IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 March 2024, accepted 11 April 2024, date of publication 19 April 2024, date of current version 3 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3391416

RESEARCH ARTICLE

Building Precision: Efficient Encoder–Decoder Networks for Remote Sensing Based on Aerial RGB and LiDAR Data

MUHAMMAD SULAIMAN[1], ERIK FINNESAND[1], MINA FARMANBAR[1], AHMED NABIL BELBACHIR[2], (Member, IEEE), AND CHUNMING RONG[1,2], (Senior Member, IEEE)

[1]Department of Electrical Engineering and Computer Science, University of Stavanger, 4021 Stavanger, Norway
[2]NORCE Norwegian Research Centre, 5008 Bergen, Norway

Corresponding author: Muhammad Sulaiman (muhammad.sulaiman@uis.no)

This work is supported by the project EMB3DCAM, "Next Generation 3D Machine Vision with Embedded Visual Computing," and co-funded under grant number 325748 of the Research Council of Norway.

ABSTRACT Precision in building delineation plays a pivotal role in population data analysis, city management, policy making, and disaster management. Leveraging computer vision technologies, particularly deep learning models for semantic segmentation, has proven instrumental in achieving accurate automatic building segmentation in remote sensing applications. However, current state-of-the-art (SOTA) techniques are not optimized for precisely extracting building footprints and, specifically, boundaries of the building. This deficiency highlights the need to leverage Light Detection and Ranging (LiDAR) data in conjunction with aerial RGB and streamlined deep learning for improved precision. This work utilizes the MapAI dataset, which includes a variety of objects beyond buildings, such as trees, electricity lines, solar panels, vehicles, and roads. These objects showcase diverse colors and structures, mirroring the rooftops in Denmark and Norway. Due to the aforementioned problems, this study modified UNet and CT-UNet to use LiDAR data and RGB images to segment buildings using Intersection Over Union (IoU) to evaluate building overlap and Boundary Intersection Over Union (BIoU) to evaluate precise building boundaries and shapes. The proposed work changes the configuration of these networks to streamline with LiDAR data for efficient segmentation. The batch data in training is augmented to improve model generalization and overcome overfitting. Batch normalization inclusion also improves overfitting. Four backbones with transfer learning are employed to enhance convergence and parameter efficiency of segmentation: ResNet50V2, DenseNet201, EfficientNetB4, and EfficientNetV2S. Test-Time Augmentation (TTA) is employed to improve the predicted mask. Experiments are performed using single and ensemble models, with and without Augmentation. The ensemble model outperforms the single model, and TTA also improves the results. LiDAR data with RGB improves the combined score (average of IoU and BIoU) by 13.33% compared to only RGB images.

INDEX TERMS Building precision, deep learning, LiDAR, remote sensing, semantic segmentation, U-Net, context-transfer U-Net.

I. INTRODUCTION

Terrestrial LiDAR is a sophisticated, active tool for remote sensing purposes that utilizes laser pulses to illuminate a specific area and then receives the reflected pulses bounced back by the objects within that area [1]. The process allows for the creation of precise 3D representations of the scene, which contain x and y coordinates to illustrate the location corresponding to other objects and z as the depth of the sensing objects. This tool is widely used in various fields,

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

such as geography, geology, and engineering, to obtain accurate and detailed information about the surrounding area's topography, structure, and composition.

Airborne LiDAR systems are extensively required to acquire high-resolution data over large areas, i.e., urban regions [2], [3], [4]. These systems are typically affixed to planes, helicopters, or drones and consist of four primary components: an Inertial Measurement Unit (IMU), a global positioning system (GPS), a laser scanner, and a computer. In the process of data acquisition, the airborne traverses the target area and laterally emits pulses of near-infrared light. The sensor captures the reflected light, registering the duration it takes for the light to travel from the laser to the object and back. The GPS monitors the airborne vehicle's altitude and location while the IMU keeps track of its orientation and speed. The precise determination of the location where the laser pulse was reflected becomes achievable by integrating data from the GPS for location, IMU for accurate spatial orientation with speed, and the recorded time from the sensor. Subsequently, a computer is employed to process and manage this comprehensive dataset.

In the past few decades, a huge effort has been made to design innovative models for building extractions from remote sensing images, while accurate building extraction with precise boundaries is still challenging for the computer vision community. Building extraction in remote sensing images is challenging due to three issues: first, various other objects like electricity lines and trees cover the building [5]; second, building shadows and band reflectance; third, high-resolution images make it challenging to segment building boundaries accurately. Building extraction is mainly based on aerial images [6] (RGB), Lidar-based [7], [8], and fusion-based [9].

The significance of building extraction through remote sensing LiDAR data is multifaceted, contributing to diverse fields such as urban planning [6], environmental management [10], disaster management [11], and geospatial analysis. LiDAR-derived building information is crucial in urban planning as it offers insights into land use patterns, supports zoning decisions, and monitors urban growth. In environmental management, the data aids in assessing the ecological impact of urbanization and analyzing green space distribution. For disaster management, accurate building information enhances vulnerability assessments and emergency response planning. Building extraction contributes to spatial modeling, 3D visualization, and simulations in geospatial analysis. Moreover, the data is valuable for infrastructure monitoring, supporting asset management utility planning, contributing to a comprehensive understanding of urban landscapes, fostering sustainable development, and making informed decisions across various academic disciplines.

A. PROBLEM DESCRIPTION

In recent years, advancements in remote sensing through LiDAR technology have introduced novel challenges, particularly in pixel classification based on depth information in satellite imagery. Broadly, image segmentation encounters numerous difficulties, with one of the most formidable issues arising when attempting to simultaneously utilize RGB and LiDAR data for training on a specific dataset and achieving robust generalization on a distinct test set (the same is the case with the MapAI dataset). This challenge becomes notably pronounced in satellite imagery, where images in the test set may be affected by varying illumination conditions or pertain to different geographical areas than those represented in the training set [12].

This study used the MapAI dataset, which relies on precision in building segmentation challenge held at a conference organized by the Norwegian Artificial Intelligence Research Consortium (NORA) in cooperation with the University of Agder (CAIR) [13]. The MapAI competition consists of two tasks: the first task exclusively utilizes aerial images, while the second task requires laser data (LiDAR) either alone or in conjunction with aerial images. The primary goal is to formulate models for building segmentation with precise boundary delineation. The secondary objective involves comparing the outcomes of both tasks to assess the impact of LiDAR data in conjunction with aerial images on the accurate segmentation of buildings.

B. DATASET

The dataset utilized in this study is the MapAI dataset [13], and Figure 1 offers a visual depiction of sample images from this dataset. The dataset comprises diverse components, including aerial images, LiDAR data, and ground truth masks. White pixels in ground truth illustrate buildings, while black pixels depict the ground surface and other objects except buildings.

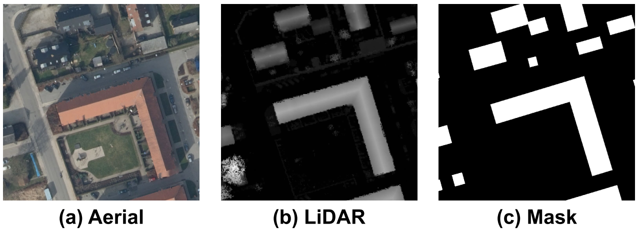

(a) Aerial (b) LiDAR (c) Mask

FIGURE 1. MapAI dataset sample images, each having 500 × 500 pixel resolution. (a) RGB format (b) Float32 (c) Binary.

The dataset consists of train, validation, task1_test, and task2_test. The training set consists of 7500 images, while the validation set comprises 1500 images, both drawn from diverse locations throughout Denmark. This diversity ensures that the training and validation sets encompass various environments and types of buildings. The two test sets consist of 2346 (1368+978) images. The test set is thoroughly curated to span seven locations across Norway, covering a mix of rural and urban settings such as Bergen, Kristiansand, etc in Norway.

The LiDAR data in the published dataset is preprocessed data that omits other information like elevation and CRS. The data originates from a production setting, implying that buildings may not be properly accurately marked in the masks, and discrepancies may exist between ground truth masks and actual buildings due to production errors. Furthermore, the ground truth masks are created using a Digital Terrain Model (DTM) in the test splits. Both the Digital Surface Model (DSM) and DTM are employed, where continuous terrain surfaces are marked using DTM. Consequently, the roofs of the buildings in the test sets may appear distorted compared to the masks. However, in the training and validation set, this issue is mitigated as the masks are created via DSM, capturing the artificial attributes of the environment to avoid distortion in the building tops [14].

C. EVALUATION METRICS

The MapAI competition evaluates image segmentation using region-based and boundary-based metrics. Region-based metrics, such as Intersection over Union (IOU) or Jaccard Index (JI), measure the similarity between two binary images: the ground truth image I_g and the predicted mask I_p. These metrics calculate the intersection area divided by the total area shown in Equation 1 [15].

On the other hand, boundary-based evaluation metrics, such as Boundary Intersection Over Union (BIOU), offer a different perspective when assessing the performance of models in tasks like image segmentation. Unlike pixel-wise evaluation metrics that focus on the classification of each pixel, BIOU evaluates the intersection over the union of the edged ground truth and edged prediction mask. It takes into account the thickness of the edge from the contour line in Equation 2 [16]. These metrics provide a way to assess the performance of image segmentation algorithms and compare them to the ground truth. Equation 3 presents score as a combined metric of IoU and BIoU as official metrics for MapAI competition. The score is the average of IoU and BIoU values for all images.

$$IoU = JI = \frac{Intersection}{Union} = \frac{|I_g \cap I_p|}{|I_g| + |I_p| - |I_g \cap I_p|} \tag{1}$$

$$BIoU = \frac{|(I_{gd} \cap I_g) \cap (I_{pd} \cap I_p)|}{|(I_{gd} \cap I_g) \cup (I_{pd} \cap I_p)|} \tag{2}$$

$$Score = \frac{IoU + BIoU}{2} \tag{3}$$

D. CONTRIBUTION

The primary goal of this study is to design an enhanced Encoder–Decoder architecture for segmenting buildings and assess the impact of LiDAR data on segmentation. Two encoder–decoder architectures with different backbones are modified, employed, and compared to achieve precise building segmentation for remote sensing. Four backbones (ResNet50V2, DenseNet201, EfficientNetB4, and EfficientNetV2S) are utilized to improve feature extraction, enhance generalization, and reduce overfitting. In experiments, batch normalization is positioned before and after Relu to determine the optimal placement in the models. Multiple initial learning rates and various loss functions are tested on the validation data to identify models best learning rate and efficient loss function. In addition to testing, models are evaluated on training and validation data to detect overfitting. Training augmentation and test time augmentation (TTA) are employed for both single and ensemble models to assess the improvement of ensemble model performance.

II. LITERATURE

Semantic segmentation models are pivotal in the accurate classification and monitoring of objects, and they play a fundamental role in various application fields such as environment protection [17], urban area management [18], [19], [20], [21] and resource management [22] and monitoring [23]. Therefore, semantic segmentation is a powerful technique that can achieve pixel-level building classification in remote-sensing images. In shallow models, XGBoost and LightGBM show promising results on the same dataset [24]. Groundbreaking technology in computer vision has emerged with Convolutional Neural Networks (CNNs) in this context, with their productive creation as Dropout [25], RasNet [26], and Batch Norm [27]. CNN achieves semantic representations using different encoder–decoder designs [28], [29], [30], [31] using convolution and pooling operation and later restores the exact image size by upsampling [32].

