Article

A Moving Object Tracking Technique Using Few Frames with Feature Map Extraction and Feature Fusion

Abeer Abdulaziz AlArfaj and Hanan Ahmed Hosni Mahmoud *

Abstract: Moving object tracking techniques using machine and deep learning require large datasets for neural model training. New strategies need to be invented that utilize smaller data training sizes to realize the impact of large-sized datasets. However, current research does not balance the training data size and neural parameters, which creates the problem of inadequacy of the information provided by the low visual data content for parameter optimization. To enhance the performance of moving object tracking that appears in only a few frames, this research proposes a deep learning model using an abundant encoder-decoder (a high-resolution transformer (HRT) encoder-decoder). An HRT encoder-decoder employs feature map extraction that focuses on high resolution feature maps that are more representative of the moving object. In addition, we employ the proposed HRT encoder-decoder for feature map extraction and fusion to reimburse the few frames that have the visual information. Our extensive experiments on the Pascal DOC19 and MS-DS17 datasets have implied that the HRT encoder-decoder abundant model outperforms those of previous studies involving few frames that include moving objects.

Keywords: moving object tracking; few frames moving object tracking; feature map fusion

1. Introduction

Tracking of moving objects is an important area of research and a very important task in the field of security automation monitoring systems. The major research of deep learning models for moving object tracking includes region-based CNNs [1] and SSDs, which are single-shot models [2]. Moving object tracking has developed excellent research propositions, namely Overfat [2–4] and R-CNN with fast training [4–7]. These models need a great amount of visual data on a span of many frames for temporal feature maps to enable better performance. Labeling data instances can produce high computational costs as they can be hard to find. This problem has induced the challenges to deep learning, to some extent, and stalled the research of artificial intelligence, which has produced the need to employ a small training data size to track objects in dynamic environments.

Human brains recognize and track moving objects when only sighting a few instances, and we consider that deep neural models should also devise this capability. Assuming a sufficient abundant class of dynamic objects, a different class with few frames can be utilized to track a moving object. As depicted in Figure 1, the training phase is partitioned into two steps: abundant training and scarce training. In abundant training, adequate data training instances are employed to create a feature map space that can characterize the dynamic feature maps of moving objects. In scarce training, a tuning process of the deep learning model allows the new scarce class to be characterized in the feature map space. The authors of [3] considered a few-frames moving object tracking system. It amended the meta-tracking properties by defining the dynamic object vector. Two stages of meta-training process [5] and meta region-based CNN [5–7] models were proposed.
The authors of [8] proposed a meta-properties trainer and a property scoring process using the Googlenetv2 model [9] to allow the detection model to adjust to new classes. The feature map learner utilizes enough instances (in the training phase) from abundant labeled data instances to mine meta-feature generalization for future dynamic object classification. Accumulating scores transforms some ancillary instances from the scarce class into vectors that specify the global significance and the high correlation of the computed meta-feature map of the analogous tracked moving object. The process of incorporating the meta-feature map is learned by the feature-map trainer with the score vector computed by the scored procedure. The regression and the prediction information of the dynamic object can be computed. Nevertheless, these models are too solitary for feature map extraction, and many feature representations are unrestricted. Further, the scores are computed by a naive convolution function, which does not take advantage of the current dynamic object classes data.

The proposed HRT encoder–decoder abundant model can tackle this problem. The HRT encoder–decoder can extract all feature representations of a given surveillance video sequence and utilize the support set to achieve feature map fusion and track the dynamic object. The HRT encoder–decoder model extracts the meta feature map of the broad-spectrum dynamic object from the abundant tracking class and then allocates the learned generalized meta feature map to predict other dynamic objects. The proposed model is trained on a large number of labeled instances to mine and compute the selective feature map discriminating vectors that can be employed for some new classes. The proposed model is designed to generate generalization vectors. When utilizing a small number of instances from the new object class to tune the model parameters, the deep learning model can predict the dynamic object of the new scarce class, which is the same for the information extracted from the abundant class transferring to the new class.

The fundamental features of the abundant class represent the common features of both the abundant and new class. Therefore, the basic features of the moving objects have multiple resemblances. For instance, a horse has four tube-shaped legs, as do the new classes of cows and dogs, and so few-frames training resembles a transfer learning concept. For more perception, our model needs two-phase training comprising abundant training with a large dataset that does not have the new class, followed by scarce training which is tuned according to the new class.
In general, the major contributions of this research are discussed as follows:

- Our research presents an HRT encoder–decoder model that extracts abundant features and an encoder–decoder for feature map fusion to support few-frame moving object tracking.
- Our research proposes a deep learning model that extracts abundant feature maps and employs parallel temporal and attention procedures.

