Robust Multiple Object Tracking Using ReID features and Graph Convolutional Networks

Christian Lusardi
cml8292@rit.edu
Robust Multiple Object Tracking Using ReID features and Graph Convolutional Networks

Christian Lusardi
ml8292@rit.edu

A Thesis Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Chester F. Carlson Center for Imaging Science.

College of Science
Rochester Institute of Technology

Approved: __________________________________________________________________________

Dr. Andreas Savakis, Advisor
Professor, Department of Computer Engineering

Approved: __________________________________________________________________________

Dr. Guoyu Lu, Committee Member
Assistant Professor, Chester F. Carlson Center for Imaging Science

Approved: __________________________________________________________________________

Dr. Anthony Vodacek, Committee Member
Professor, Chester F. Carlson Center for Imaging Science

Approved: May 11, 2021
Robust Multiple Object Tracking Using Re-Identification Features and Graph Convolutional Networks

by

Christian Lusardi

Submitted to the Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology
Abstract

Deep Learning allows for great advancements in computer vision research and development. An area that is garnering attention is single object tracking and multi-object tracking. Object tracking continues to progress vastly in terms of detection and building re-identification features, but more effort needs to be dedicated to data association. In this thesis, the goal is to use a graph neural network to combine the information from both the bounding box interaction as well as the appearance feature information in a single association chain. This work is designed to explore the usage of graph neural networks and their message passing abilities during tracking to come up with stronger data associations. This thesis combines all steps from detection through association using state of the art methods along with novel re-identification applications. The metrics used to determine success are Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), ID Switching (IDs), Mostly Tracked, and Mostly Lost. Within this work, the combination of multiple appearance feature vectors to create a stronger single feature vector is explored to improve accuracy. Different types of data augmentations such as random erase and random noise are explored and their results are examined for effectiveness during tracking. A unique application of triplet loss is also implemented to improve overall network performance as well. Throughout testing, baseline models have been improved upon and each successive improvement is added to the final model output. Each of the improvements results in the sacrifice of some performance metrics but the overall benefits outweigh the costs. The datasets used during this thesis are the UAVDT Benchmark and the MOT Challenge Dataset. These datasets cover aerial-based vehicle tracking and pedestrian tracking. The UAVDT Benchmark and MOT Challenge dataset feature crowded scenery as well as substantial object overlap. This thesis demonstrates the increased matching capabilities of a graph network when paired with a robust and accurate object detector as well as an improved set of appearance feature vectors.
Acknowledgement

I would like to start by expressing my gratitude to Dr. Andreas Savakis for supporting me and my goals, as well as allowing me to work in the Real-Time Vision and Image Processing Lab at RIT. Dr. Savakis has been instrumental in my development as a researcher and has taught me to have confidence in my work as well as have the passion to be creative and explore new ideas. Working with him has made me feel more professional and prepared for a future in the scientific field. I would also like to thank my fellow lab members, especially Abu Md Niamul Taufique, for their support and guidance as I immersed myself in advanced machine learning concepts. They have been incredibly patient and knowledgeable with all of my questions and concerns regarding the work that has been done for this thesis.

I would also like to thank my committee members, Dr. Guoyu Lu and Dr. Anthony Vodacek, for their support and expertise during the creation of my master’s thesis. I am grateful for the support and guidance of all of the faculty and staff within the Imaging Science as well as Computer Engineering departments for their support during the past 2 years. Their mentoring, whether it be academically or personally, has been instrumental in shaping me to become the individual that I am today. I would also like to thank the RIT Research Computing Cluster for their support and computing resources.

I would also like to thank my family and friends for their continuous encouragement through my many career and academic transitions. They have been the boost that I needed and I would not have been able to be do it without them.
# Contents

1 Introduction .................................................. 11
   1.1. MOT Network Varieties .................................. 11
   1.2. Contributions ........................................... 13

2 Background .................................................. 14
   2.1. Object Detectors ........................................ 14
       2.1.1. Faster RCNN ........................................ 14
       2.1.2. Deformable Part Models ............................ 16
       2.1.3. CenterNet ........................................... 17
   2.2. Single Object Tracking ................................. 19
   2.3. Motion prediction ....................................... 20
       2.3.1. The Role of Re-Identification .................... 21
   2.4. Multiple Object Tracking .............................. 22
   2.5. Graph Networks ......................................... 24
       2.5.1. Graph Convolution ................................ 26

3 Methodology - GTREID ...................................... 29
   3.1. Overview ................................................ 29
   3.2. Balanced Weights ...................................... 32
   3.3. Triplet Loss ............................................. 32
   3.4. Data Augmentation ...................................... 33
   3.5. Object Detection ....................................... 34
   3.6. Loss Functions ......................................... 34
       3.6.1. Heatmap Loss ....................................... 34
       3.6.2. Offset Loss ......................................... 35
### 3.6.3. Bounding Box Size

### 3.6.4. Total Detector Loss

### 3.6.5. Backbone Appearance Loss

### 3.6.6. Triplet Loss

### 3.6.7. Total Backbone Loss

### 3.6.8. Graph Network Loss

#### 4 Experiments

4.1. Datasets

- 4.1.1. UAVDT Benchmark
- 4.1.2. MOT Challenge Dataset
- 4.1.3. VeRi776 Dataset
- 4.1.4. Market1501 Dataset

4.2. Evaluation

4.3. Determining the Proper ReID Network

4.4. Triplet Loss

4.5. Data Augmentation Experiments

4.6. Number of Layers

- 4.6.1. Ablation Study

4.7. Training Implementation

#### 5 Results

5.1. MOT17 Results

5.2. UAVDT MOT Results

#### 6 Discussion

6.1. Speed

6.2. Additional Parameters

6.3. Bounding Box Localization

6.4. Aerial Tracking Performance

#### 7 Conclusion
List of Figures

2.1 The FairMOT model provides the backbone object detector and an appearance based ReID head that is utilized in GTREID. .............................................................. 22
2.2 GCNNMatch Architecture which is the basis for the graph association network in final GC-NMOT Model. ................................................................. 24
2.3 Image showing neighborhood edges between objects that are within \( d \) distance away from the initial track in question. Within this neighborhood, message passing will be conducted in order to classify the matches between this track and the detections in the next frame. . . . 26
3.1 LABNet [43] ReID network utilized within the proposed Appearance Feature Combination Network. Diagram details the training procedure of the network with Triplet Loss, BatchNorm, as well as Class-Balanced Loss. ......................................................... 29
3.2 Proposed graph Model for MOT, named GTREID. The appearance features consist of the combination of the FairMOT based ReID head as well as the LABNet feature vector. This combined feature is put in place of the CNN appearance feature within the GCNNMatch code below. The detections and bounding boxes are formed using the FairMOT/CenterNet predictions. ................................................................. 30
3.3 Sample of Random Erase used during training to strengthen object detector. ............... 33
4.1 Sample images from the UAVDT Benchmark. All images are taken from a drone over urban environments. Camera perspectives range from bird’s eye to head-on. Weather conditions range from sunny to foggy. Images are taken during day and nighttime. ................. 37
4.2 Sample images from the MOT Challenge dataset. Image perspectives range from surveillance camera to chest height. Dataset includes sequences in which the camera is moving as well as stationary. ................................................................. 38
4.3 Sample images from VeRi776 dataset detailing camera locations as well as sample images of vehicles from the dataset. Each image is taken from a slight aerial perspective from a surveillance camera height. Figure source: https://github.com/JDAI-CV/VeRidataset. Date of Access: February 20, 2021.

4.4 Sample images from Market1501 detailing the diversity and imaging angle of the dataset images. Figure source: [24]. Date of Access: February 20, 2021.

4.5 Detection comparison between baseline model trained without Random Erase and Random Noise (left) and baseline model trained with Random Erase and Random Noise (right). Top images show two pedestrians being blocked by the person in the red shirt, they are undetected in the baseline, but they are detected in the baseline trained with augmentation. In the bottom comparison, there is an individual who is heavily occluded by the person in the red shirt and the person in the black outfit. They are not picked up by the baseline but they are identified by the augmented baseline.

4.6 Detection comparison between baseline model trained without Random Erase and Random Noise (left) and baseline model trained with Random Erase and Random Noise (right). Both images show false positive detections on the left hand side of each image. The detections are reflections of pedestrians walking close to the windows.

4.7 Layout of the ReID Combination network that is implemented in GTREID. Features from LABNet and FairMOT are concatenated on to form a single 1024d ReID vector. This is then passed through an FC layer with dropout to form a single combine 1024d output vector that is used as the appearance feature vector.

6.1 Results from GTREID validation set in on sequence MOT17-13. Images are from frames 493, 497, 500, 507 when viewed from left to right.

6.2 Results from FairMOT validation set in on sequence MOT17-13. Images are from frames 493, 497, 500, 507 when viewed from left to right.

6.3 Sequence 1303, frames 135-137. Top Row: MDP, Middle Row: FairMOT, Bottom Row: GTREID Best Viewed In Color. Attributes: Side View, Daylight, Low Altitude

6.4 Sequence 1303, frames 26,27. Top Row: MDP, Middle Row: FairMOT, Bottom Row: GTREID Best Viewed In Color. Attributes: Side View, Daylight, Low Altitude

List of Tables

4.1 MOT dataset statistics for UAVDT and MOT17 datasets. ........................................... 39
4.2 Re-Identification dataset statistics. Datasets are used to train LABNet model. .................. 41
4.3 Comparing LABNet feature size versus MOT performance when inserted into FairMOT backbone. Results were tested on the MOT17 validation set. All networks have the same output feature dimension. Backbone model trained on GTX1080 with a batch size of 7. ... 43
4.4 ReID network comparison between LABNet and FairMOT ReID head. Checkmark indicates which network had a higher MOTA performance for each video sequence. Attributes for each sequence are listed in the right hand column. Sequences with two checkmarks indicate same MOTA score for both networks. Altitude attributes are as follows: low-altitude (10 - 30 meters), medium-altitude (30-70meters), and higher-altitude (greater than 70 meters). 44
4.5 Breakdown of which network performed better per attribute. Results correspond to tests performed in Table 4.4. ................................................................. 44
4.6 MOT performance comparison between baseline network and network trained with triplet loss on the MOT17 validation split. Models were trained on a P4 graphics card at a batch size of 4 and a ReID dimension of 512. ................................................................. 46
4.7 MOT performance comparison between baseline network and network trained with triplet loss on the UAVDT Benchmark training and testing set. Models were trained on a P4 graphics card at a batch size of 8 and a ReID dimension of 128. ................................. 46
4.8 Performance difference between the baseline model with triplet loss, and the baseline trained with Random Erase and Random Noise as well as triplet loss. Training and testing parameters were the same between both runs. Random Erase performed 40% of the time with a scale ranging from 0.02-0.25 percent of the bounding box and an aspect ratio of 0.2, 0.5. Random Noise was performed on 30% of images with a mean and standard deviation of 2. Test was conducted on the MOT17 train/validation set. Results tested on the RIT RC Cluster[36] on a P4 GPU with a batch size of 4.

4.9 Performance comparison of feature vector combination network. Both models trained and tested with the same parameters and output dimensions of 1x1,024. Both models have been trained and tested on the UAVDT train and test sequences.

4.10 min box area 200. Performance progression of model throughout each successive feature addition. All performance tests were trained and tested on the UAVDT Benchmark. Baseline Tests were performed with a batch size of 12, whereas the GTREID tests were performed at a batch size of 8. Triplet indicates the usage of the triplet loss during training and Augment indicates the use of Random Erase and Random Noise during training. Augmentation parameters are the same as those from Table 4.8. Values displayed in this table are for a minimum tracked area of 200 square pixels.

4.11 Performance progression of model throughout each successive feature addition. All performance tests were trained and tested on the UAVDT Benchmark. Baseline Tests were performed with a batch size of 12, whereas the GTREID tests were performed at a batch size of 8. Triplet indicates the usage of the triplet loss during training and Augment indicates the use of Random Erase and Random Noise during training. Augmentation parameters are the same as those from Table 4.8. Values displayed in this table are for a minimum tracked area of 100 square pixels.

5.1 MOT17 leaderboard for private detections as of March 16, 2021. Scores are ordered based on MOTA performance as seen in the MOTChallenge website.
5.2 FairMOT Baseline comparison results on MOT17 when using different training datasets. MOT17 entails only the MOT17 training set was used during training. 'Mix' details a training dataset that utilizes Caltech Pedestrian [8], CityPersons [54] (sub section of [5]), CUHK-SYSU [49], PRW [58], ETHZ [10], MOT17 [30], and MOT16 [30]. CrowdHuman+Mix includes 60 epochs of pretraining on the CrowdHuman dataset [40]. This pretraining takes place on the backbone dla-34 network. Results sourced from FairMOT github on March 16, 2021.

5.3 UAVDT Results from [9]. Scores are rank ordered based on the MOTA score reported in the paper. All scores are reported using the provided Faster R-CNN detections.

5.4 Results of independent MOT network testing. Results computed on Tesla v100 GPU [36].

6.1 Breakdown of network performance on UAVDT Benchmark per attribute based on MOTA score per sequence.
Chapter 1

Introduction

Multi-Object Tracking (MOT) is the process of continuously tracking multiple objects of interest without mixing the identities of each object as they coexist within a sequence. This topic has applications ranging from target tracking, to surveillance, thru autonomous navigation and beyond. The goal of MOT is to correctly match each object track from frame $t-1$ with a single detection in frame $t$. The model not only has to match objects that it has seen before, but also has to add new objects that it has not seen or remove objects that have left the frame. All of these tasks pose unique challenges for the network, as it needs to be able to utilize all of the data at its disposal to make proper assessments.

