COSC 6373: Computer Vision

Title:

UAV Tracking for Smart Cities

Spring 2022

Team ID:

LSV

Darshan Lakhankiya

Sneha Seenuvasavarathan

Vedant Vohra

Mentor:

Charles Manick Livermore

Dr. Ioannis A. Kakadiaris

May 10, 2022

TABLE OF CONTENTS

1. ABSTARCT

Pedestrian identification has become increasingly important, particularly to enable smart cities. While mounted cameras exist in many cities, use of unmanned aerial vehicles (UAV), such as drones, is becoming increasingly common. The first step to enabling use of video footage, from these types of elevated platforms, is to perform pedestrian detection. This step will then enable other tasks such as pedestrian tracking. For example, imagine a situation in which a pedestrian being hit by a car could be detected, so that emergency services could be automatically dispatched to the scene.

The task is to perform person re-identification on the PDESTRE [1] dataset. We implement object detection using Faster RCNN [2] and re-identification using the Siamese network [3] with Inception V3 [4] as the base model. We also implement two semi-novelties: (i) Storing and using the feature vector of images in the re-identification task; and (ii) Storing the recent frame image of a person detected instead of their first occurrence.

This report describes a two-part pipeline for performing pedestrian detection and re-identification using video footage captured by a UAV. The research, methodology, experiments, and observations have been outlined.

Keywords: Unmanned aerial vehicle, pedestrian detection, video surveillance, person reidentification.

2. INTRODUCTION

This project addresses two related challenges:

(i) Pedestrian Detection

This refers to the detection of unique individuals, relative to other pedestrians in the scene.

(ii) Pedestrian Re-Identification

Specifically, once a unique pedestrian has been detected once, then lost temporarily, the methods must successfully identify them again as the same individual.

There are 2 primary types of occlusions that pose a challenge for re-identification.

(a) person-person occlusion
Here, the person being tracked is occluded by another person. In this case, both individuals must be detected and re-identified correctly post-occlusion.

Fig 1. Person-person occlusion [5]

(b) person-object occlusion
Here, the person being tracked is momentarily occluded by an object in the scene. In this case, the person must be re-identified and continued to be tracked post-occlusion.

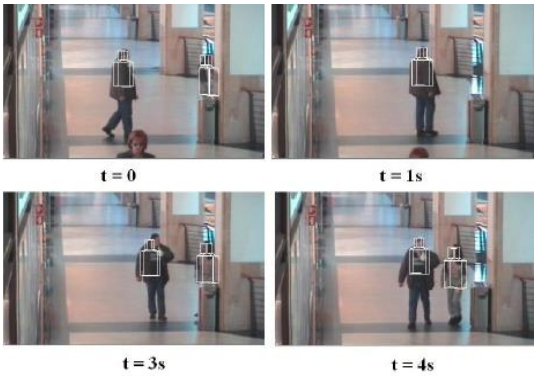


Fig 2. Person-object occlusion [6]

The main objective of this project is to correctly re-identify individuals, regardless of such momentary occlusions.

3. THE DATASET

P-DESTRE (Pedestrian Detection, Tracking, and Re-Identification) is a collaborative effort between research groups at two institutions in University of Beira Interior in Portugal and JSS Science and Technology University in India to promote future breakthroughs in video/UAV-based pedestrian analysis.

The dataset consists of 75 video files with a resolution of 3840 x 2160 @ 30 FPS. These videos were captured by a human-controlled "DJI phantom 4" drone in an outdoor location, flying at a height of 5.5 to 6.7 meters, with pitch angles between 45º and 90º degrees.

Fig 3. An illustration and an example of the data collection method [1]

The dataset also includes bounding boxes, cropped images and annotations for 269 individual volunteers detected in the footage.

Volunteers:

Age	Between 18 to 24 (more than 90%)
Gender	175 (65%) – Male 94 (35%) – Female
Accessories	28% with glasses 10% with sunglasses
Ethnicity	White and Indian

Table 1. A summary of volunteer information [1]

The dataset is fully annotated at the frame level. 25 different attributes are provided. All annotations are stored in a text file with each row in this file providing the information of one corresponding bounding box.