The large quantity of data also opens an opportunity to use Deep Learning (DL) to minimize the need for expert domain knowledge. Many Deep Learning model calculations can be parallelized on modern hardware, such as graphical processing units, substantially reducing computation time. The deep learning model is widely used in the segmentation of large-scale publically available datasets of building footprints in the African continent [33], the United States [34], [35], other countries [36], and this study used datasets collected from Denmark and Norway.

Liu et al. [37] proposed a Context-Transfer-UNet (CT-UNet) network to address the poor recognition of high-resolution images and intra-class inconsistency. The CT-UNet network design includes the Dense Boundary Block (DBB) to refine features and solve the fuzzy boundary problem and the Spatial Channel Attention Block (SCAB) to handle intra-class inconsistency. The CT-UNet achieves promising results on the WHU and Massachusetts datasets, outperforming baseline methods and existing approaches. The improved performance is attributed to the feature refinement capability and the defect compensation ability of CT-UNet [37].

Khan et al. [38] designed an encoder–decoder framework to automatically extract building footprints. The encoder utilizes a dense network with convolutional and transition blocks for capturing global multi-scale features, while the decoder employs deconvolution layers to recover lost spatial information, resulting in a dense segmentation map. Training the network in an end-to-end fashion using a hybrid loss enhances overall performance [38].

Zhu et al. [39] propose an Edge-Detail-network (E-D-Net) specifically designed for building segmentation in visible aerial images. E-D-Net comprises two subnetworks: E-Net captures and preserves edge information, while D-Net refines E-Net results to achieve predictions with higher detail quality. Additionally, a fusion strategy combines the outputs of both subnetworks, integrating edge information with fine details [39].

Yang et al. [34] compare four state-of-the-art CNN architectures for extracting building footprints across the entire continental United States to exploit the scalability of convolutional neural networks (CNNs) and leveraging areas with abundant building footprints. The evaluated CNNs, including Branch-out CNN, fully convolutional network (FCN), conditional random field as recurrent neural network (CRFasRNN), and SegNet, specialize in semantic pixel-wise labeling and emphasize capturing multiscale textural information. The evaluation employs 1-meter resolution aerial images from the National Agriculture Imagery Program and compares the extraction results of the four methods. Additionally, it proposed enhancing SegNet, identified as the preferred CNN architecture through extensive evaluations, by combining signed-distance labels to advance building extraction results to the instance level. Furthermore, the utility of incorporating additional near-infrared information is demonstrated in the building extraction framework [34].

Hong et al. [40] proposed a Multi-Task Learning (MTL) framework using a Swin transformer as a backbone and lightweight Building Extraction (BE) and Change Detection (CD) heads as a decoder. The MTL architecture utilizes a Siamese network with shared weights for feature extraction, allowing effective handling. The model employs different heads for tasks, such as using a multilayer perceptron (MLP) for building labels and a convolution-based head for CD [40].

Wei et al. [41] proposed BuildMapper for producing effective building polygons, comprising a contour initialization module for creating building contours and a contour evolution module for contour vertex enhancement to bypass complex post-processing [41].

Luo et al. [42] proposed a domain generalization method, Batch Style Mixing (BSM), to combine classic data augmentation techniques with a new style-mixing method. BSM addresses complex generalization challenges and can be seamlessly integrated into existing building extraction models [42].

Hodne and Furdal [43] utilize two encoders, Resnest26d and EfficientNet-B1, pre-trained on ImageNet, to enhance ensemble diversity. The training process involves multiple models per task, incorporating additional datasets and varying resolutions. An evolutionary algorithm is used to optimize weights for ensemble combinations. Post-processing steps include resizing predictions, filtering based on building area, and employing bilinear interpolation [43].

ATTransUNet, a deep learning model for building segmentation, is proposed by Bicakci and Sarica [44]. The model combines Attention Gated Networks and TransUnet. It employs a hybrid loss function that merges dice and focal losses with a scaled factor. ATTransUNet utilizes a Vision Transformer with specific configurations, including group normalization and GeLu activation function [44].

Borgersen and Grundetjern [45] utilize UNet with ResNet50 as the backbone. They increase the batch size and decrease the learning rate to approximately $1e^{-5}$ while employing the Dice Loss as the chosen loss function. Stable diffusion-based dataset augmentation was introduced as a novel approach, leveraging diffusion models to generate entirely new image features for segmentation challenges [45]

Mrozik et al. [46] identify an autoencoder with a ResNeXt101_32×8d backbone as the most successful model. The authors employ bilinear interpolation for up-scaling feature maps and utilize a series of convolutional and residual blocks. The model is trained with Soft mIoU as the loss function and augmented with random rotations and mirroring to improve performance. The effectiveness of the architecture is illustrated through its ability to produce optimal results on the MapAI dataset, showcasing its potential for image segmentation tasks [46].

Kong et al. [47] proposed a boundary-enhanced network inspired by Zhu et al.'s E-D-Net [39], which adopts the classic encoder–decoder structure with U-Net as the backbone for building segmentation. The first enhancement involves modifying the original bridge block of U-Net by incorporating dilated convolutional layers (DCs) with varying dilation rates (1, 2, 5). The second improvement introduces a new segmentation head after the decoder, featuring two branches that output two classification results for background, boundary, and building. This enhancement improves the network's capability for capturing boundary information [47].

Sørensen et al. utilized a U-Net architecture followed by a conditional random fields denoiser for building segmentation [48]. Losses were applied to both the U-Net raw output and the denoised output, with final predictions based on the denoised output. Performance improvement was achieved by ensembling the top three models for tasks 1 and 2. To address size constraints, mirror-padding of input images and clipping network output were implemented. The ensemble models were implemented in PyTorch using the PyTorch Lightning framework and exported to the Open Neural Network Exchange (ONNX) format. Following the approach by Moshkov et al. [49], test time augmentation was employed by averaging predictions for the final ensemble. Table 1 presents a summary of comparative methods with limitations, which provides insights into the proposed work's advantages over other closely related methods.

TABLE 1. Summary of comparative methods.

Ref	Network	Backbones	Ensembles	Dataset	Limitation
[37]	CT-UNet	ResNet34	No	Inria [50], WHU [51], Massachusetts [52] dataset	Model ensembles with deep backbones could have improved results. ResNet34 is shallow compared to ResNet50V2 used in the proposed. The batch norm after ReLu would be better according to the experiments proposed.
[38]	UNet	ResNet50, DenseNet121, InceptionV1, VGG16, AlexNet	No	Massachusetts [52] and Inria [50] dataset	Ensembling the backbones may improve results, and batch normalization may be missing for generalization.
[39]	E-D-Net	VGG11	No	Inria [50], and ISPRS Vaihingen 2-D [53] dataset	Model ensembles could have improved results further. The batch norm after ReLu would be better according to the experiments proposed. VGG11 is shallow compared to ResNet50V2 used in the proposed.
[40]	MTL	ResNets	Yes	WHU-CD	In addition with Change Detection (CD) label Building Detection (BE) label also required.
[41]	Build Mapper	CNN	No	WHU-Mix	Background of the hollow building cannot be eliminated, and cars are mistaken as buildings.
[42]	AT-MAFCN	VGG-16	No	WHU-Mix	Ensembling and enhanced backbone may improve the results.
[43]	UNet	ResNest26d, EfficientB1	Yes	MapAI [13]	UNet architecture is not modified, and lightweight backbones increase parameters.
[44]	Modified UNet	Vision Transformer (ViT)	No	MapAI [13]	ViT in Ensemble should have improved results.
[45]	UNet	ResNet50	No	MapAI [13]	UNet architecture is not modified. Generalization is limited. The ensemble not utilized.
[46]	Modified UNet	ResNet101	No	MapAI [13]	Ensemble and data augmentation could improve results.
[54]	UNet	ResNet34	Yes	MapAI [13]	ResNet34 is shallow as compared to ResNet50V2 used in the proposed. Batch Normalization missing for generalization.
[55]	UNet	ConvNext, SegFormer-B0, SegFormer-B4, SegFormer-B5	No	MapAI [13]	Ensembling the backbones may improve results.
Proposed	Modified UNet	ResNet50V2, DenseNet201, EfficientNetB4, and EfficientNetV2S	Yes	MapAI [13]	Random ensemble weights are tested and best selected, which could be optimized through an evolutionary algorithm.

III. METHOD

From input dataset to output segmentation, the overall workflow is shown in Figure 2. The dataset block is explained in I-B section. In the preprocessing block, input data is tweaked to be fitted as input for models. Data from each batch is augmented to improve training and overcome the overfitting of the model. Configuration of the proposed models is listed in the second block of training, where red text represents modification as compared to the original models. Test time augmentation improves the robustness and generalization of model prediction by creating ensembles of individual input in test data to minimize noise and uncertainties. In the testing block, the predicted masks are evaluated using mean Intersection over Union (IoU) and Boundary Intersection over Union (BIoU); IoU measures the degree of overlap for the building, and BIoU measures accurate alignment and delineation of building boundaries.

This section is structured to start with dataset preprocessing, where we prepare the dataset to align with the requirements of the models. The second segment focuses on batch data augmentation for training. The third part focuses on developing the model architecture and training, involving configuring diverse model architectures and applying ensemble techniques and data augmentation to enhance performance. The fourth component elaborates on Test-Time Augmentation. The fifth section delves into testing and evaluation, where we test model predictions against ground truth masks, employing various metrics. This section comprehensively evaluates single model predictions with and without TTA and ensemble models with and without TTA.

A. PREPROCESSING

This section provides an in-depth discussion of the preprocessing methods applied to the data. Figure 3 provides an informative flowchart outlining the methodology used in this part of the approach. In the proposed models, downsampling (in the encoder) and upsampling (in the decoder) are performed using max-pooling and transpose convolution (or upsampling) layers, respectively. These operations are typically easier to implement when the input and output sizes are powers of 2, as they allow for clean and predictable resizing of feature maps. In the case of original images with a resolution of 500 × 500 pixels, resizing them to dimensions like 512 × 512 pixels or 448 × 448 pixels is a practical and well-aligned approach. Therefore, in the preprocessing step, images (Aerial, LiDAR, and Mask) are resized to 512 ×

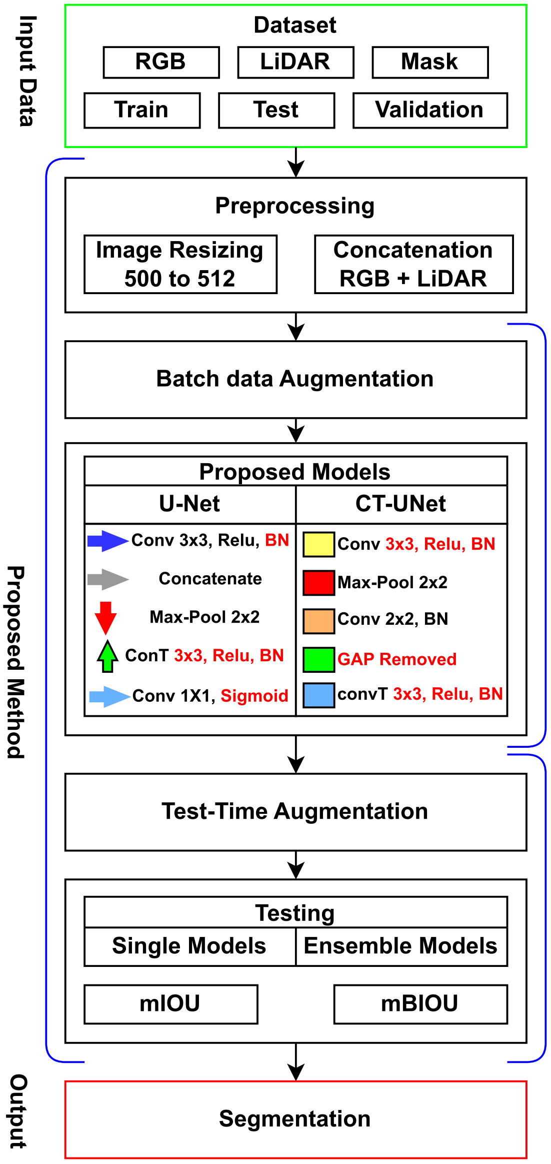


FIGURE 2. Workflow diagram. The input block shows the dataset content. The proposed method block consists of preprocessing (resizing and fusion), training (data augmentation and proposed models), testing (Test-time augmentation and evaluation metrics), and segmenting the image as output. The red text represents modifications for each model architecture to adopt for segmentation on RGB+LiDAR data.