Multi-Object tracking has a number of challenges that make it a particularly interesting area of research. One of the main issues is dealing with objects that experience occlusion. When an object becomes occluded, it may or may not reappear shortly after occlusion. This poses a challenge to the object tracker as it must decide whether an object in question is a continuation of a previous track, or if it is a new object instance that needs to be treated as a unique object. This re-association necessitates the use of either a motion association or appearance feature for matching. Finding the proper combination of these resources is vital for MOT.

1.1. MOT Network Varieties

MOT tasks are typically accomplished using either online or offline approaches. The online approaches [55, 32, 22, 26, 45] are only able to access features from past frames or the current frame. Offline approaches are able to access all frames during association, they are not limited when making associations in the way that online approaches are. They have the ability to make use of any frames in order to solidify its decision regarding a track. Online approaches are much more challenging as they are limited in the amount of information that they can work with at any given time. These networks are typically are more relevant applications in the real world such as autonomous navigation and real-time surveillance.
Multi-object trackers tend to come in single stage and multi-stage variants. Single stage variants [55, 26, 60, 45, 25] will perform object detection, feature extraction, and bounding box regression all in a single end-to-end trainable model. These models tend to focus much more on speed with a single model able to handle multiple tasks at once through the use of shared weights. The drawback of methods such as this is that their accuracy suffers slightly given the more lightweight nature of their model. Single stages networks tend to sacrifice accuracy in exchange for speed.

Multi-stage systems [22, 32, 46] utilize multiple networks working together to build an MOT tracker. These models will implement different networks to handle detection, feature extraction, and associations. This has some benefits as well as some drawbacks. For example, multi-stage systems are typically more accurate because they are able to cherry pick the networks that they want to work with for each stage of the tracking process. The downside is that they are significantly slower due to the information passing between the different networks. Multi-stage networks also tend to struggle if their pretrained weights are not suited for the data at hand. They may be trained for general feature extraction which could be used for image classification or object detection but they are not designed with object re-identification in mind. For example, [32] has a feature extractor that is trained on ImageNet [7] but it could be applied on an aerial imaging dataset. ImageNet pretraining will provide more generalized features for multiple classes, but it may not be as useful as a single stage network where all aspects of the model are trained on the target dataset. Single stage networks are also capable of achieving near real-time speeds whereas two stage systems may struggle to break single digit frames per second.

Graph based networks have become more common in MOT settings over the past few years. They provide greater associative abilities over standard matching techniques. They are unique in their ability to learn the balance between physical information such as bounding box locations with the visual information coming from a feature extractor. This allows the network to know how features are coexisting, rather than treating each object as a separate entity. When graph networks are utilized in tracking [32, 22, 26, 39], they typically utilize appearance features as nodes and a relationship between the different nodes as the edges of the graph. At each frame, the network must solve a bipartite assignment problem to assign the new nodes to the correct IDs and thus continue the correct graph. With most object trackers, the association networks are strong, but they often struggle with poor re-identification networks [32, 26] which hurt their visual associations.

To handle motion prediction and bounding box location predictions, most MOT networks utilize either a Kalman Filter [23] or a variation of the Lucas Kanade Optical Flow [33] motion estimation. Both methods have shown success in motion prediction during tracking but the Kalman Filter is typically used more often due to its ease of computation and general accuracy. Optical Flow methods have begun to show their
strengths yet they are less prominent due to the issue that they typically need to be trained into a network in order to be effective.

1.2. Contributions

The proper usage of node features and edge connections are important to creating the most efficient and accurate model possible. For this thesis, a single graph that incorporates characteristics from both the bounding box and appearance feature vectors are used. The visual information comes from a combination of two dedicated feature extractors in order to create the most robust output feature possible.

The main contributions of this thesis are as follows:

- Novel usage of multiple Re-Identification networks whose feature vectors are combined to create the most robust appearance model possible within the graph association.

- Incorporation of a class-label triplet loss during training to aid in the discriminating capabilities the built-in ReID network.

- Addition of data augmentation techniques to make the detector training more robust to occlusion and illumination change.
Chapter 2

Background

2.1. Object Detectors

Arguably one of the most important aspects of any tracking model are the object detections at the beginning of the processing chain. The detections that come into the association network can drastically affect the score of the entire system. The best association networks in the world will suffer if they are given poor information to work with. If a detector is able to present more bounding boxes with higher confidence, this will allow the association network to more accurately match a known track with an incoming track hypothesis. A stronger detector will also reduce the number of false negatives which plague many MOT networks. Reducing the number of false negatives will give the model more information to accurately update known locations and features of each track, a benefit that will also reduce the number of ID switches. In the process, the higher rate of object proposals will likely lead to a slight increase in false positives. However, this increase in false positives ideally won’t overshadow the performance increases in the other two metrics. All together this leads to a higher Multi-Object Tracking Accuracy score as this metric is composed of the number of false positives, false negatives, and ID switches (see Equation 4.1). A strong object detector should be able to provide tightly cropped bounding boxes around an object and do so with a high degree of confidence. On top of its ability to detect occluded objects, the object detector should also be able to refute and reject any false detections of objects that are distractors. This will ensure that the rest of the MOT network will not be bombarded with useless object detections that could lead to poor tracking results.

2.1.1. Faster RCNN

In most MOT datasets, a set of object detections are typically provided for MOT networks that do not have their own built-in detector. The most commonly used network is the Faster R-CNN (FRCNN) [34] detector.
FRCNN is a real-time object detection network which utilizes the success of region-based object proposal networks such as R-CNN [13] and Fast R-CNN [12]. Faster R-CNN outperforms the previous iterations by taking advantage of Region Proposal Networks which share layers with the Convolutional Neural Networks (CNN) that form the backbone of the network. This usage of shared layers allows for a significant decrease in compute time per image at roughly 10ms per image, down from 0.2 seconds per image of old state of the art. Another speed contribution comes from the usage of anchor boxes. Anchor boxes are a series of region proposals that represent different scales and aspect ratios. There are 9 anchors placed at each step of a sliding window that traverses across the feature map of the last convolutional layer.

Anchor boxes have proven to be a much more efficient use of resources than previous methods which use multi-scaled images, or multiple filter sizes to create their region proposals. Anchors eliminate the need for resizing images during runtime, which saves resources. The anchor boxes in FRCNN are based on a constant sliding window size and a single output feature vector size. The anchors themselves are the components that are of different size.

The region proposals coming from the middle stage of the network provide a simple binary classification of ‘object’ or ‘not object’ in a particular area of the image. The region proposal network then builds a bounding box around the target that was labeled as an ’object’, but it does not classify what type of object is in the location.

The loss function for the RPN network trains both the bounding box regression as well as the binary classifier. The total loss function is as follows:

\[
L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)
\]  

(2.1)

where \(i\) is the index of an anchor and \(p_i\) is the predicted probability that the anchor \(i\) is an object. The ground truth binary label is \(p_i^*\) which can either be 1 for a positive anchor or 0 for a negative anchor. \(t_i\) is a vector which has the coordinates of the bounding box and \(t_i^*\) houses the ground truth coordinates for that bounding box. The classification loss at the beginning of Equation 2.1 is a log loss over the binary classes. In the second part of the equation, the loss function is the smooth L1 loss [12]. In the equation, the \(p_i^*\) indicates that this regression function is only applied to anchors that are considered positive. The \(\lambda\) parameter is a balancing parameter which is set to 10 in the released version of the Faster R-CNN code.

MOT datasets provide Faster R-CNN detections because they have been a staple object detector since the network’s inception. The detections from the network are both reliable and accurate with relatively high confidence. That being said, due to the fixed locations of the anchor boxes, there will always be a slight
inaccuracy in the bounding box location if it does not perfectly align with the object in the image. This is
not problematic for most use cases as the detections are close enough to where they are not noticeable by a
typical observer. MOT requires that the object locations are as accurate as possible to avoid issues gathering
re-identification features or measuring bounding box locations for motion prediction.

2.1.2. Deformable Part Models

The Deformable Part Models (DPM) [14] is another detector that is provided by the MOT Challenge
[30] as a source of bounding box coordinates. This model was unique in that it uses a set of structures to
represent objects. Each object is broken down into a series of parts, essentially smaller boxes of information
based within the original bounding box of the object. Each of the parts is then connected to close neighbors
in order to develop a structure.

The network operates by passing a root filter across an image to get a baseline response map of gradients.
The model then samples each of the parts at a slightly higher resolution in order to get a classification at
different scales. The filters are moved across the image at different locations and scales to create a feature
pyramid. The root filter location defines the bounding area for all of the remaining part filters. Within the
filter size, the higher resolution part filters are used to extract more exact details from the sampled location.

When it comes to classifying an object, an overall score is computed for each root location. The highest
scoring root locations in an image will define the object detections whereas the locations of the highest
scoring parts are what define the full object hypothesis. The scoring of a hypothesis is detailed in Equation
2.2

\[
score(p_0, ..., p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=0}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b
\]  

(2.2)

where the score of each filter (in each location) is subtracted by a deformation cost that is dependent on the
relative position of the part filter within the root filter. The deformation cost is a 4 dimensional vector that
describes how far the part is from the anchor location of the root.

\[
\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)
\]  

(2.3)

The locations of each of the part filters are learned during training but they are not explicitly given during
training. Six locations are initialized in a manner to cover the highest energy locations of the root filter. A
part itself is anchored either along the central vertical axis of the root filter or off axis. If the filter is off axis,
a symmetric filter is placed on the other side of the axis to even out the distribution. The final filters may move as the accuracy and hinge loss function determines better places for each filter to exist.

While this network is accurate for most detection tasks, it struggles in situations where objects occlude one another. In cases such as two pedestrians crossing paths, the non-maximal suppression of this network will incorrectly treat two highly confident proposals as a single object detection, resulting in a false negative for one of the pedestrians. This will cause a fragmentation in the tracking and could lead to an ID switch if the occluded object appears again after a few frames.

2.1.3. CenterNet

As an alternative to traditional bounding box based object detectors, the CenterNet [61] architecture is based on the detection of a single point at the center of the object’s bounding box. This architecture utilizes a learned heatmap which highlights the central location of each object in an image. The approach is similar to the region proposals of FRCNN, but instead, uses each pixel location as a region proposal. In addition, CenterNet does not categorize each pixel as ‘background or not’ like the region-based networks do. Its object proposal technique is solely based on having a single peak per object proposal, which eliminates the need for traditional non-maximal suppression. Instead, the network takes the element wise maximum Gaussian level of the heatmap in order to find the central point for each object. The output feature map of the network is also higher resolution which further eliminates the need for multiple region proposals for a single object, instead, everything is taken care of in a single stage.

For this thesis, the Deep Layer Aggregation (DLA) [50] backbone was used as the feature extractor as seen in [55]. The DLA network uses multiple feature map resolutions during processing to create a multi-layer dense prediction of each image. CenterNet then uses deformable convolution layers [62] instead of the normal skip connections between feature resolutions in order to increase performance.

From the feature extractor, the heatmap predictor is trained using a series of Gaussian keypoints placed where the ground truth objects are located. The network is then trained using a pixel wise logistic regression with focal loss.

\[
L_k = -\frac{1}{N} \sum_{xyc} \begin{cases} 
(1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\
(1 - Y_{xyc})^\beta (Y_{xyc})^\alpha \log(1 - Y_{xyc}) & \text{otherwise} 
\end{cases}
\]

where

\[
Y_{xyc} = \exp \left( -\frac{(x - \hat{p}_x)^2 + (y - \hat{p}_y)^2}{2\sigma_p^2} \right)
\]
and $\tilde{p}$ is the keypoint location divided by the output prediction factor. $\alpha$ and $\beta$ are the hyper-parameters of the focal loss while $N$ is the number of keypoints in the particular image.

Given that the output feature map is of a lower resolution than the original image, an offset had to be learned in order to relate the feature map back to the original image locations. The offset is trained with a simple L1 loss between the predicted point and the ground truth point

$$L_{off} = \frac{1}{N} \sum_p \left| \hat{O}_p - \left( \frac{p}{R} - \tilde{p} \right) \right|,$$

(2.5)

in which $R$ is the output stride of the network. This value is set by default to 4 as demonstrated in the paper.

The network also learns a bounding box size parameter which is able to build a suitably sized bounding box around a given heatmap peak location. The size prediction head is also trained using an L1 loss at the center location.

$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{pk} - s_k \right|$$

(2.6)

where the centerpoint $p_k$ is:

$$\left( \frac{x_1^k + x_2^k}{2}, \frac{y_1^k + y_2^k}{2} \right)$$

and the object size $s_k$ is

$$s_k = (x_2^k - x_1^k, y_2^k - y_1^k)$$

The output of all of these heads are then combined to build accurate and exact bounding box locations for each object.

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off}$$

(2.7)

One flaw of the CenterNet architecture is that there is a possibility that two or more objects could share exactly the same centerpoint location in an image. In this case, CenterNet would draw a single box around one of the objects rather than both. Although this is the case, it happens in less than 0.1 percent of the objects in the COCO dataset [27]. That being said, according to [61], region based networks such as Fast R-CNN miss roughly 2 percent of detections due to the inaccuracy of their designs. Given CenterNet’s ability to predict a potential object detection using a single point, it was chosen for the backbone of [55] and showed great success in their network. This reasoning stands for this thesis as one of the most important characteristics is gathering a robust feature using the most accurate bounding box locations as possible.
2.2. Single Object Tracking

Single Object Tracking (SOT) has a number of parallels to MOT and the fundamental concepts can be carried over to help assist an MOT model. SOT is a task in which a single bounding box is given to a model in the first frame of a sequence, and the task of the tracker is to follow that single object throughout the remaining frames. The primary task in SOT is to not lose track of the object or lock onto a different object, as this would be considered a failure. This task relies on a strong object detector as well as a strong association network that is able to distinguish between a target and any distractions.