2. Related Works

2.1. Moving Object Tracking

Machine learning models for moving object tracking utilize surveillance video processing [6–12]. Moving object tracking utilizes moving object predictions abundant in moving object positions. The current models of moving object tracking comprise of one or many phases for tracking moving objects [13–17], which are defined using the surrounding borders’ geometry computations. The multi-phase model requires the computation of nominee borders with possible moving objects, and then predicts and validates the boundaries of the mined borders [18]. The former model executes a grid technique focused on the moving object to compute prediction parameters to identify the moving object without computing any of the borders. The multi-phase model has a higher performance [19–21]. The single phase model is lower compared to the multi-phase model in terms of precision, but it outperforms other models in speed, and it is characterized by a real-time paradigm. Nevertheless, these models have a shortcoming wherein they require more labelled data instances for model training.

2.2. Few-frames Moving Object Tracking

Few-frames learning is a state-of-the-art model that utilizes only a few frames for object tracking in both training and prediction classification [22–24]. The few-frames model includes the few-frames prediction for computing the moving object’s position in surveillance videos. It is used to identify the moving objects with little training data. Usually, we can define this technique as a scarce learning model [25].

The scarce learning model defines the moving object tracking attained from a few analogous data as the starting learning phase. It can be modified to new tracking situations using regression models. The authors of [26] presented a learning algorithm to enhance the performance of the classification of the moving object tracking with little data. The authors of [27] designated moving object information from the moving object province while performing adaptation to augment the cases of few-frames moving object tracking. The model of presenting a preprocessing learning algorithm is usually applied in the moving object tracking paradigm. The notion of a transfer learning technique highly decreases the model training epochs, but it can yield to an overfitting difficulty, which can be solved by the systematic model. Further, while transfer learning with scarce data can converge faster, the classifier still faces an unsatisfactory generalization setting.

Transfer learning allows the model to learn faster. Transfer learning utilizes the pre-learning of similar tasks, and transfer learning is utilized to produce hyper parameters from previous training to attain a worthy model initialization. The transfer learning model presented in [28] syndicates the transfer learning phase and the moving object classification, which can acquire knowledge and fast learning techniques, and train the classifier on how to utilize few data instances. The transfer learning CNN presented in [6] utilizes transfer learning on specific region features and enhances the moving object division through an R-CNN. The authors of [29] use the grouping of moving object tracking and few-frames tracking in an independent transfer learning to offer an approach for adjusting the classifier, and they proposed a higher accuracy tracking system and embraced an optimization technique for few data instances. The authors of [30] proposed a transfer learning initiator to a CNN. Once utilized, given a few surveillance video frames of new moving object, the transfer initiator can yield the classifier impact in the prediction stage of augmented learning of new instances in a feed loop manner. Compared with the Google net model, the
authors of [30] utilized the data of many classes, and they utilized the transfer features and adjusted them to obtain fast learning of other classes.

2.3. Attention Model

Encoder–decoder models (E-DM) are always utilized in natural language processors and utilize a transformer model [29–32]. E-DM models transform the input map M into the following matrices and vectors: support matrix V, key vector K, and value vector VA. E-DMs compute the dot product between V and K to yield the attention score of input instances. This procedure is known as the attention computation, and it is the principal process of encoder–decoder models. The score action of feature map M using the attention scores yields the feature map outputs with specific links between input words. Encoder–decoder models and their deviations have accomplished better results in NLP paradigms. An encoder–decoder model such as the BERT transformer model depicted in [33] trains transformers for unknown translations by mutually restricting the contexts.

Attention models such as the one depicted in [33] employ an encoder–decoder model for spatial data. It was proven that such an encoder–decoder model can enhance the performance of deep learning, especially in computer vision. The authors of [34] introduced a CNN architecture that utilizes an encoder–decoder pair for moving object tracking. In [35], the authors presented an encoder–decoder model to tackle computer vision and accomplished a high accuracy on the data of several surveillance video frame detection inputs. Due to the high accuracy of encoder–decoder models, many computer vision algorithms that depend on the encoder–decoder pair have been introduced. To utilize the encoder–decoder models for few-frames moving object tracking, we propose some alterations to the methodology of the typical encoder–decoder pair model.