512 pixels, driven by three considerations. First, its minimal difference, compared to 448 pixels, ensures the data retains a high-resolution quality. Second, 512 pixels match the power of 2, which is more convenient for encoder–decoder architecture, simplifying the implementation of operations. Third, due to the usage of pre-trained models in the training step, it is common to use input image size as a power of 2 within the deep learning approaches.

The proposed work transfers LiDAR data from the float32 to uint8 format, which is driven by two primary considerations. First, Tensorflow only supports 8-bit data formats for images with four channels when loading batches from the directories. Second, unit8 is computationally more efficient than float32 values. However, this transition results in a minor loss of information and detail, which is visible in Figure 4. Specifically, we observe a subtle reduction in the regularity of shapes for building tops and pixels near objects at ground level, such as cars and fences. However, it is important to emphasize that the building's shape, location, and overall height remain unaltered, resulting in a limited impact on the training outcomes. Following the pre-processing phase, LiDAR data is integrated into the model input by concatenating it with an RGB image, creating a 4-channel data structure.

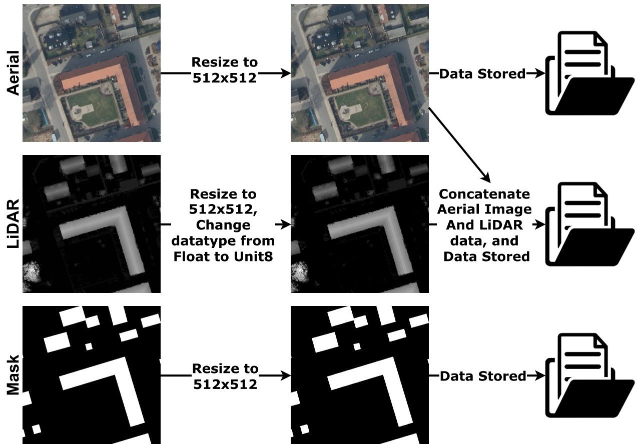
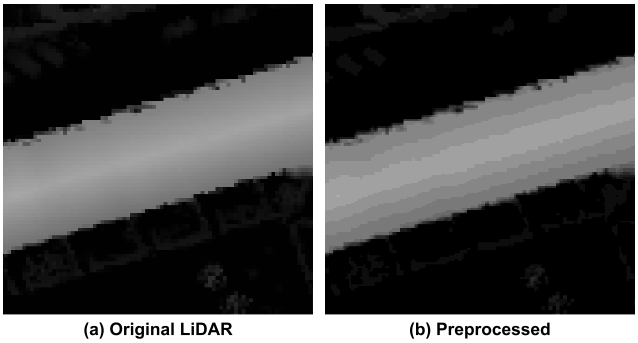

Aerial
Resize to 512x512
Data Stored
LiDAR
Resize to 512x512, Change datatype from Float to Unit8
Concatenate Aerial Image And LiDAR data, and Data Stored
Mask
Resize to 512x512
Data Stored

FIGURE 3. Pre-processing Overview, to align data with model requirements.


(a) Original LiDAR

(b) Preprocessed

FIGURE 4. Zoomed illustration of Original and Preprocessed LiDAR data of a building roof. Almost negligible roughness in contour appeared.

B. BATCH DATA AUGMENTATION

Due to the massive size of the dataset, the Keras data generator is used to load one batch at a time. To further enhance model performance, the online data augmentation first block in Figure 2 is used to avoid overfitting and reduce memory overhead compared to offline data augmentation [56]. Techniques incorporated in data augmentation are - 90 to 90 degrees rotation, width, and height shift alone x-axis and y-axis, respectively, in the interval of 0.7 to 1.3, shear, zoom, horizontal flip, and vertical flip. Figure 5 shows sample images that have undergone online data augmentation.

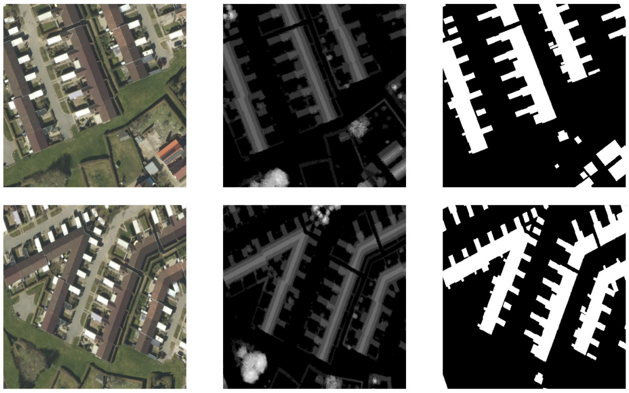


FIGURE 5. Original images in the first row. Augmented images in the second row.

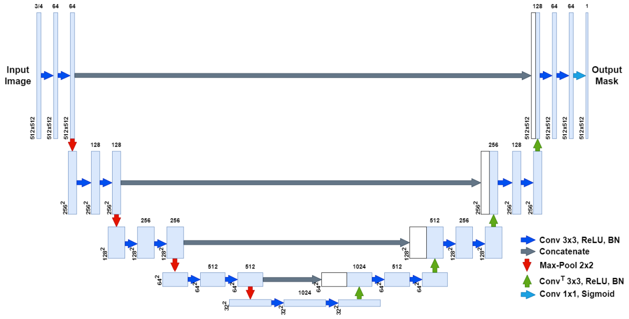


FIGURE 6. Illustration of proposed U-Net architecture adapted for Building Segmentation. Batch Normalization was added after Conv 3 × 3 and Relu. Transpose convolution 3 × 3 is used instead of 2 × 2, followed by Relu and Batch Normalization. For output mask convolution 1 × 1 followed by sigmoid.

C. MODELS

Two different architectures are proposed based on the U-Net model due to its popularity in medical imaging [57], [58]. The first model used the baseline of the original U-Net architecture [31], and the second used the baseline of Context Transfer UNet (CT-UNet) [37]. The proposed work uses U-Net and CT-UNet because they share a foundational encoder–decoder architecture that enables them to capture high-level contextual information and fine-grained spatial details, which is crucial for segmentation tasks. These architectures incorporate skip connections that connect the encoder and decoder, help in precise localization, mitigate the vanishing gradient problem, and facilitate information transfer. The U-shape design makes them versatile enough to adapt to various segmentation tasks. CT-UNet, in particular, incorporates mechanisms for capturing and transferring contextual information across the image, making CT-UNet the best segmentation option.

1) PROPOSED U-NET

Figure 6 illustrates the proposed architecture using baseline U-Net. The input layer is set up with a dimension of 512 × 512 pixels, and the number of channels is either 3 for RGB or 4 for RGB with LiDAR. The encoder side of the original U-Net at each depth is modified with double 3 × 3 convolution, ReLU, and batch normalization. For every convolution, zero-padding has been added to preserve shape consistency between the input and output. A Gaussian normal distribution is utilized to initialize kernels in order to handle the problem of dying ReLU. The initial network depth utilizes 64 kernels for each convolution, doubling this number for subsequent depths. Following these operations, 2 × 2 max-pooling has been employed to shrink the feature maps' spatial dimensions.

On the decoder side, a transposed convolution of 3 × 3 is employed, as opposed to the original U-Net architecture, which uses 2 × 2. The decision to implement this change stemmed from experimental findings revealing that 2 × 2 transposed convolutions could generate a checkerboard pattern in the expanded feature map. This occurrence had the potential to introduce misleading edges, thereby compromising the overall performance of the network. Following the transposed convolutions, ReLU activation and batch normalization are applied. Each depth on the encoder side is concluded with batch normalization. After the concatenation of feature maps, two sequences of 3 × 3 convolutions are executed, mirroring the operations performed on the encoder.

Following that, the architecture undergoes the last convolution using a 1 × 1 singular kernel, and the sigmoid is used as an activation function after the final series of 3 × 3 convolutions. Backbones can be easily integrated into a U-Net network by replacing the encoder part of the architecture. The final feature map obtained from the backbone for each spatial dimension concatenates with the associated decoder feature map.

Batch normalization is included in both the encoder and decoder parts of the original U-Net to stabilize training, faster convergence, and a small amount of reduction in overfitting. Batch Normalization can be included before and after Relu. Figure 7 depicts the outcomes of experiments (utilizing batch size of 6, dice as loss function, and the learning rate of 0.0001) comparing batch normalization before and after. Applying

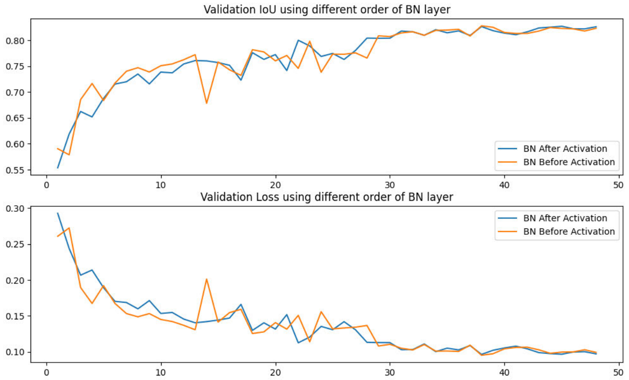


FIGURE 7. U-Net: For different orders of BN, the IoU and Loss curves on the validation set are computed during training.

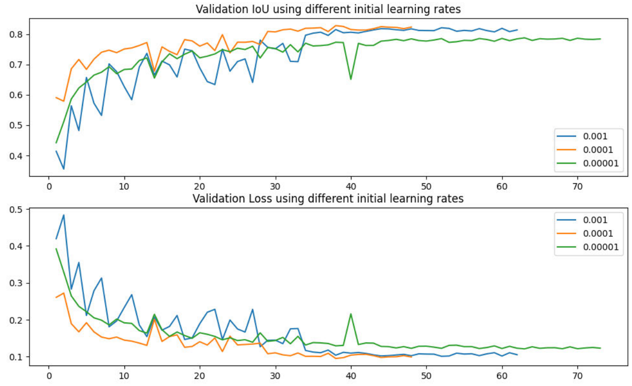


FIGURE 8. U-Net: IoU and Loss curves are computed on the validation set for different initial learning rates during training.