Older networks performed their tracking by convolving an initial target appearance feature across each successive frame and measured the location of the peak response of the convolution [42]. This approach was okay for instances in which there were not many distractors that could confuse the tracker.

However, as [3] points out, distractors can become an issue if a neighboring object has nearly identical visual characteristics. This is why they began to focus more on keeping track of distractors during testing. They call their method *scene aware* tracking, in which they look back at the previous frame and find and potential objects that could lead to a tracking switch. Once they locate these objects, they propagate the location information to the current frame in order to reject the potential mismatch.

This idea of motion awareness helping to aid in the visual appearance matching carries over to MOT. Appearance features alone are not enough to accurately keep track of an object between frames. The motion of the object plays a crucial role in ensuring that distractors are rejected, and the correct track is continued.

In addition to utilizing location information, Taufique [44] utilizes appearance information for rediscovering lost tracks as well as eliminating potential matches with local distractors. During tracking, the template image is kept alongside a series of image samples from neighboring objects. At given intervals, the objects are re-sampled and added to a dictionary of object information. During inference, each of the potential dictionaries is averaged and compared to one another, along with their bounding box locations, to make a proper assignment. In times of heavy occlusion, just the known appearance features are used to compare detections because the camera and vehicle may have shifted enough to make object location irrelevant. Utilizing their modified version of LABNet, they are able to achieve state of the art performance on the aerial-based single object tracking UAVDT Benchmark.

The concept of continuously re-sampling and adapting is also implemented in the FCAT [31] tracker. This tracker was able to update the exemplar image during tracking to ensure that the template image remained relevant to the most recent appearance of the object being tracked. The tracker carries similar ideas to the SiamReID network in that it is based on a Siamese Tracker [2] with major improvements made to the feature extraction of the template image. The basic concepts of the single object tracking such as: feature updating,
motion information, and distractor awareness all carry over to MOT scenarios. In MOT, every object is surrounded by its own set of distractors and as such, it needs to utilize all of the information that it has available to it to maintain a proper track without switching.

2.3. Motion prediction

Bounding box movement is just as important as appearance features when it comes to an assignment problem. In situations such as those seen in Figure 4.1, many of the vehicles look quite similar on their own. The matching problem becomes significantly harder if the object goes under illumination change or occlusion. For this reason, it is not practical to just use appearance features as the matching method. This is where bounding box motion and motion prediction come into play. Bounding box estimations are also crucial in that they allow networks to have a reasonable understanding of where an object might be if it undergoes occlusion or if there is a momentary discontinuity in a track due to a false negative detection.

Many traditional MOT models utilize the Kalman Filter [23] as their motion prediction model. This model uses prior information gathered over time to estimate the position of an object in a future frame. It does so using the state matrix in Equation 2.8:

\[ x_k = F x_{k-1} + B u_{k-1} + w_{k-1} \]  

(2.8)

in which the state transition matrix \( F \) is multiplied with the previous state matrix from the previous update. \( B \) represents the motion updating matrix, and it is applied to the motion vector from the previous frame. Finally, \( w_{k-1} \) is the previous process noise vector that represents the errors that are inherent in any measurement system [17].

From there, the output of Equation 2.8 is placed into Equation 2.9 where it is multiplied with a measurement model to create a measurement vector.

\[ z_k = H x_k + v_k \]  

(2.9)

In the equation above, \( H \) is the measurement model and \( v_k \) represents the measurement noise. The Kalman filter has two stages, update and prediction. Within the prediction stage, \( x_k \) is estimated using gathered information from previous measurements. Each estimation is accompanied by an error calculation to account for any noise. Once a measurement is made, the difference between the actual and predicted values are made in the form of:
\[
\tilde{y}_k = z_k - H\hat{x}_k
\]

The Kalman Gain is an important aspect of the updating phase as well. The job of the gain is to dictate the balance of influence between the predicted measurement and actual measurements from the system. The Gain is as follows:

\[
K_k = \frac{P_k^- H^T}{R + HP_k^- H^T}
\]

where \(P_k^-\) is the previous error covariance for the last update. The gain is then used along with the state estimate to come up with the next predicted location. The updated state measurement is as follows:

\[
\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y}.
\]

Other methods such as [39] utilize a pyramidal Optical Flow [53] method for their motion prediction. This method learns an image velocity vector between two images. Pyramidal Optical Flow also computes an affine matrix to handle any deformation of an object’s bounding box between frames. It does this calculation for multiple image scales in order to calculate pixel motion more accurately. This works in a similar manner to the DLA feature extraction network that is used in the backbone of [55]. Having a pyramid of velocity and deformation descriptions is useful for having the most accurate predictions possible.

This method is becoming more popular as its efficacy and accuracy are realized. The application of Optical Flow methods were not tested for this thesis, but in future work it could be experimented with in order to determine whether it would improve tracking.

### 2.3.1. The Role of Re-Identification

Re-Identification plays a large role in the performance of MOT networks. In instances where the model does not use any appearance information [60, 4], the number of identity switches is often significantly higher as any overlaps in tracks can lead to the network losing track of which ID belonged to each object. In [55], the authors demonstrated how effective adding a simple ReID network can be to the reduction of ID switching. This model uses an incredibly lightweight ReID model to build their feature vectors. In other cases [32, 22], a much more powerful CNN feature extractor is used to build vectors. In these instances, the models are typically pretrained on a larger dataset such as ImageNet [7]. While the model is trained on a large dataset, this may lead to issues if the images in the MOT dataset are vastly different than those from the pretraining set.
Other models such as [25] use more advanced feature extractors that are trained in an end to end manner, much like [55]. However, this model utilizes features extracted at multiple scales to ensure the strongest identity feature possible. Models like these are much stronger than the simple single scale ID features like those in [55], yet they are still designed to work quickly, potentially leaving some performance out for the sake of speed.

For this purpose, models such as LABNet [43] can be used. A full-size ReID model can provide much more accurate and relevant feature vectors for the task at hand. Within LABNet, the use of graph connections and triplet loss [38] allow this model to become illumination and rotation invariant which is crucial in MOT tasks where subjects are free to move about without any predetermined paths.

2.4. Multiple Object Tracking

![Figure 2.1: The FairMOT model provides the backbone object detector and an appearance based ReID head that is utilized in GTREID.](image)

One of the more successful single stage MOT methods is the FairMOT network [55] which tackled the issue of accuracy by using a state-of-the-art object detector. They implemented an anchorless object detection network based on [61]. This model is built to detect objects using center point regression of the object as opposed to a series of region proposals that need to be classified into a "foreground" or "background" pixels. The lack of region proposals also eliminates the need to perform non-maximal suppression on any given boxes as each object is only given a single detection hypothesis. The CenterNet architecture has no manual thresholds within the network, everything is learned during training and this results in much stronger object locations and bounding boxes.

Another MOT architecture that is built upon CenterNet is the CenterTrack [60] method. This method
is an example of a network which tracks objects exclusively through the motion of the bounding boxes, without the use of any appearance models. Their network still achieves high MOTA scores because of the strength of the detection bounding boxes and optical flow based motion predictions, but it suffers from high ID switching because it has no access to the appearance information of the target. This means that when the object in question goes under occlusion or crosses paths with another object, it has a much higher likelihood of not being matched properly in the next frame. FairMOT built upon this idea by adding in the appearance features that allow it to perform much more successfully. Their appearance feature is built using an extra two layer convolutional network on top of the DLA-34 feature extractor. The appearance feature is located at the same point as the peak of the heatmap and as such, this model creates a single ReID feature that is located at the centerpoint of the object.

In order to train the ReID head of the network, the peak heatmap location is obtained and the class distribution is learned using the Cross Entropy Loss in Equation 3.7

\[ L_{identity} = - \sum_{i=1}^{N} \sum_{k=1}^{K} L^i(k) \log(p(k)). \]  

(2.10)

\( K \) is the number of classes within the dataset, and \( L^i(k) \) is the one-hot representation of the ground truth class labels. The ReID loss is combined with the CenterNet losses to create a final loss

\[ L_{total} = \frac{1}{2} \left( \frac{1}{\epsilon w_1} L_{detection} + \frac{1}{\epsilon w_2} L_{identity} + w_1 + w_2 \right) \]  

(2.11)

in which

\[ L_{detection} = L_k + \lambda_{size} L_{size} + \lambda_{offset} L_{offset} \]

and \( w_1 \) and \( w_2 \) are learnable parameters during training.

FairMOT demonstrates how appearance features can play a large role in the overall performance of an MOT network. A strong feature vector will allow a model to accurately distinguish between objects that are similar in appearance as well as physically close to one another. The appearance features are also crucial in handling situations in which an object momentarily leaves the frame and comes back in a different location. As was seen in [60], the lack of appearance features can cause a spike in Identity Switches which will hurt the overall score of the network.
2.5. Graph Networks

Graph networks are beginning to show prominence as a matching architecture. The ability to pass information between nodes in the graph via pre-defined edges has tremendous benefit over traditional methods that do not consider the association between points. Overall graph network structure can vary between different graph based MOT networks. While almost all networks use the appearance features of each object as the nodes of the graphs [32, 22, 26, 39], they tend to differ in regards to how they build the edges between nodes. [32] builds their edges using a concatenation of the bounding box overlaps and feature appearance to create what they call an interaction feature. This feature is then used in a cosine similarity function that feeds into a Sinkhorn normalization algorithm [41] before reaching a Hungarian Assignment algorithm [20] for tracking.

Other models such as [22] utilize two separate graph models in their approach. They use a separate model for the appearance features and one for the motion features. Their motion features are made up of the bounding box coordinates and the distance between the boxes makes up their edges. The same architecture is used in their appearance model except the feature vectors are the nodes and the feature distances are the edges. The output of each model is then combined before being fed into a Hungarian Algorithm [20] for final assignment.

The novelty of [32] comes from their use of the Sinkhorn algorithm during training and testing. The Sinkhorn algorithm helps to solve the problem of optimal transport between a set of known tracks and the new set of detections. It is not a hard assignment algorithm, meaning it does not make final assignments between each pairing, but it is commonly used to solve graph matching problems [37]. The algorithm works...
by normalizing the initial assignment cost matrix along the rows and columns to make all rows and all columns sum to one. This final matrix takes the form of a doubly stochastic matrix. This matrix is useful because it is fully differentiable, meaning it can be used during training and testing. This allows networks such as [32] to use it within their objective function that feeds into their binary cross-entropy loss.

Other models such as [26] utilize extra conditionals to determine whether a connection between nodes is made. In their layout, a spatio-temporal condition has to be met in order build a neighbor graph among the set of detections. The objects in question must appear in the same frame and be within a certain Euclidean distance of each other in order to have a connection formed. This concept, while strict and accurate in ideal cases, causes problems if the object detector does not detect all objects in the same frame. If there is a cluster of 5 people walking into a frame, but only the first 3 show up in frame $t$ and the other two do not appear until frame $t + 10$, this could result in the group of 5 to not get connected. This means that their information would not be passed amongst the nodes.

Graph convolutional networks such as [32] can be used to increase the association abilities of an already strong MOT network. As previously mentioned, GCNNMatch (Figure 2.2) network utilizes the appearance and physical features in a single feature vector that allows it to match objects to detections based on both their physical location between frames as well as their appearance features. Where this differs from FairMOT and [22] is that the similarity metrics are measured based on a learned balance of features as opposed to the two features used in traditional methods. The IOU distance, along with the appearance distance between all objects assists the model with proper association. For example, if there is an object that is close in appearance, but has almost no overlap between their respective bounding boxes, then this is likely not a good match, and thus will be given a lower similarity score. In traditional MOT networks, the appearance feature distances are measured first, then the bounding box IOU is measured. The areas where the matches line up determine the assignments. With a graph network, the balance between the IOU and the appearance feature vector is known after all information is passed between the tracks and detections.
2.5.1. Graph Convolution

![Diagram of Graph Convolution](image)

Figure 2.3: Image showing neighborhood edges between objects that are within $d$ distance away from the initial track in question. Within this neighborhood, message passing will be conducted in order to classify the matches between this track and the detections in the next frame.

The key that makes a graph neural network powerful is the ability to share information among different nodes. Graph neural networks are able to build neighborhood connections between nodes using a predefined criteria and make associations through these shared connections. This gives the neural network an idea of how an individual object differs from its neighbors. This is in contrast to most MOT networks which are unable to learn any matching parameters during training. When associations are made during inference, scores typically come in the form of similarities between objects. The detection that has the highest similarity is chosen to be the match for that track.

In Figure 2.3, the black vehicle in the center of the image is connected to all vehicles which are within a certain distance threshold around it. The appearance information and bounding box overlap are then passed between all objects to find which vehicle in frame $t + 1$ is the most similar. Once this is completed, the linear assignment can take place and a match is found. This is repeated for each object in the tracking image to find their matches. If there are any objects that are beyond a similarity threshold, they will be marked as either a lost track or they will be added as a new track.

For this thesis, the graph association network used is based on GCNNMatch. This model builds edge connections between nodes based on the Euclidean distance between a node and its neighbors. In PyTorch Geometric [11], the layer that is responsible for passing information is called a GCNConv layer and is defined as

$$H^{(l+1)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right), \quad (2.12)$$
where $\tilde{A} = A + I_N$ is the adjacency matrix for the graph $G$ and $I_N$ is the identity matrix. In Equation 2.12, $\tilde{D}$ is the degree matrix of $\tilde{A}$ while $W^l$ is trainable weight matrix. In this equation, node features are summed through all possible neighboring edges that are within a predefined Euclidean Distance. The final output matrix $H^{(l+1)}$ houses the activations from layer $l$.