3. The Proposed Approach

This research proposes a deep learning model using an abundant encoder–decoder (high resolution transformer (HRT) encoder–decoder). An HRT encoder–decoder employs a feature map extraction that focuses on high-resolution feature maps that are more representative of the moving object. In addition, our research employs the proposed HRT encoder–decoder for feature map extraction and fusion to reimburse the few frames that have the visual information. In the proposed model, we present an abundant class with plenty of data features, while a new scarce class is represented by a small amount of data. Our aim is to utilize the abundant and scarce classes to induce model learning that can predict moving objects in both classes. In Figure 2, the employed double training phase is depicted. The first phase utilizes the preceding information to define the surveillance video frame feature map learned from the abundant class (A). The second phase is the adjusting phase, where tuning of the training of the scarce class occurs to provide adaptation of the neural model to the new moving objects in the scarce class (S). Double data inputs are defined as the support data \(b\) and the support data \(v\). The abundant learning phase depicts the definition of the data of the support set to \(A_b\) and the support data set to \(A_v\), which is similarly defined in the new scarce class. If the number of classes in the new class is \(M\) and the count of frames in each class is \(f\), the problem is formulated as \(M\)-way \(f\)-frames moving object tracking.
The proposed HRT encoder–decoder model: a support phase and a support learning phase. The HRT encoder–decoder has three processes: support extraction module, auto encoder, and encoder–decoder.

### 3.1. The Deep CNN Model

The proposed deep CNN model mainly encompasses a feature map extraction process from the abundant representations, which is utilized to extract feature maps from the support set. Subsequently, the extracted feature representations are depicted into feature map flat vectors, and they are fed to the encoder–decoder input layer. In addition, these vectors are utilized in the auto encoders and decoders to perform feature map fusion. Flat vectors computed from the support and the support sets are utilized to acquire the fusion vectors.

To extract the abundant feature map and decrease the feature map loss of the surveillance video frames, our model employs an abundant feature map training model, as depicted in Figure 3 below.

![Feature map extraction model consisting of multiple parallel phases followed by a fusion model.](image)

**Figure 3.** Feature map extraction model consisting of multiple parallel phases followed by a fusion model.

### 3.2. The Proposed Support Model

In the proposed support model, the partitioning of the model into multiple parallel phases of self-attention encoders is followed by pooling. Each self-attention encoder is fed
by abundant class images and only the last two phases are fed by the scarce class images. Multiple feature sets are extracted from each phase. All feature sets are collected and utilized through a stride fusion model of the feature maps. The resolution from the pooling modules is of a lower resolution than the auto encoder stage. Each parallel phase extracts feature map vectors and then the fusion model performs multi-stride fusion. The feature maps of both the abundant and scarce resolutions are merged, and our model employs stride encoders for the abundant surveillance video frames and down-sampling for the pooling layers. The fusion technique utilizes pixel averaging, and the multiple channels of the resolution feature maps are adjusted to the same value. In addition, to enhance the receptive value, stride convolution computes the feature map vectors in the first phase. In order to improve the focus of the locus of the feature map, we utilize parallel temporal attention at the completion of each phase.

The PTAP process is depicted in Figure 3 and the algorithm is as shown in Algorithm 1. The process encompasses parallel temporal attention threads and is linked via a structural residual model. The attention usually consists of a pooling layer and two $3 \times 3$ convolution layers, and in between them, an ReLU activation is utilized. The temporal attention process has four convolutions. The attention PTAP centers the attention to decide which channels encompass the main features of the moving dynamic object. The temporal attention process emphasizes the temporal trajectory and detects which frame includes the main data of the moving dynamic object. The detail of the process is depicted as follows:

\begin{align*}
    \text{Map}_F &= I_C \psi(\text{Map}_{TA} + F_{CA}) \\
    \text{Map}_{SA} &= C_P(C_G(I_c)) \\
    \text{Map}_{CA} &= C_u(\mu(C_D(\text{Map}_P(I_c))))
\end{align*}

\(\text{Map}_F\) defines the feature map of the output and the input \((I_C)\). The activation functions \(\psi\) and \(\mu\) indicate Sigmoid and ReLu activation functions. \(C_P\), \(C_D\), and \(C_u\) define convolution functions, and \(\text{Map}_P\) indicates the maximum pooling average value. \(\text{Map}_{TA}\) represents the temporal attention output value and \(\text{Map}_{CA}\) depicts the channel attention output.