Attributes	Values
Frame	0,1,2,...
ID	-1: 'Unknown' | 1,2,...
Bounding Box	x, y, h, w (Top left column, Top left row, height, weight)
Head Pose	flag, yaw, pitch, roll
Gender	0: Male, 1: Female, 2: Unknown
Age	0: 0-11, 1: 12-17, 2: 18-24, 3: 25-34, 4: 35-44, 5: 45-54, 6: 55-64, 7: >65, 8: Unknown
Height	0: Child, 1: Short, 2: Medium, 3: Tall, 4: Unknown
Weight	0: Thin, 1: Medium, 2: Fat, 3: Unknown
Ethnicity	0: White, 1: Black, 2: Asian, 3: Indian, 4: Unknown
Hair Color	0: Black, 1: Brown, 2: White, 3: Red, 4: Gray, 5: Occluded, 6: Unknown
Hairstyle	0: Bald, 1: Short, 2: Medium, 3: Long, 4: Horse Tail, 5: Unknown
Beard	0: Yes, 1: No, 2: Unknown
Moustache	0: Yes, 1: No, 2: Unknown
Glasses	0: Normal glass, 1: Sun glass, 2: No, 3: Unknown
Head Accessories	0: Hat, 1: Scarf, 2: Neckless, 3: Cannot see, 4: Unknown
Upper Body Clothing	0: T Shirt, 1: Blouse, 2: Sweater, 3: Coat, 4: Bikini, 5: Naked, 6: Dress, 7: Uniform, 8: Shirt, 9: Suit, 10: Hoodie, 11: Cardigan, 12: Unknown
Lower Body Clothing	0: Jeans, 1: Leggins, 2: Pants, 3: Shorts, 4: Skirt, 5: Bikini, 6: Dress, 7: Uniform, 8: Suit, 9: Unknown
Feet	0: Sport Shoe, 1: Classic Shoe, 2: High Heels, 3: Boots, 4: Sandal, 5: Nothing, 6: Unknown
Accessories	0: Bag, 1: Backpack Bag, 2: Rolling Bag, 3: Umbrella, 4: Sport Bag, 5: Market Bag, 6: Nothing, 7: Unknown
Action	0: Walking, 1: Running, 2: Standing, 3: Sitting, 4: Cycling, 5: Exercising, 6: Petting, 7: Talking over the Phone, 8: Leaving Bag, 9: Fall, 10: Fighting, 11: Dating, 12: Offending, 13: Trading

Table 2. Annotation protocol for the P-DESTRE dataset [1]

In this project, we have decided to use the first 3 attributes (Frame, ID, Bounding Box), since in a real-world scenario, we are unlikely to be able to generate the remaining attributes automatically.

4. METHODOLOGY

The workflow is divided into 2 stages, (a) person detection and (b) person re-identification.

(a) Person Detection

The objective of this stage is to detect the unique individuals in the video and assign them labels and generate an intermediate dataset that can be used by the re-identification model.

The following steps are involved in this process:

1. Detect pedestrians in each frame, generate bounding boxes and assign them numerical labels (ID)

2. Crop images of pedestrians based on the bounding boxes and store them.
3. Repeat 1 and 2 for subsequent frames.
4. Pairs of images are generated by taking the first frame of an individual and each subsequent detection. These pairs are passed on to the re-identification model to measure similarity.

Model

We use faster RCNN to perform object detection on the video frames. Faster R-CNN implements an object detection algorithm in which the network itself learns the region proposals. It consists of 3 parts. The first part is the convolution layer, where filters are trained to extract features from an image. The second part is the Region Proposal Network that sliding a small window over the feature map of the last convolution layer, determine the presence of an object and predict bounding boxes of the predictions. The last part is a fully connected layer that takes the region proposals as input and predicts its classification and bounding boxes. [2]