Equation 2.12 features some key components which allow it to be used in machine learning applications. The addition of the identity matrix is crucial because it creates the self loops that are necessary for the node to pass information to itself. If $I_N$ was not there, then the network would not consider each node to be a neighbor to itself and therefore would not match tracks and detections properly. The other important addition is the use of the inverse degree matrices. These act as a normalization parameter for the adjacency matrix. For instance, in Figure 2.3, the black car is connected to all 6 detections in frame $t + 1$, yet the silver car in the top right corner might only be connected to its closest 3 vehicles because the remainder are too far away. This would mean that the product of $\tilde{A}H^{(l)}W^{(l)}$ would result in a stronger weight for the vehicle with more connections. The degree matrices re-scale this product to ensure that this does not happen.

Within [11], activation layers can be stacked together to make a larger message passing network to propagate information through nodes that share connections with one another [19]. An example of a multi-layer classification network is further explained in the literature as

$$Z = f(X, A) = \text{softmax} \left( \tilde{A} \text{ReLU} \left( \tilde{A}XW^{(0)} \right)W^{(1)} \right),$$

(2.13)

where $\tilde{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ as described earlier. Within this equation, $W^{(0)}$ is the initial weight matrix that goes from the input to a hidden state for the hidden layer. $W^{(1)}$ is the weight matrix that goes from the hidden state to the output state. The softmax function that is described is the standard softmax used in neural network classification as

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)};$$

This softmax function is applied row-wise to the output matrix to create a classification output for a set of graph data. This data is then passed through a Cross-Entropy Loss to find the error. The final output and classification work in the same way that a normal classification network would and as such, graph data can be used for classification.

For this reason, graph neural networks can be used to perform matching tasks in MOT networks. The final assignment of a detection to a known track is analogous to assigning a label to an image or a bounding box for a classifier. GCNNMatch performs their final activation in a slightly different manner than the original
literature. Instead of using a softmax classifier, they use the Sinkhorn Algorithm to create soft assignments for each track and detection. The output similarity scores are effectively the same as the probabilities that result from the softmax function. Their output is scaled between a 1 for a perfect match and a 0 for a perfect rejection. For this reason, they utilize a weighted binary cross entropy loss in order to push the model to predict either a 0 or a 1, making the hard assignment more concrete during testing.

In the case of [43], the graph network is used to aggregate features from different locations within the image to limit the effects of rotation and scaling on the classification. Their network builds a series of graphs that aggregate patches of an image. This is done to reduce the memory consumption and computation issues that are inherent in graph networks. Their model also does away with the weight matrix that is described in the equations above because it caused overfitting.

When considering building a graph neural network for association. One of the major aspects to consider is the amount of memory that is typically consumed by a GCN during training and testing. Given the graph connections and message passing between nodes, a tremendous amount of memory is used during this process. The memory consumption also swells and recedes during operation. This is because different images have a different number of detections present in them which results in a varying number of connections between objects. This, along with the number of objects that are being stored in memory can easily exceed the hardware limits of a graphics card. As mentioned earlier, this is why [43] uses an aggregation of local information when training and testing, as opposed to building a fully connected network.
Chapter 3

Methodology - GTREID

3.1. Overview

As mentioned, GTREID incorporates the usage of a graph convolutional neural network for strengthened data association between object tracks and new object detections. The graph architecture is based on [32]. This model uses a Sinkhorn Algorithm [41] during training and testing in order to make initial soft assignments during training and testing. This model also combines the appearance and location features into one network for more efficient usage of all available information during inference. The GCNNMatch code does suffer from poor appearance features as a pretrained DenseNet-121 [15] network is used to draw feature vectors from the bounding boxes. This model is not trained for aerial tracking applications and as such suffers greatly from Identity Switches and False Negatives. The network also lacks a dedicated object detector, instead relying on detections from a provided source.

![Diagram of LABNet ReID network](image)

Figure 3.1: LABNet [43] ReID network utilized within the proposed Appearance Feature Combination Network. Diagram details the training procedure of the network with Triplet Loss, BatchNorm, as well as Class-Balanced Loss.

For this work, a Re-Identification (ReID) model based off Taufique’s [43] LABNet ReID network (Figure 3.1) is used to assist in the creation of robust appearance feature vectors. LABNet utilizes a class-balanced
loss [6] during training to ensure that the underrepresented objects in the dataset are not unfairly discounted in comparison to the more prevalent identities. It also implements a triplet loss when training to increase the differentiating abilities of the network. Triplet loss works by sampling positive and negative examples with each iteration. These are presented during training confuse the final assignment labels. It works by showing the network that not every example that is given during training is going to be a positive one, instead there may be some confusers or distractors. LABNet features a message passing GCN network as well which evenly splits each image into a 20x20 grid and passes aggregated image information amongst all of the regions to ensure that the vehicle can be properly re-identified even if it goes under a large rotation or scale change since the last time that it was imaged.

Figure 3.2: Proposed graph Model for MOT, named GTREID. The appearance features consist of the combination of the FairMOT based ReID head as well as the LABNet feature vector. This combined feature is put in place of the CNN appearance feature within the GCNNMatch code below. The detections and bounding boxes are formed using the FairMOT/CenterNet predictions.

GTREID combines the FairMOT, LABNet, and GCNNMatch into a single network as shown in Figure 3.2. The object detections from FairMOT’s CenterNet architecture can provide accurate object locations for use by the other networks. In order to strengthen the detection quality of the CenterNet architecture, an additional level of data augmentation is implemented. This includes random erasing and random noise.
These augmentations help the detector become more robust to occlusions as well as camera noise shifts that can be present when the illumination conditions change.

Re-Identification is handled by a combination of the built in feature vector from the FairMOT backbone and the LABNet feature extractor. This combination network will provide accurate detections for a more diverse range of scenarios where a single model might struggle. The LABNet model was chosen because of its state-of-the-art performance on the VeRi776 dataset and the strong accommodation of rotation and illumination changes. These scenarios are typical in the UAVDT Benchmark where vehicles commonly turn at intersections, causing their appearance to rapidly change with respect to the camera. The benchmark also has sequences which are imaged at night or vehicles go under the shadows of a building, attributes which the LABNet features are designed to handle. For GTREID, LABNet was pretrained on dedicated ReID datasets and further finetuned on the MOT datasets that are outlined in the chapter below.

For use in GTREID, the architecture of LABNet was shifted slightly to make the appearance features more generalized. During MOT challenges, it is not uncommon for the initial ReID feature to be pulled from an occluded position. This means that the first feature that is pulled is an inaccurate representation of what the object truly looks like. If the ReID feature is too strict, then the model will look for an object that looks identical to the occluded object, and if it cannot find it, then it may mark that as a false negative. On top of this, the full size LABNet model is much more memory intensive as it has to be passed through a ResNet-101 backbone before, causing an unnecessary amount of resources to be allocated to the feature extractor. For this reason, LABNet was trained with a ResNet50 backbone and a smaller overall depth with each channel dimension being scaled down to one-quarter of the original dimension.

In order to avoid using predefined thresholds to determine which appearance network to use, a combination network is tasked with learning the proper mixture of features from each network to provide the most accurate feature vector. This network consists of the feature vector from the FairMOT network as well as the feature vector from LABNet, the output feature vector is a mixture of the two input vectors and is the same size as the combined input feature.

From there, the accurate bounding boxes and robust Re-Identification features will be passed into a network resembling GCNNMatch to be used in a graph convolution based association. Here, the physical relationship between the objects in the past and present will be used to make more accurate matching connections for future frames.
3.2. Balanced Weights

As used in GCNNMatch, the loss function used to measure the performance of the graph network is a Weighted Binary Cross-Entropy Loss. This is used because the model is trying to push positive associations towards 1 and negative associations towards 0. Given the nature of the graph data, there will be far fewer positive associations than negative. This requires a higher weight to be assigned to the positive labels in order to avoid having the network learn to only assign everything to 0 and thus reduce the original loss.

3.3. Triplet Loss

Triplet Loss [38] plays an important role in both LABNet and GTREID. As mentioned earlier, the loss presents the network with both positive and negative examples of a specific class during training. The goal is to ensure that a specific anchor image \( x^a_i \) is closer classified closer to another positive example of that class \( x^p_i \) and further away from any negative class examples \( x^n_i \). This is detailed in Equation 3.1

\[
L_{tri} = \| f(x^a_i) - f(x^p_i) \|_2^2 + \alpha < \| f(x^a_i) - f(x^n_i) \|_2^2
\] 

(3.1)

where

\[
\forall (f(x^a_i), f(x^p_i), f(x^n_i)) \in \tau
\] 

(3.2)

and \( \alpha \) is a margin that is the target between the positive and negative examples. \( \tau \) represents the set of all triplets that are presented during training and \( N \) is the number of elements in that set. The triplet loss in [38] is designed to find harder examples to use during training otherwise the model would solve the problem too quickly, resulting in slower training times without any additional performance boost. This is solved by finding the triplets within the training batch. It is important to ensure that there are enough representations of each class label to make the triplet effective. For this reason a large mini batch is used during training and all positive anchors are used during the triplet computation. The negative examples are chosen in a fashion that makes them less difficult to solve in order to prevent the model from collapsing. The negative examples that are used are chosen so that their distance is farther away than the positive examples, but their squared distance is still close enough to be difficult for the model solve.

This loss is used during the ReID training of LABNet in order to prevent the model from grouping all similar looking identities together. Instead, the identity that is truly the closest is the one that is used in the end. In GTREID the triplet loss is applied to the output classifications of the built in ReID model from the backbone network. The goal is to push the classification probabilities that are incorrect farther away from the classifications that are correct.
3.4. Data Augmentation

![Image of people walking]

Figure 3.3: Sample of Random Erase used during training to strengthen object detector.

Having a strong object detector plays a crucial role in the success of an MOT system. Having accurate and robust detections will greatly improve both the MOT Accuracy and MOT Precision scores of the model. False negatives plague many MOT networks as the detections are missed or their confidence score isn’t high enough to be used for association. On the other hand, if the confidence threshold is further decreased, then the model runs the risk of adding false positive detections that will hurt the tracking scores.

In order to reduce the number of false negatives as much as possible, a series of data augmentations have been implemented to create a more robust object detector. Random Erase \[59\] and Random Noise have demonstrated significant performance increases in detection performance.

Random Erase is the process of blocking part of the annotated bounding box with a solid color during the training of an object detector. This effect can be seen in Figure 3.3 where small gray boxes are covering part of each pedestrian in the image. The boxes themselves are only designed to occlude parts of the annotated bounding boxes, not the image as a whole. The color patches may vary in size and shape across the bounding box. The boxes themselves are set to be a gray tone as this represents a neutral occlusion color that is solely designed to block the image without changing any other characteristics. The goal of random erase is to train an object detector to be more robust to occlusions and partial disappearance. Random Erase is designed to prevent a detector from overfitting to the training data, thus ruining the ability to generalize to situations where the object may go under partial occlusion. This is especially common in pedestrian MOT datasets where multiple individuals will cross paths or get in each other’s way while walking. Having a detector that
can not only identify the occluded individual, but still draw an accurate bounding box helps to reduce the potential for false negatives to spike in the resulting evaluation.

Random Noise has been implemented to help deal with situations in which lighting conditions may be different than those experienced during training. Darker scenes are present in the UAVDAT dataset and as such, the noise increase in the camera is more prevalent. This can lead to not only inaccurate detections but also poor ReID features from the built in ReID head of the backbone network. To combat this, adding noise into the training data allows the system to be more accurate during testing.

### 3.5. Object Detection

As mentioned previously, the detector used for this thesis is the CenterNet based architecture from FairMOT. This detector is used due to the accuracy of the central pixel localization and bounding box size head. This feature is important because the bounding box information is further used by the network to extract image crops that are passed through the LABNet model for rotation invariant feature extraction. The bounding boxes then go into not only the motion prediction component, but also the overlap calculation that is within the graph network.

The accurate detector also plays a role in extracting features from the centerpoint of each detection that is used in the appearance combination network that is present in the graph model. The combination of all of these factors demonstrate how important the object detector is in the success of GTREID.

### 3.6. Loss Functions

The loss functions used in GTREID are the same as those used in the backbone models during training with the addition of the triplet loss.

#### 3.6.1. Heatmap Loss

The heatmap head is trained using a pixel-wise logistic regression with focal loss.

\[
L_k = -\frac{1}{N} \sum_{xyc} \begin{cases} 
(1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\
(1 - Y_{xyc})^\beta (Y_{xyc})^\alpha \log(1 - Y_{xyc}) & \text{otherwise}
\end{cases}
\]

where

\[
Y_{xyc} = \exp\left(-\frac{(x - \tilde{p}_x)^2 + (y - \tilde{p}_y)^2}{2\sigma_p^2}\right)
\]

and \(\tilde{p}\) is the keypoint location divided by the output prediction factor. \(\alpha\) and \(\beta\) are the hyper-parameters of the focal loss while \(N\) is the number of keypoints in the particular image.
3.6.2. Offset Loss

The offset is trained with a simple L1 loss between the predicted point and the ground truth point

\[ L_{off} = \frac{1}{N} \sum_{p} \left| \hat{O}_p - \left( \frac{p}{R} - \tilde{p} \right) \right|, \]  

(3.4)

in which \( R \) is the output stride of the network. This value is set by default to 4 from the baseline [55].