Our attention model employs a parallel design to learn surveillance video frame complex feature representations. The parallel temporal attention modules can compute the channel features correlation at each point and adjust them to improve the representative ability of feature representations mapping.

3.3. The Encoder–Decoder Model

The encoder–decoder model was introduced by Google [35–40]. An HRT encoder–decoder utilizes a self-attention method to compute the feature maps in a parallelized mechanism from the surveillance video input. To keep the input correlation, the model utilizes position-coding to compute the location coordinates. Consequently, the encoder–decoder model can guarantee the correlation between the previous and subsequent data; but, due to the parallel nature of the input, the model training period is decreased. The encoder–decoder model has a transformer structure encoder. When extracting the feature map, the parallel input is fed to the encoder for the correlation computation and other data feature maps are acquired and decoded.

3.3.1. The Encoder Structure

The key module of the encoder in the encoder–decoder model is the self-attention method. To compute the attention vector, the three inputs of the location, namely \(I_1, I_2, I_3\), are utilized as depicted in Algorithm 1.
**Algorithm 1: The Encoder Framework**

1. Compute the correlation in the input. The correlation is computed by the dot product, which is to compute the dot product for the vectors in $I_1$ and each vector in $I_2$. The specific formula is:

$$\text{Corr} = I_1 \cdot I_2^T$$  \hspace{1cm} (4)

2. The computed correlation is divided by the parameter $d$ to alleviate the gradient in the learning phase, as depicted in Equation (5):

$$\text{Corr} = \frac{\text{Corr}}{d}$$  \hspace{1cm} (5)

where $d$ defines the distribution parameter of the classifier softmax and denotes the model degeneration learning curve.

3. Change the normalized correlation vector into a value in the range of zero and one using the softmax classifier. The correlation is converted into a probability matrix $Z$ with values in the range of zero and one, as follows:

$$Z = \text{Attention}(I, J) = \text{Softmax}\left(\frac{II^T}{d}\right)$$  \hspace{1cm} (6)

4. Compute the value of the dot product of $Z$ and $K$.

$$X = Z \cdot K$$  \hspace{1cm} (7)

The purpose of accumulating a residual instance RES is to avoid the degradation in the deep neural model of the training model. Degradation implies that with the accumulation of the layers’ number in the deep neural model, the loss yields to attain saturation and the layer count increases.

Normalization can speed up the learning process and enhance the learning curve stability. Nevertheless, normalization has to solve the small size of the data. The normalization layer is linked to the input size, and if the input is small, it will face high interference. The mean and variance of the input will yield a misrepresentation of the data distribution. This can yield to usage of a large amount of memory, as well as an extended learning time. The learning phase may fail due to the static gradient path. In this case, we can utilize channel normalization, which splits the channel into sub-channels and computes inside the batch. The computation does not depend on the sub-channel size, and the performance can be more stabilized in bigger batches. Channel normalization can prevent batch normalization problems. For surveillance video frames with batch sizes of $M$, $G$, $H$, and $C$, channel normalization defines the channels into sub-channels and computes averages and standard deviations in each sub-channel, forcing each layer input to follow the range of zero to one distribution, which resolves the covariance problem of relocation and speeds up the model’s convergence. This is depicted as follows:

$$r = \frac{I - E[I]}{\sqrt{\text{SD}[I]} + \epsilon} \ast p1 + p2$$  \hspace{1cm} (8)

where $I$ is the input, $r$ is the normalized input, $E[I]$ is the expected value, $\text{SD}[I]$ is the standard deviation, $p1$ and $p2$ are the training parameters, and $\epsilon$ is the threshold that stops the denominator from reaching zero.

3.3.2. The Decoder Structure

In the encoder–decoder model the decoder structure transmits the support map to the support feature map. The support set vector and the support vector are fed as inputs to the encoder’s $I$ and $J$. Concurrently, we subdue the background external to the support moving object, and we use the label of the support vector as the training phase input with the computed support vector from the equation $B \otimes M$. Then, we compute the transformed
feature map using the attention equation of $Z_{B \rightarrow I}(B \otimes M)$. The computation is depicted as follows:

$$I_{\text{channel}} = \text{channel.Norm} \left( Z_{B \rightarrow I}(B \otimes M, J, I) + I \right)$$  \hspace{1cm} (9)

Compared with $I$, the feature-map-enhanced $I_{\text{channel}}$ groups various moving object maps from the support feature map $I$ to enhance its value.

