, S., Ebler, J. et al. Structural variation in 1,019 diverse humans based on long-read sequencing. Nature 644, 442–452 (2025).

[7] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, S. Aganezov, S. J. Hoyt, M. Diekhans, G. A. Logsdon, M. Alonge, S. E. Antonarakis, M. Borchers, G. G. Bouffard, S. Y. Brooks, G. V. Caldas, et al., "The complete sequence of a human genome," Science, vol. 376, no. 6588, pp. 44–53, 2022.

[8] E. Espinosa, R. Bautista, R. Larrosa, and O. Plata, "Advancements in long-read genome sequencing technologies and algorithms," Genomics, vol. 116, no. 3, p. 110842, 2024.

[9] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," bioinformatics, vol. 25, no. 14, pp. 1754–1760, 2009.

[10] M. Mahmoud, N. Gobet, D. I. Cruz-D´avalos, N. Mounier, C. Dessimoz, and F. J. Sedlazeck, "Structural variant calling: the long and the short of it," Genome Biology, vol. 20, p. 246, 2019.

[11] D. R. Zerbino and E. Birney, "Velvet: algorithms for de novo short read assembly using de bruijn graphs," Genome research, vol. 18, no. 5, pp. 821–829, 2008.

[12] K. Cleal and D. M. Baird, "Dysgu: efficient structural variant calling using short or long reads," Nucleic Acids Research, vol. 50, no. 9, pp. e53–e53, 2022.

[13] K. Shafin, T. Pesout, P.-C. Chang, M. Nattestad, A. Kolesnikov, S. Goel, G. Baid, M. Kolmogorov, J. M. Eizenga, K. H. Miga, et al., "Haplotype-aware variant calling with pepper-margin-deepvariant enables high accuracy in nanopore long-reads," Nature methods, vol. 18, no. 11, pp. 1322–1332, 2021.

[14] A. M. Wenger, P. Peluso, W. J. Rowell, P.-C. Chang, R. J. Hall, G. T. Concepcion, J. Ebler, A. Fungtammasan, A. Kolesnikov, N. D. Olson, et al., "Accurate circular consensus long-read sequencing improves vari- ant detection and assembly of a human genome," Nature Biotechnology, vol. 37, pp. 1155–1162, 2019.

[15] M. Jain, S. Koren, K. H. Miga, et al., "Nanopore sequencing and assembly of a human genome with ultra-long reads," Nature Biotechnology, vol. 36, pp. 338–345, 2018.

[16] F. J. Sedlazeck, P. Rescheneder, M. Smolka, H. Fang, M. Nattestad, A. von Haeseler, and M. C. Schatz, "Accurate detection of complex structural variations using single-molecule sequencing," Nature Methods, vol. 15, no. 6, pp. 461–468, 2018.

[17] R. Jian, X. Chen, and et al., "cutesv: A sensitive and fast tool for detecting short and long structural variations," Bioinformatics, vol. 36, no. 4, pp. 1233–1235, 2020.

[18] D. Heller et al., "Svim: Structural variant identification using mapped long reads," Bioinformatics, vol. 35, no. 18, pp. 2907–2915, 2019.

[19] Z. Chong et al., "Debreak: Efficiently detecting structural variations in long-read sequencing data," Genome Biology, vol. 24, no. 1, p. 14, 2023.

[20] W. Rowell et al., "Comprehensive variant detection in a human genome with highly accurate long reads," 2019. PacBio White Paper.

[21] M. Kolmogorov et al., "Nanovar: Accurate structural variant detection with ont and pacbio data," BioRxiv, 2020.

[22] N. Dierckxsens, T. Li, J. R. Vermeesch, and Z. Xie, "A benchmark of structural variation detection by long reads through a realistic simulated model," Genome Biology, vol. 22, no. 1, 2021.

[23] X. Zhang, N. Han, and J. Zhang, "Comparative analysis of vgg, resnet, and googlenet architectures evaluating performance, computational efficiency, and convergence rates," Applied and Computational Engineering, vol. 44, pp. 172–181, 03 2024.

[24] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), vol. 1, pp. 96–99, 2021.

[25] N. Zakaria, F. Mohamed, R. Abdelghani, and K. Sundaraj, "Vgg16, resnet-50, and googlenet deep learning architecture for breathing sound classification: A comparative study," in 2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI- CSP), pp. 1–6, 2021.

[26] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," Bioinformatics, vol. 34, no. 18, pp. 3094–3100, 2018.

[27] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup, "The sequence alignment/map format and samtools," bioinformatics, vol. 25, no. 16, pp. 2078–2079, 2009.

[28] N. Dierckxsens, T. Li, J. R. Vermeesch, et al., "A benchmark of structural variation detection by long reads through a realistic simulated model," Genome Biology, vol. 22, no. 342, 2021.

[29] H. Elrick, C. M. Sauer, J. Espejo Valle-Inclan, K. Trevers, M. Tanguy, S. Zumalave, S. De Noon, F. Muyas, R. Cascão, A. Afonso, A. G. Rust, F. Amary, R. Tirabosco, A. Giess, T. Freeman, A. Sosinsky, K. Piculell, D. T. Miller, C. C. Faria, G. Elgar, *et al.*, "SAVANA: reliable analysis of somatic structural variants and copy number aberrations using long-read sequencing," *Nature Methods*, vol. 22, no. 7, pp. 1436–1446, 2025.

[30] R. Li, H. Chu, K. Gao, H. Luo, and Y. Jiang, "SUMMER: an integrated nanopore sequencing pipeline for variants detection and clinical annotation on the human genome," *Functional & Integrative Genomics*, vol. 25, no. 1, p. 21, 2025.

[31] T. Jiang, Z. Zhou, Z. Zhang, S. Cao, Y. Wang, and Y. Liu, "MEHunter: transformer-based mobile element variant detection from long reads," *Bioinformatics*, vol. 40, no. 9, p. btae557, 2024.

[32] A. Daigle, L. S. Whitehouse, R. Zhao, J. J. Emerson, and D. R. Schrider, "Leveraging long-read assemblies and machine learning to enhance short-read transposable element detection and genotyping," *bioRxiv*, p. 2025.02.11.637720, 2025.

[33] H. Ding and J. Luo, "MAMnet: detecting and genotyping deletions and insertions based on long reads and a deep learning approach," *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac195, 2022.

[34] Y. Zheng and X. Shang, "SVcnn: an accurate deep learning-based method for detecting structural variation based on long-read data," *BMC Bioinformatics*, vol. 24, no. 1, p. 213, 2023.