3.6.3. Bounding Box Size

The size prediction head is also trained using an L1 loss at the center location.

\[ L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{pk} - s_k \right| \]  

(3.5)

where the centerpoint \( p_k \) is:

\[ \left( \frac{x^k_1 + x^k_2}{2}, \frac{y^k_1 + y^k_2}{2} \right) \]

and the object size \( s_k \) is

\[ s_k = (x^k_2 - x^k_1, y^k_2 - y^k_1) \]

3.6.4. Total Detector Loss

The output of all of these heads are then combined to build accurate and exact bounding box locations for each object.

\[ L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \]  

(3.6)

3.6.5. Backbone Appearance Loss

In order to train the ReID head of the network, the peak heatmap location is obtained and the class distribution is learned using the Cross Entropy Loss in Equation 3.7

\[ L_{identity} = - \sum_{i=1}^{N} \sum_{k=1}^{K} L^i(k) \log(p(k)). \]  

(3.7)

\( K \) is the number of classes within the dataset, and \( L^i(k) \) is the one-hot representation of the ground truth class labels.
3.6.6. Triplet Loss

The triplet loss used in GTREID is designed to increase the separability between the classification outputs of the backbone ReID network. Triplet Loss[38] is defined as

$$L_{triplet} = \| f(x^a_i) - f(x^p_i) \|^2_2 + \alpha < \| f(x^a_i) - f(x^n_i) \|^2_2.$$  (3.8)

where $x^a_i$ is an anchor classification, $x^p_i$ is a positive classification, and $x^n_i$ is a negative classification. $\alpha$ represents the margin that is enforced between the positive and negative images in order to prevent the model from collapsing.

3.6.7. Total Backbone Loss

The ReID loss is combined with the CenterNet losses to create a final loss

$$L_{total} = \frac{1}{2}( \frac{1}{e^{w_1}} L_{\text{detection}} + \frac{1}{e^{w_2}} L_{\text{identity}} + L_{\text{triplet}} + w_1 + w_2)$$  (3.9)

in which

$$L_{\text{detection}} = L_k + \lambda_{\text{size}} L_{\text{size}} + \lambda_{\text{offset}} L_{\text{offset}}$$

and $w_1$ and $w_2$ are learnable parameters during training [55].

3.6.8. Graph Network Loss

A Weighted Binary Cross-Entropy Loss is used to train the graph network within GTREID. The goal of the graph network is to assign a tracklet-detection pair with a similarity score of 1 and a non-matching pair with a similarity score of 0. Given the inherently imbalanced nature of graph data, there will be a far greater number of mismatched pairs in comparison to the matching pairs. Therefore, the matches need to be weighted higher to avoid the possibility of the model settling and constantly choosing a non-match. GTREID uses the same Weighted Binary-Cross Entropy Loss as GCNNMatch [32]

$$loss = -w_0 \ast (y \ast loss(x)) - w_1 \ast ((1 - y) \ast \log(1 - x))$$  (3.10)

where $w_0$ and $w_1$ are predefined weights that are set to (10,1) as seen in GCNNMatch.
Chapter 4

Experiments

4.1. Datasets

4.1.1. UAVDT Benchmark

Figure 4.1: Sample images from the UAVDT Benchmark. All images are taken from a drone over urban environments. Camera perspectives range from bird’s eye to head on. Weather conditions range from sunny to foggy. Images are taken during day and nighttime.

The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking (UAVDT) [9] is one of the only aerial based Multi-Object benchmarks that is shot from 3 altitude levels at varying times of day with diverse weather. The benchmark is shot at altitudes ranging from low-altitude (10 - 30 meters), medium-
altitude (30-70 meters), and higher-altitude (greater than 70 meters). The change in altitude provides a unique challenge for a MOT network because of the drastically changing resolution of vehicles on the ground. When the drone is in the low-altitude setting, a vehicle’s image area could easily surpass 1000 pixels but at the higher-altitude setting, it is not uncommon to see vehicles whose area is less than 100 pixels. This means the tracker has to be able to make strong detections at all heights as well as draw meaningful features from those small to large bounding boxes. The dataset also features three kinds of camera object view points: front-view, side-view, and bird-view. As a result, camera can be facing anywhere from head on, to directly above the objects that are being tracked. The final attribute within the dataset is the weather condition, which ranges from daylight, night, and fog. The training and testing sequences can have any combination of these attributes at once, which makes them especially difficult to perform well on.

An important note for the UAVDT dataset is that it is entirely available to the public and thus has no evaluation server with which to compare other methods to. The paper does perform a couple of tests on other MOT methods but they are older methods dating back to 2015.

4.1.2. MOT Challenge Dataset

![Sample images from the MOT Challenge dataset. Image perspectives range from surveillance camera to chest height. Dataset includes sequences in which the camera is moving as well as stationary.](image)

The MOT Challenge dataset [30] is a much more commonly referenced Multi-Object Tracking benchmark that is used as a standard for performance. This dataset is designed to only track pedestrians which
makes it a sharp transition from the UAVDT Benchmark. However, this benchmark only has an evaluation server for testing so its leader board is much more clear and serves as a definitive proof of superior performance. The MOT Challenge has multiple iterations from 2015, 2016, 2017, and 2020. MOT15[21] and MOT17[30] are most commonly cited in the recent literature. The viewpoints for the videos range from the chest level of a pedestrian, to a surveillance perspective above a city street. Unlike the UAVDT benchmark, these are not listed attributes of the dataset, and MOT Challenge is strictly a tracking benchmark. In this thesis, only the challenge from 2017 is evaluated on due to its prevalence in literature and the impressive performance of the baseline FairMOT network.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frames</th>
<th>Boxes</th>
<th>Evaluated Targets</th>
<th>Illumination</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAVDT</td>
<td>80,000</td>
<td>841,500</td>
<td>Cars</td>
<td>Day, Night, Fog</td>
</tr>
<tr>
<td>MOT17</td>
<td>11,200</td>
<td>&gt;292,000</td>
<td>Pedestrians</td>
<td>Day, Night</td>
</tr>
</tbody>
</table>

Table 4.1: MOT dataset statistics for UAVDT and MOT17 datasets.

4.1.3. VeRi776 Dataset

Figure 4.3: Sample images from VeRi776 dataset detailing camera locations as well as sample images of vehicles from the dataset. Each image is taken from a slight aerial perspective from a surveillance camera height. Figure source: https://github.com/JDAI-CV/VeRidataset. Date of Access: February 20, 2021.

One of the datasets used to pretrain the LABNet ReID [43] network for vehicle identification was the VeRi776 dataset [29]. Within this dataset is 50,000 images with 776 unique vehicles which are imaged using 20 separate cameras. The cameras are spread out across a 1.0 km² area and all images were taken within 24 hours.

The cameras were set up to record at 25 frames per second at a resolution of 1920x1080 pixels. The cameras were positioned in arbitrary directions and angles in relation to the roadway. In order to make the
dataset challenging, each vehicle in question is imaged by at least two cameras from different viewpoints and background, during different lighting conditions throughout the day. This ensures that the vehicles are not identical in subsequent frames, which would be far too easy for the ReID network. Although this dataset is not the largest vehicle ReID dataset available, it is immensely diverse. Other datasets such as VehicleID [28] contain 221,763 images of 26,267 vehicles. However, this dataset only contains images of either the front of the back of a vehicle, not up to 20 different angles that can be seen in VeRi776. This not only makes the ReID task more challenging, but it also makes the feature more applicable to MOT tasks where vehicle orientation changes drastically from the beginning to the end of the sequence. This is especially true in situations where the camera is moving independently of the target, such as the case with UAVDT and KITTI MOT.

The dataset is further broken down into a training set which consists of 37,781 images of 576 vehicles and a testing set that has 11,579 images with 200 vehicles. Within the testing set, a single sample of each vehicle is chosen as a query image that the ReID network has to use as a reference within the testing set. The query search is made up of 1,678 images. During testing, the query images cannot be used as a match within the testing set. The image that is pulled from the test set is discarded as a potential option, instead the network has to choose from one of the remaining images in order to find a matching vehicle ID. The match can be from any of the camera location of the vehicle itself.

The reason why this dataset was chosen for this task was due to the performance of the LABNet architecture on the testing set as well as the nature of the images themselves. Currently, LABNet achieves state of the art mAP and Rank-1 Accuracy scores on this particular dataset, outperforming the previous state of the art by 4.42% mAP and 1.18% in Rank-1 Accuracy. Ensuring that the ReID network is working with a dataset with which it performs well is an important step to a strong overall MOT system.

VeRi776 is also advantageous in that it is taken from a slightly elevated viewpoint for each image. This allows it to be applicable to both UAVDT and KITTI MOT benchmarks. Both of these datasets focus on vehicles as the main interest. However, they differ slightly in that UAVDT is taken from an aerial perspective and KITTI is taken from a dashboard perspective. The VeRi776 dataset serves as an in between perspective between the two of these. This allows a single pretrained model to perform well for both MOT datasets when used as a feature extractor.

4.1.4. Market1501 Dataset

In order to perform well on a dataset that involves pedestrians rather than vehicles, LABNet had to be pretrained on a second dataset that involved pedestrians. Two of the most common benchmarks are
the Market1501 [57] and Duke MTMC [35] ReID datasets. Both datasets are based around pedestrian re-
identification in a busy urban square. However, the Duke MTMC dataset has since been removed from usage
due to security concerns regarding surveillance applications. Given that this is the case, the Market1501
dataset has been chosen as the primary dataset for training LABNet for pedestrian detection.

The Market1501 dataset consists of 1,501 unique identities from 32,668 bounding boxes with up to 6
(or at least two) unique cameras per person ID. The cameras used include five 1280x1080 cameras and one
720x576 camera. There is slight overlap between each of the cameras that capture all of the images. In
order to draw boxes and crop individuals, a Deformable Part Model (DPM) [14] was utilized. In order for
a bounding box to be marked as “good,” it has to have at least a 50% overlap with a ground truth bounding
box that was hand labeled. In the case where the box has less than 20% overlap, it is labeled as ”distractor.”

The Market1501 dataset has not officially been tested using LABNet for ReID purposes. However, in
initial testing, using the LABNet model in replacement of the standard ReID head of the backbone network,
the results showed improvement over the baseline and as such will be used to pretrain the final LABNet
network. This model will be used to aid the tracking on pedestrian based MOT datasets such as MOT17.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Images</th>
<th>Objects</th>
<th>Cameras</th>
<th>Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERI [29]</td>
<td>49,325</td>
<td>776</td>
<td></td>
<td>Vehicle</td>
</tr>
<tr>
<td>Market1501 [57]</td>
<td>32,668</td>
<td>1501</td>
<td>At least 2, up to 6</td>
<td>Pedestrian</td>
</tr>
</tbody>
</table>

Table 4.2: Re-Identification dataset statistics. Datasets are used to train LABNet model.
4.2. Evaluation

The primary metrics that are used to judge the success of a Multi-Object Tracker are the CLEAR Metrics [1]. These metrics include MOT Accuracy (MOTA), MOT Precision (MOT Precision), Mostly Tracked (MT), Mostly Lost (ML), False Positive (FP), False Negatives (FN), and ID Switching (IDs). Mostly Tracked is defined as the number of objects that are tracked for at least 80 percent of their time in frame. Mostly Lost counts the number of objects who were tracked for less than 20 percent of their lifetime in frame. ID Switching is total number of times that a tracked trajectory changes its matched GT identity.

MOTA is the most telling judgement of an MOT network’s performance as it encompasses a couple of other metrics in a single number. [1] defines MOTA as:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$  (4.1)

where $m_t$ is the number of misses, $fp_t$ is the number of false positives, and $mme_t$ is the number of mismatches for time $t$.

A miss describes a situation in which no hypothesis was made for a potential track when the opportunity was there. A False Positive on the other hand is a situation in which a track hypothesis was made where there wasn’t one. This is typically seen when an advertisement or a street sign is detected as a person and their location is falsely tracked as a potential match. Finally a mismatch is an example of when a tracking hypothesis for an object changes compared to the previous frame. This could take place when the network falsely associates one individual with another, or when an object is assigned a new ID after an occlusion takes place.

Another important metric is the MOT Precision which demonstrates the total error in the position of a ground truth object and the network’s hypotheses per frame. This metric is independent of any of the tracker’s re-identification skills or trajectory skills. MOTP is defined as:

$$MOTP = \frac{\sum_{i,t} d_{it}^2}{\sum_t c_t}$$  (4.2)

where $d_{it}^2$ is the distance between the ground truth object and the corresponding hypothesis and $c_t$ is the number of matches found for time $t$.

Other common metrics used in MOT performance evaluation include Precision and Recall. [30] details these metrics as follows:

$$Recall = \frac{TP}{TP + FN}$$
\[ Precision = \frac{TP}{TP + FP} \]

Where TP and FP are the total number of false positive and false negatives over all frames, respectively.

IDF\(_1\) is another common metric which is the ratio of correctly identified detections over the average number of ground-truth and computed detections.

\[ IDF_1 = \frac{2IDTP}{2IDTOP + IDFP + IDFN} \tag{4.3} \]

where

\[ IDFN = \sum_{\tau \in AT} \sum_{t \in \tau} m(\tau, \gamma_m(\tau), t, \Delta) \]

and

\[ IDFP = \sum_{\tau \in AC} \sum_{t \in \tau} m(\tau_m(\gamma), \gamma, t, \Delta) \]

The final two metrics reported in the MOT Challenge include FAF and Fragmentations. FAF is the average number of false alarms per frame and fragmentations are the total number of times a trajectory is interrupted during tracking. For both metrics, lower scores indicate better performance.

