ChangeMambaVision: Adapting MambaVision for Building Change Detection

Thomas Gozalie
Department of Computer Science
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
thomas.gozalie@binus.ac.id

Andrea Stevens Karnyoto
Bioinformatics and Data Science Research Center
Bina Nusantara University
Jakarta, Indonesia 11480
andrea.karnyoto@binus.ac.id

Edy Irwansyah
Department of Computer Science
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
edirwan@binus.ac.id

Bens Pardamean
Computer Science Department
Master of Computer Science Program
Bina Nusantara University
Jakarta, Indonesia 11480
bpardamean@binus.edu

***Abstract*—Change Detection (CD) is important for analysis of land cover change within a geographical area. Accurate manual classification and labeling of change areas is difficult, and change results are used for subsequent data analysis. Therefore, there is a need for highly accurate automated methods for CD. The recent Mamba architecture has seen growing usage in vision applications due to its lower time complexity compared to Transformer, which provides better scalability for very high-resolution (VHR) imagery. MambaVision is a novel architecture that incorporates Mamba blocks alongside convolution blocks to enhance feature extraction of finer details, and Transformer blocks to enhance the capture of global dependencies. MambaVision has shown reported state-of-the-art performance on general vision tasks, as well as achieving unparalleled throughput compared to other vision backbones. In this paper, we propose ChangeMambaVision, where we adapt the MambaVision backbone with minimal modifications for Change Detection with a lightweight ChangeFormer decoder to complement its strengths. We show that ChangeMambaVision on the Base checkpoint achieves better performance than the current state-of-the-art (CDMamba) in the LEVIR-CD (+0.56 F1 / +0.92 IoU) and WHU-CD (+0.49 F1 / +0.77 IoU), while having faster throughput (68.5% increase) and lower training memory costs (23.32% decrease) than the current state-of-the-art.**

***Index Terms*—change detection, mamba, transformer, high-resolution imagery, remote sensing**

I. Introduction

Change detection (CD) is a dense prediction task where the objective is to detect areas of change at the pixel level given two images taken in the same area but at different time periods [1] [2]. Change detection has been growing in popularity recently due to its applicability to various research and data analytics for urban planning [3], urban growth monitoring [4], and disaster assessment [5]. Accurate manual classification and labeling of change areas is difficult, prone to human error, and usually done by experts. Classification accuracy of change or no-change areas are important, as change classifications are used for subsequent geographical data analysis. Therefore, research in change detection aims to provide methods that can be trusted to provide highly accurate predictions.

Research on deep learning methods, particularly deep computer vision models [6] [7], has enabled automatic feature extraction from images that can model complex feature dependencies. Early models used convolutions [8] that are capable of extracting features from a feature map while aggregating information from neighboring features through a sliding filter window. The U-Net architecture [9], which has since been widely used in image segmentation in general [10], geosensing [11] and medical [12] [13] contexts, has inspired a number of early change detection models such as FCNN [14] and HRSCD [15] due to its simplicity and strong ability of extracting longer-range dependencies while maintaining high-resolution feature extraction capability and efficiency through downsampling and skip connections between encoder and decoder layers.

More recently, the Transformer architecture [16] [17], which has seen extensive usage in other vision tasks [18] [19], has been adapted to CD through various models [20] [21]. Transformers can capture global dependencies between image patches with quadratic time complexity through self-attention and positional encoding. However, quadratic complexity incurs a significant computational cost, especially for dense prediction tasks such as CD where pixel-level information is considered.

The Mamba architecture [22] [23] offers a linear-time alternative in capturing global dependencies through a hardware-aware selective scan on sequence of image patches, which speeds up computation and is more scalable to larger resolution feature maps. The Mamba architecture has been adapted

to recent top-performing models in change detection [24] [25].

The MambaVision architecture [26], is recently introduced to be a top-performing backbone model in image classification, object detection, and segmentation tasks, achieving state-of-the-art performance at various throughput values (img/sec). Its efficiency and strong performance makes it a promising model to be adapted and used for Change Detection.

