Semi-Supervised Semantic Segmentation of UAV Videos

AKSHAY KUMAR CHAPRANA

JUNE, 2023

SUPERVISORS:
Dr. Michael Yang
Prof. Dr. Ing. F.C Nex (Francesco)

Semi-Supervised Semantic Segmentation of UAV Videos

AKSHAY KUMAR CHAPRANA

Enschede, The Netherlands, June, 2023

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics (GFM)

SUPERVISORS:
Dr. Michael Yang
Prof. Dr. Ing. F.C Nex (Francesco)

THESIS ASSESSMENT BOARD:
Prof. Dr. Ir. George Vosselman

EXTERNAL ASSESSOR:
Dr. Roderik Linderbergh (Delft University of Technology)

This M.Sc. thesis is dedicated to my parents,

SUSHMA,

AJAY VEER SINGH.

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

"Why Should I become someone's shadow
I want to create my own identity
If I have to become a shadow
Then I will become as great as my Father"

- Akshay Kumar Chaprana

ABSTRACT

Unmanned Aerial Vehicles (UAVs) have become crucial in various fields, collecting vast amounts of aerial data. Semi-supervised semantic segmentation techniques play an important role in extracting valuable information from this data. Combining labelled and unlabelled data enables efficient classification and segmentation of objects in UAV imagery. This integration has revolutionized decision-making processes in environmental monitoring, agriculture, infrastructure inspection, disaster management, and security surveillance. The extracted information aids in precision agriculture, urban planning, risk detection, and real-time situational awareness. The synergy between UAVs and semi-supervised semantic segmentation holds immense potential for advancing data analysis and decision support systems.

Semi-supervised semantic segmentation has emerged as a powerful technique in computer vision for extracting precise and accurate object boundaries from images. Unlike traditional approaches that rely solely on labelled data, semi-supervised semantic segmentation leverages labelled and unlabelled data to enhance the segmentation efficiency. By merging the strengths of supervised and unsupervised learning, this approach addresses the challenges of limited labelled data availability while harnessing the abundance of unlabelled data. Various algorithms, such as self-training methods and deep learning models, have been evolved to exploit the potential of semi-supervised learning in semantic segmentation effectively.

This research explored the scope of using semi-supervised techniques for semantically segmenting the UAVid dataset. This dataset brings high-resolution videos in 4K and unique challenges like dynamic object recognition, wide-scale disparity, and temporal consistency continuation. The BiMSANet model is used to segment the semantic classes in the UAVid dataset, as this model is very efficient in dealing with the challenges mentioned above. Only 10 labelled images with 5-sec intervals in each video sequence are available in the dataset. The interval between two frames is reduced from 5 sec to 1 sec to get higher temporal resolution and additional valuable information, which results in more frames in each video sequence. Pseudo-labelling is performed on the newly extracted frames through the use of a trained BiMSANet model on the original labelled images. Three experiments were conducted with different combinations of pseudo-labelled frames and original labelled images to assess the optimum condition for the selection of frames in semantic segmentation of the UAVid dataset. The results from the experiments are presented in the mIoU metrics.

The findings from the experiments of our approach show improvement in the segmentation of most of the semantic classes present in the UAVid dataset. For instance, Exp – 1 shows the best results in Building and Road class, Exp – 2 shows the best results in Static_car and Moving_car classes, and Exp – 3 shows better accuracy in the Vegetation class. In Exp – 3, due to balanced segmentation across all the semantic classes, achieved the best mIoU score of 76.51% (mIoU of 7 classes, excluding Human class). Exp – 3 outperforms the previous best BiMSANet (76.43%) by a margin of 0.08%.

Keywords: Semi-supervised, Semantic segmentation, UAVs data, Self-training, Pseudo labels, Deep learning.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude and appreciation to everyone who contributed to the successful completion of my research/thesis at the ITC Faculty, University of Twente. Their support, guidance, and encouragement have been invaluable throughout this interesting journey.

First and foremost, I am immensely grateful to my first supervisor, Dr. Michael Yang, for his unwavering support and guidance. His expertise, patience, and regular meetings were instrumental in shaping my research and resolving any queries I had. I am truly grateful for his invaluable insights and constant encouragement during these challenging times.

I would also like to extend my sincere thanks to my second supervisor, Prof. Dr. Ing. Francesco Nex, who willingly shared his knowledge and expertise whenever I approached him. His willingness to clarify my doubts and provide guidance played a significant role in enhancing the quality of my work.

A special mention goes to my technical guide, Mr. Kun Li, whose assistance was crucial in overcoming obstacles and solving complex problems. His unwavering support, timely suggestions, and ideas were instrumental in helping me complete my thesis successfully. I am grateful for his dedication, mentorship, and friendship throughout this process.

I would also like to express my gratitude to my friends, Ashutosh Umale, Rishi Pareek, Ashik Shivaprasad, and Tanuj Pareek, who accompanied me on this journey. We shared not only academic discussions but also moments of relaxation, shared meals, and unwavering support. Their presence and constant motivation were pivotal in keeping me motivated and focused on achieving the best possible results.

Last but not least, I am deeply thankful to my family and friends back at home for their unwavering support, understanding, and encouragement. Their belief in me and their continuous encouragement throughout this endeavour provided the necessary strength to overcome challenges and strive for excellence.

I am truly blessed to have had such an incredible support system throughout this research/thesis, and I am sincerely grateful to each and every individual who has played a part in my journey. Without their assistance and encouragement, this accomplishment would not have been possible.

Thank you all for your invaluable contributions, support, and belief in me.!!!!!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background

The growing usage of Unmanned Aerial Vehicles (UAVs) in multiple fields has generated massive amounts of aerial video data (Ahmed et al., 2022). It is critical to extract helpful information from this data for applications such as surveillance, urban planning, environmental monitoring, and infrastructure inspection. Semantic segmentation plays a critical role in interpreting the content of these videos by splitting UAV footage into semantically significant sections (Gadde et al., 2017). Semantic segmentation gives a detailed and accurate scene representation by assigning a specific class name to each pixel or region, like buildings, clutter, vegetation, or tree (Lyu et al., 2020).

Semantic segmentation of UAVs videos has various advantages and benefits. It provides an in-depth knowledge of the scene by allocating semantic labels to each pixel or region, allowing fine-grained evaluation and object-level recognition. This degree of precision is useful for jobs like urban land cover mapping, where the precise demarcation of distinct classes is critical for urban planning and development (L.-C. Chen et al., 2017). Semantic segmentation aids in recognizing and monitoring objects and their motion across time from the scene (Kim et al., 2018). It is critical for applications like surveillance and monitoring, where identifying anomalies or changes in the scene is critical for early warning systems or emergency response.

Traditional supervised learning algorithms depend heavily on labelled training data for semantic segmentation, which can be costly and time-consuming to gather (L. C. Chen et al., 2020). Researchers have adopted semi-supervised learning strategies to solve these restrictions, which use labelled and unlabelled data to augment the efficiency of semantic segmentation models (Zhu, 2008). Semi-supervised semantic segmentation in UAV videos has various advantages over traditional supervised methods. These methods can efficiently harness the information available in the unlabelled samples to improve the models' segmentation accuracy and generalization capabilities by including enormous amounts of unlabelled data that are readily available (Jin et al., 2023). These techniques are beneficial when acquiring large-scale labelled datasets for UAV videos is challenging or not feasible.

The primary concept of semi-supervised semantic segmentation is to use unlabelled data to complement labelled data while training the models (Yuchao Wang et al., 2022). One popular method is "pseudo-labelling" or "self-training." Initially, the model use the provided labelled data for training. A trained model is subsequently applied to the unlabelled data to create predictions, regarded as pseudo-labels (Dong-Hyun Lee, 2013). The unlabelled samples having high-confidence predictions are then added to the labelled dataset, increasing the size of the dataset which will use in training. After that, the model is retrained with the supplemented dataset, and the iterative process tends to improve the model's predictions and efficiency.

This research aims to investigate and evaluate semi-supervised techniques for semantic segmentation in UAV videos. We hope to design a successful and effective approach for extracting comprehensive and accurate semantic information from drone videos by leveraging advanced semi-supervised learning techniques and adapting them to the special challenges of UAV video data. Our work promotes semi-supervised semantic segmentation approaches, allowing for a more efficient analysis of UAV video data and aiding the creation of intelligent systems for various UAV-based applications.

The following sections will examine the approach, datasets, experimental results, and prospective uses for semi-supervised semantic segmentation in UAV videos. By emphasizing advances in this field, we hope to shed light on the benefits and consequences of these techniques in real-world settings involving UAV video analysis.

1.2. Research Problem

Semantic segmentation task is one of the essential functions in machine vision, which has substantially advanced with the progress of deep learning in scene understanding (Yuchao Wang et al., 2022). Semantic segmentation entails assigning each pixel a label in the image to understand the picture/scene (Lyu et al., 2020). Recent developments in visual scene understanding through deep learning provide a perceptive base for numerous industries, including robots, medical image analysis, and autonomous driving. Since deep learning can extract significant semantic information from the training data (Lyu et al., 2021), it serves as the basis for the majority of successful and efficient techniques for scene comprehension problems. However, this learning approach does have a limitation: it needs many training samples to learn relevant information, particularly for real-world applications (Lyu et al., 2020).

Aside from the limited labelled dataset, the semantic segmentation of UAVid videos faces several challenges, including wide scale disparity among objects at different distances or of diverse classes, dynamic object recognition (differentiate the cars, moving or static) in the urban road scene, and temporal consistency preservation for better prediction across frames.

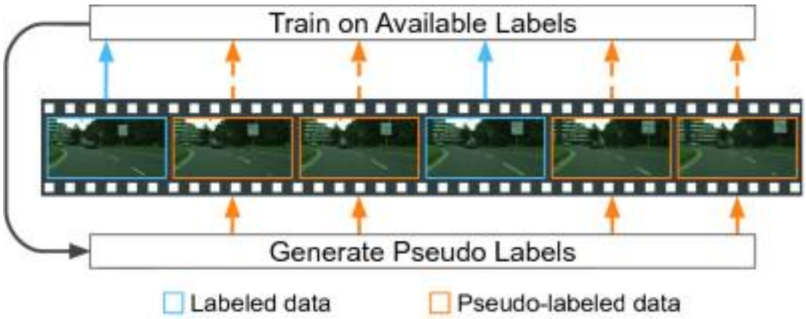


Figure 2: Illustration of Semi-supervised technique used in segmentation of video sequence.(Naive-Student)

A semi-supervised semantic segmentation method is the solution to the above problem (Yuchao Wang et al., 2022). This method learns a model using a limited number of labeled data and many pseudo labels generated on unlabeled ones. In a scientific context, this research aims to optimize the process of semantic segmentation through a semi-supervised approach using UAV videos. This study will use UAVid dataset (Lyu et al., 2021), which has 4K resolution UAV videos in oblique views.

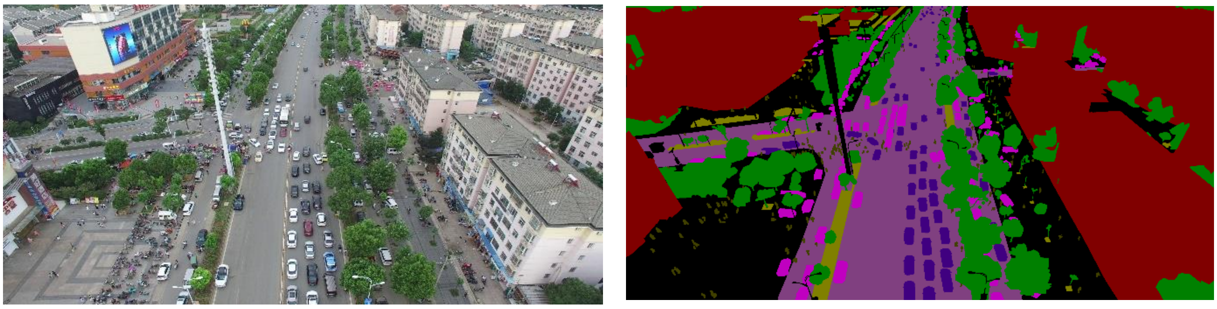


Figure 1: UAVid 2020 Dataset (uavid)

1.3. Research Objectives & Questions

An assessment of the Usability of Semi-Supervised learning for semantic segmentation of UAV video sequences dataset from Urban areas (street scenes).

1. **Sub-objective (1):** To evaluate the effect of time interval or gap between two frames on semantic segmentation.

   1.1. How does the density of labeled images affect the process of semantic segmentation?
   1.2. How many frames do we need to extract high-level semantic information from video sequences?
   1.3. What will be the time interval between two labeled images?

2. **Sub-objective (2):** Examine the implication of pseudo labels usability for semantic segmentation process.

   2.1. How to check the quality of generate pseudo-labels using semi-supervised learning?
   2.2. How can pseudo-labels help to do semi-supervised semantic segmentation?
   2.3. How can the speed of UAVs during data collection affect the quality of generated pseudo labels?

Novelty

It is quite interesting to explore how the time interval or gap between video frames affects the semantic segmentation process. The originality of this research is in determining the best combination of frames from UAV videos to improve the final segmentation outcome. To the best of our knowledge, no one has tried the frames combination with a time interval of less than 5 seconds. We decreased the time interval in this study to 1 second in order to get more information from the videos.

1.4. Scientific and Societal Relevance

The manual labeling of static images under the supervised learning paradigm necessitates a crucial investment of resources and time, whether in the kind of dedicated annotation tools or labor hours. This method would not scale well enough to classify real-time video frames completely (L. C. Chen et al., 2020). It would be ideal if a training method could learn more independently, especially on videos, similar to how visual learning works in humans (Arazo et al., 2020).

Labeling videos is complex and challenging in the context of self-driving automobiles, object tracking, and real-time monitoring, which has led to the emergence of an industry of specialist data annotation firms (Lyu et al., 2019). A video is a temporal and spatial data stream like a picture. To adequately evaluate a video, it is necessary to recognize the items in the scene at particular frames in the video, label their appearance, and track their activity over time. It would be challenging even if a human annotates videos frame-by-frame under close monitoring and control. A person's capacity to manage all the knowledge they must represent for the work at hand is constrained (Enfuse, 2021).

Semi-supervised training will be a good choice to contour the above challenge as it requires fewer labeled datasets for building a deep-learning approach to do semantic segmentation. One can train a model with less labeled training data through semi-supervised learning as compared to supervised learning. For instance a model called Naïve-Student (L. C. Chen et al., 2020) use both self-training learning and semi-supervised learning, where the model is trained using predictions from unlabelled data and where limited labelled data is provided to train the model with unlabeled data respectively.

By introducing the task of urban scene image segmentation to the UAV platform, researchers might learn more about the visual comprehension job in the UAV sceneries and provide the groundwork for more advanced intelligent applications like traffic management, crowd monitoring, disaster management, etc. The semantic segmentation problem utilizing UAV data demands more significant consideration because the information from UAVs has distinct peculiarities (Lyu et al., 2020).

1.5. Key Concepts

1.5.1. Semantic Segmentation

The technique of identifying each pixel of the image using a predetermined set of classes is known as semantic image segmentation (Lyu et al., 2021). We often wish to identify which pixel corresponds to which class in an image containing multiple classes. For instance, in a street scene, one can segment the roadways, pedestrians, trees, bikes.

Semantic segmentation is quite advantageous for autonomous vehicles like self-driving cars and drones. Semantic segmentation allows self-driving cars to recognize items in their environment and make decisions based on that information, such as applying the brakes when another vehicle or object is very close to them. It is beneficial in the autonomous flight and landing of drones. (Bit, 2019)(hackabit).

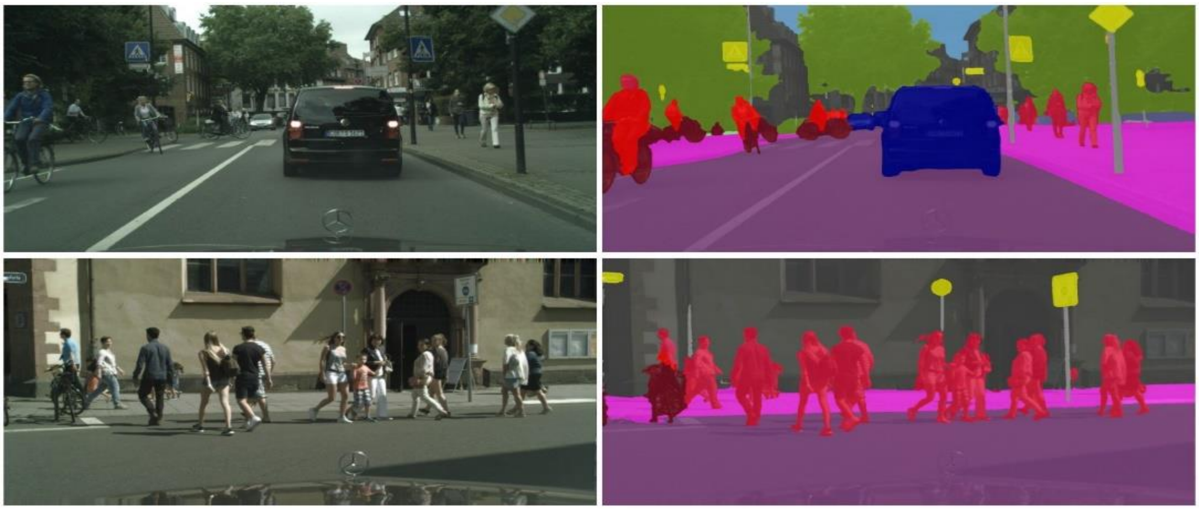


Figure 3: Semantic Segmentation of Images

For instance, the accompanying image has various things that can be utilized as classes for image segmentation tasks, including automobiles, trees, people, road signs, etc.

1.5.2. UAVs

Unmanned aerial vehicles, often known as drones or pilotless aircraft, are operated using cutting-edge components, such as a contemporary sensor, replica models, Ground Stations, communication system, and a platform (Ahmed et al., 2022). UAVs can acquire shots from angles not possible from the ground, and they often operate in dangerous or inaccessible areas for people. The UAVs enable picture retrieval in broad regions more affordable and practical, enabling rapid access to helpful knowledge about a specific region. UAVs acquire photos from the sky with higher spatial resolution and flexible flying schedules than satellites and aircraft, giving the opportunity to quickly monitor and evaluate the terrain at specific moments and locations (Lyu et al., 2020). In the recent years, UAVs have been utilized for both civilians and military purposes, including search and rescue operations, surveillance, weather forecasting, and mapping (Ahmed et al., 2022). UAVs are now also utilized for precise farming (Lottes et al., 2017), cadastral mapping (Crommelinck et al., 2017), and emergency rescue missions during natural catastrophes such as storms, floods, bushfires, etc.

1.5.3. Semi-Supervised Learning

Semi-Supervised learning is a form of process which focuses on performing specific learning tasks utilizing unlabeled data and labeled. Semi-supervised learning techniques are beneficial in situations where labeled data is limited. It allows the utilization of the substantial amount of unlabeled data present on numerous occasions in addition to the more common limited sets of labelled data (van Engelen & Hoos, 2020). The core function of semi-supervised semantic segmentation is to give appropriate pseudo-labels to each pixels in an unlabeled images (Yuchao Wang et al., 2022). Pseudo-label is one kind of label that is generated from a trained model with limited labeled data to be used to label real-time video frames (Dong-Hyun Lee, 2013).

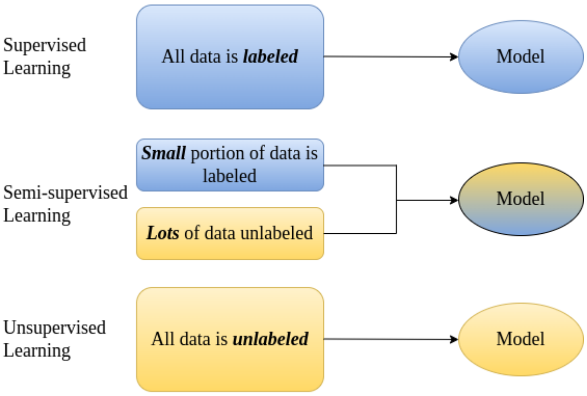

Supervised Learning
All data is ***labeled***
Model
Semi-supervised Learning
Small portion of data is labeled
Lots of data unlabeled
Model
Unsupervised Learning
All data is ***unlabeled***
Model

Figure 4: Represent Semi-supervised vs Supervised vs Unsupervised learning(Semi-supervised)

The major challenge in semi-supervised learning is finding the proper ways to provide reliable and efficient supervision signals for unlabeled input (Jin et al., 2023). To address the issue, two major branches of approach are proposed: entropy minimization and consistency regularization. Entropy minimization, promoted by self-training, works simply by giving pseudo labels to unlabelled data and then mixing it with original labelled data for supplementary re-training. Furthermore, consistency regularization assumes that the prediction of an unlabeled sample is invariant to various types of perturbations (Yang et al., 2022).