The input of the combined feature maps form a feed forward network (FF) with ‘Avoid Connection’. Its importance lies in the process of the ReLU layer. The feature map vector is extracted by the attention process to a feature map adaptation, which enhances the model’s expressiveness. The FF model is a double/multi-layer perceptron model (D-MLP), which has a fully connected (FC) layer and an ReLU activation layer, which is employed to each location distinctly. The computation is as follows:

$$FF(O') = \text{ReLU} \left( O'C^1 + a^1 \right)C^2 + a^2$$  \hspace{1cm} (10)

Here, $O'$ is the output from the prior layer, and

where, $C^1 \in S^{D_w \times D_f}$,

$C^2 \in S^{D_f \times D_w}$,

$a^1 \in S^{D_f}$, and

$a^2 \in S^{D_w}$.

These are all the hyper-parameters of the training phase. The value of the parameter $D_f$ is higher than the value of $D_w$. After the transitory stage of the FF network, we utilize the Acc and the channel normalization processes.

4. Experiments

In this section, we will compare and test the HRT encoder–decoder model through a model simulation. In this article, an encoder–decoder model is utilized to identify few-frames moving object tracking. The experiments are depicted in the following subsections.

4.1. Datasets

We utilized public data for moving object tracking surveillance to train and test our model. There were two datasets: DOC19 and DS17. The data set description of the HRT encoder–decoder model is depicted in [12].

4.1.1. The Dataset DOC19

We utilized the DS17 and DOC19 datasets for model training using 12,000 video frames. The validation process used video frames from both datasets (5300). The training dataset chose abundant classes, while the prediction process used new instances. The abundant classes contained many labelled surveillance video frame data, and the new class had few surveillance video frames. For the N-class and M-frames surveillance tracking process, we defined the new class as N classes, and each single class had M video frames with tagged labels. In the beginning, we accomplished model training on the abundant classes to get a primary model score, and at the next stage, we performed a fine tuning of the model on the new class. In the new class, we accumulated the moving object in the abundant class so that the trained encoder–decoder model could identify both the new and the abundant classes. To avoid the non-generality of the model processing tracking, we split the dataset into three subsets to train and test the model. In each subset, for the 22-class, five classes were selected as the new classes, and the other classes were utilized as the abundant-class data. For each subset, we took 3, 7, and 9 for the K parameter of the new class for training and validation. When evaluating the datasets, we utilized the mean accuracy of the new class to test. When the join and difference ratio between the result and the true label was higher than 0.5, then the result was correct, that is, JD50.
4.1.2. The Dataset DS17

The DS17 dataset enclosed rich classes and a large number of video frames. It will be utilized for testing surveillance video frame object tracking. In the paradigm of the moving object tracking process, DS17 included 76 different classes with 10,000 video frames for training and 5000 video frames for validation. We selected 18 classes as the new class sets and the remaining classes were the abundant class set.

4.1.3. Training Process

The simulation environment of our experiment was a TX208 GPU with 64 GB of memory. It was executed using Python on Linux sun stations and utilized deep learning PyTorch to construct the encoder–decoder models. The model parameter gradient utilized stochastic descent computation with an energy of 0.8, and score tuning was defined as being of value equal to 0.0005, with batch defined as 32 in size. In addition, the training surveillance video frames were handled by a model of horizontal, vertical overturning, and color exposure to increase the training data size.

4.2. Comparison

4.2.1. Results on the Dataset DOC19

In this section, we describe the experimental results. Table 1 illustrates the results of the model when trained on the dataset DOC19 of the HRT encoder–decoder model on the new class. We also compare the results with current single-phase models, such as SPD [41], Meta Googlenet [42], and Det [43]. The proposed HRT encoder–decoder predictor performs at higher tracking results when the number of video frames of the new class is high. In the first subset, we developed 1.4% more than the others in five frames, 3.3% points more than the others in seven frames, and 1.3% over the others in eleven frames.