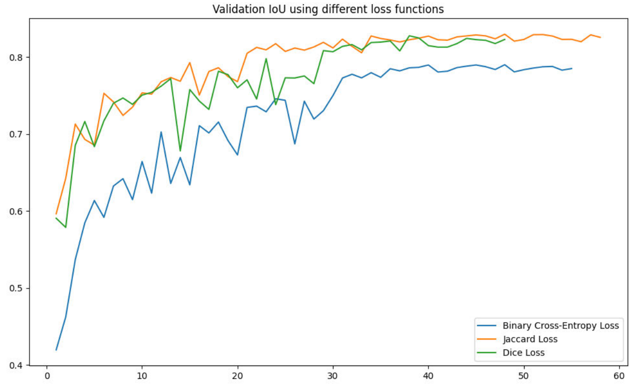


FIGURE 9. U-Net: IoU is computed for Binary Cross-Entropy, Jaccard, and Dice loss on the validation set during training.

batch normalization after ReLu results in a higher validation of 0.0029 in Intersection over Union (IoU). Considering the observed impact of batch normalization from the experiment, the decision has been made to execute batch normalization after the Relu function for the proposed models.

Two experiments are performed on validation data for IOU and loss to establish the U-Net model with optimal initial learning rate. Figure 8 shows experiments with different learning rates with the same dice loss and batch size of 6. The graphs indicate that using ADAM as the optimizer, the highest IoU and fast convergence were achieved with a learning rate of 0.0001.

To determine the best loss function for the models, experiments are performed using three loss functions: Binary Cross-Entropy, Jaccard Loss, and Dice loss, having the best initial learning rate of 0.0001, with ADAM as the optimizer and batch size of 6. The experiment is evaluated on the base of validation IOU. Figure 9 illustrates that Dice and Jaccard loss exhibit similar performance, with Dice loss having a validation Intersection over Union (IoU) slightly higher by 0.0001 than Jaccard loss. Furthermore, unlike Jaccard loss, Dice loss ends the training ten epochs earlier.

2) PROPOSED CT-UNET

A modified version of U-Net called Context-Transfer-UNet (CT-UNet) was created to improve remote sensing image segmentation. In CT-UNet, the encoder portion of the network is replaced with a backbone. Compared to U-Net, CT-UNet has an extra level; it can capture more intricate and abstract patterns. It introduces the Dense Boundary Block (DBB), which is composed of two components called Dense Block (DB) and Boundary Block (BB), along with a Spatial Channel Attention Block (SCAB). DB refines the feature before transferring it to the decoder. DBB transfers low-level features to high-level features to ensure the presence of boundary information at a high level to improve BIOU metrics, and SCAB tries to improve intra-class consistency.

To reduce the training time and increase the network's performance, four backbones, ResNet50V2, DenseNet201, EfficientNetB4, and EfficientNetV2S, are used in CT-UNet as encoders. ImageNet [59] dataset is used to train these backbones using a 3-dimensional input, which makes them appropriate for Task_1, while Task_2 input is 4-dimensional, due to which training from scratch is required. Based on the specific task, the network takes an input size of 512×512 with either three channels for RGB only or four channels for both RGB and LiDAR. As we reach the lower layers of the network, the feature maps have a spatial dimension of 16×16, and the channel count corresponds to the output of the chosen backbone.

Referring to Figure 3: Architecture of CT-UNet in [37], Global Average Pooling (GAP) is replaced with a Squeeze-and-Excitation (SE) block and later removed. The feature map's spatial dimension is eventually reduced to 1×1, leading to a dimension mismatch in the decoder section of the network. This study attempts to substitute the Global Average Pooling (GAP) with a Squeeze-and-Excitation (SE) block. However, the network exhibited improved performance when neither GAP nor SE block was utilized. Notably, in the network's decoder section, the ReLU activation function and Batch Normalization (BN) block swap positions compared to the original network. The final up-sampling block from the original CT-UNet network is eliminated to match input and output dimensions. In the decoder section of the network, modifications have been made to the convolution and transposed convolution operations, utilizing a 3×3 kernel size. Having half as many channels as this particular block, the output of the Dense Block (DB)/DBB block from the previous stage is used as the second input of the Dense Boundary Block (DBB) in the original CT-UNet design. A 1×1 convolution is applied to the second input to adjust its channel count to match that of the DBB block. After the DBB block, features undergo max-pooling to align dimensions with the Spatial Channel Attention Block (SCAB) and the subsequent DBB block. All convolutional blocks utilizing ReLU activation follow the He normal initialization [60],

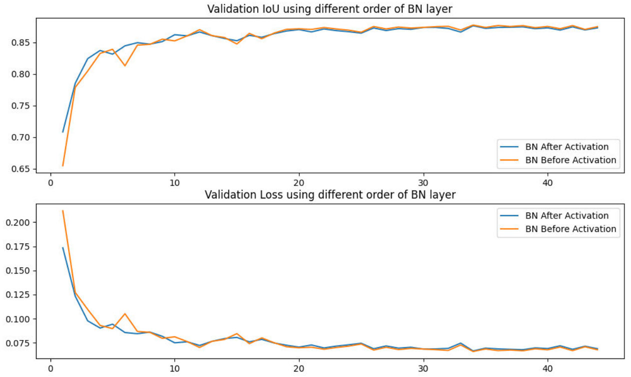


FIGURE 10. **CT-UNet: For different orders of BN, the IoU and Loss curves on the validation set are computed during training.**

TABLE 2. **CT-UNet results on the test set for various BN orders.**

Order of BN	IoU	BIoU	Score
BN After Activation Function	**0.7785**	**0.6050**	**0.6918**
BN Before Activation Function	0.7612	0.5907	0.6759

while blocks having sigmoid are employed with Glorot normal.

Figure 10 shows the results of experiments (batch size of 6, dice as loss function, and a learning rate 0.0001) for batch normalization before and after based on IOU and Loss. An increase of 0.0019 is observed in the validation Intersection over Union (IoU) when Batch Normalization (BN) is applied prior to the activation function. However, BN implementation follows the ReLu, which results in a more consistent training progression. Due to almost similar results, another experiment is performed by testing the model on a test dataset. Table 2 shows results for the BN position based on IoU and BIoU. The final score suggests that BN after ReLu is the best option.

In order to establish the optimal initial learning rate for the CT-UNet model, two experiments are performed on validation data for IOU and loss. Figure 11 shows results from the experiments with four different learning rates that have the same dice loss and batch size of 6. The graphs show that the highest validation Intersection over Union (IoU) is produced with an initial learning rate of 0.00005.

3) MODELS COMPLEXITY

The complexity of the model while training depends on the number of parameters, the batch size, and the complexity of the model. Memory is directly proportional to the complexity of the model. A larger batch size necessitates more memory because gradients for every sample in the batch must be stored in the model. In instances where memory is limited, complex models often resort to using smaller batch sizes in comparison to smaller models. Experimentation results for all model variations in this work are presented in Table 3, which denote that complex models require less batch size and vice versa.

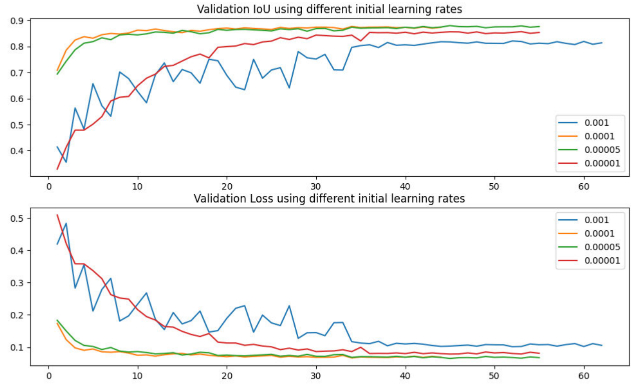


FIGURE 11. **CT-UNet: IoU and Loss curves are computed on the validation set for different initial learning rates during training.**

TABLE 3. **Models batch size and parameter.**

Network	Backbone	Batch Size	Parameters
U-Net	Baseline	6	34,540,737
	EfficientNetB4	12	18,253,672
	EfficientNetV2S	12	19,069,561
	ResNet50V2	12	17,541,441
	DenseNet201	12	24,746,881
CT-UNet	EfficientNetB4	10	17,831,856
	EfficientNetV2S	12	19,413,761
	ResNet50V2	12	17,803,401
	DenseNet201	10	24,831,177

4) SINGLE AND ENSEMBLE MODELS

A total of 9 different models listed in Table 5 are tested as single models. Five variations of U-Net have a baseline, and four different backbones are used. Four different variations of CT-UNet with different backbones are used in experiments.

Ensemble model prediction refers to combining the predictions of multiple individual models to make a final prediction or decision to improve the overall performance, accuracy, and robustness of predictions compared to using a single model. All models are trained on the same dataset and predicted once on the whole dataset. Each model's prediction is weighted differently based on its performance on a validation set. Better-performing models receive higher weights in the final prediction.

D. TEST TIME AUGMENTATION

Test-time augmentation (TTA) in the first block of Figure 2 is employed to avoid misclassification caused by slight variations in input data [61] during testing. It introduces a slight increase in testing time. In the image segmentation context, TTA generates multiple augmented variations of a test image, predicts each version, and afterward counts the average score as indicated in Figure 12 after reversing the augmentation of each anticipated mask to the original image.

IV. EXPERIMENTATIONS AND RESULTS

The experimentation section is structured into two main sections, namely Task_1 and Task_2, each concentrating on a

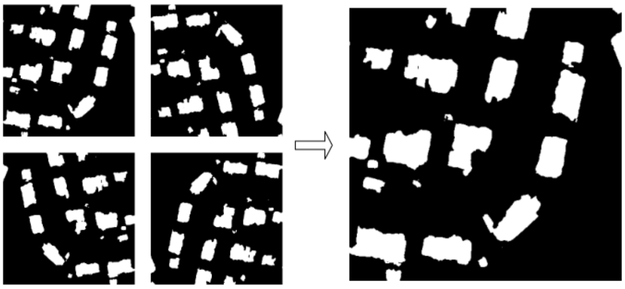


FIGURE 12. Left side: Predicted masks (original, 90 degrees, 180 degrees, and 270-degree mask). Right side: reverse and the average of predicted masks.

distinct aspect of the research. Within each section, a detailed breakdown is further divided into three parts: training and validation evaluation, single model performance analysis, and ensemble model results. This structured approach allows for a comprehensive systematic exploration of the research's objectives and outcomes.

A. TASK_1

This section exclusively presents and analyzes the experimental results conducted using RGB/aerial images only.

1) TRAINING AND VALIDATION EVALUATION

The training and validation progress curve for all the models listed in Table 3 is visually shown in Figure 13. This analysis leads to the conclusion that most models exhibit stable training curves, maintaining consistent performance during the training process. However, two models, U-Net DenseNet201 and CT-UNet ResNet50V2, demonstrate a drop in validation IoU that lasts for the fourth epoch before reaching a new peak, followed by a new high peak IoU value. CT-UNet shows lower IoU in the starting epoch as compared to U-Net because the CT-UNet models' learning rate is lower.

Table 4 provides a comprehensive summary of the best training and validation IoU scores achieved by each model, along with the number of epochs used to reach the highest peak. It is concluded that U-Net EfficientNetV2S outperforms other models with 63 epochs in both training and validation IoU. Moreover, there is an average difference of approximately 2% in IoU between training and validation across all models. This signifies the extent to which the models demonstrate generalization and reduced overfitting to the training set despite the training and validation sets belonging to distinct regions.