In our case, we could utilize self-training approach where pseudo labels help in training the deep neural networks to fulfill the purpose of semi-supervised semantic segmentation.

1.5.4. Pseudo Labelling

“The process of predicting labels for unlabelled data from the aid of the labelled data model is known as pseudo-labeling” (Lee, 2013). In this case, a model was initially trained on the labeled dataset; then, it was employed to provide pseudo labels to the unlabeled dataset. Eventually, both the datasets, labelled and the pseudo labels, are mixed for training the model. As these labels might or might not be real, and as we are producing them based on an identical data model, it is dubbed pseudo (Dong-Hyun Lee, 2013).

The following steps are commonly included in the pseudo-labelling process: First, using regular supervised learning techniques, the training of the model is on the small labelled dataset. The model that has been trained is subsequently utilized to predict labels for unlabelled data, resulting in pseudo-labels (Arazo et al., 2020). These bogus labels are screened to assure their quality and dependability. The filtered pseudo-labels are mixed together with the original labeled data to form an augmented dataset. The model is retrained using this updated dataset, which includes both labelled and pseudo-labeled samples. The pseudo-labelling and training processes are often repeated iteratively to improve the model's predictions and enhance its performance (L. C. Chen et al., 2020).

1.6. Applications

A wide range of applications support the efficacy of semantic segmentation like robotic navigation and autonomous driving benefit immediately; such applications include terrain recognition, path planning, and obstacle identification (Lyu et al., 2019).

Particularly semantic segmentation of UAV datasets is mostly used for surveillance and monitoring in the target region. They have already been employed for weed monitoring, cadastral mapping, smart farming, and precision agriculture, but there hasn't been much research on urban scene analysis. Applications for monitoring traffic, including car accidents and traffic jams, population tracking, and city greenery, including vegetation blooms and damages, may be built on the study of urban scene image semantic segmentation (Lyu et al., 2020).

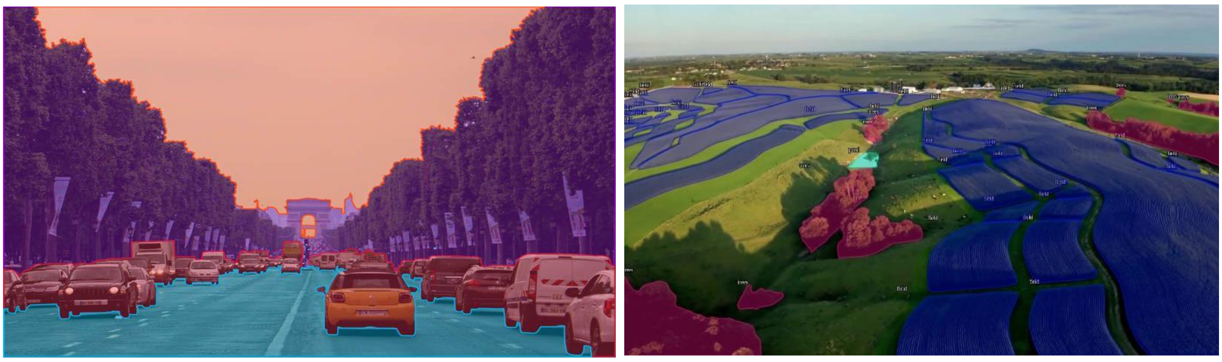


Figure 5: Application of Semantic Segmentation of UAV datasets in traffic management and crop field detection(Application)

1.7. Thesis Structure

This report is primarily divided into 7 chapters, with multiple sub-chapters in each chapter. The State of the Art is reviewed in the succeeding chapter. The overview of the UAVid data is included in the third chapter. In Chapter 4, the research methodology is discussed. It provides an explanation of the purpose and process behind each experiment. The findings are recounted in the fifth chapter, together with the analytical discussions that surrounded them in sixth chapter. The conclusion and recommendations, which includes the research's contributions and shortcomings as well as its future directions, is briefly discussed in the final chapter.

2. STATE OF THE ART

This chapter briefly describes Semi-supervised learning methods in section 2.1 and reviews some networks that have evolved to resolve the multi-scale problem for semantic segmentation in section 2.2. And there is a short introduction to the algorithm used in the research in section 2.3.

2.1. Semi-supervised learning (SSL)

The essential purpose of SSL is to use unlabelled data to build improved learning algorithms. However, this is not every time simple nor entirely feasible. Unlabelled data can only be valuable when it provides information essential to label predictions that are either not available in labelled data or is challenging to extract (Zhu & Goldberg, 2009). The algorithm must gather this information to employ any semi-supervised learning strategy in the application.

(Oliver et al., 2018) used two picture classification challenges to compare multiple semi-supervised neural networks, namely the virtual adversarial training, mean teacher model, and a wrapper method termed as pseudo-labelling. They found that when an extra unlabelled data were added (no eliminating of labelled data), the error rates frequently decreased. Efficiency depreciation appeared only when it was evident that there was a discrepancy between the categories found in the data with labels and the categories available in the unlabelled data. Aforementioned findings are encouraging, indicating that neural networks can use unlabelled data to increase performance in image classification tasks consistently.

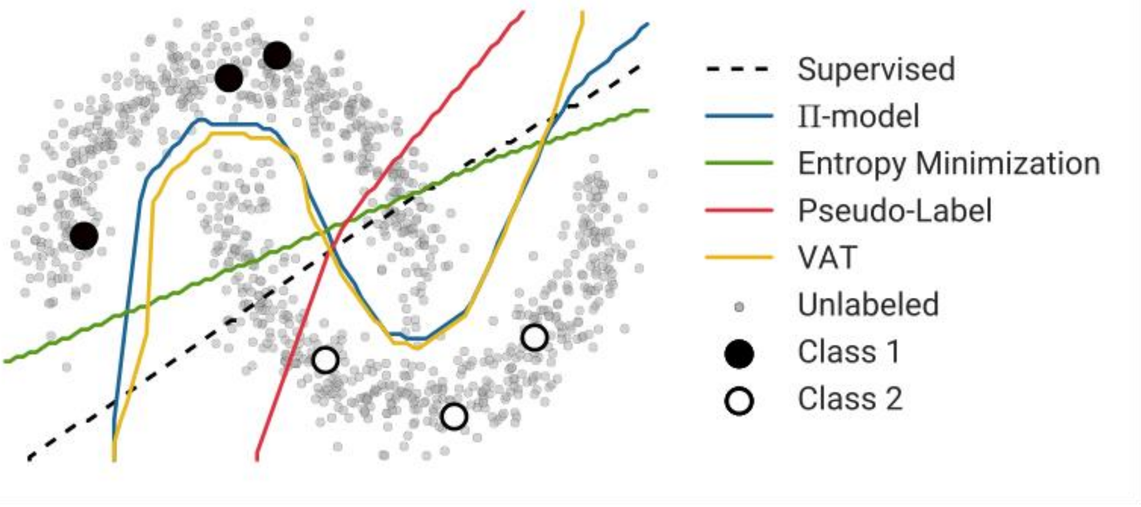


Figure 6: Two moon dataset separated by decision lines, using various SSL techniques, with 6 labelled samples and remaining unlabelled samples.(Oliver et al)

The primary problem in SSL is determining how to create accurate and efficient supervision indicators for unlabelled data. To address the issue, two key approaches are offered: **entropy minimization** (Pham et al., 2021) and **consistency regularization** (Jeong et al., 2019). Entropy minimization, promoted by **self-training** (Lee, 2013), it works simply by labelling unlabelled data with pseudo labels and merging it into human-labelled data to perform re-training. Furthermore, consistency regularization assumes that the prediction of an unlabelled sample is invariant to various types of **perturbations**. FixMatch (Sohn et al., 2020), for example, recommends injecting strong perturbations into unlabelled images and supervising the training process using forecasts from weak perturbation ones to combine the benefits of each techniques. FlexMatch (Zhang et al., 2021) and FreeMatch (Yidong Wang et al., 2022) recently considered the learning degree of distinct semantic classes and then filtered poor-confidence label using classes-specific thresholds. Before describing the self-training and perturbation-based methods we need to know the different types of Semi-supervised learning techniques.

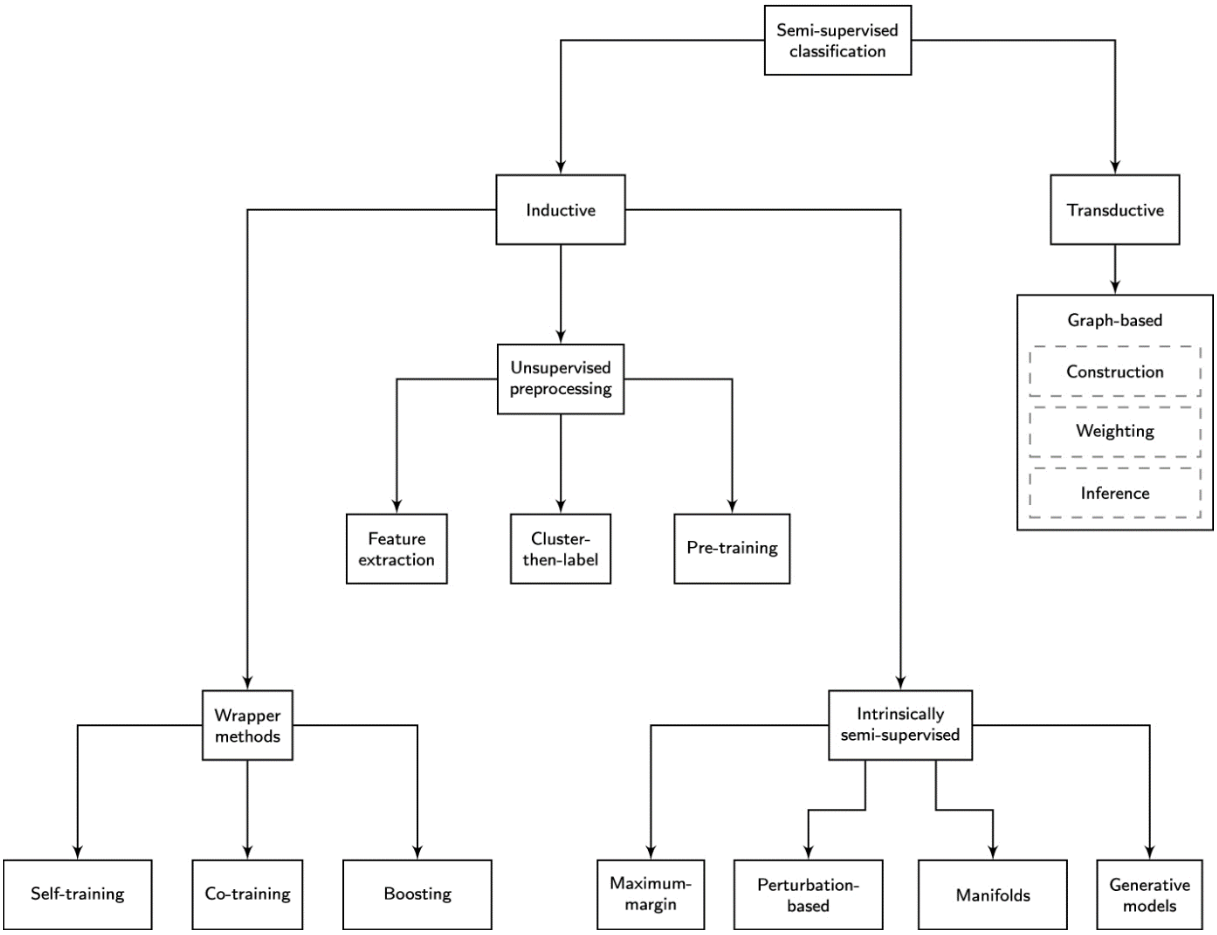


Figure 7 : Illustration of taxonomy for the semi-supervised categorization. Each leaf represents different type of techniques used in segmentation methods using unlabelled data. (I king et al)

From figure 7, The initial differentiation in semi-supervised classification taxonomy, which separates inductive and transductive approaches, which is widespread during dividing the semi-supervised techniques into further parts (Zhu & Goldberg, 2009). Like supervised learning approaches, the former produces a model for classification of semantic class which can be utilized to generate pseudo-labels for hitherto unknown data points. The latter do not produce such models but rather offer straight forecasts (van Engelen & Hoos, 2020). In a nutshell, given a data set $\mathbf{X_L}$, $\mathbf{X_U} \subseteq \mathbf{X}$, having labels $\mathbf{y_L} \in \mathbf{Y_l}$ to the $\mathbf{l}$ labelled data samples, inductive approaches provide a classifier $\mathbf{f: X \rightarrow Y}$. At the same time, transductive methods gives predictions in terms of labels $\mathbf{\hat{y}_U}$ for the unlabelled samples in $\mathbf{X_U}$. As a result, inductive approaches involve model optimization, while transductive methods improve directly across the predictions $\mathbf{\hat{y}_U}$. Here, $\mathbf{X_L}$, $\mathbf{X_U}$ represents labelled data and unlabelled data, respectively. $\mathbf{y_L}$ is the labels for the labelled samples and $\mathbf{\hat{y}_U}$ is the label prediction on $\mathbf{X_U}$.

2.1.1. Inductive methods

The primary aim of inductive approaches is to construct a classifier to generate labels for any data in the input area. Although unlabelled data can be utilized to train this classifier, the prediction for numerous new, hitherto unseen instances are distinct after training is complete (van Engelen & Hoos, 2020). The aforementioned relates to the goal of supervised learning methods: the model is developed in the process of trainings for predicts the labels on unseen or fresh data. The inductive methods can be separated into 3 categories like; Unsupervised pre-processing, Wrapper methods, and Intrinsically semi-supervised methods.

Wrapper methods belong to the simplest and most extensively used semi-supervised learning algorithms (Zhu, 2008). To expand the current supervised approaches to the semi-supervised environment, it initially trains the classifier on data with labels and afterward employs the prediction of the resulting classifier to generate more labelled data. The same classifier can be retrained using the pseudo-labels dataset and the previously utilized labelled data. Wrapper methods are used when pseudo-labels are generated on unlabelled data by a wrapper approach, and the final inductive classifier is constructed by a fully supervised learning technique, unaware of the distinction between initially labelled data and data with pseudo-labels. (van Engelen & Hoos, 2020). This method is further categorised into three parts: Self-training, Co-training and Boosting.

The **unsupervised pre-processing methods**, which are different from wrapper and intrinsically semi-supervised methods, the way of using the unlabelled data and labelled data for training the model in two different steps. In general, the unsupervised phase consists of one or the other the automated extraction or modification of samples features from unlabelled data (features extraction), unsupervised data clustering (clusters-then-labels), or the initiation of the learning procedure's parameters (pre-trainings) (van Engelen & Hoos, 2020). Features extraction, Clusters-then-labels and Pre-trainings are the different categories of unsupervised pre-processing methods. An example of features extraction method that is autoencoder is illustrated in Figure 8. **Intrinsically semi-supervised** algorithms that head on includes unlabelled data into the learning method's objective functions or optimizations process (Zhu, 2008). Most of the methods are straight semi-supervised expansions from supervised learning methods: they expand the supervised classifier's objective function to incorporate unlabelled data. For example, semi-supervised support vector machines (S3VMs) (Ding et al., 2017) go beyond supervised SVMs by maximizing the margins on labelled and unlabelled samples. Many popular supervised learning algorithms, such as SVMs (Cervantes et al., 2020), Gaussian processes, and neural networks, have inherent semi-supervised extensions. This method is further divided into four parts such as: Maximum-margins, Perturbation-based, Manifolds and Generative models.

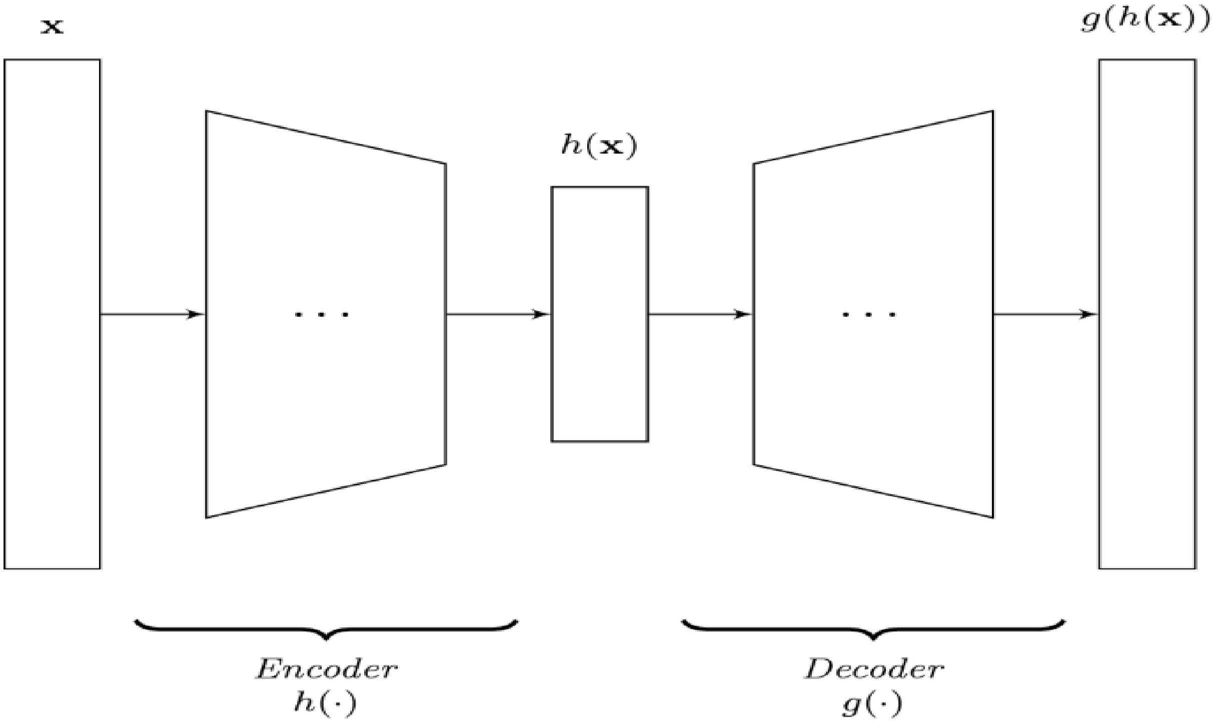


Figure 8: Simplified representation of an autoencoder, an example of semi-supervised feature extraction methods. The networks having multiple layers shown as rectangles and the trapeziums used to show the encoder and decoder.(I king et al)

2.1.2. Transductive methods

In contrast to inductive approaches, transductive methods do not build classifiers for the complete input area. Alternatively, they can only predict things encountered during the training phase (van Engelen & Hoos, 2020). As a result, transductive approaches lack separate train and evaluate phases. As supervised learning approaches, by properties, are not provided with unlabelled data before the evaluation stage, there are no clear correlation of transductive methods with supervised learning. Because transductive learners lack an input space model, the transmission of information must be by direct links between data samples. This realization soon led to the development of a graph-based technique for transductive learning; if the graph is constructed in which similar data points are linked together, information may be communicated via the graph's edges as represented in Figure 9. In general, all transductive approaches are either explicitly or implicitly graph-based (Zhou & Li, 2005).

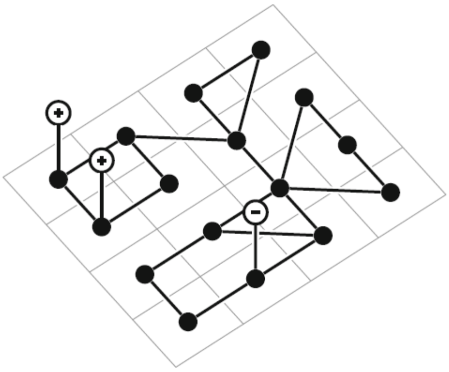


Figure 9: Representation of graph-based techniques used in undirected graphical model. Where black dots are edges and nodes from main graph. White dotes having sign of minus, plus are nodes have labelled data.(I king et al)

Transductive graph-based approaches typically have three phases: graph building, graph weighted, and inferencing the graph. In the initial stage, a collection of items, X, is utilized to build a graph, with each node indicating a data sample and pairs of comparable data samples joined by an edge. In the following stage, these edges are given a weight to indicate the degree of pairwise similarities among the corresponding data samples. In the final stage, the graph assists in assigning labels to unlabelled data samples (Zhou & Li, 2005).