### 4.3. Determining the Proper ReID Network

<table>
<thead>
<tr>
<th>ReID Network</th>
<th>MOTA</th>
<th>MOTP</th>
<th>MT</th>
<th>ML</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>45.56</td>
<td>22.84</td>
<td>40</td>
<td>81</td>
<td>804</td>
<td>10,561</td>
<td>109</td>
</tr>
<tr>
<td>LABNet Market 1501 Only</td>
<td>45.70</td>
<td>22.84</td>
<td>40</td>
<td>81</td>
<td>792</td>
<td>10,560</td>
<td>93</td>
</tr>
<tr>
<td>LABNet Finetuned MOT17</td>
<td><strong>45.77</strong></td>
<td><strong>22.86</strong></td>
<td>40</td>
<td>81</td>
<td>793</td>
<td><strong>10,548</strong></td>
<td><strong>89</strong></td>
</tr>
</tbody>
</table>

Table 4.3: Comparing LABNet feature size versus MOT performance when inserted into FairMOT backbone. Results were tested on the MOT17 validation set. All networks have the same output feature dimension. Backbone model trained on GTX1080 with a batch size of 7.

During the initial testing for GTREID, a test was conducted to see if it is worthwhile to change the feature extractor used during association. To test this, the baseline FairMOT model was trained with an output feature dimension of 1x512. During inference, this model was then replaced with a modified version of LABNet that was detailed in the Methodology chapter. As shown in Figure 4.3, the LABNet model improves upon the MOTA and MOTP performance of the baseline. The most significant improvement is in the number of ID switches which dropped by 20 when compared to the baseline.

This improvement is due to LABNet being a more advanced ReID model that is designed to provide more
accurate, rotation invariant features than the backbone model, despite LABNet not being directly integrated into the backbone. While there was an improvement in overall performance, the number of mostly lost and mostly tracked objects did not improve as the changes to the False Positive, False Negative, and Identity Switching was not enough to overcome the lifespan thresholds that are detailed for the tracking metrics.

Despite the fact that the LABNet model used in these experiments was a trimmed down version, it still outperformed the baseline due to the better training metrics and graph convolution layers on top of the feature extractor. The lightweight nature of LABNet with ResNet-50 means that it is easier to train with as well and thus would fit well into the GTREID backbone.

Table 4.4: ReID network comparison between LABNet and FairMOT ReID head. Checkmark indicates which network had a higher MOTA performance for each video sequence. Attributes for each sequence are listed in the right hand column. Sequences with two checkmarks indicate same MOTA score for both networks. Altitude attributes are as follows: low-altitude (10 - 30 meters), medium-altitude (30-70 meters), and higher-altitude (greater than 70 meters).

<table>
<thead>
<tr>
<th>Sequence</th>
<th>LABNet</th>
<th>FairMOT</th>
<th>Light</th>
<th>Height</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0203</td>
<td>✓</td>
<td></td>
<td>Night</td>
<td>Low</td>
<td>Bird</td>
</tr>
<tr>
<td>M0205</td>
<td>✓</td>
<td>✓</td>
<td>Night</td>
<td>low</td>
<td>Side/Front</td>
</tr>
<tr>
<td>M0208</td>
<td>✓</td>
<td>✓</td>
<td>Sunny</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M0209</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Low</td>
<td>Front/Side</td>
</tr>
<tr>
<td>M0403</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Medium</td>
<td>Front/Side</td>
</tr>
<tr>
<td>M0601</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>High</td>
<td>Front/Side</td>
</tr>
<tr>
<td>M0602</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M0606</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Medium</td>
<td>Front/Side</td>
</tr>
<tr>
<td>M0701</td>
<td>✓</td>
<td>✓</td>
<td>Fog</td>
<td>Medium</td>
<td>Bird</td>
</tr>
<tr>
<td>M0801</td>
<td>✓</td>
<td>✓</td>
<td>Sunny</td>
<td>Low</td>
<td>Bird</td>
</tr>
<tr>
<td>M0802</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Low</td>
<td>Side</td>
</tr>
<tr>
<td>M1001</td>
<td>✓</td>
<td>✓</td>
<td>Night</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M1004</td>
<td>✓</td>
<td></td>
<td>Fog</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M1007</td>
<td>✓</td>
<td></td>
<td>Night</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M1009</td>
<td>✓</td>
<td>✓</td>
<td>Fog</td>
<td>Medium</td>
<td>Bird</td>
</tr>
<tr>
<td>M1101</td>
<td>✓</td>
<td></td>
<td>Night</td>
<td>Low</td>
<td>Front/Side</td>
</tr>
<tr>
<td>M1301</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Medium</td>
<td>Front</td>
</tr>
<tr>
<td>M1302</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Medium</td>
<td>Side</td>
</tr>
<tr>
<td>M1303</td>
<td>✓</td>
<td></td>
<td>Sunny</td>
<td>Low</td>
<td>Side</td>
</tr>
<tr>
<td>M1401</td>
<td>✓</td>
<td>✓</td>
<td>Sunny</td>
<td>Low</td>
<td>Front</td>
</tr>
</tbody>
</table>

Table 4.5: Breakdown of which network performed better per attribute. Results correspond to tests performed in Table 4.4.

<table>
<thead>
<tr>
<th>Network</th>
<th>Night</th>
<th>Sunny</th>
<th>Fog</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Bird</th>
<th>Side</th>
<th>Front</th>
</tr>
</thead>
<tbody>
<tr>
<td>LABNet</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>FairMOT</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4.5: Breakdown of which network performed better per attribute. Results correspond to tests performed in Table 4.4.
A further inspection of the ReID models was performed between LABNet and the built-in FairMOT backbone. In this test, the MOTA score for each sequence was measured. In Table 4.4, the network that achieved the higher MOTA score is marked with a ✓. The areas where both boxes are checked indicates a sequence in which the models scored the same. The reason MOTA was used as the defining metric is due to the metrics that it encompasses. MOTA includes the number of False Positives, False Negatives, and ID Switches for the model. Therefore, it is a good indication of overall network performance. To the right of each sequence is the list of attributes that are associated with that sequence including the lighting conditions, camera height, and perspective of the camera.

As shown in Table 4.4, it is difficult to pick a single ReID network to use for all sequences of the UAVDT dataset. Illumination and altitude are the only definitive difference between the two networks with LABNet outperforming FairMOT consistently in the dark sequences and only narrowly beating it in sunny situations. Other attributes such as height were also evenly matched between the two different ReID networks with the only clear winner being FairMOT in the highest altitudes. Point of view had different results depending on the network, however, GTREID has no definitive way of determining the point of view of the camera. This makes this attribute very difficult to account for. With this being the case, the idea of combining the output of the two networks into a single feature vector was formed. Given that the two output feature vectors proved to be most effective at a dimension of 1x512, the two vectors could be concatenated to form a single 1x1024 feature vector. Once the outputs were concatenated, the output is passed through to a combination network before returning as a 1x1024 output feature vector. The output vector is a blended combination of the two feature vectors from the network.

In order to combine the two vectors together, the feature were normalized on their own before passing through the FC layer. The output of the layer was then sent through Batch Normalization [16] layers, followed by ReLU activation function. During training, a dropout layer was used to prevent overfitting one set of weights from a single network. This proved beneficial due to the fact that the ReID head of the backbone network was still being trained as the graph network began training. Using dropout prevented the network from defaulting to only using the fully trained output of LABNet. This lead to the best combination of ReID features that were available.

4.4. Triplet Loss

In order to test the effectiveness of the triplet loss, a test was conducted on the validation set of the MOT17 dataset as well as the testing set of the UAVDT Benchmark. The results proved to be mixed as it added performance on UAVDT and detracted from the results of MOT17.
Table 4.6: MOT performance comparison between baseline network and network trained with triplet loss on the MOT17 validation split. Models were trained on a P4 graphics card at a batch size of 4 and a ReID dimension of 512.

<table>
<thead>
<tr>
<th>MOT17</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
<th>MT</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o triplet loss</td>
<td>44.2</td>
<td>22.7</td>
<td>1,484</td>
<td>10,165</td>
<td>111</td>
<td>45</td>
<td>75</td>
</tr>
<tr>
<td>w/ triplet loss</td>
<td>43.5</td>
<td>22.2</td>
<td>1,027</td>
<td>10,780</td>
<td>98</td>
<td>41</td>
<td>87</td>
</tr>
</tbody>
</table>

Table 4.7: MOT performance comparison between baseline network and network trained with triplet loss on the UAVDT Benchmark training and testing set. Models were trained on a P4 graphics card at a batch size of 8 and a ReID dimension of 128.

<table>
<thead>
<tr>
<th>UAVDT</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
<th>MT</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o triplet loss</td>
<td>46.18</td>
<td>79.33</td>
<td>36,341</td>
<td>146,890</td>
<td>238</td>
<td>507</td>
<td>238</td>
</tr>
<tr>
<td>w/ triplet loss</td>
<td><strong>46.36</strong></td>
<td>78.29</td>
<td>38,162</td>
<td>144,420</td>
<td>298</td>
<td>524</td>
<td>319</td>
</tr>
</tbody>
</table>

As seen in Table 4.6, the addition of the triplet loss does not help the model in any significant fashion, instead it ended up hurting the model in the end. Table 4.7 shows a slightly better result with the triplet loss helping to improve the results of the MOTA scores as well as the number of vehicles that were mostly tracked.

The reason that the triplet loss added performance to one dataset and not the other is due to the larger number of vehicles that are within the dataset. The UAVDT benchmark has 1,368 unique identities whereas the MOT17 train and validation split has just 363 identities. This means that the triplet loss that is implemented in this network is unable to push the label probabilities far enough away from one another to get a useful triplet loss. As mentioned earlier, it is able to solve the triplet labels too easily and it end up hurting the overall test results.

4.5. Data Augmentation Experiments

<table>
<thead>
<tr>
<th>MOT17</th>
<th>IDs</th>
<th>MT</th>
<th>ML</th>
<th>FP</th>
<th>FN</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/ RE + RN ReID Dim 512</td>
<td><strong>96</strong></td>
<td>41</td>
<td>79</td>
<td>1,060</td>
<td>10,424</td>
<td><strong>45.1</strong></td>
<td>22.4</td>
</tr>
<tr>
<td>w/o RE + RN ReID Dim 512</td>
<td>111</td>
<td><strong>45</strong></td>
<td>75</td>
<td>1,484</td>
<td>10,165</td>
<td>44.2</td>
<td><strong>22.7</strong></td>
</tr>
<tr>
<td>w/o RE + RN ReID Dim 128</td>
<td>99</td>
<td>44</td>
<td>84</td>
<td>1,343</td>
<td>10,719</td>
<td>42.3</td>
<td>22.2</td>
</tr>
</tbody>
</table>

Table 4.8: Performance difference between the baseline model with triplet loss, and the baseline trained with Random Erase and Random Noise as well as triplet loss. Training and testing parameters were the same between both runs. Random Erase performed 40% of the time with a scale ranging from 0.02-0.25 percent of the bounding box and an aspect ratio of 0.2, 0.5. Random Noise was performed on 30% of images with a mean and standard deviation of 2. Test was conducted on the MOT17 train/validation set. Results tested on the RIT RC Cluster[36] on a P4 GPU with a batch size of 4.
Figure 4.5: Detection comparison between baseline model trained without Random Erase and Random Noise (left) and baseline model trained with Random Erase and Random Noise (right). Top images show two pedestrians being blocked by the person in the red shirt, they are undetected in the baseline, but they are detected in the baseline trained with augmentation. In the bottom comparison, there is an individual who is heavily occluded by the person in the red shirt and the person in the black outfit. They are not picked up by the baseline but they are identified by the augmented baseline.

Table 4.8 shows the performance increase when using Random Erase and Random Noise on the validation set of the MOT17 dataset. The MOTA score increased by 0.90% which is mostly attributable to the decrease in false positives which fell by 424 from the Baseline with a ReID dimension of 512 and 283 from the baseline with the ReID dimension of 128. There was a slight increase in the number of false as compared to the baseline model with a ReID dimension of 512. Although the number of false negatives increased, the lower number of identity switches along with the decrease in false positives meant that there was an overall improvement in MOTA despite a lower MOTP score.

Figure 4.5 shows examples in which the detector was able to identify and draw bounding boxes around objects that the baseline model was not able to see. The top set of images shows earlier detection of two pedestrians on the left hand side of the man in the red shirt. His arms slightly occlude the two people walking towards the camera, causing them to not get a detection in the original baseline model. Likewise in the bottom frame, the individual between the man in the red shirt and the man in the black outfit is identified
after training with the augmentations when he was not identified in the baseline. The backbone network still outperforms the augmented model in the number of ID switches, MOTP, as well as the number of tracks that are categorized as mostly tracked. This could be due to the decrease in bounding box accuracy that is causing poor appearance feature extractions leading to larger ID switches or missed tracks in the validation sequences. The appearance feature issue is aided in the final model by the use of two appearance feature extractors as well as the association learning within the graph network.

Figure 4.6: Detection comparison between baseline model trained without Random Erase and Random Noise (left) and baseline model trained with Random Erase and Random Noise (right). Both images show false positive detections on the left hand side of each image. The detections are reflections of pedestrians walking close to the windows.

Figure 4.6 shows examples of the false positives which took place in both sequences of the MOT17 validation set. In each frame, a false positive detection is made of a pedestrian’s reflection in the store’s glass. Both detections occur on the left hand side of each image. While each example technically has a person within the bounding box, the evaluation metrics do not count reflections as tracks, therefore these were listed as false positive detections. In the bottom image comparison, while the detector did identify the reflection of the individual in the glass, it also identified more people in the lefthand side of the background by the cross-walk. This additional detection would end up helping the overall score of the model as this
additional detection would help the MOTA score as well as the number of False Negatives.

4.6. Number of Layers

Once the output dimension of the appearance feature vector was settled, the next step was to determine the number of layers that would be necessary to get the best overall result for the network. A simple test was conducted on the KITTI-MOT dataset in which models were trained back to back using the same backbone and association techniques. The only difference between the two was the number of layers in the combination network. In each case, each layer had its own ReLU activation function, as well as dropout and BatchNorm implementations.