2) SINGLE MODEL RESULTS

Table 5 represents the results of single models with and without TTA for testing datasets. The results presented in the table indicate that the CT-UNet EfficientNetB4 model demonstrates the highest performance in single-image prediction, achieving 78.65% Intersection over Union (IoU), 61.48% Boundary Intersection over Union (BIoU), and 70.07% average score. This superior performance is also evident in the case of Test-Time Augmentation (TTA) predictions, where the model attains an IoU of 79.14%, BIoU of 61.89%, and an average score of 70.52%. Across all models considered, it is noteworthy that TTA consistently enhances both IoU, BIoU, and total score by an average of 0.67%, 0.6%, and 0.84%, respectively.

TABLE 4. Task_1: IoU for both U-Net and CT-UNet models on training and validation set with different backbones after the last epoch.

Network	Backbone	Epoch	Train IoU	Val IoU	Overfit
U-Net	Baseline	57	0.842	0.8146	0.0274
	EfficientNetB4	51	0.8978	0.8745	0.0233
	EfficientNetV2S	63	**0.901**	**0.8822**	0.0188
	ResNet50V2	48	0.8701	0.8516	0.0185
	DenseNet201	63	0.8911	0.8724	0.0187
CT-UNet	EfficientNetB4	56	0.8868	0.8713	0.0155
	EfficientNetV2S	42	0.889	0.8691	0.0199
	ResNet50V2	68	0.8706	0.8469	0.0237
	DenseNet201	63	0.8914	0.872	0.0197

TABLE 5. Task_1: Single model predictions results for both U-Net and CT-UNet models with different backbones on test datasets with and without TTA for all models.

Single Model		Without TTA		With TTA	
Network	Backbone	IoU	BIoU	IoU	BIoU
U-Net	Baseline	0.7471	0.5690	0.7529	0.5734
	EfficientNetB4	0.7836	0.6132	0.7898	0.6183
	EfficientNetV2S	0.7675	0.6018	0.7763	0.6100
	ResNet50V2	0.7617	0.5789	0.7677	0.5829
	DenseNet201	0.7632	0.5990	0.7685	0.6045
CT-UNet	EfficientNetB4	**0.7865**	**0.6148**	**0.7914**	0.6189
	EfficientNetV2S	0.7824	0.6113	0.7904	**0.6199**
	ResNet50V2	0.7609	0.5952	0.7690	0.6022
	DenseNet201	0.7612	0.5977	0.7680	0.6045

TABLE 6. Task_1: Results of Top 5 best ensembles of two models with TTA.

Models (Architecutr_Backbone) (weights)	IoU	BIoU
U-Net_DenseNet201 (0.5), CT-UNet_EfficientNetB4 (0.5)	**0,7954**	**0,6219**
U-Net_DenseNet201 (0.5), CT-UNet_EfficientNetV2S (0.5)	0.7938	0.6217
U-Net_EfficientNetV2S (0.5), CT-UNet_EfficientNetB4 (0.5)	0.7931	0.6210
CT-UNet_EfficientNetB4 (0.5), CT-UNet_EfficientNetV2S (0.5)	0.7937	0.6202
CT-UNet_EfficientNetB4 (0.5) CT-UNet_DenseNet201 (0.5)	0.7930	0.6208

3) ENSEMBLE MODELS

The ensemble model consists of different models with weights that represent the influence of the model in prediction. The sum of the weights should be 1. Each model's prediction is multiplied by its weight, and then the values are summed up. According to a specified threshold, depending on the dataset, each pixel value is assigned to 0 or 1, which results in a predicted mask. The appropriate threshold for this experiment is 0.5. Table 6 shows results for the top five best combinations of two model ensembles. The data presented in the table conclude that the most influential ensemble configurations were achieved by assigning equal weight, which is 0.5, to both models. The top-performing ensemble was composed of U-Net DenseNet201 with 79.54% IoU and CT-UNet EfficientNetB4 with 62.19% BIoU and 70.86% average score.

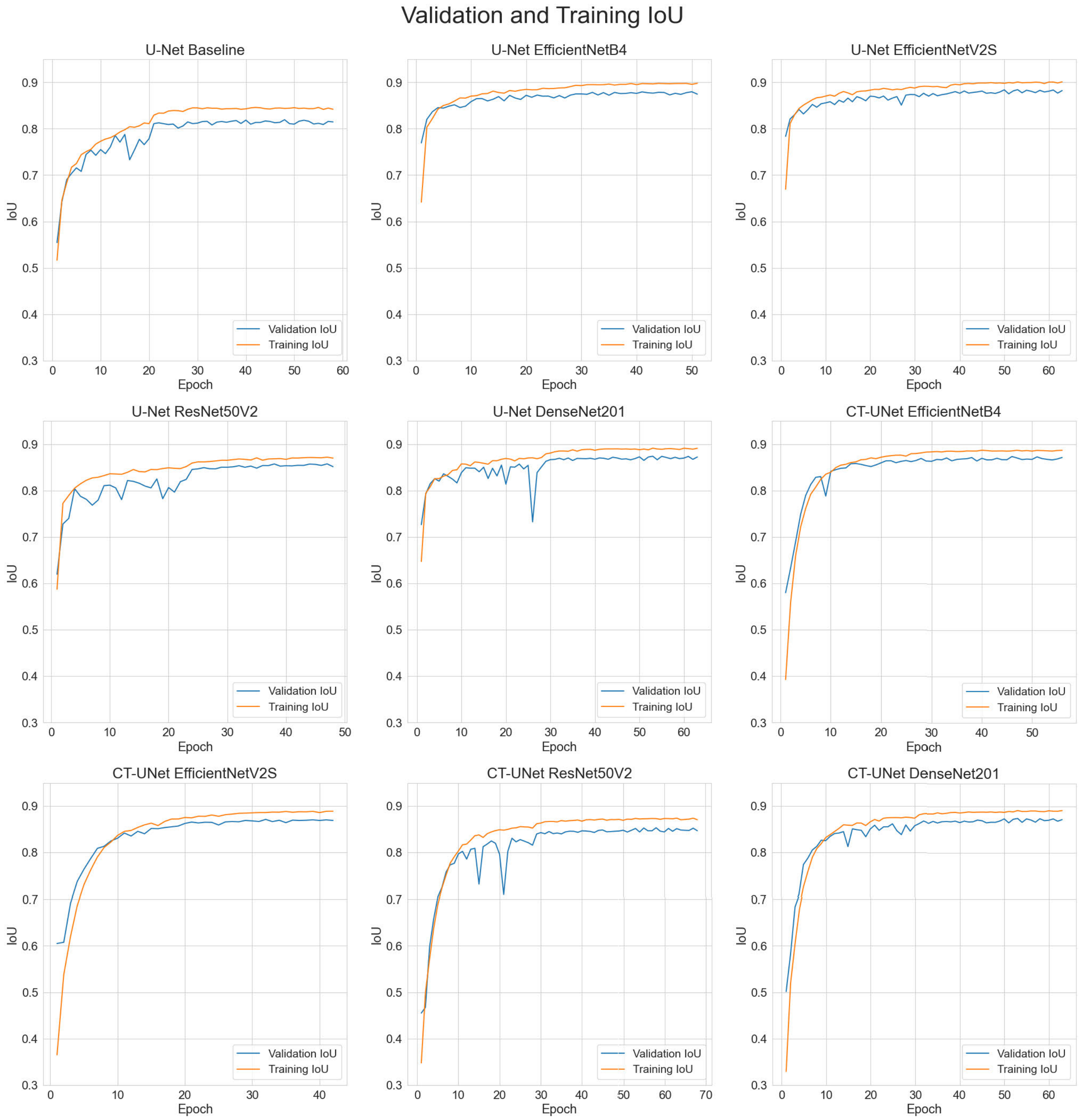


FIGURE 13. Task_1: Validation and Training IoU for all models during training.

The most effective model ensemble within the top five, utilizing Test-Time Augmentation (TTA), comprises U-Net with DenseNet201 encoder having 0.4 weight, CT-UNet with EfficientNetB4 encoder having 0.3 weight, and CT-UNet with EfficientNetV2S encoder having 0.3 weight achieved 80.11% IoU, 62.95% BIoU, and 71.53% score with a threshold of 0.3. This ensemble model is used for Task_1 as the proposed solution. TTA demonstrates an average improvement in IoU and BIoU across the top 5 ensembles by 0.61% and 0.74%, respectively.

B. TASK_2

This section presents experimental results performed on the combination of both LiDAR data and RGB aerial images. Both data are fed to the network as input simultaneously,

TABLE 7. Task_1: Top 5 best ensembles of three models with TTA.

Models (Architecutr_Backbone) (weights)	IoU	BIoU
U-Net_DenseNet201 (0.4), CT-UNet_EfficientNetB4 (0.3), CT-UNet_EfficientNetV2S (0.3)	**0.8011**	**0.6295**
U-Net_EfficientNetB4 (0.2), U-Net_DenseNet201 (0.4), CT-UNet_EfficientNetV2S (0.4)	0.8001	0.6292
U-Net_EfficientNetB4 (0.2), U-Net_DenseNet201 (0.4), CT-UNet_EfficientNetB4 (0.4)	0.8000	0.6275
U-Net_EfficientNetV2S (0.3), U-Net_DenseNet201 (0.4), CT-UNet_EfficientNetB4 (0.3)	0.7995	0.6280
U-Net_EfficientNetV2S (0.2), U-Net_DenseNet201 (0.4), CT-UNet_EfficientNetV2S (0.4)	0.7988	0.6284

with the first three channels belonging to RGB images and the fourth channel belonging to LiDAR.

1) VALIDATION AND TRAINING SET EVALUATION

The same models presented in Table 3 are trained using a combination of LiDAR data and RGB aerial images for this specific task. In Figure 14, we observe the learning curves based on IoU for all these models. Compared to Figure 13 of Task_1, Figure 14 shows a very small difference in training and validation IoU for all models, which depicts that models are less overfitted for Task_2 as compared to Task_1.

Figure 14 illustrates notable instability in the training progress for the U-Net with ResNet50V2 encoder, U-Net with DenseNet201 encoder, and CT-UNet with DenseNet201 encoder. This is evident in a sudden and pronounced drop in validation Intersection over Union (IoU) during the initial epochs of training. However, it's noteworthy that the validation IoU manages to recover and align with the IoU of training towards the conclusion of the training process. The CT-UNet models, having similarity to Task_1, exhibit a lower initial starting point and take more time to reach the IoU plateau of the training set, primarily because of the lower initial learning rate.

Table 8 presents the IoU scores obtained after the last epoch for both training and validation datasets. Remarkably, the U-Net Baseline demonstrates superior performance on the training dataset, achieving an impressive IoU score of 89.46%. Conversely, the U-Net EfficientNetB4 surpasses others on the validation dataset, attaining 89.18% of the IoU score.

TABLE 8. Task_2: IoU on training and validation set for all models after the final epoch.