The contributions of this paper is as follows:

1) We propose ChangeMambaVision, a model that utilizes MambaVision [26] as the backbone model for feature extraction, and the lightweight ChangeFormer [20] decoder to produce a change map from the extracted features.
2) We show that this simple approach attains better performance than the SOTA baseline (CDMamba [25]) in the LEVIR-CD [27] and WHU-CD [28] datasets,
3) We show that the best performing checkpoint (ChangeMambaVision-B) still provides faster throughput and lower training memory usage than the SOTA baseline given the same feature map size in addition to the strong performance.

II. Related Work

Early deep learning techniques in change detection utilize fully-convolutional neural networks (FCNNs) due to their ability to tackle dense prediction tasks such as image segmentation. Daudt et al. proposed three architectures, namely FC-EF, FC-Siam-conc, and FC-Siam-diff [14] for change detection, utilizing FCNNs for end-to-end inference. FC-EF utilizes early fusion, where feature maps are concatenated along the channel dimension before passing it to the model. FC-Siam-conc and FC-Siam-diff used a Siamese network for feature extraction, and fusing extracted features by concatenation with the upsampling path. FC-Siam-diff takes the absolute difference of feature values from the siamese network before concatenating with the upsampling path.

Inspired by the success of the U-Net architecture on image segmentation tasks [9], Daudt et al. proposed FC-EF-Res [15] which builds upon FC-EF by utilizing a U-Net-like architecture with early fusion of feature maps. Fang et al. proposed SNUNet-CD [29] which combined the Siamese network with NestedUNet, and proposed an Ensemble Channel Attention Module (ECAM) for deep supervision.

The success of Transformers in NLP [16] [30] has inspired a number of models utilizing Transformer blocks for feature extraction. ChangeFormer [20] proposed by Bandara and Patel utilized Transformer blocks at each resolution of the downsampling path, and a lightweight decoder using fully connected layers. Chen et al. proposed Bitemporal Image Transformer (BIT) [31] which maps CNN-encoder to decoder features by tokenizing pixel-based features, process said tokens through a transformer encoder, before converting back tokenized features into pixel-based features through a transformer decoder. Song et al. proposed ACABFNet [32] which uses CNN and Transformer branch for feature extraction, and crosses over intermediate features from one branch to another using a single convolution and BatchNorm layer.

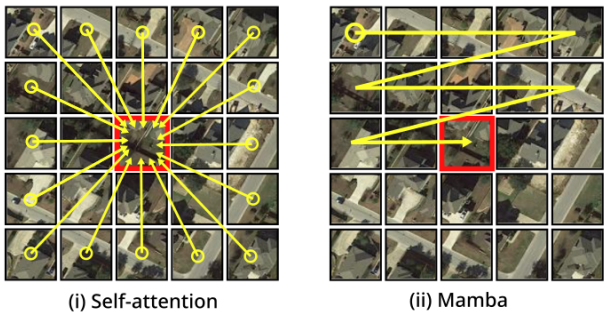

Fig. 1: (i) Self-attention in Transformer has a time complexity of $O((HW)^2)$, as every patch attends to every other patch. (ii) Mamba has a time complexity of $O(HW)$ due to its autoregressive formulation.

Mamba's linear-time scaling to sequence length [22] makes it potentially effective for image processing, where each feature pixel in height and width dimensions are considered tokens. Chen et al. proposed ChangeMamba [24] which utilizes a VMamba [23] backbone for feature extraction, and proposed an STSS block that uses VMamba's VSS blocks for fusion of feature tokens generated by intermediate feature maps. Zhang et. al proposed CDMamba [25] that combines convolutions and Mamba SSM at the block level with their proposed SRCM block, incorporated global-local guided fusion of intermediate encoder features, and a decoder utilizing the SRCM block for processing the fused bitemporal feature map.

III. Experimental Method

A. MambaVision

MambaVision is a recently proposed vision backbone model that incorporates convolution, Transformer, and Mamba blocks.

For context, the Transformer architecture is initially introduced in [16] which uses self-attention to process dependencies between features at a global scale with a time complexity of $O((HW)^2)$ for a feature map with spatial size (H, W). Meanwhile, the newer Mamba architecture introduced in [22] processes each token sequentially in an autoregressive nature by iterating over every pixel and processing the output at that token's location w.r.t past tokens. Mamba has a time complexity of $O(HW)$ which offers a significant speed up compared to self-attention especially for spatially larger feature maps. A visual comparison of how self-attention and Mamba processes image patches is shown in Figure 1.