2.1.1.1 Self-training

Self-training techniques (also known as "self-learning" techniques) are one of the utmost fundamental of pseudo-labelling procedures (Triguero et al., 2015). These techniques are built up from a unitary supervised model, trained iteratively on a combination of labelled and pseudo-labelled data, which is generated in earlier iterations of the method. A supervised model is trained with the labelled data only at the start of the self-training process. The trained model is employed to predict the unlabelled data samples. Highly accurate prediction or high-quality pseudo-labels join the data set with original labels, and the supervised model is retrained using the combination of the original labelled data and the freshly generated pseudo-labels. This method works in a loop until all the unlabelled data get pseudo labels (van Engelen & Hoos, 2020).

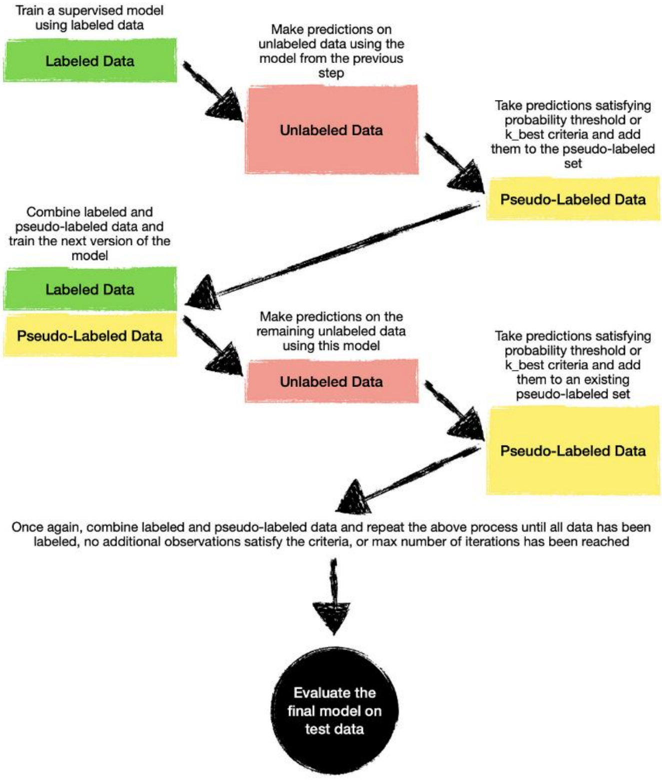

Train a supervised model using labeled data
Labeled Data
Make predictions on unlabeled data using the model from the previous step
Unlabeled Data
Take predictions satisfying probability threshold or k_best criteria and add them to the pseudo-labeled set
Pseudo-Labeled Data
Combine labeled and pseudo-labeled data and train the next version of the model
Labeled Data
Pseudo-Labeled Data
Make predictions on the remaining unlabeled data using this model
Unlabeled Data
Take predictions satisfying probability threshold or k_best criteria and add them to an existing pseudo-labeled set
Pseudo-Labeled Data
Once again, combine labeled and pseudo-labeled data and repeat the above process until all data has been labeled, no additional observations satisfy the criteria, or max number of iterations has been reached
Evaluate the final model on test data

Figure 10: The iterative process of Self-Training.(Self-training)

David Yarowsky presented self-training as a method for the first time, extracting the knowledge from words by examining their context in text sources (Yarowsky, 1995). Following then, various uses and variants of self-training evolved. For example, a self-training was employed to address the challenges of object detection tasks by Rosenberg (Rosenberg et al., 2005). They achieved the highest accuracy during that period by developing the best object detection model. In another instance, the self-training methods were evolved to classify hyperspectral pictures (J. Li et al., 2013). They utilized their domain expertise to choose a collection of prospective unlabelled data and the most significant pseudo-label examples from the predictions of the trained algorithm.

The self-training approach allows for a wide range of pattern considerations, such as data selection to pseudo-label, reusing pseudo-labels dataset in subsequent training of the algorithm, and pause principles (Triguero et al., 2015). The technique for selecting the data used to generate pseudo-labels is critical as it decides the set of data that will be included in the training dataset for the model training. In conventional self-training circumstances wherein this decision is based on prediction probability, the degree of probability estimations significantly influences the algorithm's effectiveness. Prediction probability ordering for unlabelled data, in particular, should indicate the actual confidence level (Yidong Wang et al., 2022).

The primary classifier in self-training are, by principle, unaware of the existence of the wrappers technique (van Engelen & Hoos, 2020). As a result, they must be entirely trained again during every self-training cycle. When a classification model is trained progressively (i.e., by maximizing the objective functions on specific data samples or portions of the available data), an iterative pseudo-labelling technique identical to self-training might be useful (Lee, 2013). Instead of training the whole system again in every repetition, each data sample might be pseudo-labelled throughout the training process. Lee, who pioneered the pseudo-label approach, adapted the aforementioned approach to neural networks (Lee, 2013). As the pseudo-label anticipated in previous training rounds tend to be uncertain, the weights for the specific pseudo-labels grow over time. The methodology of pseudo-labelling is similar to self-training but different in that the classification model is not trained afresh with every pseudo-labelling phase; however, it is optimized with latest pseudo-labelled data and so formally varies from the wrapper methods concept.

2.1.1.2 Perturbation-based methods

Perturbation-based techniques are a type of approach that is often employed in semi-supervised learning. These strategies seek to boost a machine-learning model's efficiency by utilizing labelled and unlabelled data. The unlabelled data is disrupted or manipulated in some way in perturbation-based approaches to generate synthetic labels or more training instances (van Engelen & Hoos, 2020).

Standard perturbation-based procedures include "consistency regularization" or "augmentation-based methods" (Yang et al., 2022). Techniques for data augmentation are utilized in this approach to generate several augmented variants of each unlabelled data point. The model is taught to predict consistently across various upgraded versions. These augmented variants motivate the model to develop more resilient and generalizable representations (Sohn et al., 2020). The augmented data act as a form of regularization, using the unlabelled data to enhance the model's efficiency.

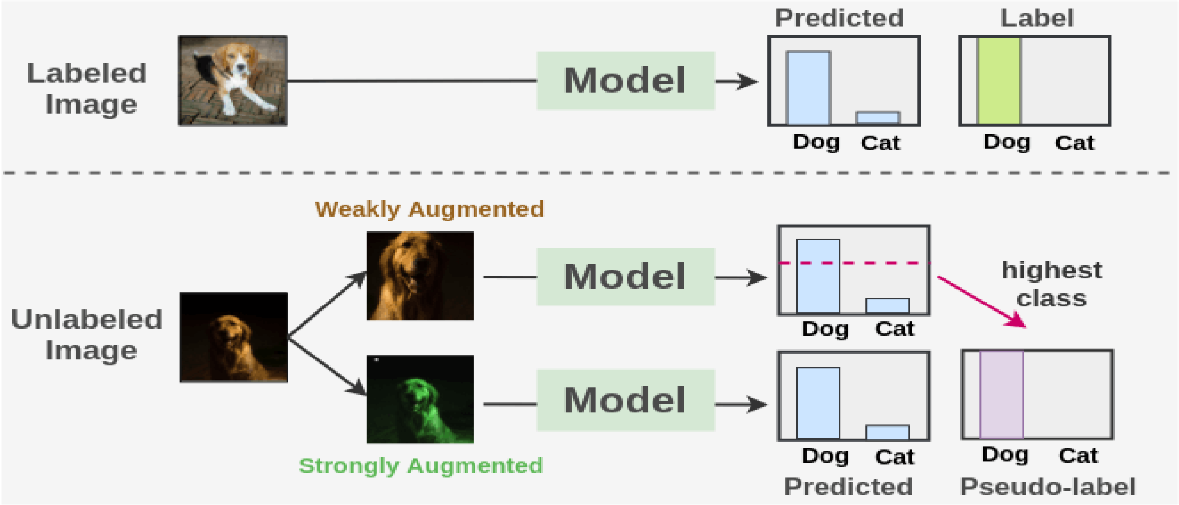


Figure 11: Perturbation based technique used in FixMatch Pipeline, unlabelled image is perturbated into weak and strong augmentation.(FixMatch)

In this manner, Perturbation-based approaches can benefit semi-supervised learning as they leverage the knowledge contained in unlabelled data, which tends to be numerous and easier to get than labelled data. By using this additional information, these strategies may enhance the model's generalization and precision, especially when labelled data is sparse or expensive to get (Arazo et al., 2020).

2.2. Networks for Semantic Segmenation

Modern semantic segmentation techniques count on potent deep neural networks efficiently extracting significant semantic information to identify the class kinds for every pixel (L.-C. Chen et al., 2018). There is typically a performance trade-off when developing deep neural networks for objects of various scales. In remote sensing images, for instance, the feature extraction of a small vehicle is more accessible and handled at higher resolutions when finer elements, such as wheels, can be seen. Since their entire outlines should be seen for semantic segmentation, larger things like highways and buildings benefit from having more global context when recognized (Lyu et al., 2021).

There are numerous ways to address the multiscale issue while designing deep neural networks. The initial approach gradually reshapes features from coarse to fine scales(Shelhamer et al., 2016). The other method is constructing a multiscale feature detection block in the center of the deep learning models (L.-C. Chen et al., 2018). Graph networks, Self-attention have also been employed to collect data globally and strengthen the characteristics of each pixel (Li & Gupta, 2018).

To address the multiscale challenge in semantic segmentation of the UAVid dataset, we explore some existing networks based on the approaches mentioned above to deal with multiscale issues in the dataset. Here, we review some of the network architecture designs to apprehend the networks' working better.

2.2.1. Multi-Scale-Dilation Net (MSDNet)

As the very prime bid to handle the multiscale challenges in the UAVid dataset, the multi-scale-dilation net is introduced (Lyu et al., 2020). The basic concept is derived from the ideology of multiscale image inputs, in which each image's input dataset undergoes scaling by the "scale to batch and batch to scale" operations. The intermedial feature is then concatenated across "coarse to fine scales" before being used to generate the ultimate semantic segmentation result. Figure 12 illustrates the MSDNet architecture. In the following diagrams, the Trunk is representing the extraction of features, the Feat shows the features, and the Seg is the final segmentation head.

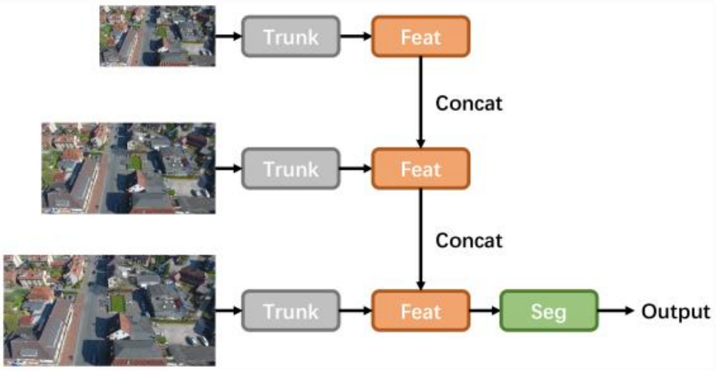
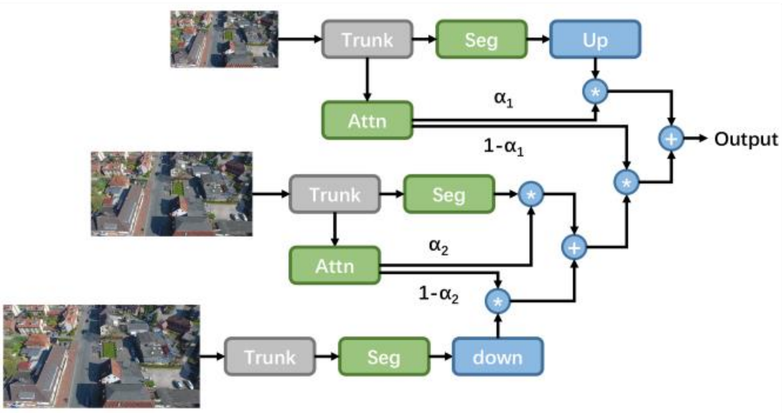


Figure 12: Multi-Scale-Dilation Net. Concatenating the features from coarse to fine scales(.Lyu et al)

2.2.2. Hierarchical Multi-Scale Attention Net (HMSANet)

The Hierarchical multiscale attention net (Tao et al., 2020) is suggested as a hierarchical attention mechanism for learning to merge semantic segmentation outputs from adjacent scales. Deep neural networks are taught to classify images while anticipating weighted masks for score map merging. This technique performs best in the multiscale pixel-level semantic segmentation challenge of the Cityscapes dataset (Cordts et al., 2016). The hierarchical method permits for variable network architectures during training and inference; for example, during training, the network may include just 2 branches of 2 adjacent scales; however, during testing, the network may include 3 branches of 3 adjacent scales, as illustrated in Figure 13. The terms up refer to bilinear upsampling and down refers to bilinear downsampling.

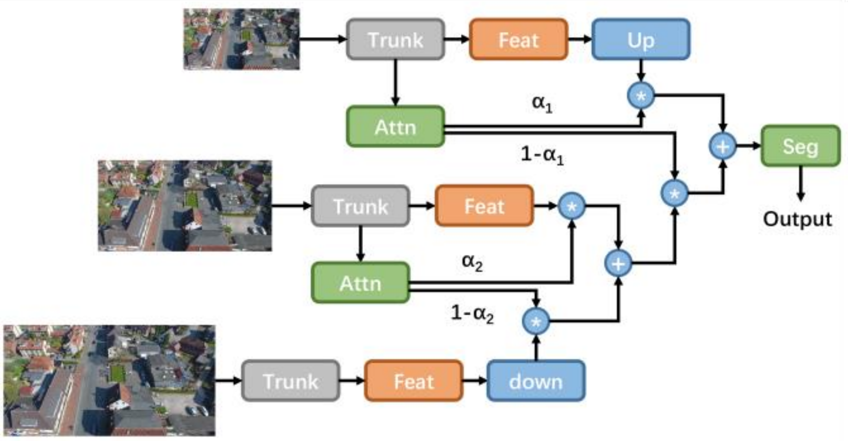
Figure 13:Illustration of Hierarchical Multi-scale Attention net architecture.(Lyu et al)



2.2.3. Feature Level Hierarchical Multi-Scale Attention Net (FHMSANet)

The FHMSANet is the upgraded version of HMSANet; the latter has a limitation: When mixing score maps, it considers the score maps of adjacent scales through linear interpolation, whereas the former has better score maps through interpolation of features contrary to the previous (Lyu et al., 2021). It moves the Seg that is segmentation head to the last of the fused features, which makes the difference and produces better score maps. Figure 14, shows the architecture of FHMSANet where segmentation head is moved after the fused features.

Figure 14:Illustration of Feature Level Hierarchical Multi-Scale Attention net architecture.(Lyu et al)



2.3. Self-Training using BiMSANet

Bidirectional multiscale attention networks (BiMSANet) is a good solution for the multiscale challenges in the semantic segmentation task (Lyu et al., 2021). This method merges the features directed by the attentions of multiple scales along bi-directional routes, i.e., fine-to-coarse and coarse-to-fine. It is motivated by the multiscale attention technique (Tao et al., 2020) and the feature-level fusion strategy (L.-C. Chen et al., 2017). This method was tested over the new UAVid2020 dataset (Lyu et al., 2020). Owing to its oblique seeing style, one of its difficulties is the significant scale variance across and within classes for various things. This approach delivers the best results with an mIOU value of 70.8% (Lyu et al., 2021). This method exceeds the current top-ranked solution, which focuses on addressing the multiscale problem by over 0.8%.

To achieve elegance and efficacy, we drew inspiration from both the Naive-Student model (L. C. Chen et al., 2020) and Bidirectional Multi-Scale Attention Networks (BiMSANet) (Lyu et al., 2021)(discussed in section 4.4). To address the constraints of insufficient labelled data or more unlabelled data, the self-training and semi-supervised learning approaches from the Naive-Student model were investigated. BiMSANet is utilized to handle the multi-scale challenge in the semantic segmentations of the UAVid dataset. Furthermore, Ye Lyu's (the author) code for this network is a publicly available and adequately maintained library with ample documentation for modifying its use. As the network gains cutting-edge efficiency, it has been broadly adapted for other activities by other individuals, with keen community involvement providing an additional benefit in adapting this network for our research.

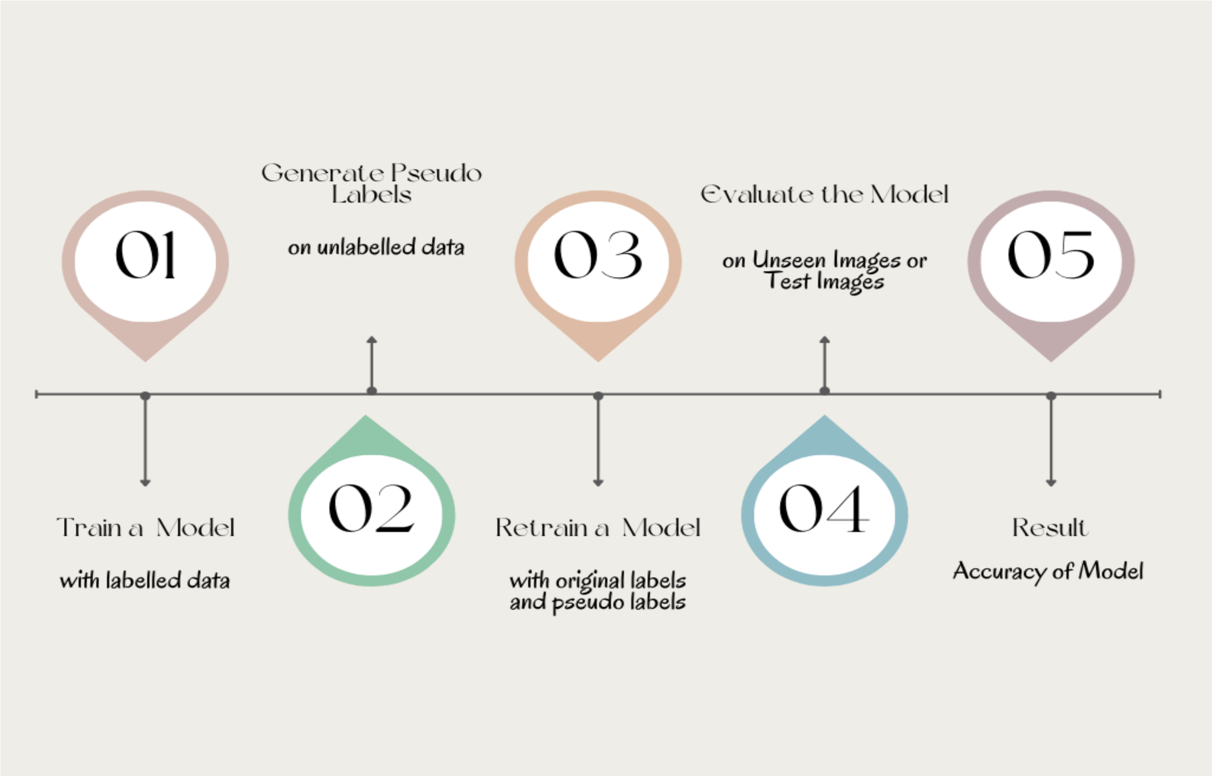


Figure 15: Framework of the model

3. DATA DESCRIPTION

This chapter contains a short summary of the UAVid dataset (Lyu et al., 2020), the class definition used in semantic segmentation, and the dataset splits arrangement used in this work under section 3.1, 3.2, and 3.3, respectively.

3.1. UAVid Dataset

The segmentation of urban scenes, particularly the street environment, will be addressed in this work using a unique UAVid semantic segmentation collection that includes high-resolution UAV photos in oblique angles. A high-resolution dataset for semantic segmentation of UAV videos that focuses on the street segment is called UAVid. The collection comprises 42 sequences (sequences 1 through 42), each recorded in oblique views at a 4K high resolution. A total of 420 photos have been densely classified, with eight classifications to complete the work of semantic tagging. The eight categories include background clutter, moving cars, static cars, low vegetation, roads, buildings, trees, and low vegetation (Lyu et al., 2020).

This dataset is distinctive in that it poses various challenges, such as the wide-scale difference between semantic classes at different distances or belonging to other categories. To Identify movable objects in urban street scenes (distinguishing moving cars from static cars) and the conservation of temporal uniformity for greater forecasting across frames. These difficulties highlight the originality of this dataset. Eight item classes were assigned to 420 high-resolution photos from 42 video sequences. Concerning the number of labeled pixels, this dataset is 10 times larger than the Vaihingen dataset (Rottensteiner et al., 2014), and 5 times larger than the CamVid dataset (Brostow et al., 2008), and twice as large as the Potsdam dataset.