Model Architecture

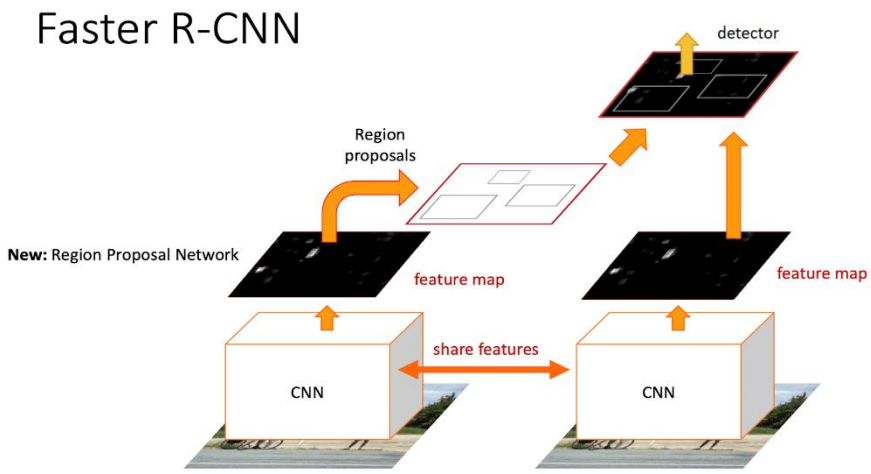


Fig. 4. Architecture of Faster R-CNN [7]

Implementation

- The object detection model returns 35,000 objects for an image.
- We take the top 200 objects.
- Out of the 200 objects, we filter people, and then filter down to 20 people per image/frame.
- In order to filter out unnecessary/erroneous/overlapping boundaries, we use a technique called Non-max suppression that selects the best bounding box

out the set of bounding boxes returned by the object detection model, for the image, using a metric called Intersection over Union(IOU) [8].

- We only consider a maximum of 20 people per frame with detection threshold = 0.6 and IoU (threshold = 0.5)
- Detection is deemed to be successful if the predicted probability > 0.5 and the IOU > the 0.5.

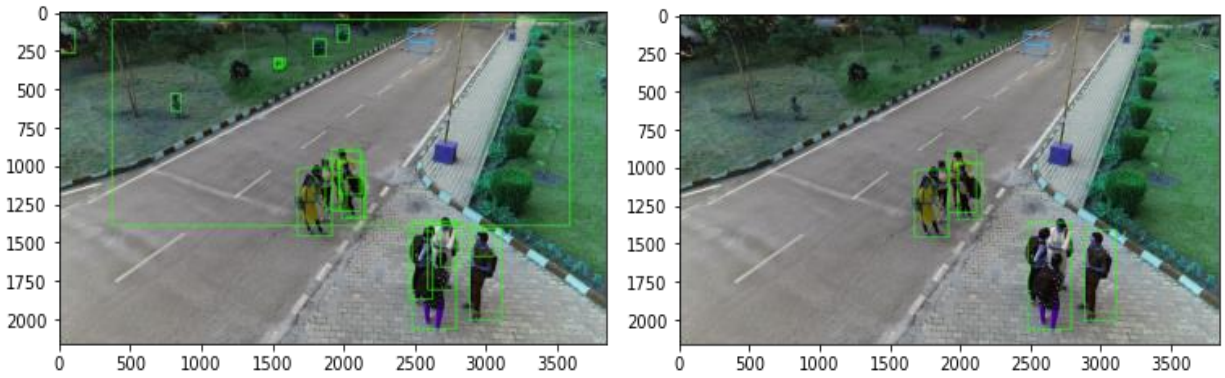


Fig. 5. Frames Without IoU vs with IoU

(b) Person Re-Identification

In this stage, we take the image pairs generated by the detection model and determine if the two images in each pair contain the same person or not, by estimating the similarity between them. Our re-identification model is based on the Inception V3 CNN and is trained via transfer learning. The scope of the project is re-identification within the video.

Base Model – Inception V3

Deep layers of convolution results in overfitting of the data. In order to combat this, inception V3[4] employs multiple filters of varying sizes at the same layer, thereby making the model wider than deeper.