However, the authors noted the main issue with Mamba's autoregressive formulation is that it treats the entire feature map as a sequence where each feature is assumed to have sequential dependencies with each other, and cannot capture dependencies in the next timestep and beyond as Mamba processes the feature sequence step-by-step, which limits the ability to capture global context.

For the macro-architecture of MambaVision, the first two levels use purely convolution blocks in order to extract finer

details and short-range dependencies while aggregating information from surrounding pixels, while the deeper levels use a combination of Mamba and Transformer blocks, where the Transformer blocks enhance the model's ability to capture global context and long-range dependencies. Specifically, with N blocks in a level, there are N/2 Mamba blocks and N/2 Transformer blocks arranged sequentially where the Transformer blocks are placed in the last half of all the blocks, as this setup attains the best performance in their ablation study. For the micro-architecture of MambaVision, the author redesigned the original Mamba mixer to make it more suitable for vision tasks by firstly replacing causal convolutions with regular convolutions to capture dependencies in both directions, and added a symmetric branch without SSM with an additional convolution and Sigmoid Linear Unit (SiLU) activation to restore information that is lost due to the sequential nature of SSMs.

B. ChangeFormer decoder

The ChangeFormer decoder [20] utilizes convolutional blocks for extracting the feature difference map from before and after feature maps, and utilizes MLP layers to aggregate the multi-level feature difference maps to predict the change map. Firstly, features at each level are processed by an MLP layer to the same dimensionality D_{embed}. The feature difference map is produced by concatenating the before- and after-feature map pairs channel-wise, and then applying Conv2D, ReLU and BatchNorm2d (BN) to the feature difference map sequentially which compresses the channel dimension by half, effectively performing difference fusion of the feature map pair. Then multi-level feature fusion is performed, where the feature difference maps of each level is upsampled to the same size (H/4, W/4 where H and W are spatial dims of original image), channel-wise concatenated, and a single MLP layer is applied to perform dimensionality reduction back to D_{embed}. Finally, the resulting feature map is produced by upsampling with ConvTranspose2d, the upsampled feature maps are refined by residual blocks, and the change map prediction is produced by the final MLP layer that maps the channel dimension D_{embed} to the number of classes.

C. ChangeMambaVision

The proposed ChangeMambaVision model, shown in Figure 2, utilizes a minimally modified Siamese MambaVision encoder for multi-level feature extraction, and the ChangeFormer decoder for feature fusion and change map prediction. We modified the MambaVision model as follows:

1) For the initial patch embedding layer (PatchEmbed), BatchNorm2d layers are changed to LayerNorm2d.
2) LayerNorm2d is applied to the features after every level and before downsampling to the next level.
3) MambaVision.forward() is modified to return intermediate features at each level after the normalization layer introduced at modification (2).
4) kwargs['resolution'] is changed from 224 to 256 to match the size of the patched dataset images.
5) kwargs['dims'] is changed from scaling by level index to explicit (e.g. $D_{level_i} = dim \times 2^i$ is changed to $D_{level_i} = D_i$ where D_i corresponds to an array of increasing channel dimensionality e.g. [96, 192, 384, 768]).

These modifications, especially in (1), (2) and (3), are applied to make MambaVision more in line with the encoder part of original ChangeFormer (which uses LayerNorm when downsampling) without drastically altering the original behavior of the model. The visual of the applied changes is shown in Figure 3. Note that the ChangeFormer decoder remains unmodified.

The ChangeFormer decoder projects the multi-level embeddings into the same dimensionality D_{embed}. As MambaVision is available in multiple checkpoints that has differing feature dimensionalities between checkpoints, we made it so that $D_{embed} = D_2 \times 2$. This selection of D_{embed} is to ensure balance between efficiency and representational ability while being compatible with all checkpoints available in MambaVision.