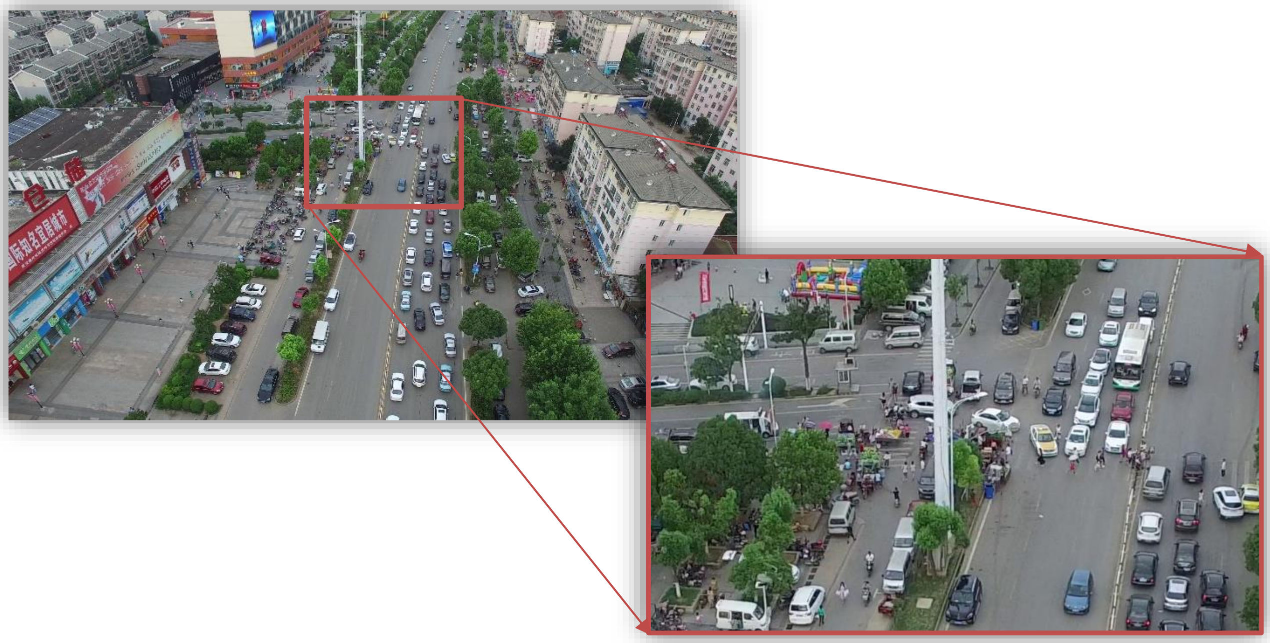


Figure 16: Scene complexity in UAVid dataset

Modern, lightweight drones like the DJI phantom3 pro and phantom4 were utilized for the collection of data. The UAVs eliminate any apparent blurring effects brought on by platform motion by flying consistently at a top speed of 10 m/s. The UAVs' standard cameras are used to capture video using only RGB channels.

This dataset contains scenes that are dynamic and complex with a variety of items. With both stationary and moving elements, the complexity of the scene in real life is what this dataset seeks to achieve. For the UAVid dataset, scenes near streets are preferred because they're complicated and have more continuous human activity. The scene has many objects, including automobiles, pedestrians, buildings, highways, plants, billboards, light poles, traffic signals, etc.

Table 1: Dataset Description

S.NO.	Features	Description
1	UAV used for data collection	DJI phantom3 pro & DJI phantom4
2	UAV flying speed and height	10m/s speed & 50m height
3	Channels of dataset	RGB channels
4	Oblique view	Camera angle 45 degrees to the vertical direction
5	High-resolution	4K resolution videos recording & 4096*2160 or 3840*2160 image resolution
6	Complex and dynamic scenes	The dataset contains both static & moving objects

3.2. Class definition

Thoroughly labelling to all variety of items in the urban scenario in the high-resolution UAV image is prohibitively expensive. As a result, only those that are most familiar and representatives categories of objects are tagged in the UAVid dataset (Lyu et al., 2020). In total, eight classes are chosen for semantic segmentation: Clutter, Building, Road, Static_car, Tree, Vegetation, Human, and Moving_car as shown in Figure 17.

Each class is defined below:

1. clutter: all items that do not correspond to one of the classes defined below.
2. building: represents dwelling house, garage, skyscraper, security booth, and under-construction structures. Walls and fences that stand alone are not featured.
3. road: the surface of a pathways or bridges on which cars can legitimately drive. There are no parking lots included.
4. static car: non-moving vehicles such as cars, buses, trucks, autos, and tractors. Motorcycles and bicycles are not counted.
5. tree: a tall tree with a canopy and a main trunk.
6. vegetation: shrubs, grass and bushes.
7. human: peoples as pedestrian, biker, and all other humans engaged in various activities.
8. moving car: vehicles include moving cars, buses, trucks, and tractors. Motorcycles and bicycles are not counted.

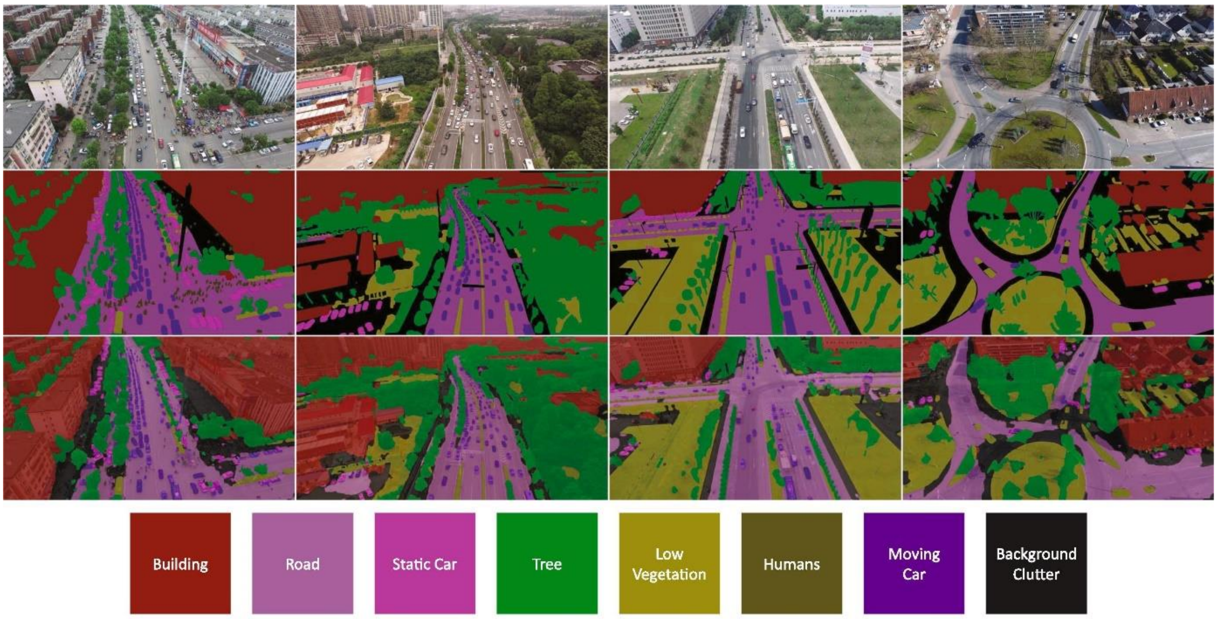


Figure 17: Illustration of different classes with their assigned colour from the UAVid dataset. Original images, ground-truth labels, and predicted labels in first, second, and third row, respectively.(uavid.nl)

3.3. Dataset splits

The 42 heavily annotated sequences of videos are organized into three parts: training, validation, and testing. UAVid doesn't split the data entirely at random, but rather in such a way that each split is representative of the variety of distinct scenes. Every class should be represented in all three splits. UAVid is divided into sequences, with each sequence coming from a distinct scene location. According to this technique, it receives 20 training sequences (200 labelled image) and 7 validation sequences (70 labelled image) for training and validation set, respectively, with publicly accessible annotations. The test set holds the remaining 15 sequences (150 labelled image), the labels of which have been restrained for benchmarking purpose. The size ratios between the training, validation, and test sets are almost 3:1:2.

4. METHODOLOGY

In this chapter, we discuss the methodology used to semantically segment the UAVid dataset using semi-supervised techniques, as shown in Figure 18. Section 4.1 elaborates the procedures for extracting frames from the UAVid dataset videos. Section 4.2 then describes the annotation approach used to label the frames. We examined how to build pseudo labels on unlabelled data frames in Section 4.3. Section 4.4 describes the segmentation model, BiMSANet. The architecture and specifics of the model are detailed further in subsection 4.4. Lastly, in section 4.5, the metrics utilized to compute the effectiveness of the suggested methodology for UAVid semantic segmentation, is described.

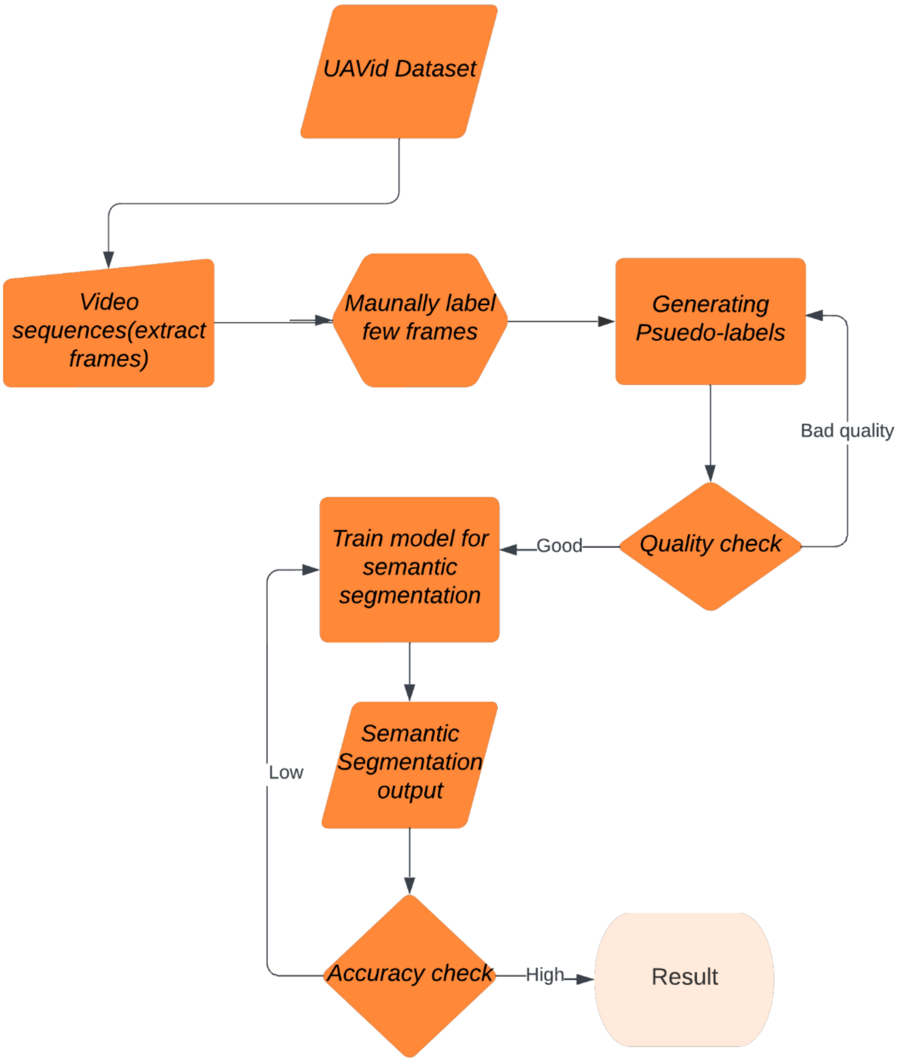

UAVid Dataset
Video sequences(extract frames)
Maunally label few frames
Generating Psuedo-labels
Bad quality
Quality check
Good
Train model for semantic segmentation
Semantic Segmentation output
Low
Accuracy check
High
Result

Figure 18: Methodology Flowchart

4.1. Frames Extraction

To obtain frames from the UAVid data videos, the frame extraction procedure was set up employing Python and the OpenCV package. The videos were fed into the program utilizing the OpenCV package, which provides a dependable and efficient approach for video manipulation and processing (Rashmi, 2020).

The frame extraction criteria were configured to extract frames at varied time intervals. Time intervals of 1 second, 2 seconds, 3 seconds, and 4 seconds were chosen to collect frames at varying rates. With a rate of motion of 20 frames per second, the frame extraction technique involved calculating the frame skip frequency for each time interval. This skip frequency defined how many frames would be omitted between extractions, ensuring that the appropriate time intervals were met. Frames were extracted and preserved based on the calculated skip frequency for each time interval using a loop that successively scanned the video frames. The retrieved frames were saved in a specific area with proper file naming standards, allowing for easier categorization and retrieval for later analysis or utilization.

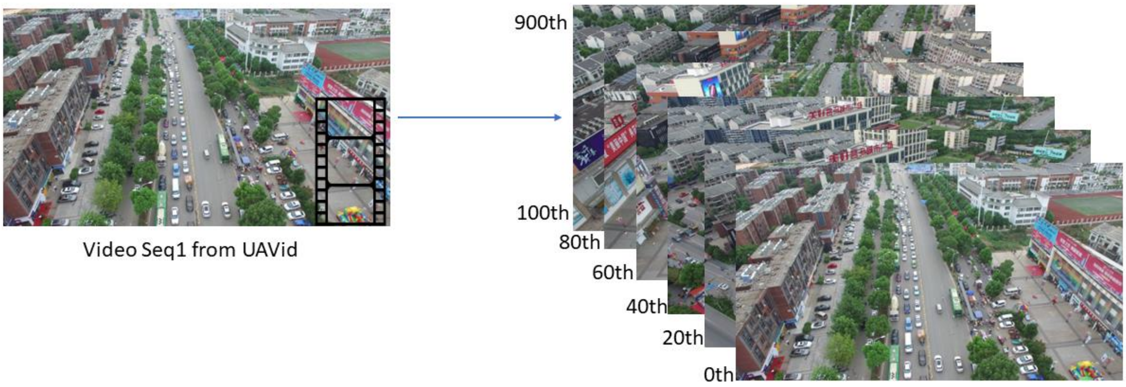


Figure 19: Frames extraction from UAVid video with a time interval of 1 second

Finally, the frame extraction algorithm successfully extracted frames from the UAVid data videos at various time intervals using Python and the OpenCV package. Python and OpenCV together created an efficient and adaptable framework for frame retrieval. Frames were accurately recovered for later analysis or uses in UAV video processing and evaluation by establishing the intended time intervals and applying frame skipping according to the video's frame rate.

4.2. Annotation method

We used annotations provided in the UAVid dataset to conduct our research. The UAVid dataset has supplied densely labelled fine-annotation for high-resolutions of UAVs photos. All of the labels were obtained using the UAVid labeller tool (Lyu et al., 2020). It took about two hours to classify total pixels in a single photo. Annotators can use pixels, super-pixels, and polygon-level annotations method, as seen in Figure 20. The UAVid approach uses a strategy similar to the COCO-Stuff (Caesar et al., 2016) dataset for super-pixel level labelling. UAVid initially uses the SLIC method (Achanta et al., 2012) to divide the photo in to super-pixels, which are groups of pixels which are spatially related and exhibits common features, such as colour and textures.

The pixels inside the identical superpixels get labels of the same sort of class. Superpixel-level annotating is particularly useful for items with sawtooth edges, such as trees. UAVid provides superpixels classification at four distinct scales, allowing labellers to better adapt to items with varying scales. Polygon annotations is especially beneficial for annotating things with straight edges, such as buildings, whereas pixel-level annotation acts as a primary annotating tool. The UAVid annotating tools also includes video play feature nearby specific images to aid in determining whether or not particular items are in motion. Because there may be overlapping items, it labels the overlapping pixel as belonging to the semantic class which is closest to the cameras (Lyu et al., 2020).

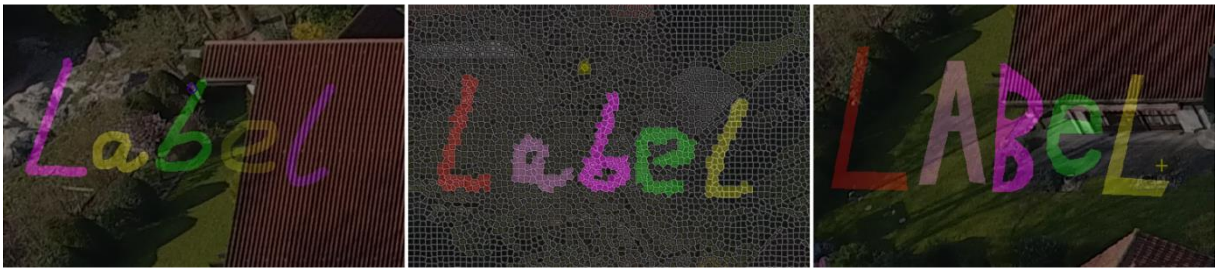


Figure 20: Example of annotation techniques, pixel-level, superpixels-level, and polygon-level annotations from left-to-right order.(uavid.nl)

4.3. Generate pseudo labels through semi-supervised learning

The main problem in semi-supervised techniques is to efficiently use the massive unlabelled data. One extensively used approach is pseudo labelling to deal with the above problem (Dong-Hyun Lee, 2013). As illustrated in Figure 21, the model provides pseudo labels to unlabelled data based on model predictions made on the fly. These data featuring pseudo labels are going to be used as supplemental supervision while training to improve performance.

The following are the main properties of semi-supervised pseudo-labelling (L. C. Chen et al., 2020):

- Unlabelled data is being used in the process of training.
- Using the model-generated pseudo labels on unlabelled dataset to augment the training dataset.
- In training, combine labelled and pseudo-labelled datasets to learn more information.

By utilizing the unlabelled data and assigning pseudo-labels on them, the process of pseudo-labelling leverages the benefits of semi-supervised learning (Dong-Hyun Lee, 2013). It uses the additional information in the unlabelled data to enhance the model's performance beyond what could be enhanced with only the limited labelled dataset. It is fundamental to observe that the efficiency of pseudo-labelling is dependent on the quality of the pseudo-labels and the distribution of unlabelled data (Arazo et al., 2020). In the ideal situation, the pseudo labels should be accurate enough to deliver meaningful supervision to the model throughout training (Iscen et al., 2019).