Table 1. Results using DOC19 where the first subset used five frames, seven frames, and eleven frames, and the JD50 result was obtained.

| Frames | New Subset 1 | New Subset 2 | New Subset 3 |
|--------|--------------|--------------|--------------|
|        | 3  | 5  | 11 | 3  | 5  | 10 | 3  | 5  | 10 |
| SPD    | 12.4| 29.1| 38.5| 5.0| 15.7| 31.0| 15.0| 27.3| 36.3|
| Meta Googlenet | 26.7| 33.9| 47.2| 22.7| 30.1| 40.5| 28.4| 42.8| 45.9|
| Det    | 28.9| 35.0| 48.8| 25.9| 30.6| 41.5| 27.9| 41.9| 42.9|
| Our HRT model | 29.4| 37.7| 49.9| 27.8| 36.8| 47.5| 28.9| 44.2| 49.2|

4.2.2. Results on the Dataset DS17

Compared with the DOC19 dataset, DS17 show more complications in moving object tracking processes due to the DS17 set having more video frames. We performed the training of the model with 60 abundant classes of DS17, and then we performed the fine-tuning process of the model when the frames were 13 or 23, separately. The results are depicted in Table 2. Our proposed model outperforms previous models. When the frame number was 13, our model enhanced the performance by 8.1% at JD45: 90, and for a frame number of 23, our model enhanced the performance by 9.3% at JD45: 90. The results are illustrated in Figures 4 and 5.
Table 2. Results of the proposed HRT encoder–decoder model using the DS17 data set with frame numbers of 13 and 23 compared to other models in few-frames moving object tracking.

| Frame Number | Model          | JD45: 90 | 0.5 | 0.75 |
|--------------|----------------|----------|-----|------|
| k = 13       | SPD            | 84       | 81  | 81   |
|              | Meta Googlenet | 82       | 88  | 86   |
|              | Det            | 83       | 86  | 81   |
|              | Our HRT encoder–decoder | 94 | 96  | 97   |
| k = 23       | SPD            | 87       | 86  | 81   |
|              | Meta Googlenet | 81       | 87  | 86   |
|              | Det            | 83       | 87  | 81   |
|              | Our HRT encoder–decoder | 95 | 96  | 98.4 |

Figure 4. Average accuracy for k = 13.

Figure 5. Average accuracy for k = 23.

4.3. Ablation Experiments

The ablation results are decisive in utilizing the encoder–decoder HRT for infusion and in using an abundant extraction model. The ablation training and testing were performed
on the DOC19 dataset, and the frame number was defined to be 7, with the abundant and the new class splitting of the dataset.

To test the temporal attention in the abundant CNN, we accumulated the ablation results of these processes. We performed the experiments in the CNN abundant model. As depicted in Table 3, the experimental results are enhanced when temporal attention was accumulated. When channel attention was accumulated in the model, the results were enhanced more. Consequently, we found that the accumulative receptive parameter of the model and a higher defined score of the feature map using the attention method is highly operative.

Table 3. The ablation results with the feature map extraction model accumulated with the channel and temporal attention.

| Temporal Attention | Channel Attention | Accumulated Temporal and Channel Attention | JD50  |
|-------------------|------------------|------------------------------------------|-------|
|                   | √                |                                          | 0.531 |
| √                 |                 |                                          | 0.542 |
| √                 | √                |                                          | 0.546 |
| √                 | √                |                                          | 0.558 |

Using the attention method in the encoder–decoder model shows an enhancement of the encoder–decoder in the computer vision model in [33]. Table 4 depicts the ablation results. The fusion of the abundant set feature map vector with the scarce vector shows a better performance. In addition, when a facade is utilized to substitute the preceding temporal location in the decoding phase of the encoder–decoder model, the moving object tracking ablation result of the model can also be enhanced.

Table 4. Ablation results of facade and attention fused with the encoder–decoder.

| Facade | Attention | JD50  |
|--------|-----------|-------|
|        |           | 0.6006|
| √      |           | 0.6220|
| √      | √         | 0.6642|
| √      | √         | 0.6462|

After the ablation simulation on the feature map computational model and encoder–decoder, the best arrangement of the models is found to perform the ablation computation on the abundant feature map model and encoder–decoder. Table 5 depicts the results after 200 epochs of training. We also confirmed the impact of the accumulated procedures with our training models. Through this accumulation, the scarce model results can be highly enhanced for moving object tracking.