Network	Backbone	Epoch	Train IoU	Val IoU	Overfit
U-Net	Baseline	41	**0.8946**	0.8845	0.0101
	EfficientNetB4	47	0.893	**0.8918**	0.0012
	EfficientNetV2S	47	0.8897	0.8866	0.0031
	ResNet50V2	40	0.8837	0.8866	-0.0029
	DenseNet201	40	0.8903	0.8866	0.0037
CT-UNet	EfficientNetB4	57	0.8851	0.8792	0.0059
	EfficientNetV2S	47	0.8851	0.8821	0.0030
	ResNet50V2	53	0.8944	0.8799	0.0145
	DenseNet201	36	0.8944	0.8765	0.0179

The discrepancy between the training and validation IoU scores typically serves as an indicator of potential overfitting to the training data. However, in this case, the average difference between the two scores is just 0.6%. This minimal gap suggests that the model does not exhibit significant overfitting on the training set nor underfitting on the validation dataset.

2) SINGLE MODEL RESULTS

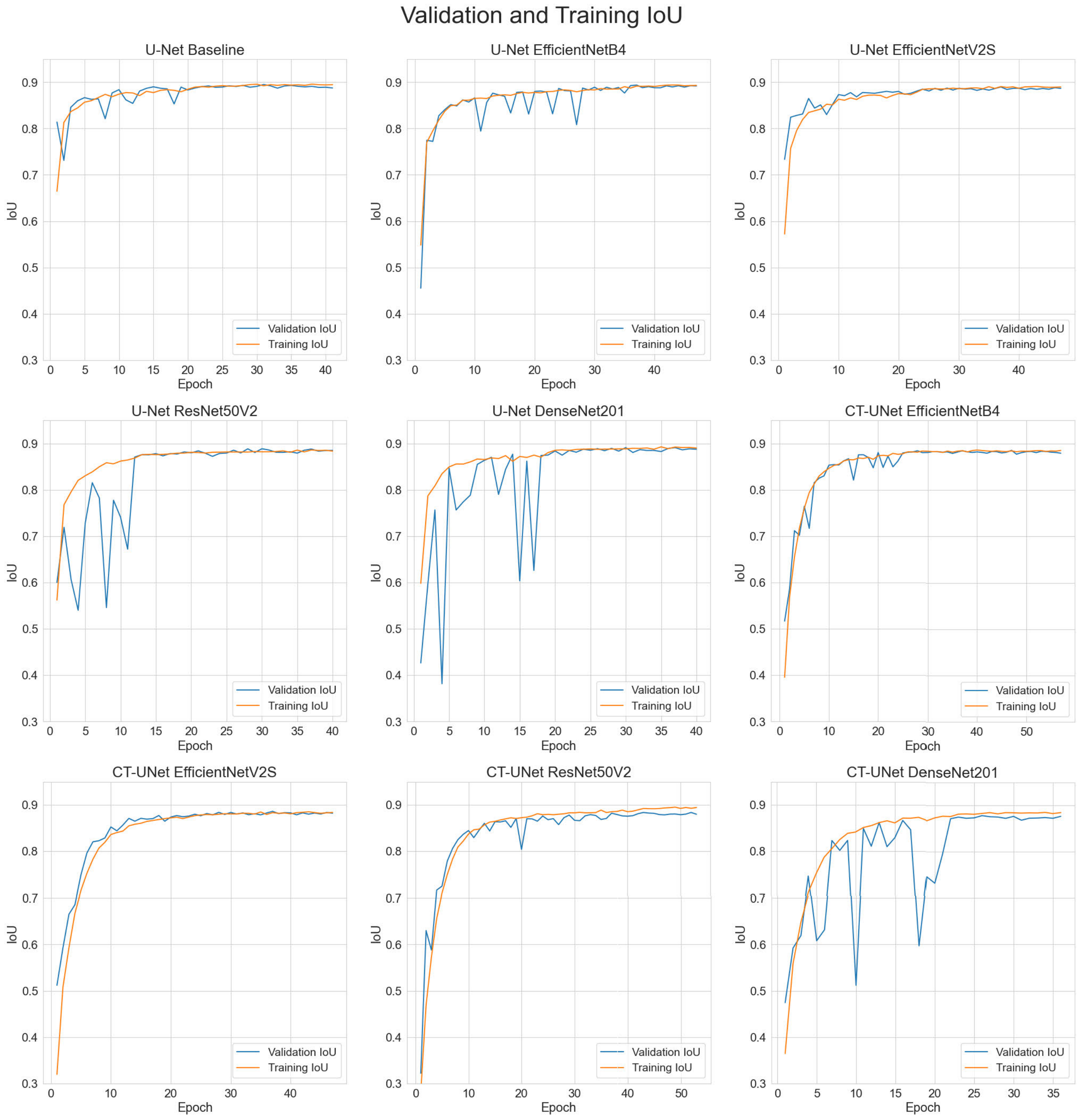
Single model results on the test dataset are presented in Table 9. The Table shows results for models having TTA and models without TTA. Notably, U-Net DenseNet201 emerges as the top performer for single-image predictions, yielding an impressive 89.09% of IoU, 79.07% of BIoU, and 84.08% overall score. This strong performance is consistent when TTA is applied, having 89.30% of IoU, 79.39% of BIoU, and 84.35% overall score. Moreover, when considering all models collectively, it's apparent that TTA yields an average improvement of 0.31% in IoU and 0.46% in BIoU across the table.

TABLE 9. Task_2: Single model prediction results for all models on test datasets with and without TTA.

Single Prediction		Without TTA		With TTA	
Network	Backbone	IoU	BIoU	IoU	BIoU
U-Net	Baseline	0.8836	0.7824	0.8873	0.7891
	EfficientNetB4	0.8804	0.7720	0.8847	0.7788
	EfficientNetV2S	0.8766	0.7798	0.8799	0.7838
	ResNet50V2	0.8875	0.7859	0.8897	0.7903
	DenseNet201	**0.8909**	**0.7907**	**0.8930**	**0.7939**
CT-UNet	EfficientNetB4	0.8688	0.7661	0.8721	0.7705
	EfficientNetV2S	0.8794	0.7783	0.8820	0.7823
	ResNet50V2	0.8788	0.7775	0.8825	0.7823
	DenseNet201	0.8751	0.7704	0.8779	0.7737

3) ENSEMBLE MODELS

Table 10 presents the top five models ensemble using TTA. The most successful model ensemble, achieved by Test-Time Augmentation (TTA), comprises three models: U-Net Baseline having 0.5 weight, U-Net with DenseNet201 encoder having 0.3 weight, and CT-UNet with EfficientNetV2S encoder having 0.2 weight. This ensemble exhibits outstanding performance, achieving 89.64% IoU, 80.09% BIoU, and 84.86% overall score. This model ensemble serves as the best option for Task_2. Among the top five ensembles, when TTA is applied, there is an average enhancement of 0.25% in IoU and 0.57% in BIoU. Notably, it's worth highlighting that the U-Net Baseline consistently emerges as the dominant component in all of the top five ensembles that utilize TTA.



Validation and Training IoU

U-Net Baseline

U-Net EfficientNetB4

U-Net EfficientNetV2S

U-Net ResNet50V2

U-Net DenseNet201

CT-UNet EfficientNetB4

CT-UNet EfficientNetV2S

CT-UNet ResNet50V2

CT-UNet DenseNet201

FIGURE 14. **Task_2: Validation and Training IoU for all models during training.**

TABLE 10. **Task 2: Illustration of results from top 5 weighted ensembles with TTA.**

Model 1	**Model 2**	**Model 3**	**Weight**	**IoU**	**BIoU**	**Score**
U-Net Baseline	U-Net DenseNet201	CT-UNet EfficientNetV2S	0.5, 0.3, 0.2	**0.8964**	0.8009	**0.8486**
U-Net Baseline	U-Net EfficientNetB4	U-Net DenseNet201	0.6, 0.1, 0.3	0.8962	0.8009	0.8485
U-Net Baseline	U-Net EfficientNetV2S	U-Net DenseNet201	0.5, 0.2, 0.3	0.8961	**0.8010**	0.8485
U-Net Baseline	U-Net ResNet50V2	U-Net DenseNet201	0.6, 0.1, 0.3	0.8962	0.8008	0.8485
U-Net Baseline	U-Net EfficientNetV2S	U-Net ResNet50V2	0.5, 0.2, 0.3	0.8954	0.7999	0.8472

C. ANALYSIS

The experiments clearly demonstrate the superiority of model ensembles over single-model predictions in both tasks. This advantage is derived from the ensemble's ability to mitigate prediction variability by employing diverse model architectures. Given that each model possesses distinct strengths and weaknesses, one model's strengths can compensate for another's weaknesses. Additionally, leveraging multiple models enables the capturing of a broader spectrum of features since each model has learned different features based on its underlying architecture.

Furthermore, the experiments reveal that TTA predictions slightly outperform single-instance predictions. TTA achieves this by reducing prediction uncertainty, capturing a wider representation of the input image, and exhibiting superior edge localization between the background and foreground. The predictions gain accuracy and resilience when TTA and ensemble modeling are combined.

The effects of several prediction models on Task_1 and Task_2 of the test set are shown in Figures 15 and 16, respectively. The best models are used to create these predictions. In Figure 15, the top row of predictions reveals that the left building is erroneously connected with the right building by the model within the masks, resulting in red pixels that serve as false positives. However, while employing model ensembles or TTA, the majority of these false positive pixels are successfully eliminated. Notably, the ensemble with TTA demonstrates the ability to differentiate shadows and small gaps within the building, eliminating all scattered false positive artifacts.

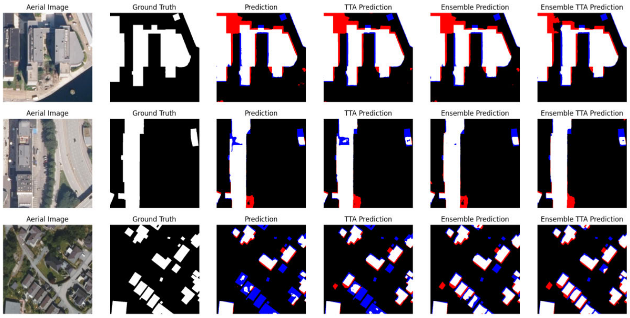


FIGURE 15. Illustraction of task 1 predicted masks generated from the test dataset. White: True Positives, Black: True Negatives, Red: False Positives, Blue: False Negatives.

In the middle row, it is evident that as we progress to more advanced prediction models, the gaps and false negatives within the primary building are progressively filled. Conversely, in the bottom row, the single model prediction encounters challenges in correctly identifying the building row as foreground at the bottom. TTA proves beneficial by aiding in the correct classification of two of the buildings. The majority of the buildings in the bottom row are accurately classified when the ensemble is used, despite some uneven borders. This is a substantial improvement. These irregularities are subsequently rectified when TTA is employed.

Examining Figure 16, subtle variations are observed in the generated masks distinguished from those in Task_1. The overall geometry of predicted buildings in the masks remains consistent; the primary distinction lies in the removal of minuscule false positive artifacts as we transition toward more sophisticated methods. This phenomenon primarily arises from the models assigning greater importance to the LiDAR data compared to the RGB data. As a result, the projections closely match the building's geometry and positioning, as shown in the LiDAR data.

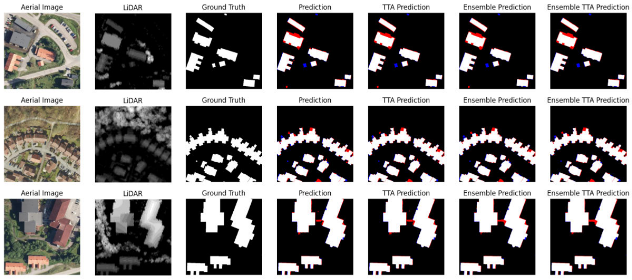


FIGURE 16. Illustration of task 2 predicted mask using the test dataset. White pixel: True Positives, Black pixel: True Negatives, Red pixel: False Positives, and Blue pixel: False Negatives.