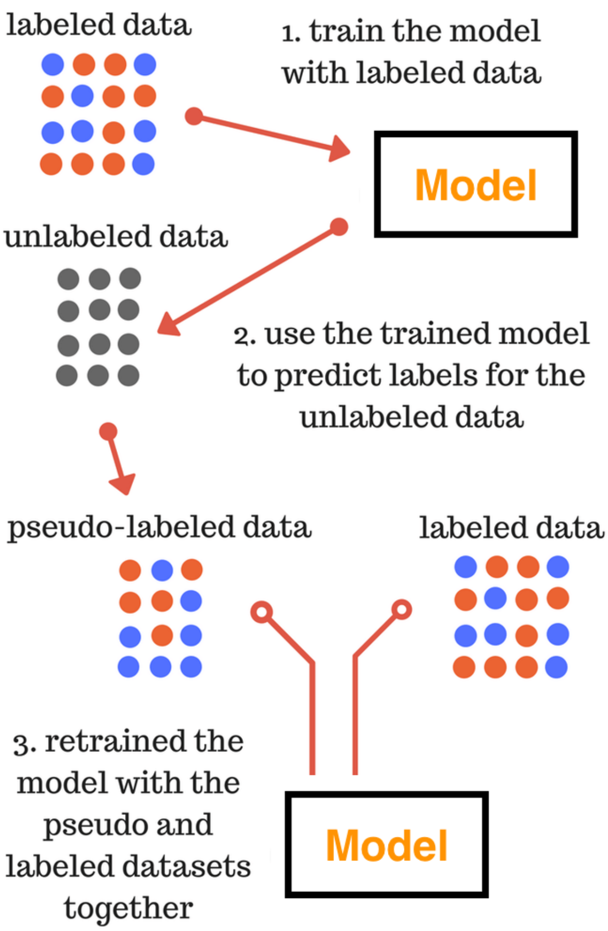
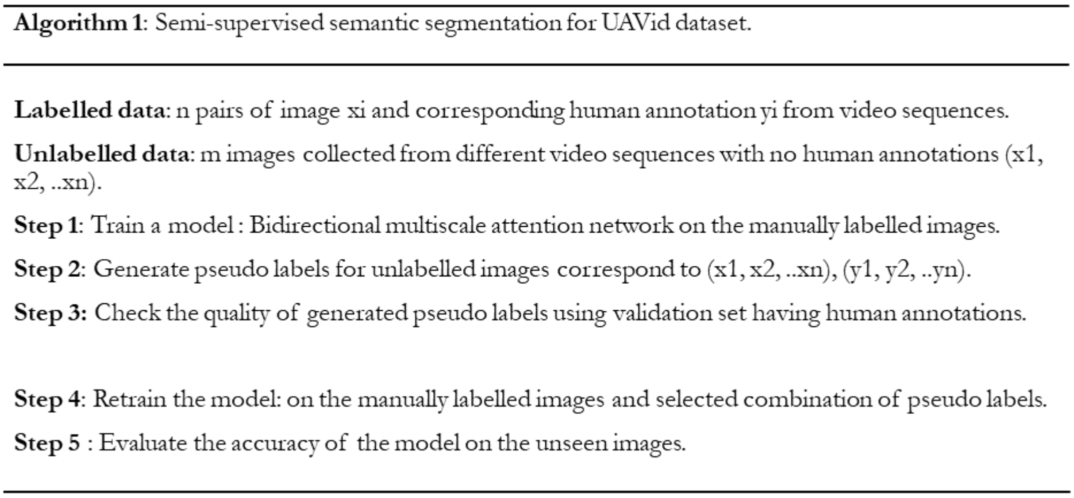

labeled data
1. train the model with labeled data
Model
unlabeled data
2. use the trained model to predict labels for the unlabeled data
pseudo-labeled data
labeled data
3. retrained the model with the pseudo and labeled datasets together
Model

Figure 21: Pseudo Labelling, Semi-supervised learning

We assume the merits of generated pseudo-labels will be comparable to model prediction on the validation set having original labels. Suppose the unseen images are comparable to the training images. In that case, the accuracy on the validation set is more likely to be a better predictor of the model's performance on the unseen images with pseudo labels. The model's generalization ability should be more trustworthy when the unseen images have comparable properties and distribution to the training data.

Overall, pseudo-labelling is a semi-supervised learning strategy since this blends original-labelled and pseudo-labelled data to improve training and capitalize on the advantages of using unlabelled data in a scenario of shortage of fully labelled dataset (Zhu & Goldberg, 2009).

In Figure 22, there is an illustration of the algorithm used to predict pseudo labels and to get the final output of semantic segmentation of UAVid videos.



Algorithm 1: Semi-supervised semantic segmentation for UAVid dataset.

Labelled data: n pairs of image xi and corresponding human annotation yi from video sequences.

Unlabelled data: m images collected from different video sequences with no human annotations (x1, x2, ..xn).

Step 1: Train a model : Bidirectional multiscale attention network on the manually labelled images.

Step 2: Generate pseudo labels for unlabelled images correspond to (x1, x2, ..xn), (y1, y2, ..yn).

Step 3: Check the quality of generated pseudo labels using validation set having human annotations.

Step 4: Retrain the model: on the manually labelled images and selected combination of pseudo labels.

Step 5 : Evaluate the accuracy of the model on the unseen images.

Figure 22: Algorithm used in Semi-supervised semantic segmentation of UAVid dataset. Here 'xi' represent labelled images and 'yi' represent correspond annotation.

4.4. Model: Bidirectional multi-scale attention networks (BiMSANet)

In this Sub-section, we will introduce the model used to predict pseudo-labels and to final semantic segmentation of the UAVid dataset.

4.4.1. BiMSANet Architecture

The BiMSANet approach also considers the hierarchical attention mechanism and feature level fusion. There is an illustration of the framework of the BiMSANet in Figure 23. “The image pyramid is formed for the input image **I** of size H × W, by adding two additional images, $I_{2\times}$ and $I_{0.5\times}$, which are obtained by bi-linear up-sampling **I** to size 2H × 2W and bi-linear down-sampling **I** to 1/2H × 1/2W” (Lyu et al., 2021). The BiMSANet have 2 hierarchical paths for features fusion. The structure for each pathway is similar as for feature-level hierarchical multiscale attention nets. The arrangement of the two paths permits feature fusion from two sides, which helps to determine the fusion weights on a better scale.

The network requires different characteristics for two paths; thus, it employs feature level fusion. When the score maps are utilized for mixing, the Feat1 and Feat2 in both paths will be identical, limiting the representation capability of the paths. Both paths make use of their specific attention features and branches. Attn1 and Feat1 represent the coarse-to-fine path, whereas Attn2 and Feat2 represent the fine-to-coarse path (Lyu et al., 2021). The Feat1 and Feat2 from the two paths are integrated hierarchically among scales to generate the ultimate feature, which concatenates the properties of the two paths.

BiMSANet also uses parameter pooling between different branches. Trunk, Attn1, and Attn2 are three branches that correlate to the three different scale and share the identical network parameters. Feat1 and Feat2 in all the 3 branches differ because they represent the product of distinct images inputs via the similar trunk.

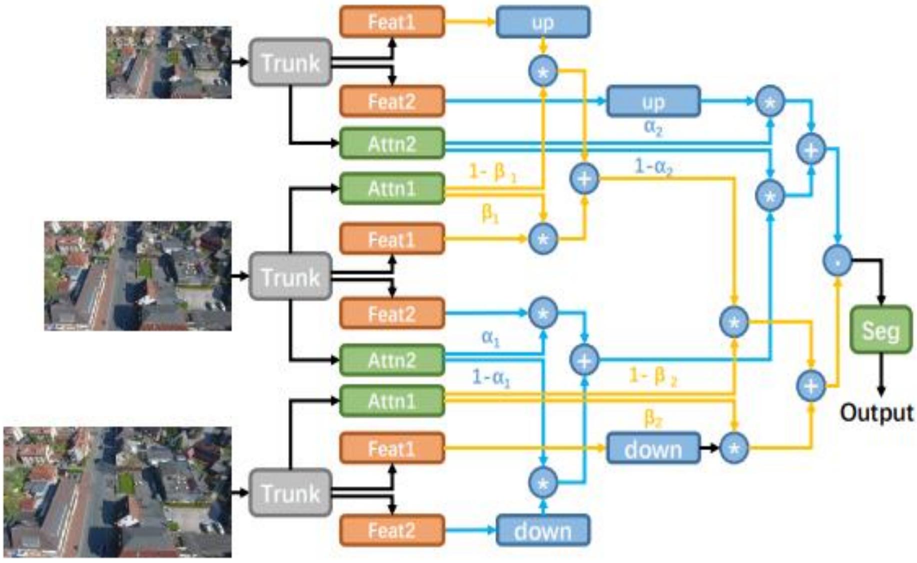


Figure 23: Architecture of the BiMSANet. (BiMSANet)

4.4.2. Architecture Details

In this Sub-section, the description of the components used in this architecture are mentioned.

Trunk: BiMSANet chose the deeplabv3+ (L.-C. Chen et al., 2018) as the trunk to extract features from each scale properly. As the backbone, the model uses wide residual networks (Zagoruyko & Komodakis, 2016), specifically the WRN-38, previously trained on the imagenet data (Deng, 2009). Deeplabv3+'s ASPP module offers convolutions with atrous rates of 1, 6, 12, and 18. The deeplabv3+ features called $\mathbf{f_b}$ are improved further with the following modules: **Conv3 × 3(256)- > BN- > ReLU- > Conv3 × 3(256)- > BN- > ReLU- > Conv1 × 1(nc)** (brackets contains the number of output channels), which correlates to the modification of features in the **Seg** of the HMSANet preceding the final segmentation (Lyu et al., 2021). Conv, BN, and ReLU are convolution, batch norm, and rectified linear unit acronyms (L.-C. Chen et al., 2018).

An image input **I** is transform by the trunk **T** into feature maps **f** with **nc** channels, i:e., **f = T(I). nc =** $\mathbf{n_{class}}$ **× d**, where $\mathbf{n_{class}}$ represents the number of classes. The channel expansion rate is represented by **d.** Here expansion rate d is set to 4. The first 1/2**nc** channels are dedicated to the Feat1, whereas the second 1/2**nc** channels are dedicated to the Feat2 (Lyu et al., 2021).

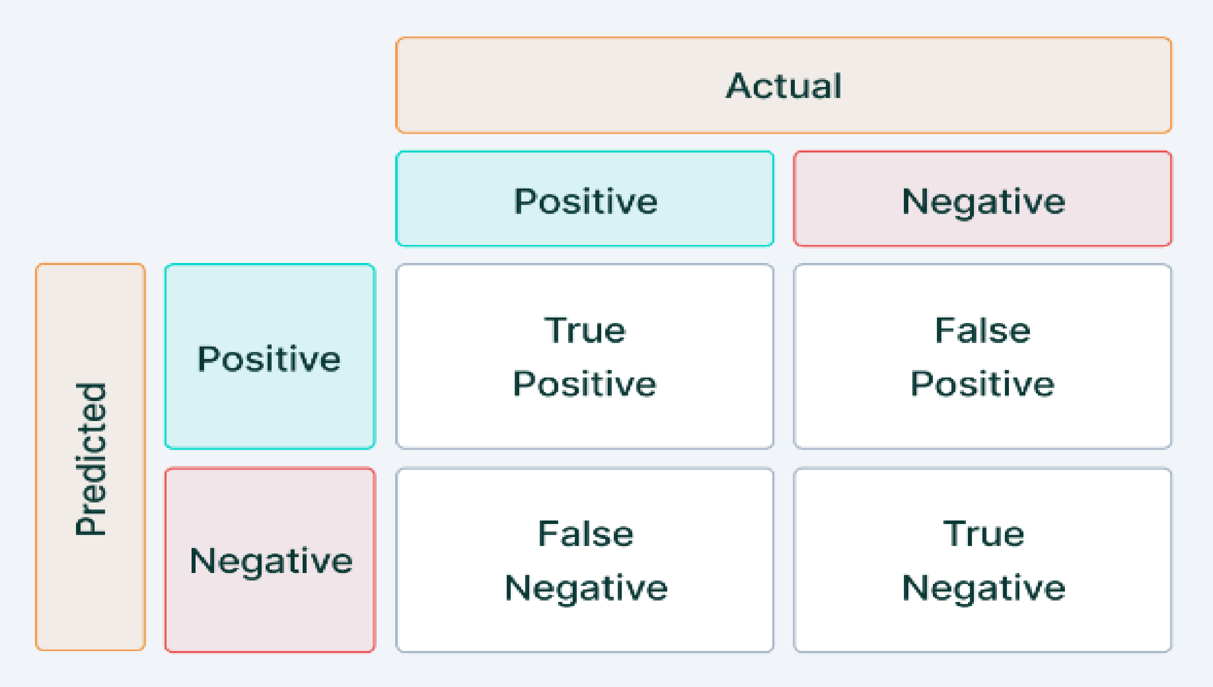
Attention head: The Attn1 and Attn2 have the same structure but differ in their specifications. The attention heads translate the deeplabv3+ features $\mathbf{f_b}$ to the attention weights α, β (ranging between 0.0 to 1.0 with 1/2 nc channels) for the two paths. The structure is composed of the following parts for each attention head: **Conv3 × 3(256)- > BN- > ReLU- > Conv3 × 3(256)- > BN- > ReLU- > Conv1 × 1(1/2 nc)- > Sigmoid** (brackets contains the number of output channels) (Lyu et al., 2021).

Segmentation head: The segmentation head called **Seg** is responsible to transform the input fused feature maps $\mathbf{f_{fused}}$ to a score maps **S** (8 channels in case of UAVid dataset), which correlates to the probabilities of class for each pixels, i.e., **S = Seg(**$\mathbf{f_{fused}}$**)**. The segmentation head **Seg** is basically the convolution of size 1×1, **Conv1 × 1(**$\mathbf{n_{class}}$**)**. The final class labels for all pixels are output by the Argmax operation alongside the channel dimension (Lyu et al., 2021).

Auxiliary semantic head: Model also uses the semantic segmentation auxiliary head for every branch in training, as in (Tao et al., 2020) , which is composed of 1×1 convolution, **Conv1 × 1(**$\mathbf{n_{class}}$**)**.

4.5. Evaluation Metrics

The comprehensive confusion matrix for each semantic classes created from the network's test semantic segmentation results can be used to calculate the quantitative efficiency of the developed algorithm. The resulting confusion matrix includes an extensive description of each class's accurate and incorrect classifications. Figure 24 illustrates a binary classification in a basic confusion matrix, with green column indicating right (positives) classifications and red column indicating incorrect (negatives) classifications.



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 24: Confusion matrix with green column as positive and red column as negative.

To evaluate the model output, we initially must determine what constitutes a prediction as positive detection or a true positive (Jordan, 2018). The model will calculate the IoU score for every (targets, predictions) and then identify which mask combinations have an IoU score higher than a predefined threshold value. Figure 25 shows, a mask combination, if the IoU score is higher than threshold value of 0.5 only then it is true positive, otherwise it is treated as false negative. In our case of the UAVid dataset, we put this threshold value to 1.0, which means if the predicted and ground truth mask entirely overlap, only then is that pair treated as true positive.

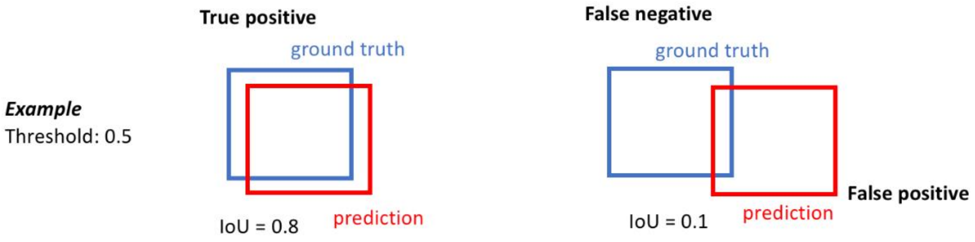


Figure 25: Threshold value define True positive and False negative.(Evaluating image segmentation model)

Pixel or Overall Accuracy: The most often used statistic for evaluating semantic segmentation is simply reporting the share of pixels in the test images that was identified correctly (Jordan, 2018). Pixel accuracy is frequently given independently for each class and globally throughout all classes. We are simply evaluating a binary mask when calculating per-class pixel accuracy; a true positive reflects the pixels that are accurately predicted to belong to the particularly given classes (depending on the target masks), whereas a true negative reflects the pixels that have been correctly identified as not falling within the specific given classes (Nitr, 2020).

$$\text{Overall Accuracy} = \frac{\text{TP + TN}}{\text{TP + TN+ FP + FN}}$$

Equation (4.1)

When the class appearance is minimal within the image, or the pixel number distribution of multiple classes is imbalanced, this metric can produce deceptive findings because the measure is biased in reporting how effectively the model identifies negative cases (i.e., when the specific class is absent) (Shorten & Khoshgoftaar, 2019). In a case, where few classes lead the dataset, the model can achieve high accuracy by predicting the leading class for most pixels while failing miserably on the minority classes (Lyu et al., 2020). However, when working with class-imbalanced data, other measures such as Mean intersection over union (mIoU), F1-score, Recall, and Precision are more effective evaluation metrics (Marsocci & Scardapane, 2022).

Mean Intersection over Union (mIoU): In the case of semantic segmentation with many classes, mIoU is typically thought to be a more useful and reliable indicator than total accuracy (Jordan, 2018). It assists in determining how well the model can segment each class, such as both dominant and minority classes.

mIoU gives a more thorough evaluation of the model's ability to capture the borders and spatial interactions of various objects or regions by examining per-class performance (Nitr, 2020).

The Intersection over Union (IoU) statistic, commonly known as the Jaccard index, compute the percentage overlap between the ground truth and our prediction output (Rezatofighi et al., 2019). Simply put, the IoU metric counts the quantity of pixels shared by the ground truth and the predicted masks divided by the overall pixels shared by both the masks.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$

Equation (4.2)

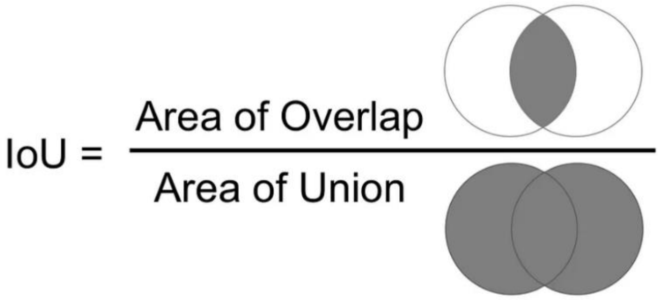


Figure 26: Representation of IoU formula-the area of overlap as TP; the area of union as TP+FP+FN.

The IoU score is generated separately for each semantic class and then meant across all classes to generate an overall **mIoU** score used in our semantic segmentation prediction.

5. RESULTS

The interpretation of the research outcomes is presented in this chapter through Experiments 1-3, using different combination of the frames extracted from the UAVid videos in section 4.1. Here, BiMSANet is the base classifier used for semantic segmentation in all the experiments, which perform segmentation for all 8 classes in the dataset as shown in Figure 27. We conducted 3 experiments with distinct combinations of frames having original labels and pseudo-labels generated by the trained model on original labels. In section 5.1 and 5.2, we presented model performance results and quality of pseudo labels, respectively. We discuss experiment – 1, 2, and 3 in section 5.3, 5.4, and 5.5 respectively and report accuracy of all experiments through mIoU metrics on test set. We have 10 original labels with 5 sec time interval in each sequence.

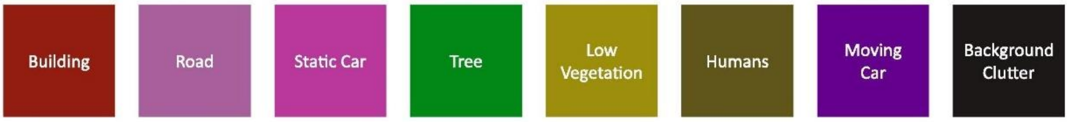


Figure 27: Representation of semantic classes with their identity colour.

All models used in the experiments were built with Pytorch (Paszke et al., 2019) and performed on an exclusive Nvidia A40 GPU with 48GB of memory and a batch size of two images. The model makes use of mixed precision and synchronous batch normalization (Lyu et al., 2021). The training optimizer used stochastic gradient descent with momentum 0.9 and weight decay of $5e^{-4}$. A "polynomial" learning rate policy is used (Liu et al., 2015), with a poly exponent of 2.0. The initial rate at which learning begins is set at 5e-3. The model undergoes training for 175 epochs using random image selection. We use random scaling to scale the photos from 0.5× to 2.0×. Image patches of 896 × 896 pixels in size are obtained through random cropping.

Even though the model in all experiments was trained for the 8 semantic classes presented in the dataset, the **mIoU** is calculated for 7 semantic classes, excluding humans. We exclude Human class from the evaluation part, as it bring class imbalance in the dataset because the size of this class is very small and have very less pixel share in the dataset. Most important is when we produce pseudo labels this class performs poor accuracy and only become the potential source of noise which negatively effects the model's efficiency. In section 5.2, we presented the quality assessment of the semantic classes in pseudo labels. We selected the classes with an accuracy of more than 50% in pseudo labels to ensure that the model feed with good quality information for better generalization.

5.1. Model Performance

Our research uses BiMSANet as a base model to predict the semantic segmentation for multiple classes in the UAVid dataset. Initially, the base model is trained with ground-truth labels, in our case, 10 labels with 5 sec time intervals from each sequence. We used 200 labelled images for model training phase, and we have 70 labelled images for validation. We used a validation set to assess the efficiency of the trained model on unseen samples; it estimates how well the model can generalizes to new samples. It aids decision-making, such as choosing the optimum model or hyperparameter configuration. The validation set serves as an interim checkpoint to confirm the model's performance before the final test set evaluation.

Table 2: Accuracy of the model in mIoU metrices for semantic classes on validation set. Mean, maximum and minimum scores are in blue, green and red respectively.

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
BiMSANet	78.06	73.51	94.07	85.65	80.88	81.76	74.33	49.76	84.55

Table 2 depicts the semantic segmentation results on the validation set, and the measuring metric is mIoU. The IoU score for all eight semantic classes can be observed; the building class has the highest accuracy while the Human class has the lowest accuracy, and the total accuracy for the model is mIoU is 78.06, which shows that the model can generalize to new instances with this accuracy.