<table>
<thead>
<tr>
<th>Experiments</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>MT (%)</th>
<th>ML (%)</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Layer</td>
<td>47.627</td>
<td>79.511</td>
<td>41.32</td>
<td>23.06</td>
<td>38,461</td>
<td>139,780</td>
<td>307</td>
</tr>
<tr>
<td>2 Layer</td>
<td>48.269</td>
<td>79.337</td>
<td>45.06</td>
<td>22.82</td>
<td>42,005</td>
<td>134,050</td>
<td>269</td>
</tr>
</tbody>
</table>

Table 4.9: Performance comparison of feature vector combination network. Both models trained and tested with the same parameters and output dimensions of 1x1,024. Both models have been trained and tested on the UAVDT train and test sequences.

As seen in Table 4.9, the 2-Layer network increased performance on the UAVDT Benchmark. The MOTA score increased by 0.642% as well as a 3.74% increase in mostly tracked with a 0.24% decrease in the number of mostly lost objects. The ID switching also went down by 38 which indicates that the appearance feature is helping to improve the output in regard to keeping the identities constant during tracking. The 1-Layer network had a slight improvement in MOTP as well as False Positives but these improvements were not significant enough to overcome the remaining improvements from the 2-Layer network.
Figure 4.7: Layout of the ReID Combination network that is implemented in GTREID. Features from LABNet and FairMOT are concatenated on to form a single 1024d ReID vector. This is then passed through an FC layer with dropout to form a single combine 1024d output vector that is used as the appearance feature vector.

Because of the results of this test, the final appearance combination model is formed as seen in Figure 4.7. This model combines both appearance vectors from LABNet as well as the FairMOT backbone into a single vector, then passes the values through a dual layer combination network, implementing Dropout, BatchNorm, and ReLU layers before being sent to the rest of the network.

4.6.1. Ablation Study

<table>
<thead>
<tr>
<th>Trial</th>
<th>Triplet</th>
<th>Augment</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
<th>MT (%)</th>
<th>ML (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>✓</td>
<td>✓</td>
<td>47.56</td>
<td>78.855</td>
<td>43.769</td>
<td>134,720</td>
<td>279</td>
<td>44.01</td>
<td>23.78</td>
</tr>
<tr>
<td>Baseline</td>
<td>✓</td>
<td>✓</td>
<td>47.195</td>
<td>79.059</td>
<td>42.266</td>
<td>137,470</td>
<td>285</td>
<td>42.54</td>
<td>23.31</td>
</tr>
<tr>
<td>Baseline</td>
<td>✓</td>
<td>✓</td>
<td>47.209</td>
<td>78.708</td>
<td>43.144</td>
<td>136,530</td>
<td>293</td>
<td>42.87</td>
<td>23.31</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td>✓</td>
<td>47.455</td>
<td>78.78</td>
<td>43,776</td>
<td>134,990</td>
<td>363</td>
<td>43.68</td>
<td>22.57</td>
</tr>
<tr>
<td>GTREID</td>
<td>✓</td>
<td>✓</td>
<td>48.269</td>
<td>79.337</td>
<td>42,005</td>
<td>134,050</td>
<td>269</td>
<td>45.06</td>
<td>22.82</td>
</tr>
</tbody>
</table>

Table 4.10: **min box area 200**. Performance progression of model throughout each successive feature addition. All performance tests were trained and tested on the UAVDT Benchmark. Baseline Tests were performed with a batch size of 12, whereas the GTREID tests were performed at a batch size of 8. Triplet indicates the usage of the triplet loss during training and Augment indicates the use of Random Erase and Random Noise during training. Augmentation parameters are the same as those from Table 4.8. Values displayed in this table are for a minimum tracked area of 200 square pixels.
Table 4.11: Performance progression of model throughout each successive feature addition. All performance tests were trained and tested on the UAVDT Benchmark. Baseline Tests were performed with a batch size of 12, whereas the GTREID tests were performed at a batch size of 8. Triplet indicates the usage of the triplet loss during training and Augment indicates the use of Random Erase and Random Noise during training. Augmentation parameters are the same as those from Table 4.8. Values displayed in this table are for a minimum tracked area of 100 square pixels.

Table 4.10 shows the performance progression of GTREID on the UAVDT Benchmark set. While GTREID does not prove to have the best overall result on each individual metric, it has the best overall result across the board. An interesting note is the independent effects of the triplet loss and data augmentations versus their combined improvement.

The triplet loss and data augmentations, when used on their own, tend to decrease the performance of the baseline model with the data augmentations dropping the MOTA performance by 0.246% and the MOTP performance by 0.072% when the minimum box area is set to 200. The data augmentations only improved the false positive rate and the number of identity switches. Similar results are seen when triplet loss is used by itself, with the MOTA score dropping by 0.26%. However, the MOTP got a slight bump in performance from the baseline as well as the false positives and the ID switches.

When the two are combined together, the MOTA and MOTP scores both improve due to the improved performance in false positives, false negatives, and ID switches. The number of mostly tracked objects also improves as the rest of the detector performance improves.

These improvements combine with the 2-Layer ReID combination network to give an overall MOTA boost of 0.814% as well as a 0.5% improvement in MOTP over the baseline. This is due to the further improvements in every other metric except for the number of Mostly lost tracks where the baseline model still holds a 0.25% better score.

When the minimum box area is set to 100, as shown in Table 4.11, the performance differential between GTREID and the baseline are slightly less significant. This is due to the issue of having bounding boxes which may be too small for LABNet to build an accurate feature from. This causes the network to reject more potential matches as well as increase the number of ID switches with a greater change than the Baseline. Although GTREID does not have the same level of performance increase when the minimum bounding box...
size is less, it still comes in either first or second in all categories except False Negative and IDs. This indicates that the overall performance boosts can be attributed to the graph association network, not just the data augmentations.

4.7. Training Implementation

Training the entire GTREID network takes place over multiple steps. The first step requires the training of the LABNet [43] ReID network. Separate training took place for the VeRi776 and Market1501 datasets in order to tackle the two tracking categories that will be tested. The training for this model followed the same instruction as [43] with the only difference being the size of the network that was being used. As seen in Table 4.3, further finetuning on the MOT datasets proved beneficial to the output of the MOT network as well.

The main FairMOT backbone was trained for 24 epochs with Random Erase and Random Noise implemented from epoch 3 to epoch 13. The reason for the limited data augmentation was to reduce the possibility of negatively affecting the training of the ReID head within the backbone. If the augmentations took place for all epochs, ablation studies proved that the ReID network would sample too many gray bounding boxes such as those seen in Figure 3.3. This negatively impacted the learning outcome and performance of the feature extractor.

Starting on the twenty-fourth epoch, the GCN model began training. Given the increased number of parameters in the GCN network, the entire model had to be trained at a slightly lower batch size than the original mode with a batch size of 8 rather than 12. The GCN network is trained using feature extractor inputs from both LABNet and FairMOT, therefore each batch required the ReID models to be set to their evaluation mode momentarily in order to pass their ReID features into the GCN. Once both ReID features were passed into the GCN, their feature vectors are combined and passed through the combination network before passing through the rest of the network.

The reason why the GCN model began training later in the epoch cycle was to prevent weak features from the FairMOT ReID head. If the model began training at the same time as the feature extractor, there was a chance that the combination model would favor too heavily towards the LABNet features and discard the potentially useful features coming from the FairMOT network (Figure 4.7. In addition to this, the baseline GCN model was able to converge within 7 training epochs which worked well for the final application within the backbone. In order to further limit the potential for overfitting within the combination network, layer dropout was used with a probability of 0.5. With this, the fully connected layers would randomly disconnect, forcing the network to use the remaining options which improves accuracy of the network. The GCN layers
also featured Batch Normalization and ReLU activation functions to help improve performance.

The learning rate for the backbone network started at $1^{-4}$, then dropped to $1^{-5}$ at epoch 20 and remained at that value until the end of training. As for the Graph Network training, the learning rate was set to $1^{-3}$ for epochs 24 thru 26, then dropped to $1^{-4}$ for the remainder of the training. The Adam Optimizer [18] was used to train all portions of the final network. Separate optimizers were used for the backbone and the GCN model.
Chapter 5

Results

5.1. MOT17 Results

<table>
<thead>
<tr>
<th>Tracker</th>
<th>MOTA (%)</th>
<th>IDF1 (%)</th>
<th>MOTP (%)</th>
<th>MT</th>
<th>ML</th>
<th>FP</th>
<th>FN</th>
<th>IDs</th>
<th>Frag</th>
<th>Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGGMOT</td>
<td>77.1</td>
<td>74.1</td>
<td>81.1</td>
<td>1,083</td>
<td>315</td>
<td>23,307</td>
<td>102,369</td>
<td>3,480</td>
<td>6,939</td>
<td>9.9</td>
</tr>
<tr>
<td>ReMOT_box</td>
<td>77.0</td>
<td>72.0</td>
<td>80.3</td>
<td>1,218</td>
<td>324</td>
<td>33,204</td>
<td>93,612</td>
<td>2,853</td>
<td>5,304</td>
<td>1.8</td>
</tr>
<tr>
<td>TTS</td>
<td>76.7</td>
<td>75.1</td>
<td>82.0</td>
<td>1,200</td>
<td>387</td>
<td>36,231</td>
<td>93,150</td>
<td>2,346</td>
<td>7,719</td>
<td>1.1</td>
</tr>
<tr>
<td>TLR [56]</td>
<td>76.5</td>
<td>73.6</td>
<td>81.2</td>
<td>1,122</td>
<td>300</td>
<td>29,808</td>
<td>99,510</td>
<td>3,369</td>
<td>6,063</td>
<td>15.6</td>
</tr>
<tr>
<td>FUFET [39]</td>
<td>76.2</td>
<td>68.0</td>
<td>81.1</td>
<td>1,203</td>
<td>321</td>
<td>32,796</td>
<td>98,475</td>
<td>3,237</td>
<td>5,658</td>
<td>6.8</td>
</tr>
<tr>
<td>CSTrack [25]</td>
<td>74.9</td>
<td>72.6</td>
<td>80.9</td>
<td>978</td>
<td>411</td>
<td>23,847</td>
<td>114,303</td>
<td>3,567</td>
<td>7,668</td>
<td>15.8</td>
</tr>
<tr>
<td>ort_track</td>
<td>74.8</td>
<td>73.1</td>
<td>81.4</td>
<td>1,023</td>
<td>540</td>
<td>21,261</td>
<td>119,181</td>
<td>1,632</td>
<td>2,319</td>
<td>33.0</td>
</tr>
<tr>
<td>FBMOT</td>
<td>74.0</td>
<td>74.4</td>
<td>81.3</td>
<td>1,002</td>
<td>456</td>
<td>27,861</td>
<td>116,775</td>
<td>2,166</td>
<td>4,629</td>
<td>7.8</td>
</tr>
<tr>
<td>FairMOT [55]</td>
<td>73.7</td>
<td>72.3</td>
<td>81.3</td>
<td>1,017</td>
<td>408</td>
<td>27,507</td>
<td>117,477</td>
<td>3,303</td>
<td>8,073</td>
<td>25.9</td>
</tr>
<tr>
<td>GTREID</td>
<td>70.2</td>
<td>63.71</td>
<td>80.7</td>
<td>927</td>
<td>471</td>
<td>30,825</td>
<td>132,297</td>
<td>5,076</td>
<td>9,615</td>
<td>3.61</td>
</tr>
</tbody>
</table>

Table 5.1: MOT17 leaderboard for private detections as of March 16, 2021. Scores are ordered based on MOTA performance as seen in the MOTChallenge website.

<table>
<thead>
<tr>
<th>Training Data</th>
<th>MOTA (%)</th>
<th>IDF1 (%)</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOT17 Only</td>
<td>69.8</td>
<td>69.9</td>
<td>3,996</td>
</tr>
<tr>
<td>Mixed</td>
<td>72.9</td>
<td>73.2</td>
<td>3,345</td>
</tr>
<tr>
<td>CrowdHuman + Mix</td>
<td>73.7</td>
<td>72.3</td>
<td>3,303</td>
</tr>
<tr>
<td>GTREID MOT17 Only</td>
<td>70.2</td>
<td>63.71</td>
<td>5,076</td>
</tr>
</tbody>
</table>

Table 5.2: FairMOT Baseline comparison results on MOT17 when using different training datasets. MOT17 entails only the MOT17 training set was used during training. ‘Mix’ details a training dataset that utilizes Caltech Pedestrian [8], CityPersons [54] (sub section of [5]), CUHK-SYSU[49], PRW [58], ETHZ [10], MOT17 [30], and MOT16 [30]. CrowdHuman+Mix includes 60 epochs of pretraining on the CrowdHuman dataset [40]. This pretraining takes place on the backbone dla-34 network. Results sourced from FairMOT github on March 16, 2021.
Table 5.1 displays the top scoring MOT networks on the MOT17 dataset’s leaderboard. These models are all listed as having private detections, meaning they used an outside detector as the input to their MOT association networks. Table 5.2 references the results for FairMOT when using different levels of training data. The original authors utilized a larger sum of datasets to train their model before testing on the final MOT17 test set. As shown in the table, the best performance is achieved when pretraining on CrowdHuman for 60 epochs and then adding the mixed dataset afterwards. Not all of the datasets that are used in the FairMOT training are MOT datasets, some are simply object detection sets. As mentioned previously, this is one of the benefits of a model with a shared backbone, they are able to be trained on datasets with still images. When training GTREID, only the MOT17 dataset was used for training. This is done to test the ability of the network to achieve usable performance on just one training set. Therefore, the goal for GTREID is to compete against the FairMOT model that was only trained on MOT17.