Table 5. Ablation results of abundant feature map extraction model, encoder–decoder, and scarce model utilized in the proposed HRT model.

| Scarce Model | Abundant Model | Combined Abundant and Scarce Model | JD50  |
|--------------|----------------|-----------------------------------|-------|
|              |                |                                   | 0.8545|
| √            |                |                                   | 0.8202|
| √            | √              |                                   | 0.8548|
| √            |                |                                   | 0.8482|
| √            | √              |                                   | 0.8550|

5. Conclusions

In this research, we introduced an HRT encoder–decoder model to recognize few-frames moving object tracking. In the model, we utilize an abundant feature map extraction
model to extract model feature maps, as well as an attention encoder–decoder to infuse the support set feature maps and the support feature maps. An effective predictor is proposed by merging the abundant model and the encoder–decoder model to be applied to new scarce instances. The experimental results proved that our proposed HRT encoder–decoder model performs better than the preceding classifiers when the number of video frames is higher than three. We also confirmed the impact of the accumulated procedures with our training models. Through this accumulation, the scarce model results can be highly enhanced for moving object tracking.

Author Contributions: Conceptualization, Hanan Ahmed Hosni Mahmoud and Abeer Abdulaziz AlArfaj; methodology, Hanan Ahmed Hosni Mahmoud; software, Hanan Ahmed Hosni Mahmoud; validation, Hanan Ahmed Hosni Mahmoud and Abeer Abdulaziz AlArfaj formal analysis, Hanan Ahmed Hosni Mahmoud; investigation, Hanan Ahmed Hosni Mahmoud; resources, Hanan Ahmed Hosni Mahmoud; data curation, Hanan Ahmed Hosni Mahmoud; writing—original draft preparation, Hanan Ahmed Hosni Mahmoud; writing—review and editing, Hanan Ahmed Hosni Mahmoud; visualization, Hanan Ahmed Hosni Mahmoud; supervision, Hanan Ahmed Hosni Mahmoud; project administration, Hanan Ahmed Hosni Mahmoud; funding acquisition, Abeer Abdulaziz AlArfaj. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