The U-Net network performs better on the validation set when using only RGB images, as seen in Tables 4 and 5. In contrast, the CT-UNet network excels on the test set. With the incorporation of LiDAR data, as shown in Tables 8 and 9, U-Net emerges as the top performer for both validation and test sets. However, it's important to note a significant performance gap between the sets for each task, as highlighted in Table 11. Weighted ensembles are used along with TTA for both sets to obtain the results presented in the table. Tables 7 and 10 previously identified the optimal performing models using TTA for Task_1 and Task_2, which correspond to the ensembles used.

TABLE 11. Results for the validation and test set utilizing the top-performing model with TTA for each task.

	Validation		Test	
Metrics	Task 1	Task 2	Task 1	Task 2
IoU	0.9306	0.9384	0.8011	0.8964
BIoU	0.8503	0.8639	0.6295	0.8009
Score	0.8904	0.9012	0.7153	0.8486

The disparities in the metrics are likely attributed to the methodology employed in generating the ground truth masks. Specifically, the validation masks were created with the help of DSM, whereas test sets were created using DTM. This difference introduces an inherent bias, particularly affecting the upper portions of the buildings.

Another contributing factor could be the variations in building and environmental characteristics between Denmark

TABLE 12. **The proposed approach compared with the top 3 groups of the MapAI competition.**

		Task_1			Task_2			
Placement	**Team**	**IoU**	**BIoU**	**Score**	**IoU**	**BIoU**	**Score**	**Final Score**
1	FUNDATOR [43]	0.7794	0.6115	0.6955	0.8775	0.7857	0.8316	0.7635
2	HVL-ML [54]	0.7879	0.6245	0.7062	0.8711	0.7504	0.8108	0.7585
3	DEEPCROP [55]	0.7902	0.6185	0.7044	0.8506	0.7461	0.7984	0.7514
4	UIAI [46]	0.7336	0.5780	0.6558	0.8790	0.7841	0.8316	0.7437
5	YSBS [44]	0.7551	0.5613	0.6582	0.8555	0.7127	0.7841	0.7212
6	KABORG [45]	0.7112	0.5195	0.6154	0.6890	0.5616	0.6253	06203
7	Li et al [62]	**0.8012**	0.6213	0.7112	-	-	-	-
-	Proposed Work	0.8011	**0.6295**	**0.7153**	**0.8964**	**0.8009**	**0.8486**	**0.7820**

and Norway. Given that images in the training and validation sets belong to Denmark, while images in both test datasets belong to Norway, this geographical discrepancy may introduce a performance drop due to inherent bias stemming from the differing architectural and environmental contexts.

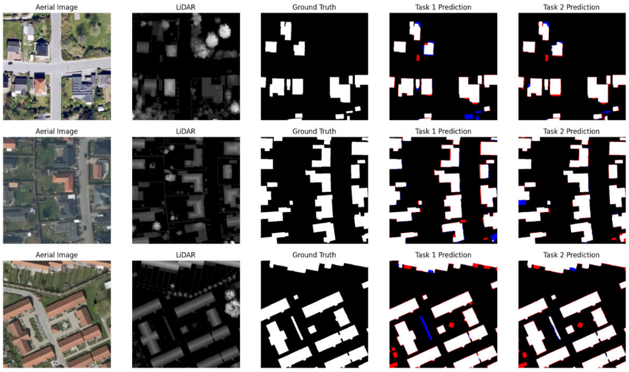


FIGURE 17. **Illustration of task 1 and 2 predicted masks from the validation dataset. White pixel: True Positives, Black pixel: True Negatives, Red pixel: False Positives, and Blue pixel: False Negatives.**

A comparison of the generated masks on the validation set for only RGB and fusion of RGB and LiDAR data is shown in Figure 17. In the top row, the prediction based on LiDAR data effectively distinguishes most buildings concealed by trees in the image's lower right corner. In the middle row, when relying solely on the RGB image, the prediction erroneously identifies a building that was a white truck, and at the right corner of the last row, the model misclassifies the building as background. These errors are rectified by utilizing LiDAR data.

The LiDAR-based prediction accurately identifies a bicycle shed that the RGB-based prediction incorrectly labels as background in the bottom row. Furthermore, the LiDAR prediction correctly classifies some false positives near the top of the building.

Superior outcomes achieved through our proposed approach are systematically contrasted with those of MapAI competitors in Table 12. This comparison is grounded in the utilization of identical datasets, tasks, and evaluation metrics. In their investigation, Hodne and Furdal [43] employed ResNest26b and efficientB1 backbones, which, despite being lightweight, possess a greater number of parameters. Consequently, the decoders in their framework exhibit high complexity compared to the backbones employed in our methodology. Furthermore, Hodne et al. utilized the original UNet architecture, whereas our proposed approach involves a tailored modification of the UNet to optimize it specifically for building precision.

Kaliyugarasam and Lundervold [54] utilize RasNet34 as a backbone in U-Net architecture and Inria Aerial image dataset [50] to pretrain the aerial model. RasNet is a shallow backbone as compared to others used in the proposed work. ResNet50V2 introduces skip connections and residual blocks, offering improved training convergence and performance to the proposed work compared to ResNet34. Furthermore, maggiori et al. didn't utilize batch normalization for generalization.

Li compared UNet, ConvNext, and different versions of SegFormer [62]. The UNet architecture effectively encapsulates global and local features. This characteristic renders it especially well-suited for tasks where preserving spatial details and context proves paramount, as exemplified in the context of building segmentation. Notably, Li et al. employ EfficientNet as the encoding backbone in their models, and models are tested on IoU only, while our proposed approach leverages more advanced and intricate backbones.

Khan et al. [38] introduced an efficient UNet architecture tailored for building segmentation, employing solely DenseNet201. This choice is motivated by the model's ability to effectively reuse feature information and optimize parameter utilization. In contrast, our proposed methodology adopts a more intricate approach. It leverages an ensemble of diverse backbones, incorporating DenseNet201 alongside EfficientNet to mitigate the overall parameter count. Additionally, ResNet is integrated to introduce skip connections and residual blocks, enhancing convergence and overall performance.

V. CONCLUSION

Two types of experiments are performed. The first part belongs to the parameter turning of the models, and the second part belongs to results analysis using the fine-tuned models. The second section of experiments is subdivided into two components: the first portion necessitated the development of a method solely employing aerial images, while the latter portion required a method utilizing LiDAR data, either independently or in conjunction with aerial images. Two distinct architectures, namely U-Net and CT-UNet, are modified and employed for model creation, each integrated with four diverse backbone architectures: EfficientNetB4, EfficientNetV2S, ResNet50V2, and DenseNet201. Techniques such as data augmentation at training, transfer learning via backbones, weighted model ensemble, and test time augmentation were implemented to bolster the model's robustness.

Three experiments are conducted for both model parameter tuning. The first experiment was to find the order of BN before or after the Relu function. Figure 7 for U-Net and Figure 10 and Table 2 for CT-UNet recommended using BN after Relu activation function. The second experiment in fine-tuning was to find the best initial learning rates for both models. Figure 8 and Figure 11 recommend 0.0001 and 0.00005 initial learning rates for UNet and CT-UNet, respectively. The Third experiment was to find the best loss function for model architecture. Figure 9 recommends Dice loss as the best loss function for the UNet model.

In the result analysis experimental setup, a total of four experiments for each task are conducted: single model prediction both with and without TTA, as well as model ensemble prediction, again with and without TTA. The experimentation result concluded that ensembles of models outperformed single models. Moreover, Test Time Augmentation (TTA) yielded a slight enhancement in the outcomes of both of these methodologies. Indeed, the experimental results suggest that CT-UNet exhibited superior performance in Task_1, whereas U-Net excelled in Task_2. The Dense Boundary Block and Spatial Channel Attention Block are less essential when employing LiDAR data, as the network finds it more straightforward to discern between the foreground and background classes.

The potential of LiDAR data can be concluded from Table 11, which presents results from the best model recognized with the help of thorough experimentation. LiDAR data improve IoU 9.53% and BIoU 17.14%, which is a significant improvement in applications of precision. In the scoring metric, the impact of LiDAR data shows 13.33% enhancement.

As evident in the table, the proposed approach outperforms competitors across all competition metrics. Several leading groups, as reported by [43] and [54], utilized U-Net in their solutions. Apart from other modifications like optimal learning rate, loss function, data augmentation, and ReLu position, the primary distinction between their approach and ours lies in their adoption of weaker backbones (EfficientNetB1, ResNet26d, ResNet34), resulting in models with a higher parameter count, indicating more complexity in their decoders. Attempts were made to raise the kernels in the convolutional layers of the decoders. Nevertheless, this approach yielded less than optimal outcomes when contrasted with the proposed method, indicating that the models became excessively complex for the given dataset.

REFERENCES

[1] X. Liu, "Airborne LiDAR for DEM generation: Some critical issues," *Prog. Phys. Geogr., Earth Environ.*, vol. 32, no. 1, pp. 31–49, Feb. 2008.

[2] R. Nelson, W. Krabill, and G. MacLean, "Determining forest canopy characteristics using airborne laser data," *Remote Sens. Environ.*, vol. 15, no. 3, pp. 201–212, Jun. 1984.

[3] S. Debnath, M. Paul, and T. Debnath, "Applications of LiDAR in agriculture and future research directions," *J. Imag.*, vol. 9, no. 3, p. 57, Feb. 2023.

[4] S. Li, L. Dai, H. Wang, Y. Wang, Z. He, and S. Lin, "Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model," *Remote Sens.*, vol. 9, no. 11, p. 1202, Nov. 2017.

[5] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan, "Semantic labeling in very high resolution images via a self-cascaded convolutional neural network," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 78–95, Nov. 2018.

[6] M. Ghanea, P. Moallem, and M. Momeni, "Building extraction from high-resolution satellite images in urban areas: Recent methods and strategies against significant challenges," *Int. J. Remote Sens.*, vol. 37, no. 21, pp. 5234–5248, Nov. 2016.

[7] R. Wang, Y. Hu, H. Wu, and J. Wang, "Automatic extraction of building boundaries using aerial LiDAR data," *J. Appl. Remote Sens.*, vol. 10, no. 1, Mar. 2016, Art. no. 016022.

[8] S. Du, Y. Zhang, Z. Zou, S. Xu, X. He, and S. Chen, "Automatic building extraction from LiDAR data fusion of point and grid-based features," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 294–307, Aug. 2017.

[9] A. Zarea and A. Mohammadzadeh, "A novel building and tree detection method from LiDAR data and aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 5, pp. 1864–1875, May 2016.

[10] C. Grecea, A. Bala, and S. Herban, "Cadastral requirements for urban administration, key component for an efficient town planning," *J. Environ. Protection Ecol.*, vol. 14, no. 1, pp. 363–371, 2013.

[11] Q. Hu, L. Zhen, Y. Mao, X. Zhou, and G. Zhou, "Automated building extraction using satellite remote sensing imagery," *Autom. Construct.*, vol. 123, Mar. 2021, Art. no. 103509.