D. Experimental Setup

We implemented our approach using PyTorch and trained the proposed ChangeMambaVision model using the NVIDIA A100 GPU as provided by Google Colab. For data augmentation, we used RandomHorizontalFlip(p=0.5), RandomVerticalFlip(p=0.5) and RandomResizedCropPair(size=(256, 256), scale=(0.8, 1.0), ratio=(1/1, 1/1)) for flip-invariance and soft scale-invariance. Training is done for 200 epochs, where the optimizer used is SGD(lr=0.01, momentum=0.9, weight_decay=5e-4) with a linear-decaying scheduler that decays the learning rate to zero across 200 epochs. The batch size during training is set to 8, and the batch size during validation and testing is set to 16. The selected loss function is Cross-Entropy Loss that takes in two classes (change/no change) without any class weighting. The selection of hyperparameters ensures that no hyperparameter tuning is used for performance optimization in order to maintain fairness of evaluation. The train-eval loop and test set evaluation is implemented with PyTorch Lightning for easy logging, resuming, and code cleanliness. For ease of reproducibility, we set the random seed to 42 for all experiments. The AdamW optimizer is used in the ablation study detailed in Section 5C.

IV. Results and Discussion

To measure the effectiveness of ChangeMambaVision on Change Detection tasks, previous state-of-the-art architectures are selected, namely the original ChangeFormer [16], ChangeMamba [20], and CDMamba [21]. Due to limitations on compute in Google Colab, we chose not to represent older CD models, however we believe the selected models are good representatives of the performance of recent CD approaches, and that the CDMamba model serves as a good state-of-the-art baseline due to its recency and strong reported metrics with respect to other CD models' reported metrics in the CDMamba paper. For fairness of comparison, every model is trained from scratch, and the selected hyperparameters as described

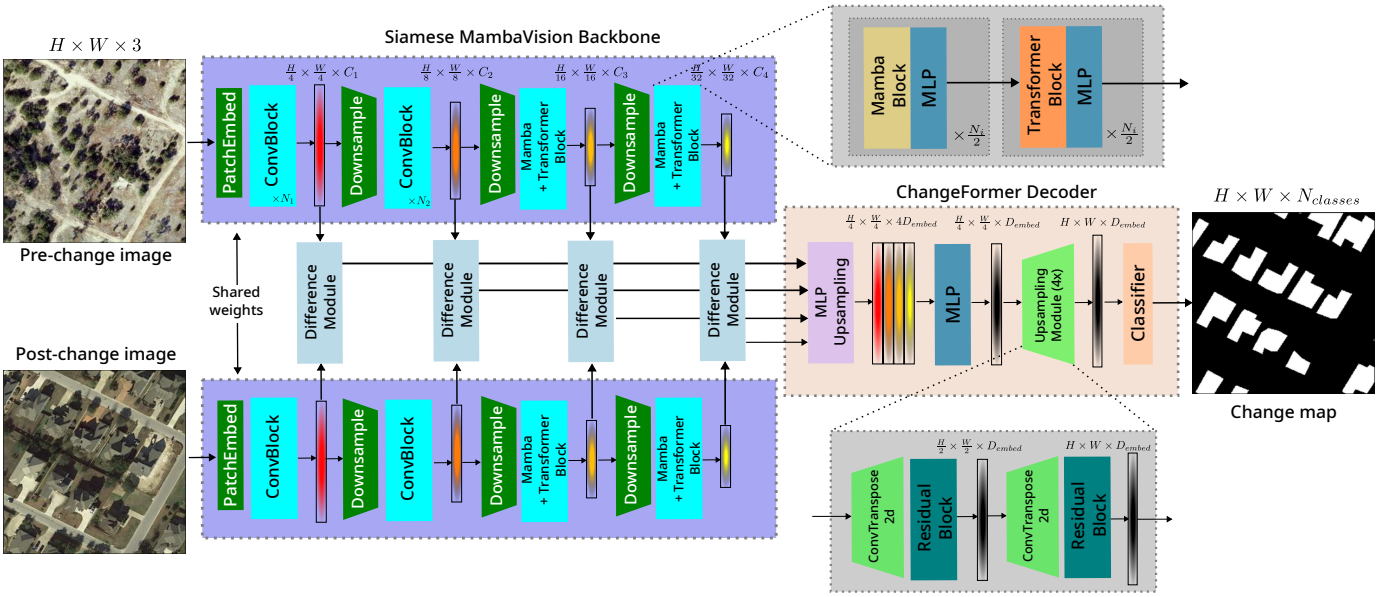

Fig. 2: The proposed ChangeMambaVision architecture, utilizing Siamese MambaVision backbone for feature extraction, and ChangeFormer decoder for difference modules, feature fusion and classification of change/no-change pixels.