Figure 28 shows the ground truth, the prediction by the base model on the validation image, and the original image from seq18 (image 000000) in the validation set. We used bounding boxes to compare the segmentation quality in the prediction label with ground truth. The blue box shows that Building and Tree classes are well segmented with clear-cut boundaries. Human class is segmented partially; the density of humans in the predicted label is reduced compared to the ground truth, as represented in the white box. The orange box shows that model predicts the road in place of the clutter class. The model generally performed well with balanced prediction to different semantic classes, whether small or large.

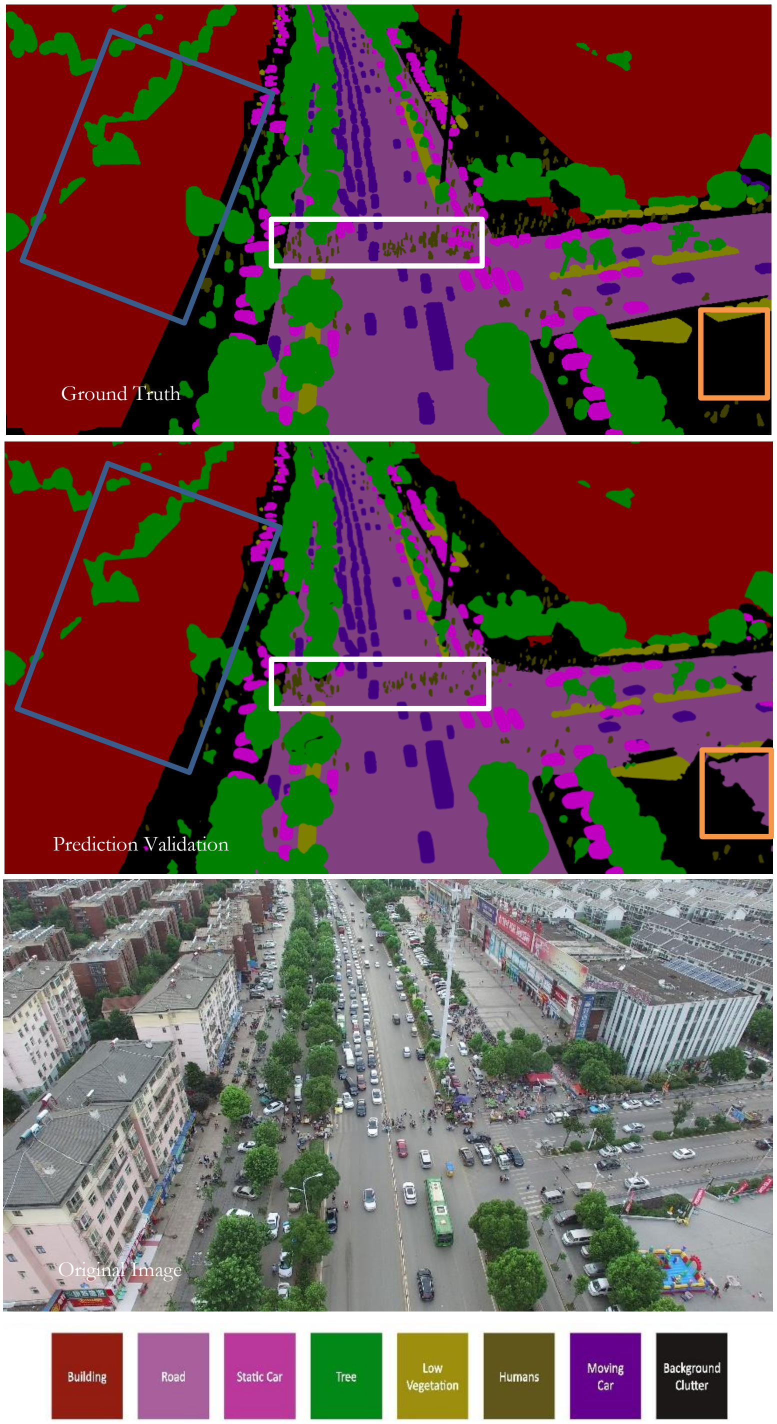


Figure 28: Semantic segmentation results on Validation set. Bounding boxes in blue, white, and orange shows comparison.

5.2. Quality of Pseudo labels

In our research, we reduced the time interval between two frames from 5 to 1 sec to enhance the temporal resolution and label consistency. Initially, we had 10 labelled images in each video sequence, but after reducing the time interval, we needed to extract more frames from the video; we extracted new frames with a time interval of 1 sec and resulting in 46 frames in each sequence and 920 frames from all the sequence. Pseudo-labels were generated on the newly extracted frames with the help of a trained base model. We performed both visual and evaluation checks to ensure the quality of the generated pseudo labels.

Table 3: Table 3: Accuracy of predicted labels on 200 common frames.

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
BiMSANet	76.72	75.33	93.74	87.49	76.15	82.98	73.41	43.73	80.92

We need the ground truth of those images to evaluate the model prediction on any images. In our case, we do not have ground truth for the newly extracted frames from the video sequence, but there are some common frames between the pseudo-labelled set and the originally labelled set. For instance, initially, we had 0th, 100th, 200th, …, 900th frame with ground truth in each sequence, and after extracting new frames, we have 0th, 20th, 40th, 60th, 80th, 100th, 120th, …., 900th frame with pseudo labels in each sequence. According to this scenario, we have 200 frames with ground truth out of 920 frames. We can evaluate the accuracy of 200 frames that are common and have ground truth. We can rely on the model's generalization ability for the remaining frames, which we already evaluated on the validation set.

Table 3 depicts the accuracy of generated pseudo-labels over the 200 common frames among the original and pseudo-labelled sets. The Human class has lowest accuracy with 43.73% in pseudo labels, it will propagate noise to the model and degrade the ability of further prediction in Human class. From Figure 29, we can visually inspect the quality of pseudo labels for other frames by comparing them with Frame 400, which is common in both sets and has an accuracy score in mIoU metrics. And some frames have poor pseudo-labels due to blur-effect in the frames because of high speed of camera rotation.

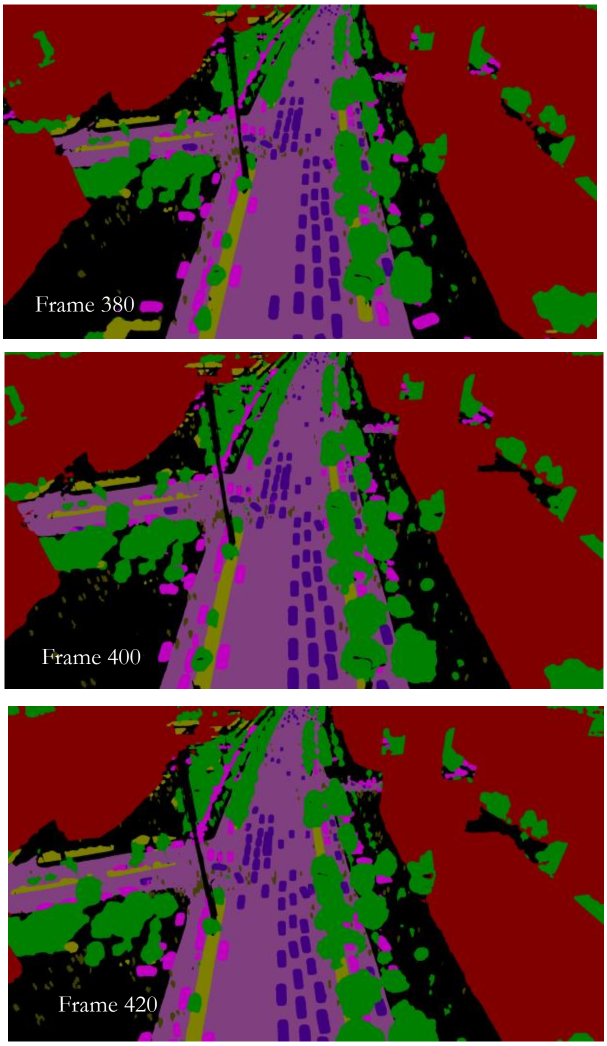


Figure 29: Quality of Pseudo labels generated. Frames are shown from seq1 of training set.

5.3. Experiment – 1 (Exp – 1)

In this particular experiment, the selection of a combination of frames, all original labels, and pseudo labels with a 1-second time gap. Each sequence has 10 original labels and 46 pseudo labels. Figure 30, depicts the chosen frame combination. We trained the model with 1120 images in total from all the sequences, driven by the notion that decreasing the time interval between frames can improve temporal resolution and label consistency, resulting in an improved model's ability to learn.

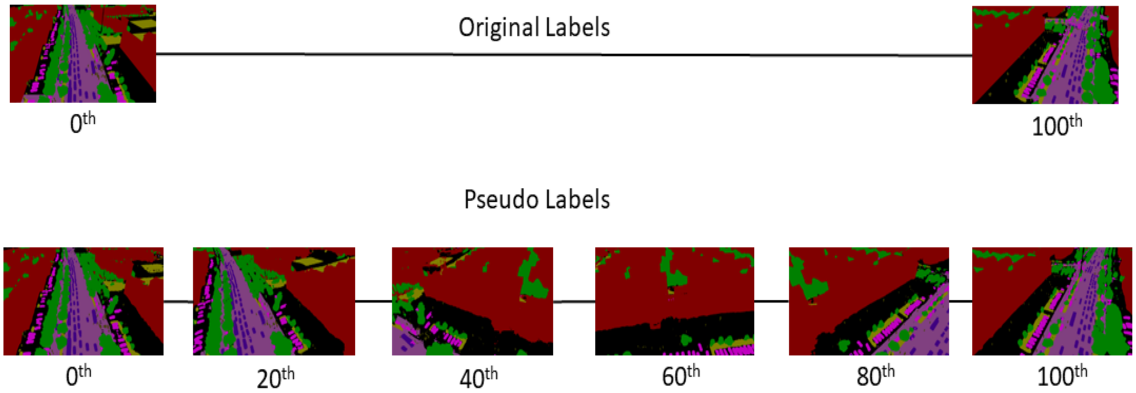


Figure 30: Selected combination of frames for Experiment -1.

In table 4, the IoU for all the semantic classes and mIoU are presented. The Building class shows the highest score with 89.13% and Vegetation class has the lowest score with 62.59%. The other classes shows prediction scores in between the above two classes and the mean of all the classes is mIoU shows 76.10%.

Table 4: Accuracy of the model in mIoU metrices for semantic classes. Mean, maximum and minimum scores are in blue, green and red respectively.

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
Experiment-1	76.10	69.33	89.13	81.86	72.81	79.48	62.59	-	77.56

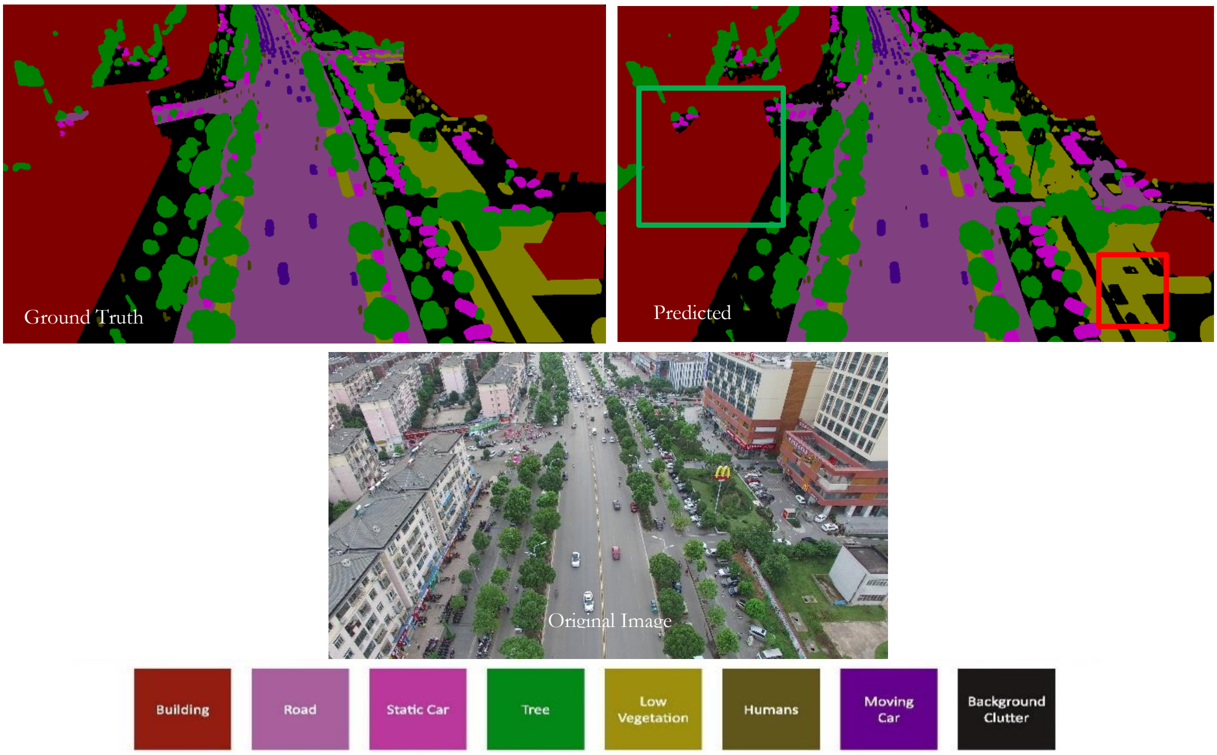


Figure 31: Semantic segmentation results with bounding boxes in green and red from predicted label compare to ground truth.

Figure 31, shows the original image, ground truth, and predicted results from the experiment – 1 for the image 000600 from seq22 in the test set. The bounding boxes in green and red colour from predicted label shows the quality of segmentation of Building and Vegetation class respectively. The red box shows that the model predicts some part of Vegetation as Clutter class; if we compare this area with the original image, the model predicts correctly. However, if we compare this area with ground truth, it is a wrong prediction. In ground truth, this area is labelled as Vegetation and not Clutter; in reality, this area is more similar to Clutter class. The other classes like Road, Tree, Moving_ car, and Clutter performed well, but Static_car showed some drop in accuracy.

5.4. Experiment – 2 (Exp – 2)

In this particular experiment, we use other combination of frames, with all the original labels, and the pseudo labels which are adjacent to the original labels. So, from each sequence we took 28 pseudo labels and 10 original labels. In total, the model was trained with 760 images. The idea is to reduce the additional noise by reducing the number of pseudo labels and maintain the temporal resolution and label consistency by having adjacent pseudo labels. The adjacent frames are similar to original frames as shown in Figure 32, the pattern of information can be easily generalize by the model and help to better understand the dynamic objects.

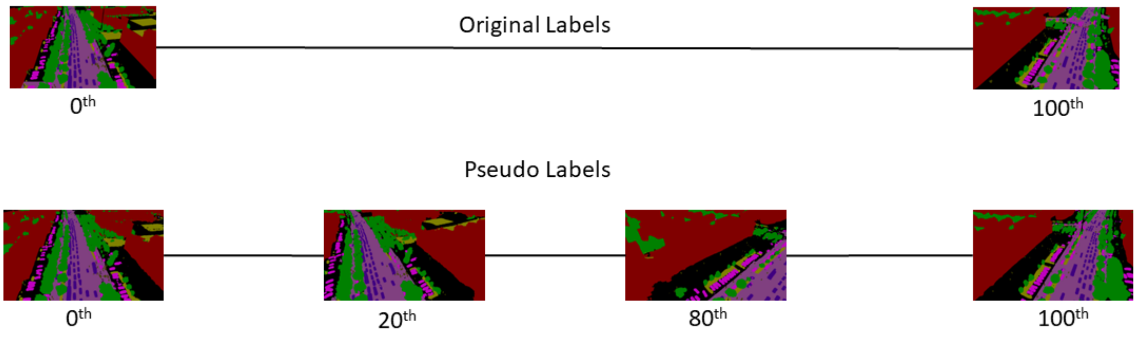


Figure 32: Selected combination of frames for Experiment - 2.

Table 5 presents the result of semantic segmentation of different semantic classes with mIoU score. The green, red, and blue colour show the highest, lowest, and mean score values for semantic classes, respectively. The orange colour shows the semantic class with maximum improvement from the previous experiment. In this condition, the Static_car class shows massive improvement from 72.81% to 76.28%. The mIoU also shows better results and increased from 76.10% to 76.28%.

Table 5: Accuracy of the model in mIoU metrices for semantic classes. Mean, maximum and minimum scores are in blue, green and red respectively, and orange box shows maximum improvement.

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
Experiment-2	76.28	68.68	88.55	80.46	76.28	79.71	62.49	-	77.77

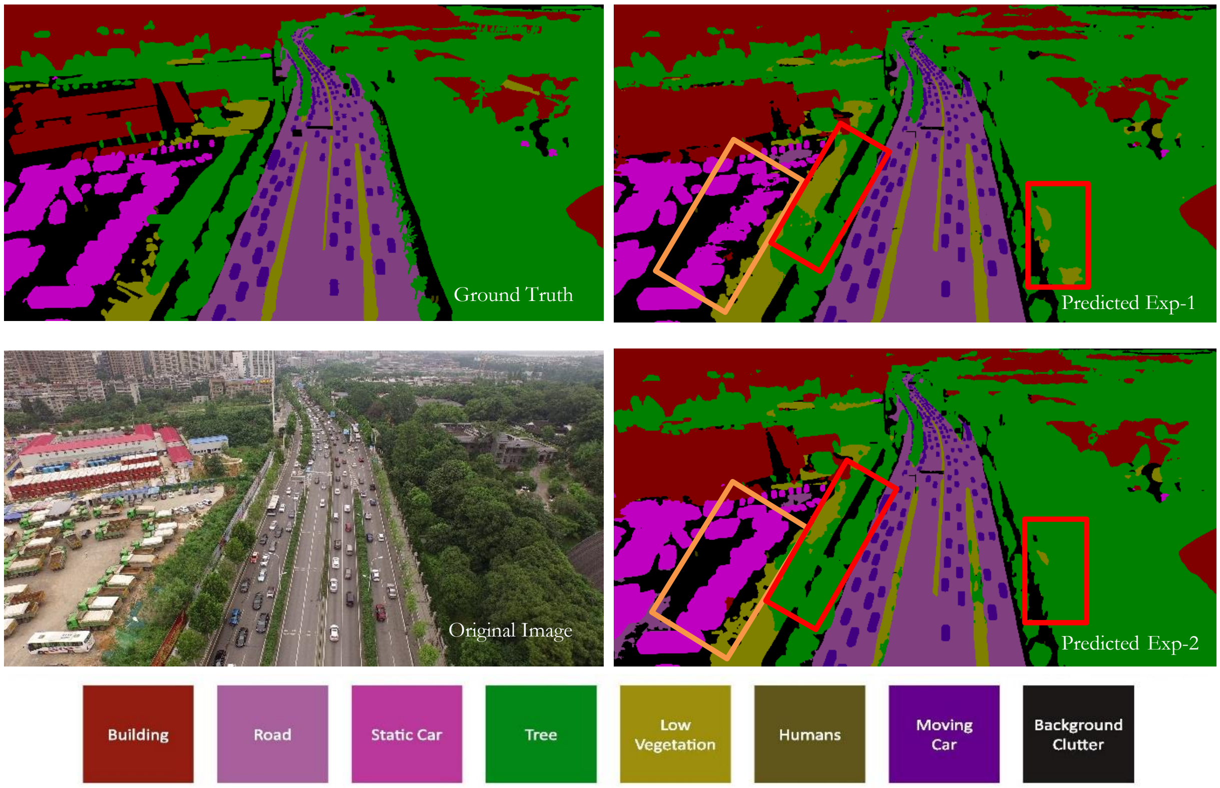


Figure 33: Semantic segmentation results with orange bounding box shows highest improvement in Static_car class and red box shows improvement in Tree class.

Figure 33 shows the ground truth, original image, and predicted labels from the experiment – 1 and experiment – 2 for image 000300 from seq28 in the test set. The orange bounding box compares the Static_car class in the experiment – 1 and experiment – 2; this class greatly improved in experiment – 2. The other classes, like Moving_car and Tree, also show improvement compared to experiment – 1, and all other classes are more or less the same. The red bounding box shows the area of the Tree and Vegetation class; here Tree class has improved compared to experiment – 1.

5.5. Experiment – 3 (Exp – 3)

In this experiment, we use another combination of frames, all original labels, and pseudo labels which are not adjacent or away to the original labels. So, from each sequence we took 28 pseudo labels and 10 original labels, which similar to experiment - 2. The model is trained on 760 images from all the sequences. Sometimes, the adjacent frames are more or less similar to each other when the time interval is small and unable to provide additional valuable information. So, we took pseudo labelled fames which are not adjacent to original labels, can bring more valuable additional information. From Figure 34, pseudo labelled frames at 0th, 40th, 60th and 100th position have different scene scenario which can bring different point of view and add more information.