Acknowledgments: We would like to thank the following for funding our project: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References
1. Engel, J.; Schops, T.; Cremers, D. Direct monocular non-alluded SLAM. Comput. Vis. 2019, 8, 834–849.
2. Murral, R.; Tardos, J. SCR: An open-source SLAM system for monocular stereo and RGB-D cameras. IEEE Trans. Robot. 2021, 33, 1255–1262. [CrossRef]
3. Bescos, A.; Facil, J.; Neira, J. DynaS: Tracking mapping and in painting in dynamic scenes. IEEE Robot. Auton. Lett. 2019, 3, 4076–4083. [CrossRef]
4. Ronzoni, D.; Olmi, R.; Fantuzzi, C. AGV global localization using indistinguishable artificial landmarks. In Proceedings of the IEEE Conference on Robotics, Cairo, Egypt, 31 May 2020; pp. 287–292.
5. Hafez, R.; David, J. SLAM2: A SLAM system for monocular stereo and RGB-D cameras. IEEE Trans. Robot. 2020, 33, 1255–1262. [CrossRef]
6.mime, J.; Bayoun, D. LSD: Large scale direct monocular model. Comput. Vis. 2020, 7, 83–89. [CrossRef]
7. Ahmed, M.; Cremers, D. Indirect deep learning odometer model. IEEE Trans. Pattern Anal. Mach. Intell. 2019, 4, 61–65. [CrossRef]
8. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometer. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 40, 611–625. [CrossRef]
9. Collins, R.; Zhou, X.; Teh, S.K. An open source tracking testbed and evaluation web site. In Proceedings of the IEEE Workshop Performance Evaluation Tracking Surveillance, Paris, France, 20 March 2020; pp. 17–24.
10. Xu, H.; Yang, M.; Wang, X.; Yang, Q. Magnetic sensing system design for intelligent vehicle guidance. IEEE/ASME Trans. Mechatron. 2020, 15, 652–656. [CrossRef]
11. Loevsky, I.; Shimshoni, I. Reliable and efficient landmark-abundant localization for mobile robots. Robot. Auton. Syst. 2021, 58, 520–528. [CrossRef]
12. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The VID dataset. J. Robot. Reason. 2021, 32, 123–127.
13. Fisher, R. The MOVSD4 surveillance ground-truth data sets. In Proceedings of the IEEE Workshop Performing Evaluation Tracking Surveillance, London, UK, 29 January 2019; pp. 12–17. [CrossRef]
14. Fuentes, J.; Ascencio, J.; Mancha, J. Visual simultaneous localization and mapping: A survey. Artif. Intell. Rev. 2019, 43, 55–81. [CrossRef]
15. Saputra, M.; Markham, A.; Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. ACM Comput. Surv. 2020, 51, 37. [CrossRef]
16. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y. Past present and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* 2016, 32, 1309–1332. [CrossRef]
17. Wang, Y.; Lin, M.; Ju, R. Visual SLAM and moving-object detection for a small-size humanoid robot. *Adv. Robot. Syst.* 2021, 7, 133–143. [CrossRef]
18. Kundu, A.; Krishna, K.; Sivaswamy, J. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In Proceedings of the IEEE/RSJ Conference on Robotics, Lafayette, LA, USA, 4–8 November 2019; pp. 436–441.
19. Li, S.; Lee, D. RGB-D SLAM in dynamic environments using static point weighting. *IEEE Robot. Automoses* 2020, 2, 223–230. [CrossRef]
20. Tan, W.; Liu, H.; Bao, H. Robust monocular SLAM in dynamic environments. In Proceedings of the IEEE Mixed Augmented Reality, Athens, Greece, 27 May 2019; pp. 209–218.
21. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End moving object tracking with Encoder-decoders. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An surveillance video frame is worth 16x16 words: Encoder-decoders for surveillance video frame recognition at scale. *arXiv* 2020, arXiv:2010.11929.
23. Sun, K.; Xiao, B.; Liu, D.; Wang, J.D. Deep abundant map learning for human pose estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
24. Park, J.; Woo, S.; Lee, J.-Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* 2018, arXiv:1807.06514.
25. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
26. Fischler, M.; Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 2021, 24, 381–395. [CrossRef]
27. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. *ACM Mix. Augment. Real.* 2017, 2, 225–234.
28. Alcantarilla, P.; Yebes, J.; Almazan, J.; Bergasa, L. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In Proceedings of the IEEE Conference on Robotics, Alexandria, Egypt, 29–31 October 2019; pp. 190–197.
29. Giordano, D.; Murabito, F.; Spampinato, C. Superpixel-abundandt video object segmentation using perceptual organization and location prior. *Comput. Pattern Recognit.* 2020, 6, 484–489.
30. Sun, Y.; Liu, M.; Meng, M. Improving RGB-D SLAM in dynamic environments: A motion removal approach. *Robot. Syst.* 2021, 8, 110–122. [CrossRef]
31. Yu, C.; Liu, Z.; Wei, Q. DS-SLAM: A semantic visual SLAM towards dynamic environments. In Proceedings of the IEEE/RSJ Conference on Robotics, Venice, Italy, 20 September 2018; pp. 1168–1174.
32. Cheng, Y.; Meng, M. Semantic mapping in dynamic environments. *Robotica* 2020, 38, 256–270. [CrossRef]
33. Hirschmuller, H. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 30, 328–341. [CrossRef] [PubMed]
34. Lowe, D. Distinctive image features from scale-invariant key points. *J. Comput.* 2004, 6, 91–110.
35. Bay, H.; Ess, A.; Gool, L.V. SURF: Speeded up robust features. *Proc. Conf. Computation. Vis.* 2008, 3, 346–359.
36. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Mach.* 2011, 32, 105–119. [CrossRef]
37. Achanta, R.; Sässbrunck, S. SLIC super pixels compared to state-of-the-art super pixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 4, 227–234.
38. Sturm, J.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the IEEE/RSJ Conference on Intelligent Robots, Beijing, China, 27 May 2017; pp. 573–580.
39. Kerl, C.; Sturm, J.; Cremers, D. Robust odometry estimation for RGB-D cameras. In Proceedings of the IEEE Conference on Robots, New Delhi, India, 14–18 October 2019; pp. 3748–3754.
40. Sun, Y.; Meng, M. Motion removal for reliable RGB-D SLAM in dynamic environments. *Robot. Auton. Syst.* 2018, 10, 115–128. [CrossRef]
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. SPD: Attention is all you need. In *Advances in Neural Information Processing Systems*; The MIT Press: Los Angeles, CA, USA, 2017; pp. 5998–6008.
42. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Meta Googlenet: Transfer learning of deep bidirectional encoder-decoders. *Trans. Pattern Anal. Mach. Intell.* 2020, 2, 22–34.
43. Wu, H.P.; Liu, Y.L.; Wang, J.W. Review of Det dataset in Deep Learning. *Comput. Mater. Contin.* 2020, 63, 1309–1321.