TABLE I: Quantitative comparison on LEVIR-CD and WHU-CD datasets. Red indicates best, blue indicates second best, and bold indicates third best.

Models	LEVIR-CD					WHU-CD				
	Pre	Rec	OA	F1	IoU	Pre	Rec	OA	F1	IoU
ChangeFormer$_{22}$	89.52	85.40	98.75	87.41	77.64	83.72	80.45	98.70	82.05	69.57
ChangeMamba$_{24}$	90.39	85.99	98.82	88.14	78.79	**86.41**	**85.43**	98.97	**85.91**	**75.31**
CDMamba$_{25}$	90.75	88.32	98.95	89.52	81.03	90.57	85.23	99.13	87.82	78.29
ChangeMambaVision-T	91.34	88.67	98.99	89.99	81.80	84.75	82.55	98.81	83.63	71.87
ChangeMambaVision-S	**90.95**	88.50	**98.97**	89.71	**81.33**	83.95	87.50	98.92	85.69	74.96
ChangeMambaVision-B	91.53	88.67	99.01	90.08	81.95	88.67	87.95	99.14	88.31	79.06

TABLE II: Number of parameters (in millions), peak training memory usage (in megabytes), throughput in the test set (in images per second) for each model in the WHU-CD dataset.

Model	#Params (M)	Memory (MB)	Throughput
ChangeFormer$_{22}$	41.03	3805	216.16
ChangeMamba$_{24}$	52.03	8293	84.32
CDMamba$_{25}$	11.91	14251	58.88
ChangeMambaVision-T	57.55	5464	153.12
ChangeMambaVision-S	87.30	6685	134.72
ChangeMambaVision-B	163.99	10928	99.20

in Section IV are kept the same across experiments. For models from past works, we chose the default configuration because 1) we use a data preprocessing pipeline (patchify and augmentation) that are similar to the ones detailed in the past models' respective papers, and 2) we assume the authors intend for future research to use the set of configurations for their models which are tuned to similar CD tasks.

A. *Quantitative Results*

Testing-set performance of each model are measured with five metrics in line with previous CD research: namely Precision (Pre), Recall (Rec), Overall Accuracy (OA), F1-score of the positive/change class (F1), and Intersection over Union of the positive/change class (IoU).

Table I shows the testing set performance of selected models in both LEVIR-CD and WHU-CD datasets. In LEVIR-CD, all ChangeMambaVision checkpoints surpass the current SOTA baseline, and in WHU-CD the proposed ChangeMambaVision-B attains the best performance across most metrics.

In the WHU-CD dataset however, ChangeMamba and CDMamba obtained better overall performance compared to the T and S checkpoints of ChangeMambaVision. ChangeMambaVision-B loses in Precision metric against CD-

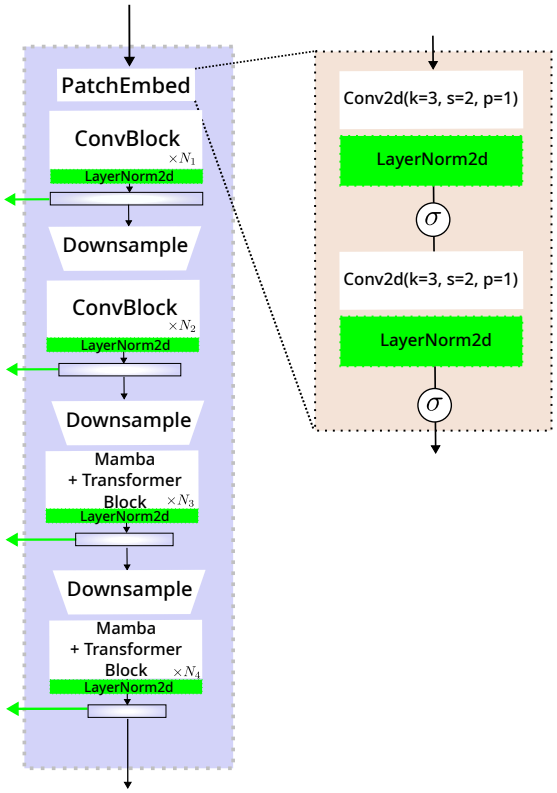

Fig. 3: The modifications applied to the model as shown by the layers and arrows in light green.