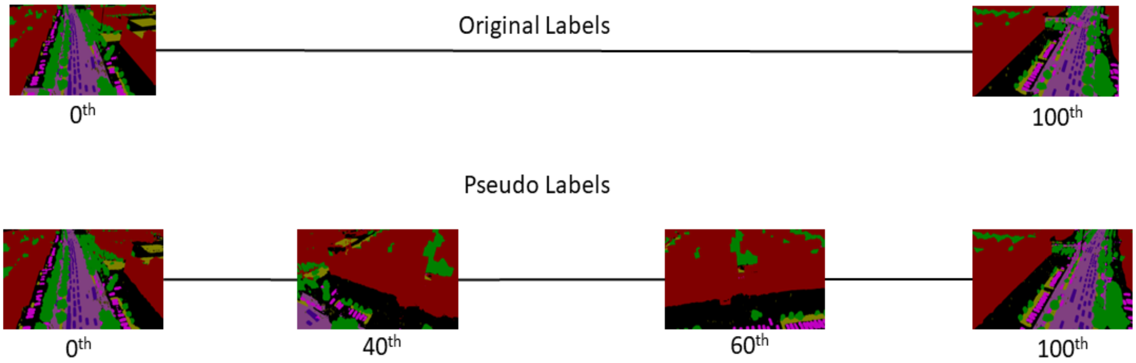


Figure 34: Selected combination of frames for Experiment - 3.

Table 6, contains the semantic segmentation results for different semantic classes in mIoU metrics. The green, red, and blue colour show the highest, lowest, and mean score values for semantic classes, respectively. All the semantic classes show improvement in accuracy except Static_car and Moving_car; the above classes lose their accuracy by a slight margin. There is a significant gain in mIoU from 76.28% to 76.51%.

Table 6: Accuracy of the model in mIoU metrices for semantic classes. Mean, maximum and minimum scores are in blue, green and red respectively

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
Experiment-3	76.51	69.28	88.57	81.54	75.97	79.73	62.87	-	77.58

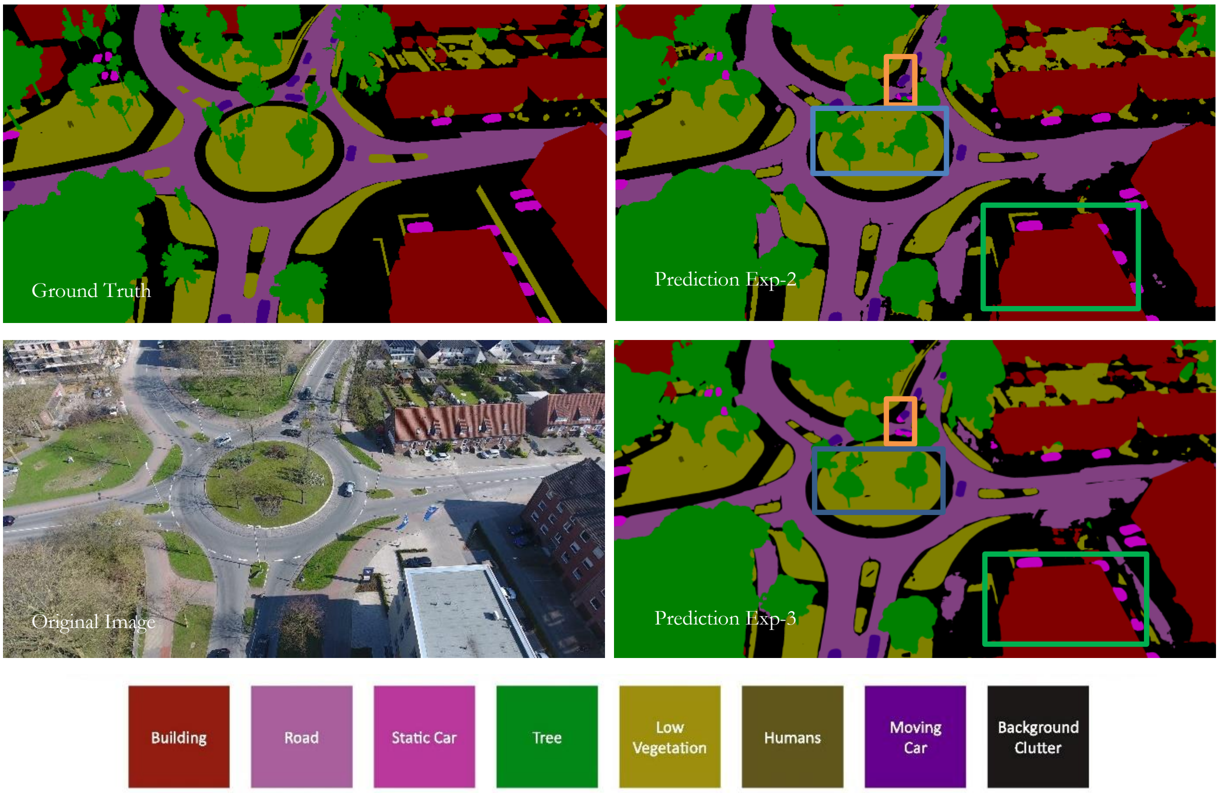


Figure 35: Results of Semantic Segmentation from Experiment – 3, bounding boxes in orange, blue and green colour shows prediction comparison of different semantic classes from experiment – 2 and experiment – 3.

Figure 35 shows the ground truth, original image, and predicted labels from experiment – 2 and experiment – 3 for image 000900 from seq30 in the test set. We show a prediction comparison between Exp – 2 and Exp – 3 with the help of bounding boxes. The blue box shows improvement in predicting Tree and Vegetation semantic class in Exp – 3 compared to Exp – 2. The green box shows that the Building class is better predicted with clear-cut boundaries in Exp – 3. The orange box shows slight depreciation in the prediction of the Moving_car and Static_car class in Exp – 3 compared to the previous.

6. DISCUSSION

In the discussion chapter, we elaborate on the semantic segmentation results for UAVid obtained from different experiments in Chapter 5. There is comparison between the results of three experiments conducted in this research with the previous best model results in section 6.1. We analyze each class's semantic segmentation results from different experiments in section 6.2.

6.1. Model Comparisons

This part presents the analysis of the semantic segmentation results on test set images from UAVid data for BiMSANet (Lyu et al., 2021) and our three separate experiments. Table 7 shows the IoU score values for each semantic class and the model's total mIoU scores. It is evident from the table 7 that Exp – 3 outperforms all other models in terms of the mIoU measure. In order to generalize each different class, the BiMSANet scores the highest for Clutter and Tree classes only, the Exp – 1 rank first for classes of Building and Road, the Exp - 2 scores maximum for classes of Static_car and Moving_car, and the Exp – 3 ranks first for Vegetation class only but received the highest mIoU for balanced prediction ability among all the different classes.

Table 7: It depicts the comparison of efficiency of different models in mIOU and for each class in IoU. Red is used to show 1st place, green is for the 2nd place, and blue represents the 3rd place.

Methods	mIoU	Clutter	Building	Road	Static_car	Tree	Vegetation	Human	Moving_car
BiMSANet	76.43	69.94	88.63	81.60	75.62	80.38	61.64	-	77.22
Experiment-1	76.10	69.33	89.13	81.86	72.81	79.48	62.59	-	77.56
Experiment-2	76.28	68.68	88.55	80.46	76.28	79.71	62.49	-	77.77
Experiment-3	76.51	69.28	88.57	81.54	75.97	79.73	62.87	-	77.58

Figure 36 depicts qualitative comparisons of predictions from different experiments with the BiMSANet model's prediction over the test set of UAVid. The predicted image is 000400 from seq30 in the test set. We used three bounding boxes of orange, white, and blue colour to show the example area from the image. The orange box region shows that the Exp-1 model struggles to provide coherent predictions for static or moving cars. However, all the other 3 models generates better predictions due to no noise in the BiMSANet model and less noise with good temporal precision in the Exp-2 and Exp-3 models, and Exp-2 outperforms all other models with the best IoU. The BiMSANet incorrectly categorized some parts of the road in yellow box as clutter. In contrast, the other two models forecasted that part of road with less error and Exp – 3 predict it with least error due to better scene comprehension. The area of parking, which is a part of the clutter class, is anticipated to be the road in the white box region by BiMSANet, Exp-2, and Exp-3, although Exp-1 explicitly forecasted this region as clutter; this is aided by the additional information offered by an extensive training set. All models incorrectly classify the clutter between the vegetation area as vegetation class in the blue box area, whereas Exp-3 makes minor mistakes and the BiMSANet makes the most.

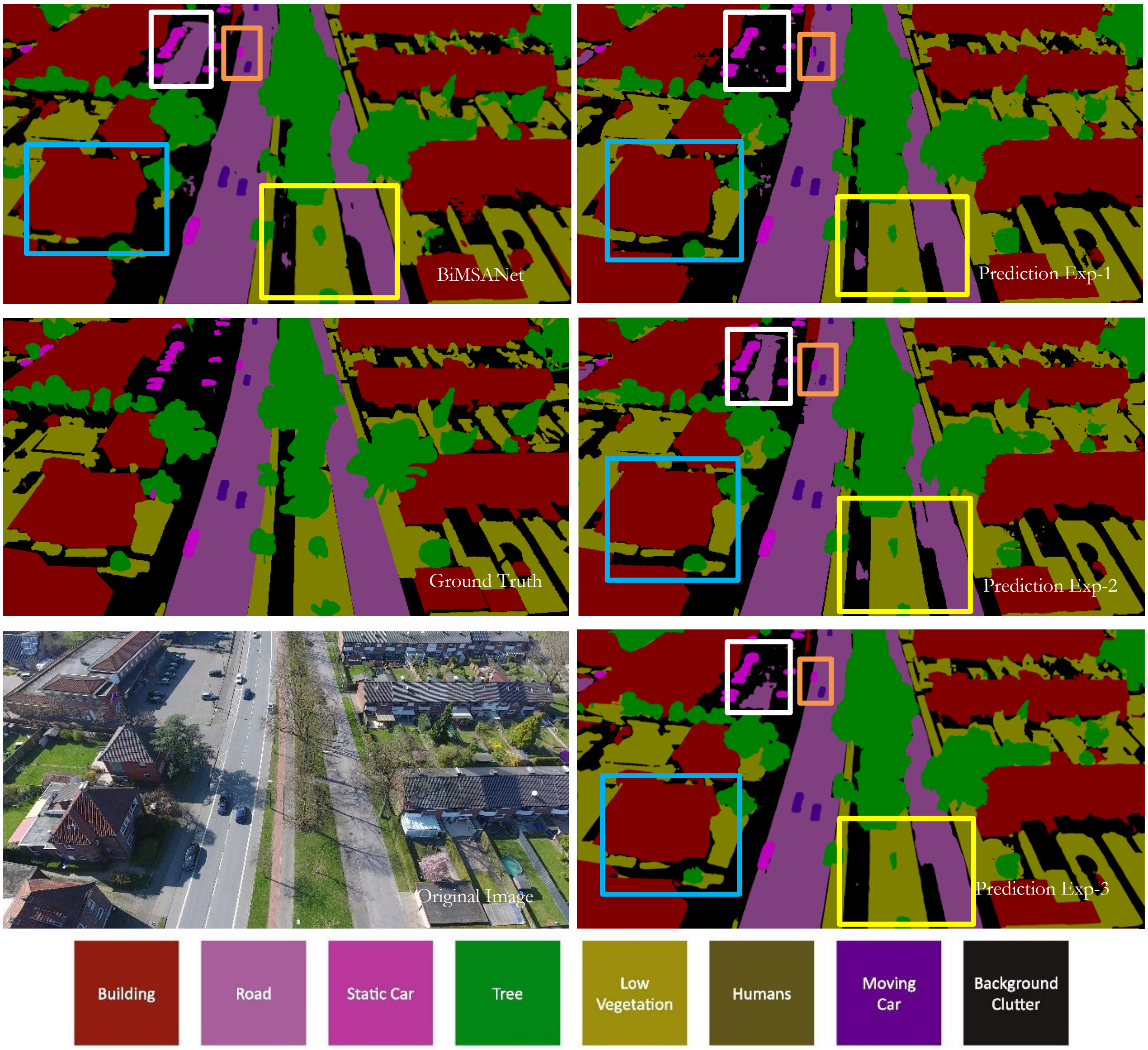


Figure 36: Qualitative comparisons of different scenarios with BiMSANet model on the UAVid test set.

6.2. Experiments Analysis

This section discusses how the model behaves or produces specific results in different experiments. All three experiments were performed under the same parameter for the model. All three experiments differ in selecting pseudo-label frames added with the original label to make large training sets. In all three experiments, we observed that selecting a particular combination of pseudo labels with original labels to enhance the training dataset significantly affects the segmentation outcomes.

- **Experiment – 1 :** Table 7 shows that Exp – 1 outperforms all the other three models in the classes of Building and Road. There is a gain of 0.5% and 0.26% in the Building and Road classes, respectively, compared to the second-best. Further, it effectively segments the classes of Vegetation, Clutter, and Moving_car but does not perform well with classes of Static_car and Tree and has the least mIoU. We added 920 pseudo labels with 200 original labels to train the model in this designed setting. When we use pseudo labels to augment the dataset, the model's performance depends on the quality of pseudo labels. We assessed the merits of pseudo labels in section 5.2; Building and Road classes have high IoU accuracy. Both the classes benefitted from additional valuable information that comes with pseudo labels, which helps to predict these classes better.

  Semantic classes of Vegetation and Moving_car were segmented well because the additional information from the pseudo labels was reliable, as these classes have good IoU accuracy in pseudo labels. Particularly for Moving_car, this class is benefitted from the better temporal resolution, which is the 1-sec interval between two frames. Static_car is the only class that drastically dropped its accuracy by 2.81%. The reason behind the poor performance for Static_car class is that the model wrongly classified static cars as moving cars due to the small interval between the two frames, which gives more dynamic information. This classification may be correct in real scenarios because mostly the static cars are not present in the middle of roads; they should be at the edge of the roads, as parked. Figure 37shows an orange box that compares the ground truth having static cars in the middle of roads and the predicted label from Exp-1, which classifies them as moving cars on behalf of better dynamic information.

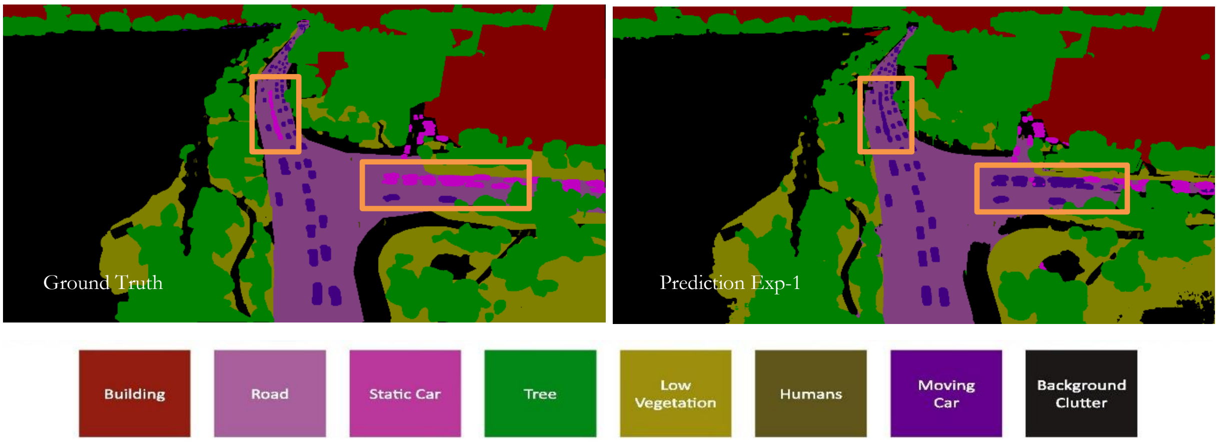


Figure 37: Wrong classification of Static_car as Moving_car, example image is 000100 from seq41 in test set.

- **Experiment – 2 :** Table 7 shows that Exp – 2 performs best for classes of Static_car and Moving_car compared to all other models. Exp -2 predictions for Static_car and Moving_car outperform the BiMSANet model's prediction by 0.66% and 0.55%, respectively. Overall performance in terms of mIoU is higher than Exp – 1. In this experiment, we only selected the pseudo labels adjacent to the original labels; this reduced the number of pseudo labels from 920 to 560. The idea is to reduce the noise that comes with pseudo labels, and selecting the pseudo labels adjacent to the original labels helps eliminate unwanted noise and maintain the temporal resolution, which is more important for understanding dynamic changes.

In this case, the model classified both the Static_car and Moving_car classes more accurately because the model learned better dynamic information from adjacent frames with short time intervals between two frames. Dynamic information plays an essential role in differentiating between static and moving cars. We believe that adjacent frames are more or less similar to each other. However, they provide better temporal information, which makes it easy to differentiate between moving and static cars by understanding the changes in their position with respect to time.

- **Experiment – 3 :** It is evident from Table 7 that Exp – 3 outperformed all the other three models in terms of mIoU. It has the highest accuracy and shows significant improvement from the previous best. This model only ranks first in the Vegetation class, but its balanced prediction to all the semantic classes makes it have the highest mIoU. The mIoU is increased from the previous best of 76.43% to the new best of 76.51%. In this experiment, we followed the same approach as in Exp – 2, reducing the number of pseudo labels from 920 to 560 but selecting the other combination of pseudo labels, which are far from the original labels or not adjacent frames. In this setting, we reduced the unwanted noise and slightly compromised the temporal resolution compared to Exp -2. As adjacent frames have similar information when the time interval is very short between two frames, they are unable to provide additional valuable information to all the semantic classes. The frames which are not adjacent to the original frames may have more diverse information because they are not similar to the original frames and have different views of the scene as in Figure 34.

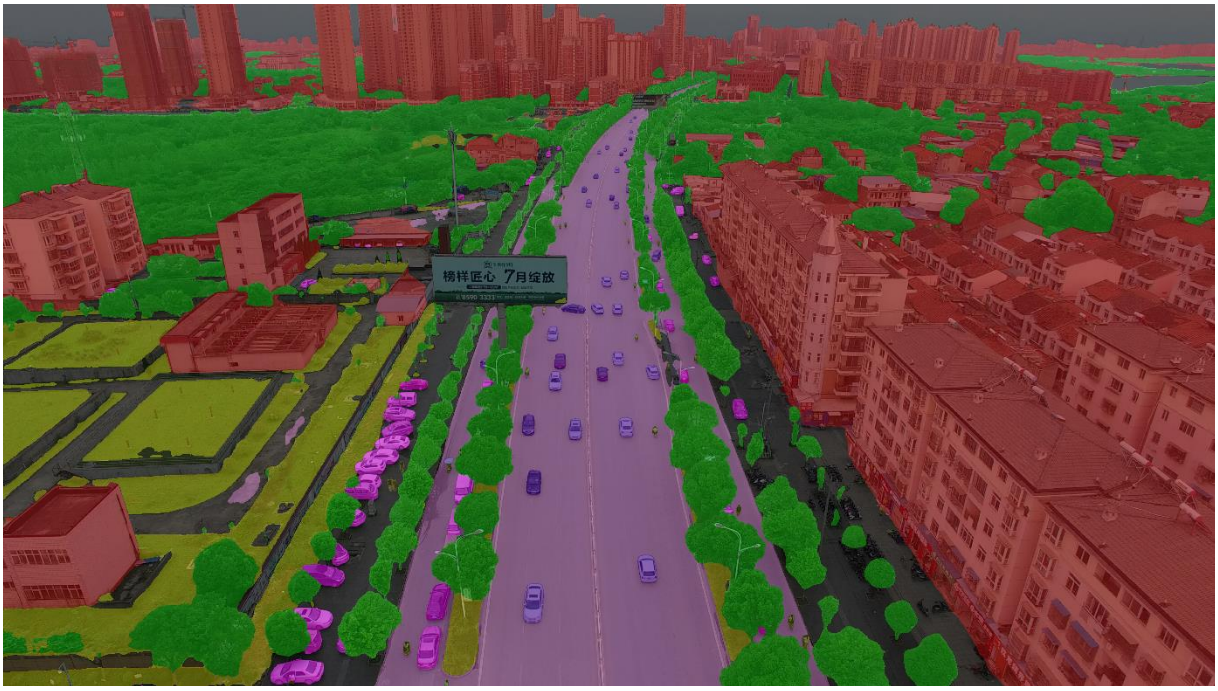


Figure 38: Blend of Original image and predicted label from Exp-3, image 000000 from seq21 in test set.