The results of GTREID on the MOT17 challenge show that GTREID is successful in the MOTA performance but struggles in other metrics when compared to the baseline that is only trained on MOT17. GTREID has significantly more ID switches than FairMOT but the larger MOTA score indicates that it has a larger success rate when it comes to false positive and false negative results per frame. This is likely due to the data augmentations providing a greater number of detections for the association network to utilize. In situations such as crowded pedestrian scenery, GTREID’s appearance feature association struggles to give accurate predictions. This is due to the overlapping bounding boxes which causes LABNet to frequently sample all of the pedestrians within the bounding box, giving an ambiguous feature vector to work with. This issue is explored further in Section 6.3.

Overall for MOT17, the performance increases for this dataset have more to do with the data augmentations and bounding box tracking capabilities of the tracker, rather than the ReID combination network that has been implemented given the increased number of ID switches.

5.2. UAVDT MOT Results

The top listed networks from the UAVDT dataset paper are:
<table>
<thead>
<tr>
<th>Method</th>
<th>IDF (%)</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>MT (%)</th>
<th>ML (%)</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP [48]</td>
<td>61.5</td>
<td>43.0</td>
<td>73.5</td>
<td>45</td>
<td>22.7</td>
<td>541</td>
</tr>
<tr>
<td>DSORT [47]</td>
<td>58.2</td>
<td>40.7</td>
<td>73.2</td>
<td>41.7</td>
<td>23.7</td>
<td>2,061</td>
</tr>
<tr>
<td>IPGAT [51]</td>
<td>49.4</td>
<td>39.0</td>
<td>72.2</td>
<td>37.4</td>
<td>25.2</td>
<td>2,091</td>
</tr>
<tr>
<td>SBMA [52]</td>
<td>48.5</td>
<td>38.6</td>
<td>72.1</td>
<td>38.9</td>
<td>24.4</td>
<td>3,489</td>
</tr>
<tr>
<td>IOUT [4]</td>
<td>23.7</td>
<td>36.6</td>
<td>72.1</td>
<td>37.4</td>
<td>25.0</td>
<td>9,938</td>
</tr>
</tbody>
</table>

Table 5.3: UAVDT Results from [9]. Scores are rank ordered based on the MOTA score reported in the paper. All scores are reported using the provided Faster R-CNN detections.

<table>
<thead>
<tr>
<th>Method</th>
<th>IDF (%)</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>MT (%)</th>
<th>ML (%)</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>FairMOT Baseline</td>
<td>68.03</td>
<td>49.63</td>
<td>78.0</td>
<td>49.96</td>
<td>17.69</td>
<td>512</td>
</tr>
<tr>
<td>GCNNMatch Baseline with</td>
<td>49.202</td>
<td>31.884</td>
<td>70.788</td>
<td>51.34</td>
<td>15.65</td>
<td>6,817</td>
</tr>
<tr>
<td>FRCNN Detections</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTREID</td>
<td>68.01</td>
<td><strong>50.00</strong></td>
<td><strong>78.8</strong></td>
<td>50.12</td>
<td>17.85</td>
<td>444</td>
</tr>
</tbody>
</table>

Table 5.4: Results of independent MOT network testing. Results computed on Tesla v100 GPU [36].

In the UAVDT dataset, GTREID performs at the top of the reported results for this benchmark. It achieves both the highest MOTA value as well as the highest MOTP value out of any of the other trackers. The number of ID switches is also significantly better than those from any of the previous trackers. This is due to not only the strong appearance feature performance, but also the bounding box overlap considerations of the graph network.

The performance of the backbone GCNNMatch network is so poor mainly due to the lack of a built in detector as well as the lack of any non-maximal suppression systems in code itself. It was tested as is, and thus suffered greatly from a large number of poor detections which results in a significant drop in performance. GTREID takes care of this with a stronger detector than the provided Faster R-CNN detections, but also the advancement of two powerful appearance feature extractors.

GTREID outperforms FairMOT in every metric aside from the Mostly Lost percentage and the IDF<sub>1</sub> score although the margins were very narrow. This slight drop in performance is likely due to objects which are very small or driving away from the drone towards the horizon, in cases such as this, GTREID struggles more than the baseline. These are the only areas in which GTREID underperforms the baseline, all others have been improved upon. As for the lifespan comparisons with GCNNMatch, GTREID performs slightly worse than the original mainly due to the shear number of detections that are made on each object. The larger number of detections simply means that they are more likely to identify each object in the frames and therefore have better matching capabilities. However, the MOTA score tells the other side of that story.
in which there are a large number of false positives as well as identity switches that drastically bring down the overall performance of the tracker. The bounding box error per object is much larger with GCNNMatch which is an indication of poor localization due to the lack of a motion model as well as the weaker detector.

The localization of the detector is important on a dataset such as this because the camera’s viewpoint and height can change so drastically between sequences. The high sequences (greater than 70 meters) can be difficult for the anchor based bounding boxes to overlap well with the ground truth. However, the higher resolution of the DLA backbone is more ideal for finding, and accurately outlining the very small objects from high above.

The decrease in ID switching is also a positive note for the GTREID network as it means the combined ReID features as well as the bounding box overlap are helping to make the associations between objects much stronger, thereby improving the MOTA by having fewer false positives and false negatives.
Chapter 6

Discussion

The results of the combined ReID networks as well as the graph association and detector upgrades have resulted in a stronger overall association network when utilized from an aerial setting and mixed results when applied to a crowded pedestrian tracking dataset.

6.1. Speed

The performance improvements of the network do come with some additional drawbacks. The chief among which is the speed decreases. The original FairMOT network was able to run through the MOT17 sequences at over 18 frames per second throughout all of the validation sequences. It was able to do this while consuming less than 1.5GB of ram on a GTX 1080 graphics card. On the other hand, the new model is slowed down significantly due to the graph association network that has been added. The model struggled to break 4 frames per second through all sequences while consuming over 10GB of RAM during testing.

While this model may have been slower than the FairMOT baseline, it was significantly faster than the GCNNMatch baseline. For that model, datasets like the UAVDT Benchmark took 3 days to test all sequences. GTREID is able to perform the same task in approximately an hour and a half on the same settings. This is mainly due to the lack of parallel processing on the baseline network. It was drastically improved for GTREID and as a result the model does not vary speeds as significantly based on the number of objects in an image.

6.2. Additional Parameters

When adding in the additional GCNN network during training and inference, it is important to consider the additional number of parameters that will come with it. The FairMOT baseline itself has 20,634,620 training parameters between the backbone network and the various heads that are on top of it. Adding in the
graph network on top of this adds an additional 593,553 parameters due to all of the additional linear layers as well as the graph convolutional layers. LABNet on its own has 2,532,440 parameters that are separated from the backbone or the GCNN network. The total number of parameters is increased by 15.15% when adding in the two additional networks, however the performance improvements show that the added capabilities of the graph network are worthwhile.

### 6.3. Bounding Box Localization

Another issue that arose during experimentation was the decreases in performance on the pedestrian tracking tasks. The cause of this issue is explored below.

![Results from GTREID validation set in on sequence MOT17-13. Images are from frames 493, 497, 500, 507 when viewed from left to right.](image)

Figure 6.1 demonstrates an issue with the GTREID tracker when tracking pedestrians. In this sequence, the camera is attached to a bus which is making a right-hand turn while the two individuals are walking towards the left of the frame. In the first frame, the detector successfully identifies the individual in front but the person behind the trashcan has yet to be identified. In the second frame, the bounding box localizes between the two individuals, encompassing both of them in a single bounding box. This bounding box has a larger percentage of the man in the black jacket than the first frame which is then sent through LABNet, causing the third frame to switch the red box to the man in the black jacket. In the fourth frame, the identity is switched again as the man in the green jacket reappears the same way that he did in the first frame.
Figure 6.2: Results from FairMOT validation set on sequence MOT17-13. Images are from frames 493, 497, 500, 507 when viewed from left to right.

Figure 6.2 shows the same set of frames as Figure 6.1 while showing the better capabilities of the center point appearance feature during times of larger overlap between subjects. The crucial difference is the earlier identification of the two independent subjects in frame two as opposed to the single merged bounding box. In this case, the tracker is able to localize more efficiently to both men. In addition to this, the identity feature is only captured from the central point of the bounding box so it is not influenced by the slight overlap in their positioning, it is able to get an accurate feature from just their jackets which is more distinguishable in each of the following frames.

A reason for the poor bounding box performance in Figure 6.1 is due to the additional data augmentations which may have caused the bounding box localization to be less accurate in situations such as this. The data augmentation caused the detector to generalize the two individuals to just one and encompassed them in a larger bounding box rather than two smaller ones. This localization issue also causes the edge weights in the graph network to be affected as the bounding box from the previous frame is closer to the man in the black shirt in the current frame, causing him to be weighted higher. Once the two men fully appear from the trash can, the appearance features can weigh more, and the identities are switched back to the way they originally were. Situations such as this happen frequently in the MOT17 dataset, resulting in a larger number of identity switches than is seen in the baseline model.
6.4. Aerial Tracking Performance

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Illumination</th>
<th>Camera Height</th>
<th>Camera Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Night</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Sunny</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Fog</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Low</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>High</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Bird</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Side</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Front</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>GTREID</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>FairMOT</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MDP</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.1: Breakdown of network performance on UAVDT Benchmark per attribute based on MOTA score per sequence.

Table 6.1 details which models perform the best per attribute based on the MOTA scores on the UAVDT Benchmark. GTREID performs better than the other top scoring models in six of the nine labeled attributes and ties for best in the remaining three. As seen in the results, the less than idea illumination conditions favor more heavily towards GTREID due to its superior appearance features and data augmentations which make it more adaptable to situations that are not sunny. GTREID also outperforms the others in side and front view scenarios. This is helped in large part to the initial LABNet training which took place on a dataset that features vehicles imaged from a slightly elevated position, looking at the front and sides of vehicles. Examples of GTREID’s performance improvements are detailed below.
Figure 6.3 shows a cropped example of Sequence 1303 in the UAVDT testing set. In this example, the full effect of GTREID’s upgrades can be seen. In the bottom left of the frame, it can be seen that MDP suffers from an identity switch as the purple yellow bounding boxes change vehicles as they drive. This is due to their heavy reliance on optical flow to perform their tracking and less utilization on the appearance feature. FairMOT and GTREID are able to correctly track the object throughout the frames due to the usage of the appearance model and Kalman Filter together. GTREID is able to differentiate itself from FairMOT in situations such as those in the top of the frame. GTREID is the only model that is able to fully track the two overlapping black cars as well as the black car that is exiting the left hand side of the screen behind the light pole. This is due to the utilization of the data augmentation and graph association assisting to track localize the vehicles even though they are occluded. The graph network is then able to share the bounding box location and appearance features in order to maintain the tracks despite the overlap.
Figure 6.4 shows another example of GTREID’s performance upgrades from earlier in the sequence. In this example, the two vehicles maintain the same identities in both frames despite heavy occlusion and bounding box overlap, but they are also properly detected in both frames. The FRCNN detection from MDP has a slightly better localization around the car but the CenterNet based detection from GTREID localizes more tightly around both vehicles than the FairMOT detections.
GTREID also performs best in fog compared to the other two trackers. As seen in Fig 6.5, GTREID and MDP are able to properly track the right most car at the top of the intersection, however, MDP has two bounding boxes covering the same vehicle whereas GTREID maintains just one. In addition, the motion model and better bounding box localization of GTREID ensures that the boxes surrounding each vehicle stay tight. Situations such as this are more difficult for the anchor based object detectors to work with as the non-maximal suppression systems are unable to maintain a close localization on the vehicle as it jumps between anchor boxes. This, combined with the fog make it more difficult for the appearance features of the other networks to keep up with the rapidly changing conditions. GTREID is able to maintain a track due to the multi-scale appearance features that come from the combination network, along with the graph
network sharing spatial location information and appearance features during association. As the drone elevates higher, the detected vehicles are separated by a smaller pixel distance, therefore they become more connected, and more information is shared between them, causing fewer identity switches.
Chapter 7

Conclusion

Graph Neural Networks are beginning to be applied in multi-object tracking applications with greater success than traditional MOT trackers. This is due to the ability to learn assignment characteristics during training, allowing the model to learn the right balance of location characteristics as well as appearance vectors. The hypothesis of combining multiple appearance feature vectors into a single feature performed well for aerial vehicle tracking. The center-point location and appearance vectors allowed the most accurate bounding boxes and feature vectors to be pulled from each image. The addition of data augmentations aided in reducing the number of weak bounding boxes that were sampled. The occlusion training allows the tracker to maintain a clear detection as the objects go under occlusion or camera resolution change due to weather conditions. The utilization of the triplet loss aided in the reduction of identity switches on large scale datasets in which a large number of independent identities could be sampled. The addition of the graph association network has resulted in better performance on datasets where the overlap of tracked objects is less common, but the change in resolution and scale is frequent. For this reason, the G(3)REID tracker developed for this thesis is able to tackle situations in which the tracked objects experience illumination and scaled change during a sequence.
Bibliography


[25] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking, 2020. 12, 22, 54


68


Rochester Institute of Technology. Research computing services, 2019. 9, 10, 46, 56


Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Tracklets predicting based adaptive graph tracking, 2020. 12, 21, 24, 54


Abu Md Niamul Taufique and Andreas Savakis. Labnet: Local graph aggregation network with class balanced loss for vehicle re-identification. 2020. 6, 22, 28, 29, 39, 52

Abu Md Niamul Taufique, Andreas Savakis, Michael Braun, Daniel Kubacki, Ethan Dell, Lei Qian, and Sean M. O’Rourke. Siamreid: Confuser aware siamese tracker with re-identification feature, 2021. 19


[53] Jean yves Bougnuet. Pyramidal implementation of the lucas kanade feature tracker, 2000. 21