Mamba which means that more false positives are predicted i.e. ChangeMambaVision-B is overpredicting towards change with respect to CDMamba. However ChangeMambaVision-B has far higher Recall compared to CDMamba which means that there are less predicted false negatives i.e. ChangeMambaVision-B misses less changes when they occur.

Additionally, in Table II we compare the number of parameters, the peak amount of memory (in MB) used during training in the WHU-CD dataset, and the testing-set throughput in image-pairs per second on an NVIDIA A100 GPU given a training batch size of 8 and a testing batch size of 16.

Despite ChangeMambaVision-B's large number of parameters compared to CDMamba, it is more efficient in memory usage during training and has a higher throughput in the test-set. The large memory training cost and slower throughput of CDMamba could be attributed to its finer granularity on forward passes, as CDMamba does not perform N/4 downsampling initially, which quadratically increases the sequence length, substantially slowing down inference, and increasing memory usage for the forward and backward passes.

When only inference is performed without training, CDMamba only uses 2347MB of memory, while ChangeMambaVision-B uses 10054MB of memory. This shows the significant memory overhead required for the backward pass on models with finer granularity, even with a significantly lower number of parameters in mind.

TABLE III: Performance comparison on the WHU-CD dataset when training with AdamW. Values in parentheses denote metric change relative to the respective SGD trained model.

Models	WHU-CD				
	Pre	Rec	OA	F1	IoU
CDMamba_{25}	84.41	84.08	98.84	84.25	72.78
	(-6.16)	(-1.15)	(-0.29)	(-3.57)	(-5.51)
ChangeMambaVision-B	**86.37**	**89.61**	**99.10**	**87.96**	**78.50**
	(-2.20)	(+1.66)	(-0.04)	(-0.35)	(-0.56)

B. Qualitative results

We selected one representative patch, its change label and change predictions of every model for each dataset in Figure 4. In the LEVIR-CD dataset patch, ChangeMambaVision-B's prediction more precisely constrains the actual changed regions compared to CDMamba, but CDMamba still has a cleaner change map with less "change islands" that are not part of the change regions. In the WHU-CD dataset, ChangeMambaVision-S and ChangeMambaVision-B have no noticeable gaps within the predicted change regions, while the other models have noticeable gaps in them. As the checkpoints of ChangeMambaVision get larger, the predicted change regions more precisely constrains the actual change regions, demonstrating its scalability. Compared with the original ChangeFormer in both datasets, only by changing the encoder/backbone to MambaVision shows a noticeable increase in the quality of the change predictions in both dataset patches even at the -T checkpoint, which suggests that the ChangeFormer decoder still can benefit from a stronger backbone and larger D_{embed} despite its simplicity.

C. Ablation Study

Ablation study: We retrained ChangeMambaVision-B and CDMamba with AdamW(lr=1e-4, betas=[0.9, 0.999], eps=1e-6, weight_decay=5e-4) optimizer which is consistent with the optimizer in the CDMamba training setup. We do not change any other hyperparameters as detailed in Section 4.3 and do not perform any hyperparameter tuning to isolate the impact of optimizer choice on the selected models' performance. ChangeMambaVision-B and CDMamba are chosen specifically due to their almost comparable performance on the WHU-CD dataset.

The testing-set performance of both models on the WHU-CD test split is shown in Table III. With the AdamW optimizer, ChangeMambaVision-B obtains slightly worse but still comparable performance to the SGD-trained metrics, while CDMamba suffers a substantial performance drop in all metrics. Recall is observed to change more positively compared to Precision, which suggests that AdamW pushes both models to predict more positive labels and often overpredicting, hence the more negative change in Precision observed in both models. AdamW also made both models overfit early on at around the 40-80 epochs range during training.