Figure 38 shows the best prediction result in the form of a blended image, made from the original image and predicted label for the image 000000 from seq21 in the test set. Most semantic classes overlap the corresponding classes in the original image accurately.

6.3. Limitations

Findings from the experiments reveals that designed approach failed to segments some classes properly or misclassifying.

- For instance, Static_car is classified as Moving_car in Exp – 1 as shown in figure 37 which leads to bad accuracy and inversely impacted the performance of the model.
- In Exp – 2, Clutter is misclassified as Road and there is error exaggeration from base model to Exp – 2 as shown in Figure 39.

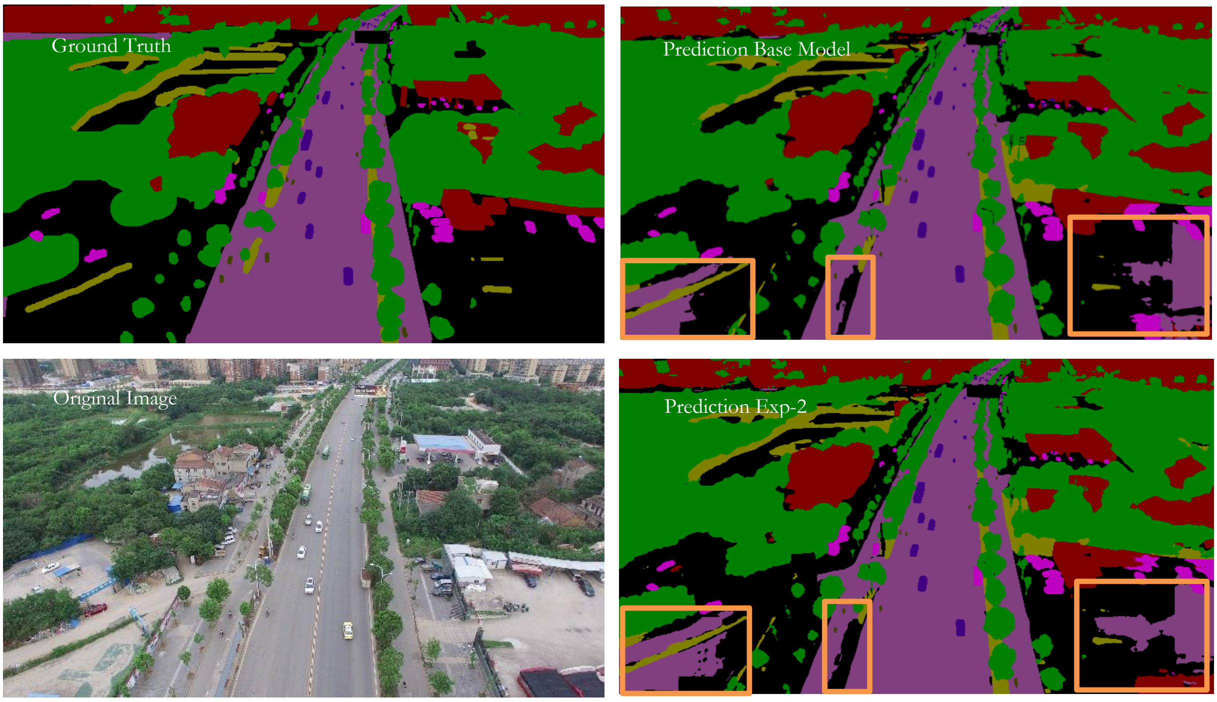


Figure 39: Misclassification between Clutter and Road, Orange bounding box shows misclassification is exaggerate from Base model to Exp-2. Image 000900 from seq21 in test set.

- Highly rely on the base model, as the quality of predicted pseudo labels depends on the generalization ability of the base model. If the base model fails to predict some classes properly, then the pseudo labels will have poor quality and affect the final model's performance. For example, the base model predicts the Human class with 43.73%, which is insufficient to add this class in pseudo labels as it will only add more noise than valuable information..

7. CONCLUSION & RECOMMENDATIONS

7.1. Conclusion

The goal of this research was to explore different combinations of frames extracted from UAVid videos with time intervals of 1 sec for semantically segmentation the UAV videos. In this context, we designed an algorithm to (i) extract frames from videos with 1-sec intervals, (ii) generate pseudo labels on newly extracted frames, (iii) quality assessment of pseudo labels and ignore pseudo labels for poor performing semantic class, (iv) perform different experiments with different combination of frames with pseudo labels and original labels, (v) accuracy assessment of the final semantic segmentation results on the test set. For semantic segmentation, we used BiMSANet as the basic model, which allows us to employ larger trunks for greater efficiency and better ability to fuse features from adjacent coarser and finer scales. We performed all of the experiments using the UAVid dataset with a wide spatial resolution range. Comparisons between all experiments and the BiMSANet model reveal that our Exp - 3 obtains better outcomes by balancing the efficiency of all semantic classes. Our Exp -3 outperforms the previous best by 0.08%.

From the experiments' findings, we observed that reducing the time interval between the frames of videos from 5 sec to 1 sec provides better temporal resolution and more information to better segment the dynamic objects; it also brings additional valuable information for other objects in the scene. For instance, our Exp – 2 shows the best results for Static_car and Moving_car classes; a better temporal resolution is needed to segment these classes efficiently. Building and Road classes achieved the best results in Exp - 1 due to additional information that comes with new frames. However, Exp – 3 shows the best results in the Vegetation class only, but due to its balanced segmentation efficiency to all the semantic classes, it outperforms all other experiments and BiMSANet. Moreover, selecting the combination of frames for video scene segmentation depends on various factors like dataset specification, video frame rate, and platform speed during data collection. Regarding UAVid videos, we have a video frame rate of 20 fps and a UAV speed of 10m/s during data collection. Therefore, the user can follow this approach in different applications where the segmentation of videos is the task. However, it is crucial to consider the factors that affect the frame selection.

In the previous work, BiMSANet used only the limited labelled data with the low temporal resolution to train the model for semantic segmentation. Our research proves that by reducing the time interval or improving the temporal resolution, we can have additional information about the scene and helps to predict the semantic classes better. We adopted a pseudo-labelling approach to deal with the limited labelled data challenge and fed the model with a large dataset to improve the model generalization ability. In our approach, three things play an essential role in improving the model efficiency: the selection of the base model or classifier, the quality of pseudo labels, and the proper combination of frames. Our research is an example of a simple and efficient way to achieve better results in the UAV video dataset. We make our research reproducible, as we used the UAVid dataset and BiMSANet as the base model, which is available publicly, and our algorithm is presented in section 4. We hope our work will help other research further to enhance the efficiency of semantic segmentation in UAV videos.

7.2. Research Questions & Answers

1.1) How does the density of labeled images affect the process of semantic segmentation?

The density of labeled images refers to the quantity of annotated or labeled images available for training the semantic segmentation model. A higher density of labeled images indicates a greater amount of training data is available. More annotated images allow the model to gain an additional understanding of numerous objects, backdrops, and settings. This can lead to better segmentation accuracy and generalization capabilities. We use pseudo labels to enhance the quantity of labelled dataset, which improves the model efficiency as mentioned in section 5.5.

1.2) How many frames do we need to extract high-level semantic information from video sequences?

The number of frames necessary to extract high-level semantic information from video sequences is determined by various factors, including the scene's complexity, the particular task or analysis being performed, and the desired level of accuracy. While no predetermined number of frames guarantees high-level semantic information extraction, here are some considerations to determine the selection of frames: Temporal Context, Frame Rate, Task Complexity, Event Duration, and Computational Constraints. In our research, Exp – 3 scores the best results for semantic segmentation of UAVid dataset, we select 200 labelled and additional 560 pseudo labelled frames to achieve this result as shown in section 5.5. It may be different for different applications, so number of frames required to extract high-level information is depends on above mentioned factors.

1.3) What will be the time interval between two labelled images?

The time interval between two labelled images in a video dataset is determined by various factors, such as the application or task's unique needs, the dynamics of the scene being taken, and the available resources. Although there is no generally applicable time interval, the interval between labelled images should be set to collect the relevant temporal information for the task at hand. A shorter time gap between labelled images is often necessary for applications requiring fine-grained analysis of motion or dynamic changes in the scene. A longer time interval, on the other hand, may be sufficient if the assignment concentrates on static or slower-changing objects like buildings or roads. In our work, we use 1 sec time interval between two labelled images, which helps to understand better the scene's dynamics, which results in better segmentation of moving and static cars. The model also generalizes well for other semantic classes due to a better understanding of the pattern and information when the gap between two labelled images is less, as mentioned in section 6.1.

2.1) How to check the quality of generate pseudo-labels using semi-supervised learning?

When utilizing semi-supervised learning to produce pseudo labels, it is critical to evaluate their quality to ensure they are dependable and can be used effectively for training. Consistency Check, Manual Inspection, Label Confidence Scores, Validation on Labeled Subset, and other approaches can be applied to assess the quality of created pseudo labels. In our research, we used manual inspection and validation on labelled subset approach and found that both are simple and effective way to assess the quality of pseudo labels. We did the quality assessment of pseudo labels in section 5.2.

2.2) How can pseudo-labels help to do semi-supervised semantic segmentation?

Pseudo-labels can help with semi-supervised semantic segmentation by using the information in both labelled and unlabelled data. Pseudo-labels enable the use of unlabelled data in the training phase for semi-supervised semantic segmentation. Mixing the restricted labelled data with the wider pool of pseudo-labelled samples allows the model to develop more robust representations and increase segmentation performance, effectively exploiting the additional information provided in the unlabeled data. We can easily see the improvements in the accuracy after adding pseudo labels from section 6.1.

2.3) How can the speed of UAVs during data collection affect the quality of generated pseudo labels?

The speed of the UAV while collecting data can affect the quality of the pseudo labels generated for semantic segmentation tasks. Higher UAV speeds can cause motion blur and object displacement, decreasing visual quality and producing pseudo-label discrepancies. Lower sample density and less frame overlap at higher speeds may decrease the availability of redundant data and fine-grained temporal information, lowering pseudo-label accuracy. Furthermore, labeling latency may rise when producing pseudo-labels after data collection. To address these difficulties, optimizing UAV flight parameters, balancing speed with image quality and temporal context, and employing post-processing techniques can all aid in improving the quality of pseudo labels. In UAVid data collection the speed of UAV was 10m/s, which is good to capture the scene without causing any motion blur or object displacement. But sometimes when Camera rotates with high speed may cause motion blur and effect the frame which results into poor pseudo label as depicts in Figure 40.

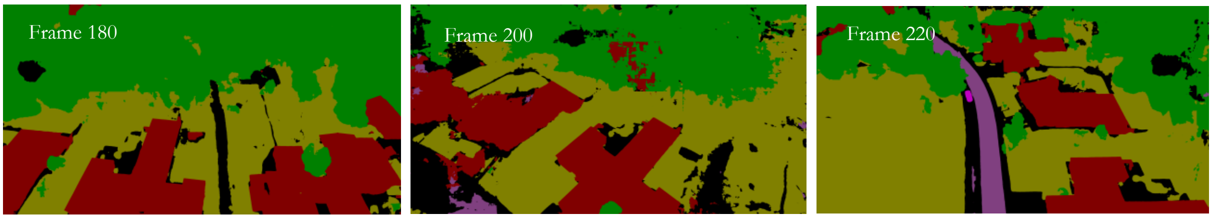


Figure 40: Motion blur due to camera rotation with high speed, frame 200 from seq13 has poor pseudo label due blur motion.

7.3. Recommendations

There are some recommendations based on this research for the future works:

- Class imbalance in the dataset, most of the image's pixels represent only some classes, like Building, Tree, and Clutter, and there is a minimal share of Human class which makes the model predict poor for this specific class. More focus is required to segment classes with a small share, ultimately enhancing the model's overall efficiency.
- Since the pseudo labelling techniques showed an improvement in segmenting the semantic classes in UAVid dataset, it depends on the combination of frames selected for training the model. In this research, the frames with time intervals of less than 5 sec are used in different combinations in different experiments, so reducing the interval below 1 sec in future work will not add more information.
- Iterative pseudo labelling will be helpful, as it tries to improve the model after each iteration.
- In this research self-training technique is used to perform the experiments as this technique is widely used and simple to implement. However, other semi-supervised techniques can also be explored to enhance the model's efficiency.

LIST OF REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274–2281. https://doi.org/10.1109/TPAMI.2012.120

Ahmed, F., Mohanta, J. C., Keshari, A., & Yadav, P. S. (2022). Recent Advances in Unmanned Aerial Vehicles: A Review. *Arabian Journal for Science and Engineering*, *47*(7), 7963–7984. https://doi.org/10.1007/s13369-022-06738-0

Arazo, E., Ortego, D., Albert, P., O'connor, N. E., & Mcguinness, K. (2020). *Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning*. https://git.io/fjQsC.

Bit, H. a. (n.d.). *Semantic Segmentation. What is Semantic Segmentation? | by Hack A BIT | hackabit | Medium*. Retrieved October 15, 2019, from https://medium.com/hackabit/semantic-segmentation-8f2900eff5c8

Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5302 LNCS*(PART 1), 44–57. https://doi.org/10.1007/978-3-540-88682-2_5/COVER

Caesar, H., Uijlings, J., & Ferrari, V. (2016). COCO-Stuff: Thing and Stuff Classes in Context. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1209–1218. https://doi.org/10.1109/CVPR.2018.00132

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/J.NEUCOM.2019.10.118

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). *Rethinking Atrous Convolution for Semantic Image Segmentation*. http://arxiv.org/abs/1706.05587

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. https://github.com/tensorflow/models/tree/master/

Chen, L. C., Lopes, R. G., Cheng, B., Collins, M. D., Cubuk, E. D., Zoph, B., Adam, H., & Shlens, J. (2020). Naive-Student: Leveraging Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12354 LNCS*, 695–714. https://doi.org/10.1007/978-3-030-58545-7_40

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., R&d, D. A., & Darmstadt, T. U. (2016). *The Cityscapes Dataset for Semantic Urban Scene Understanding*. www.cityscapes-dataset.net

Crommelinck, S., Bennett, R., Gerke, M., Yang, M. Y., Vosselman, G., Melgani, F., Nex, F., Gloaguen, R., & Thenkabail, P. S. (2017). *remote sensing Contour Detection for UAV-Based Cadastral Mapping*. https://doi.org/10.3390/rs9020171

Deng, J. (2009). *IEEE Xplore Full-Text PDF:* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5206848

Ding, S., Zhu, Z., & Zhang, X. (2017). An overview on semi-supervised support vector machine. *Neural Computing and Applications*, *28*(5), 969–978. https://doi.org/10.1007/S00521-015-2113-7/METRICS

Dong-Hyun Lee. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop: Challenges in Representation Learning, July 2013*, 1–6. https://www.kaggle.com/blobs/download/forum-message-attachment-files/746/pseudo_label_final.pdf

Enfuse. (2021). *Industry Focused Digital Service Provider - EnFuse Solutions*. https://www.enfuse-solutions.com/

Gadde, R., Jampani, V., & Gehler, P. V. (2017). *Semantic Video CNNs through Representation Warping*. http://arxiv.org/abs/1708.03088

Iscen, A., Tolias, G., Avrithis, Y., & Chum, O. (2019). *Label Propagation for Deep Semi-supervised Learning*.

Jeong, J., Lee, S., Kim, J., & Kwak, N. (2019). Consistency-based Semi-supervised Learning for Object detection. *Advances in Neural Information Processing Systems*, *32*. https://github.com/soo89/CSD-SSD

Jin, Y., Wang, J., & Lin, D. (2023). *Semi-Supervised Semantic Segmentation via Gentle Teaching Assistant. NeurIPS*, 1–15. http://arxiv.org/abs/2301.07340

Jordan, J. (2018). *Evaluating image segmentation models*. https://www.jeremyjordan.me/evaluating-image-segmentation-models/

Kim, B., Yim, J., & Kim, J. (2018). *Highway Driving Dataset for Semantic Video Segmentation*. https://sites.google.com/site/highwaydrivingdataset/

Lee, D.-H. (2013). *(PDF) Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*. https://www.researchgate.net/publication/280581078_Pseudo-Label_The_Simple_and_Efficient_Semi-Supervised_Learning_Method_for_Deep_Neural_Networks

Li, Y., & Gupta, A. (2018). *Beyond Grids: Learning Graph Representations for Visual Recognition*.

Liu, W., Rabinovich, A., & Berg, A. C. (2015). *PARSENET: LOOKING WIDER TO SEE BETTER*. https://github.com/weiliu89/caffe/tree/fcn

Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., & Stachniss, C. (2017). UAV-based crop and weed classification for smart farming. *Proceedings - IEEE International Conference on Robotics and Automation*, 3024–3031. https://doi.org/10.1109/ICRA.2017.7989347

Lyu, Y., Vosselman, G., Xia, G.-S., & Yang, M. Y. (2021). BIDIRECTIONAL MULTI-SCALE ATTENTION NETWORKS FOR SEMANTIC SEGMENTATION OF OBLIQUE UAV IMAGERY. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *V-2–2021*, 75–82. https://doi.org/10.5194/isprs-annals-V-2-2021-75-2021

Lyu, Y., Vosselman, G., Xia, G.-S., Yilmaz, A., & Yang, M. Y. (2020). UAVid: A semantic segmentation dataset for UAV imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *165*(May), 108–119. https://doi.org/10.1016/j.isprsjprs.2020.05.009

Lyu, Y., Vosselman, G., Xia, G. S., & Yang, M. Y. (2019). LIP: Learning instance propagation for video object segmentation. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, 2739–2748. https://doi.org/10.1109/ICCVW.2019.00335

Marsocci, V., & Scardapane, S. (2022). *Continual Barlow Twins: continual self-supervised learning for remote sensing semantic segmentation*. https://doi.org/10.1109/JSTARS.2023.3280029

Nitr, C. (2020). *MIoU Calculation. Computation of MIoU for Multiple-Class… | by CYBORG NITR | Medium*. https://medium.com/@cyborg.team.nitr/miou-calculation-4875f918f4cb

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., & Goodfellow, I. J. (2018). Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. *Advances in Neural Information Processing Systems*, *2018-December*, 3235–3246. https://arxiv.org/abs/1804.09170v4

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1912.01703v1

Pham, H., Dai, Z., Xie, Q., Luong, M.-T., & Le, Q. V. (2021). *Meta Pseudo Labels*.

Rashmi. (2020). *Using Python and OpenCV Extract Frames from a Video - Rashmi Erandika - Medium*. https://medium.com/@rashmierandika/using-python-and-opencv-extract-frames-from-a-video-c70ff8eba40d

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). *Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression*.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., & Jung, J. (2014). Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *93*, 256–271. https://doi.org/10.1016/J.ISPRSJPRS.2013.10.004

Shelhamer, E., Long, J., & Darrell, T. (2016). *Fully Convolutional Networks for Semantic Segmentation*. http://arxiv.org/abs/1605.06211

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/S40537-019-0197-0

Sohn, K., Berthelot, D., Li, C. L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., & Raffel, C. (2020). FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *Advances in Neural Information Processing Systems*, *2020-December*. https://arxiv.org/abs/2001.07685v2

Tao, A., Karan, N., Nvidia, S., & Catanzaro Nvidia, B. (2020). *HIERARCHICAL MULTI-SCALE ATTENTION FOR SEMANTIC SEGMENTATION*.

Triguero, I., García, S., & Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, *42*(2), 245–284. https://doi.org/10.1007/S10115-013-0706-Y/FIGURES/13 van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

Wang, Yidong, Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., Schiele, B., & Xie, X. (2022). *FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning*. https://arxiv.org/abs/2205.07246v3

Wang, Yuchao, Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., & Le, X. (2022). *Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels*. 4248–4257. http://arxiv.org/abs/2203.03884

Yang, L., Qi, L., Feng, L., Zhang, W., & Shi, Y. (2022). *Revisiting Weak-to-Strong Consistency in Semi-Supervised Semantic Segmentation*. http://arxiv.org/abs/2208.09910

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *1995-June*, 189–196. https://doi.org/10.3115/981658.981684

Zagoruyko, S., & Komodakis, N. (2016). *Wide Residual Networks*.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., & Shinozaki, T. (2021). FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. *Advances in Neural Information Processing Systems*, *22*, 18408–18419. https://arxiv.org/abs/2110.08263v3

Zhou, Z.-H., & Li, M. (2005). *Semi-Supervised Regression with Co-Training*.

Zhu, X. (2008). *Semi-Supervised Learning Literature Survey*.

Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. *Introduction to Semi-Supervised Learning*. https://doi.org/10.1007/978-3-031-01548-9