The inception V3 network largely reduces the number of parameters in the network by breaking down convolutions with large filters into smaller convolutions. The network also can replace nxn convolution layers with 1xn and nx1 layers, respectively. This gives a significant improvement in computational cost savings as n grows. Auxiliary classifiers are inserted between the network's layers during training to help with regularization. Grid sizes are also reduced to bring down computational costs.

Siamese Model

- We use a Siamese Neural Network [9] to perform re-identification on the dataset. A Siamese Network is a combination of 2 identical networks (in this case, CNNs) which are trained parallelly on 2 separate images. At the end, a distance function

is applied on the output of both networks to calculate the similarity between the two inputs.

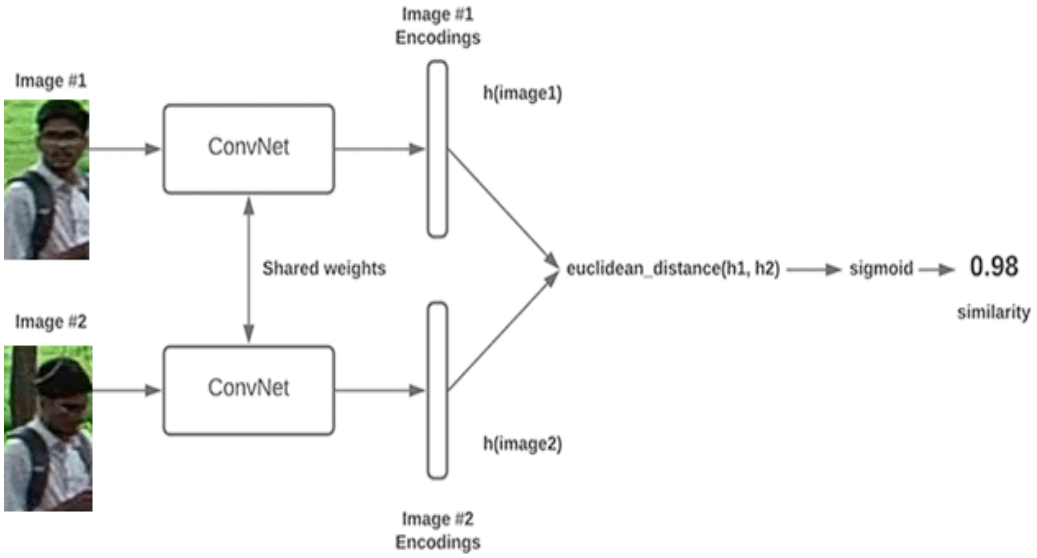


Fig 6. Example of a Siamese CNN [9]

- Training data generation
  - The dataset has 75 videos, and each video has a different number of people.
  - The number of images per person is also different.
  - We decided to take 20 different people from each video and 100 images per person for training.
  - Each image is part of two pairs - the same person pair and different person pair.
  - The different person pair is generated at random.
  - The same person pair was labeled as 0 and different person pair was labeled as 1.
  - Our data generator generates 4000 pairs per video and 300,000 pairs in total (75 videos).

- Model design
  - We employ transfer learning using Inception V3 as the base model, pre-trained on the ImageNet dataset. [10]
  - The image pairs are converted to feature vectors (encodings).
  - We fine-tune the model on our dataset.