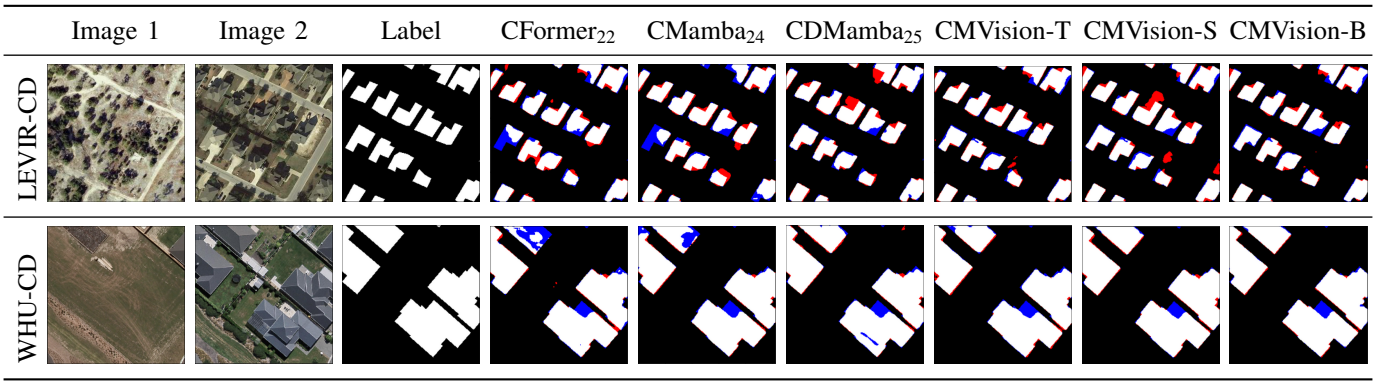

Fig. 4: Qualitative comparison on LEVIR-CD and WHU-CD datasets. Red and blue pixels indicate false positives and false negatives, respectively.

V. Conclusion

In this paper, we proposed ChangeMambaVision, a model that utilizes the efficiency and representational power of the MambaVision backbone, with a simple and lightweight ChangeFormer decoder. We applied modifications to the normalization layers on the MambaVision model for compatibility with the ChangeFormer decoder while keeping the original model's behaviour relatively unchanged. After training ChangeMambaVision with the -T, -S and -B checkpoints as well as some representative CD models, including CDMamba as our SOTA baseline, we find that ChangeMambaVision-B attained state-of-the-art metrics in LEVIR-CD and WHU-CD with lower memory costs during training and a higher throughput in the testing set compared to the SOTA baseline.

We have shown that a simple adaptation strategy of MambaVision is able to achieve strong metrics in CD. Due to limited compute, we leave more complex exploration of ablations to future work such as increasing granularity or using depth-wise convolutions for parameter efficiency. Future work can also improve current CD models by proposing custom fusion or differencing modules, different downsampling/upsampling strategies, or even a completely different decoder structure that helps push the state-of-the-art. Future work can also adapt the MambaVision and other vision backbones to other bi-temporal tasks such as Semantic CD or building damage assessment in order to support a wider range of tasks.

Acknowledgment

This is a derivative work of the original MambaVision architecture. MambaVision is provided by NVIDIA as an open-source architecture available in (https://github.com/NVlabs/MambaVision) under NVIDIA Source Code License-NC (license available at https://github.com/NVlabs/MambaVision/blob/main/LICENSE) which allows MambaVision to be used for non-commercial or research purposes. This research uses data from the LEVIR-CD dataset, available in (https://justchenhao.github.io/LEVIR/). This research uses data from the WHU-CD dataset, available in (https://gpcv.whu.edu.cn/data/building dataset.html).

Author Contribution

Conceptualization, T.G.; methodology, T.G., A.S.K., software and experiments, T.G., paper writing and editing, T.G.; paper review, T.G., A.S.K., E.I.; supervision, A.S.K., E.I., B.P.; All authors have read and agreed to the published version of the manuscript.

References

[1] G. Cheng, Y. Huang, X. Li, S. Lyu, Z. Xu, Q. Zhao, and S. Xiang, "Change Detection Methods for Remote Sensing in the Last Decade: A Comprehensive Review," May 2023.

[2] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, "Deep Learning-Based Change Detection in Remote Sensing Images: A Review," *Remote Sensing*, vol. 14, no. 4, p. 871, Jan. 2022.