[12] S. Ohleyer. (2018). *Building Segmentation on Satellite Images*. [Online]. Available: https://project.inria.fr/aerialimagelabeling/files/2018/01/fp_ohleyer_compressed.pdf

[13] S. Jyhne, M. Goodwin, P.-A. Andersen, I. Oveland, A. S. Nossum, M. Ørstavik, K. Ormseth, and A. Flatman, "MapAI: Precision in building segmentation," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 1–3, Sep. 2022.

[14] E. Finnesand, "A machine learning approach for building segmentation using laser data," Master's thesis, Dept. Elect. Eng. Comput. Sci., Univ. Stavanger, Stavanger, Norway, 2023.

[15] Y.-J. Cho, "Weighted intersection over union (wIoU): A new evaluation metric for image segmentation," 2021, *arXiv:2107.09858*.

[16] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving object-centric image segmentation evaluation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15329–15337.

[17] S. Liu and Q. Shi, "Local climate zone mapping as remote sensing scene classification using deep learning: A case study of metropolitan China," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 229–242, Jun. 2020.

[18] S. M. Azimi, P. Fischer, M. Körner, and P. Reinartz, "Aerial LaneNet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2920–2938, May 2019.

[19] J. Huang, X. Zhang, Q. Xin, Y. Sun, and P. Zhang, "Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 91–105, May 2019.

[20] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7209–7220, Sep. 2019.

[21] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "TreeUNet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 1–13, Oct. 2019.

[22] W. Liao, F. Van Coillie, L. Gao, L. Li, B. Zhang, and J. Chanussot, "Deep learning for fusion of APEX hyperspectral and full-waveform LiDAR remote sensing data for tree species mapping," *IEEE Access*, vol. 6, pp. 68716–68729, 2018.

[23] J.-D. Sylvain, G. Drolet, and N. Brown, "Mapping dead forest cover using a deep convolutional neural network and digital aerial photography," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 14–26, Oct. 2019.

[24] M. Sulaiman, M. Farmanbar, A. Nabil Belbachir, and C. Rong, "Precision in building extraction: Comparing shallow and deep models using LiDAR data," 2023, *arXiv:2309.12027*.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[29] S. Dey, A. K. Singh, D. K. Prasad, and K. D. Mcdonald-Maier, "SoCodeCNN: Program source code for visual CNN classification using computer vision methodology," *IEEE Access*, vol. 7, pp. 157158–157172, 2019.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2881–2890.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[32] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, and J. Cai, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[33] W. Sirko, S. Kashubin, M. Ritter, A. Annkah, Y. S. E. Bouchareb, Y. Dauphin, D. Keysers, M. Neumann, M. Cisse, and J. Quinn, "Continental-scale building detection from high resolution satellite imagery," 2021, *arXiv:2107.12283*.

[34] H. L. Yang, J. Yuan, D. Lunga, M. Laverdiere, A. Rose, and B. Bhaduri, "Building extraction at scale using convolutional neural network: Mapping of the United States," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2600–2614, Aug. 2018.

[35] Microsoft. *U.S. Building Footprint*. Accessed: Nov. 30, 2024. [Online]. Available: https://github.com/microsoft/USBuildingFootprints

[36] Microsofts. *Global ML Buildings Footprint*. Accessed: Nov. 30, 2024. [Online]. Available: https://github.com/microsoft/GlobalMLBuildingFootprints

[37] S. Liu, H. Ye, K. Jin, and H. Cheng, "CT-UNet: Context-transfer-UNet for building segmentation in remote sensing images," *Neural Process. Lett.*, vol. 53, no. 6, pp. 4257–4277, Dec. 2021.

[38] S. D. Khan, L. Alarabi, and S. Basalamah, "An encoder–decoder deep learning framework for building footprints extraction from aerial imagery," *Arabian J. Sci. Eng.*, vol. 48, no. 2, pp. 1273–1284, Feb. 2023.

[39] Y. Zhu, Z. Liang, J. Yan, G. Chen, and X. Wang, "E-D-Net: Automatic building extraction from high-resolution aerial images with boundary information," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4595–4606, 2021.

[40] D. Hong, C. Qiu, A. Yu, Y. Quan, B. Liu, and X. Chen, "Multi-task learning for building extraction and change detection from remote sensing images," *Appl. Sci.*, vol. 13, no. 2, p. 1037, Jan. 2023.

[41] S. Wei, T. Zhang, S. Ji, M. Luo, and J. Gong, "BuildMapper: A fully learnable framework for vectorized building contour extraction," *ISPRS J. Photogramm. Remote Sens.*, vol. 197, pp. 87–104, Mar. 2023.

[42] M. Luo, S. Ji, and S. Wei, "A diverse large-scale building dataset and a novel plug-and-play domain generalization method for building extraction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4122–4138, 2023.

[43] L. M. Hodne and E. H. Furdal, "Team fundator: Weighted UNet ensembles with enhanced datasets," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 4–6, Mar. 2023.

[44] Y. S. Bicakci and B. Sarica, "ATTransUNet: Semantic segmentation model for building segmentation from aerial image and laser data," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 7–9, Mar. 2023.

[45] K. A. Borgersen and M. Grundetjern, "MapAI competition submission for team kaborg: Using stable diffusion for ML image augmentation," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 10–12, Mar. 2023.

[46] L. F. Mrozik, A. H. Eike, and P. Alves, "Precision in building segmentation competition submission–team UiAI," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 20–22, Mar. 2023.

[47] G. Kong, C. Zhang, Y. Zhao, and H. Fan, "Building segmentation from remote sensing data using enhanced u-net," *Nordic Mach. Intell.*, vol. 2, no. 3, 2022.

[48] T. K. Sørensen, M. Vermeer, J. A. Hay, D. Fantin, and D. Völgyes, "Our MapAI approach: Focusing on data pipeline and loss functions," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 26–28, Mar. 2023.

[49] N. Moshkov, B. Mathe, A. Kertesz-Farkas, R. Hollandi, and P. Horvath, "Test-time augmentation for deep learning-based cell segmentation on microscopy images," *Sci. Rep.*, vol. 10, no. 1, p. 5068, Mar. 2020.

[50] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3226–3229.

[51] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019.

[52] V. Mnih, *Machine Learning for Aerial Image Labeling*. Toronto, ON, Canada: Univ. Toronto, 2013.

[53] I. M. Gerke, "Use of the stair vision library within the ISPRS 2D semantic labeling benchmark (Vaihingen)," Univ. Twente, Enschede, The Netherlands, Tech. Rep., 2014.

[54] S. Kaliyugarasan and A. S. Lundervold, "LAB-Net: LiDAR and aerial image-based building segmentation using U-Nets," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 23–25, Mar. 2023.

[55] L. Li, T. Zhang, S. Oehmcke, F. Gieseke, and C. Igel, "BuildSeg: A general framework for the segmentation of buildings," *Nordic Mach. Intell.*, vol. 2, no. 3, pp. 1–4, Mar. 2023.

[56] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.

[57] A. B. Gavade, R. Nerli, N. Kanwal, P. A. Gavade, S. S. Pol, and S. T. H. Rizvi, "Automated diagnosis of prostate cancer using mpMRI images: A deep learning approach for clinical decision support," *Computers*, vol. 12, no. 8, p. 152, Jul. 2023.

[58] R. Younisse, R. Ghnemat, and J. Al Saraireh, "Fine-tuning U-Net for medical image segmentation based on activation function, optimizer and pooling layer," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 5, p. 5406, Oct. 2023.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[60] L. Datta, "A survey on activation functions and their relation with Xavier and He normal initialization," 2020, *arXiv:2004.06632*.

[61] J. Nalepa, M. Myller, and M. Kawulok, "Training- and test-time data augmentation for hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 292–296, Feb. 2020.

[62] L. Li, "Segment any building," in *Proc. Comput. Graph. Int. Conf.* Cham, Switzerland: Springer, 2023, pp. 155–166.

MUHAMMAD SULAIMAN received the B.S. degree in computer science from COMSATS University Islamabad (CUI), Wah Campus, Rawalpindi, Pakistan, in 2016, and the Master of Science (M.S.) degree in computer engineering from the Ghulam Ishaq Khan Institute Engineering Sciences and Technology (GIKI), Swabi, Pakistan, in 2019. He is currently pursuing the Ph.D. degree in computer science with the University of Stavanger (UiS), Norway. His research interests include image processing, 3D point cloud, online learning, and deep learning on image and LiDAR data.

ERIK FINNESAND received the master's degree in computer science, reliable and secure systems from the Department of Electrical and Computer Science, University of Stavanger. He is currently an Automation Engineer with TietoEVRY Norway.

MINA FARMANBAR received the Ph.D. degree in computer science from Eastern Mediterranean University, Cyprus. She is currently an Associate Professor with the University of Stavanger (UiS), Norway. Her research interests include data science and machine learning applications.

AHMED NABIL BELBACHIR (Member, IEEE) received the Engineering and master's degrees in electronics from the University of Oran, Algeria, in 1996 and 2000, respectively, and the Ph.D. degree in computer science from TU Vienna, Austria, in 2005. From 2000 to 2006, he was the Technical Manager of Austrian contribution on image compression for PACS (ESA-Herschel Infrared Observatory). He joined AIT Austrian Institute of Technology, in 2006, as a Senior Scientist, with a focus on bio-inspired vision. He has been the Director of eu-robotics BoD, since 2021. He is currently the Research Director of NORCE Norwegian Research Centre and leading the DARWIN Group dealing with AI, data, and robotics for automation and autonomous systems. Among others, he co-invented the bio-inspired 360° panoramic 3D camera for robotics within the EUREKA program. He is an Editor of the Springer *Smart Cameras* book 2009 (English) translated into Chinese by China Machine Press, in 2014. He has about 140 publications, 100 invited talks, three granted patents, and two TV documentaries (Euronews/TV2 Norway). His research interests include artificial vision and machine learning for industrial and robot perception.

CHUNMING RONG (Senior Member, IEEE) received the Ph.D. degree from the University of Bergen, Bergen, Norway, in 1998. He was an Adjunct Senior Scientist leading big-data initiative with NORCE, Oslo, Norway, from 2016 to 2019, and the Vice President of the CSA Norway Chapter, from 2016 to 2017. He is currently the Head of the Center for IP-Based Service Innovation, University of Stavanger, Stavanger, Norway. He is also the Co-Founder of two startups bitYoga and Dataunitor, Norway, both the received EU Seal of Excellence Award in 2018. He has supervised 26 Ph.D. students, nine postdoctoral, and more than 60 master projects. He has an extensive contact network and projects in both the industry and academia. He has extensive experience in managing large-scale research and development projects in Norway and EU. His research interests include cloud computing, data analytics, cyber security, and blockchain. He has been honored as a member of Norwegian Academy of Technological Sciences, since 2011. He served as the steering chair from 2016 to 2019. He has been the Steering Member and an Associate Editor of IEEE Transactions on Cloud Computing, since 2016. He served as the Global Co-Chair for IEEE Blockchain in 2018. He is the Chair of IEEE Cloud Computing, an Executive Member of the Technical Consortium on High-Performance Computing, and the Chair of STC on Blockchain in the IEEE Computer Society. He is also an Advisor of the StandICT.EU to support European scandalization activities in ICT. He is the Founder and the Steering Chair of the IEEE CloudCom conference and workshop series. He is the Co-Editor-in-Chief of the *Journal of Cloud Computing* (Springer).