[3] S. Das and D. P. Angadi, "Land use land cover change detection and monitoring of urban growth using remote sensing and gis techniques: a micro-level study," *GeoJournal*, vol. 87, no. 3, p. 2101–2123, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1007/s10708-020-10359-1

[4] C. M. Viana, S. Oliveira, S. C. Oliveira, and J. Rocha, *Land Use/Land Cover Change Detection and Urban Sprawl Analysis*. Elsevier, 2019, p. 621–651. [Online]. Available: http://dx.doi.org/10.1016/B978-0-12-815226-3.00029-6

[5] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xBD: A Dataset for Assessing Building Damage from Satellite Imagery," Nov. 2019.

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Jan. 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015.

[10] T. S. Arulananth, P. G. Kuppusamy, R. K. Ayyasamy, S. M. Alhashmi, M. Mahalakshmi, K. Vasanth, and P. Chinnasamy, "Semantic segmentation of urban environments: Leveraging u-net deep learning model for cityscape image analysis," *PLOS ONE*, vol. 19, no. 4, p. e0300767, Apr. 2024. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0300767

[11] L. Garg, P. Shukla, S. K. Singh, V. Bajpai, and U. Yadav, "Land use land cover classification from satellite imagery using munet: A modified unet architecture." in *VISIGRAPP (4: VISAPP)*, 2019, pp. 359–365.

[12] T. W. Cenggoro, B. Pardamean, J. Gozali, D. Tanumihardja *et al.*, "Detection of pulmonary tuberculosis from chest x-ray images using multimodal ensemble method," *Commun. Math. Biol. Neurosci.*, vol. 2022, pp. Article–ID, 2022.

[13] N. A. Heryanto, M. Isnan, M. M. Henry, and B. Pardamean, "Semi-automated meningioma segmentation with bounding boxes," *Procedia Computer Science*, vol. 245, p. 583–590, 2024. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2024.10.285
[14] R. C. Daudt, B. L. Saux, and A. Boulch, "Fully Convolutional Siamese Networks for Change Detection," Oct. 2018.
[15] R. C. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "Multitask Learning for Large-scale Semantic Change Detection," Aug. 2019.
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Aug. 2023.
[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 2021.
[18] W. Tang, F. He, A. K. Bashir, X. Shao, Y. Cheng, and K. Yu, "A remote sensing image rotation object detection approach for real-time environmental monitoring," *Sustainable Energy Technologies and Assessments*, vol. 57, p. 103270, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.seta.2023.103270
[19] A. A. Hafiz, D. Vericho, V. J. Carter, D. C. Thio, M. Isnan, and B. Pardamean, "Vision transformer and cnns in kidney stone classification: A comparative study," *Procedia Computer Science*, vol. 269, p. 1466–1473, 2025. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2025.09.088
[20] W. G. C. Bandara and V. M. Patel, "A Transformer-Based Siamese Network for Change Detection," Sep. 2022.
[21] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
[22] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," May 2024.
[23] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "VMamba: Visual State Space Model," Dec. 2024.
[24] H. Chen, J. Song, C. Han, J. Xia, and N. Yokoya, "ChangeMamba: Remote Sensing Change Detection With Spatiotemporal State Space Model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–20, 2024.
[25] H. Zhang, K. Chen, C. Liu, H. Chen, Z. Zou, and Z. Shi, "CDMamba: Incorporating Local Clues into Mamba for Remote Sensing Image Binary Change Detection," May 2025.
[26] A. Hatamizadeh and J. Kautz, "MambaVision: A Hybrid Mamba-Transformer Vision Backbone," Mar. 2025.
[27] "LEVIR-CD," https://justchenhao.github.io/LEVIR/.
[28] "WHU Building change detection dataset," https://gpcv.whu.edu.cn/data/building_dataset.html.
[29] S. Fang, K. Li, J. Shao, and Z. Li, "SNUNet-CD: A Densely Connected Siamese Network for Change Detection of VHR Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
[30] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165
[31] H. Chen, Z. Qi, and Z. Shi, "Remote Sensing Image Change Detection with Transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
[32] L. Song, M. Xia, L. Weng, H. Lin, M. Qian, and B. Chen, "Axial Cross Attention Meets CNN: Bibranch Fusion Network for Change Detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 21–32, 2023.