Model Architecture

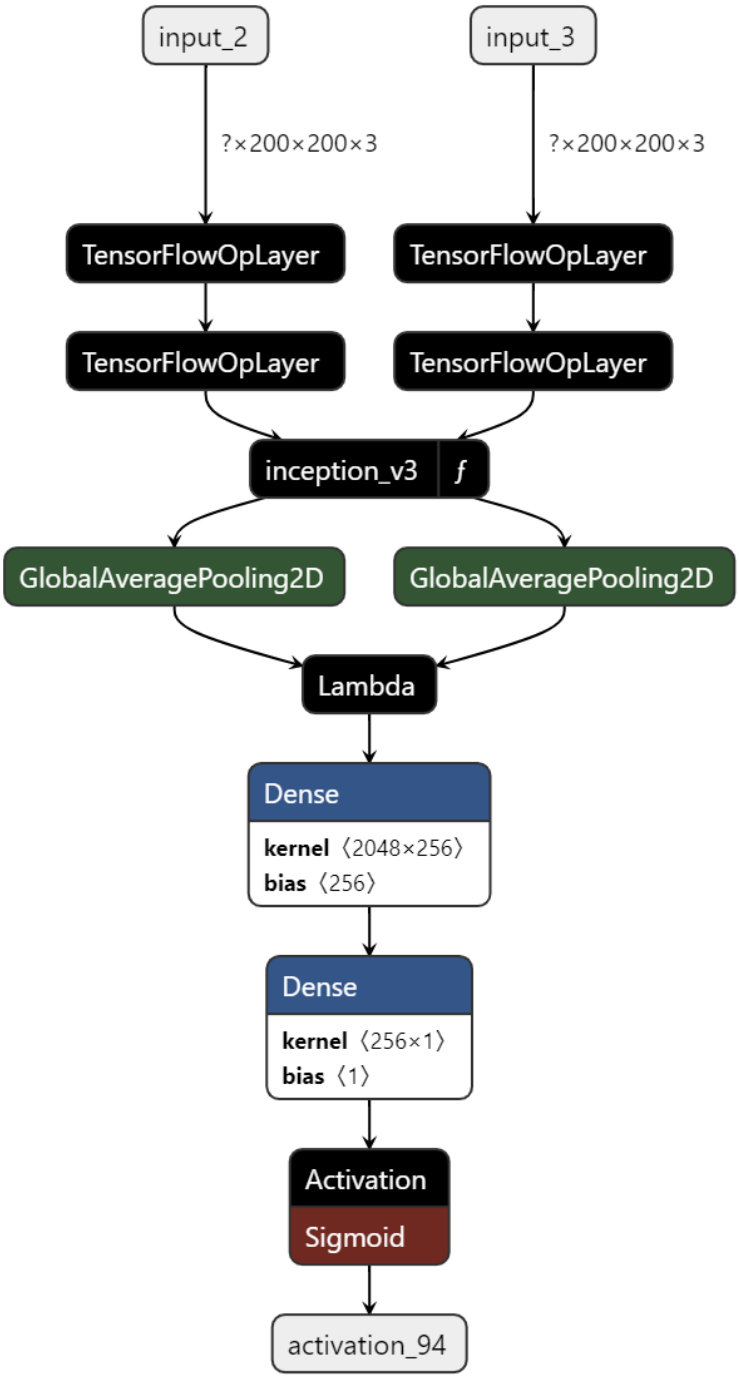


Fig 7. Siamese Network Architecture

- Input: 2 images with shape (200, 200, 3).
- Base model: Inception v3.
- Distance function: Lambda.

The model takes two input images of size 200x200 with 3 channels, followed by two TensorFlow Operation layers for each image, that perform computations with the input

tensors and returns zero or more tensors. This output from both the layers are then fed into the Inception V3 model, which returns a 4x4x2048 output. Global average pooling is performed on the 2 feature vectors(1x2048) of the images, and Euclidian distance between the feature vectors are computed(1x2048). The first dense layer, Dense(256), determines the relevance of a feature, while the last layer Dense(1) determines the probability scores of membership to the specific class, in our case, it is 'similar' or 'dissimilar'. The activation function used here is sigmoid, which transforms its input to fall between 0 and 1.

- Challenges and adjustments
    - The videos have a framerate of 30 FPS.
    - Person detection and re-identification on every frame is resource-intensive and time consuming.
    - To combat this, we take only 1 frame every second and use it for re-identification.
    - For each frame, we compare the new detections with a list of already detected pedestrians.
    - If we find a match based on a the detection threshold(0.5), then we apply the same label (ID) to the detected person. Otherwise, we assign them a new ID.

5. EVALUATION

We have evaluated the performance of our Re-Identification model and recorded the results both before and after tuning. (training/validation split: 80/20).

The training configuration used was:

- Loss function: Binary Crossentropy
- Optimizer: Adam
- Metric: Binary Accuracy
- Epochs: 15

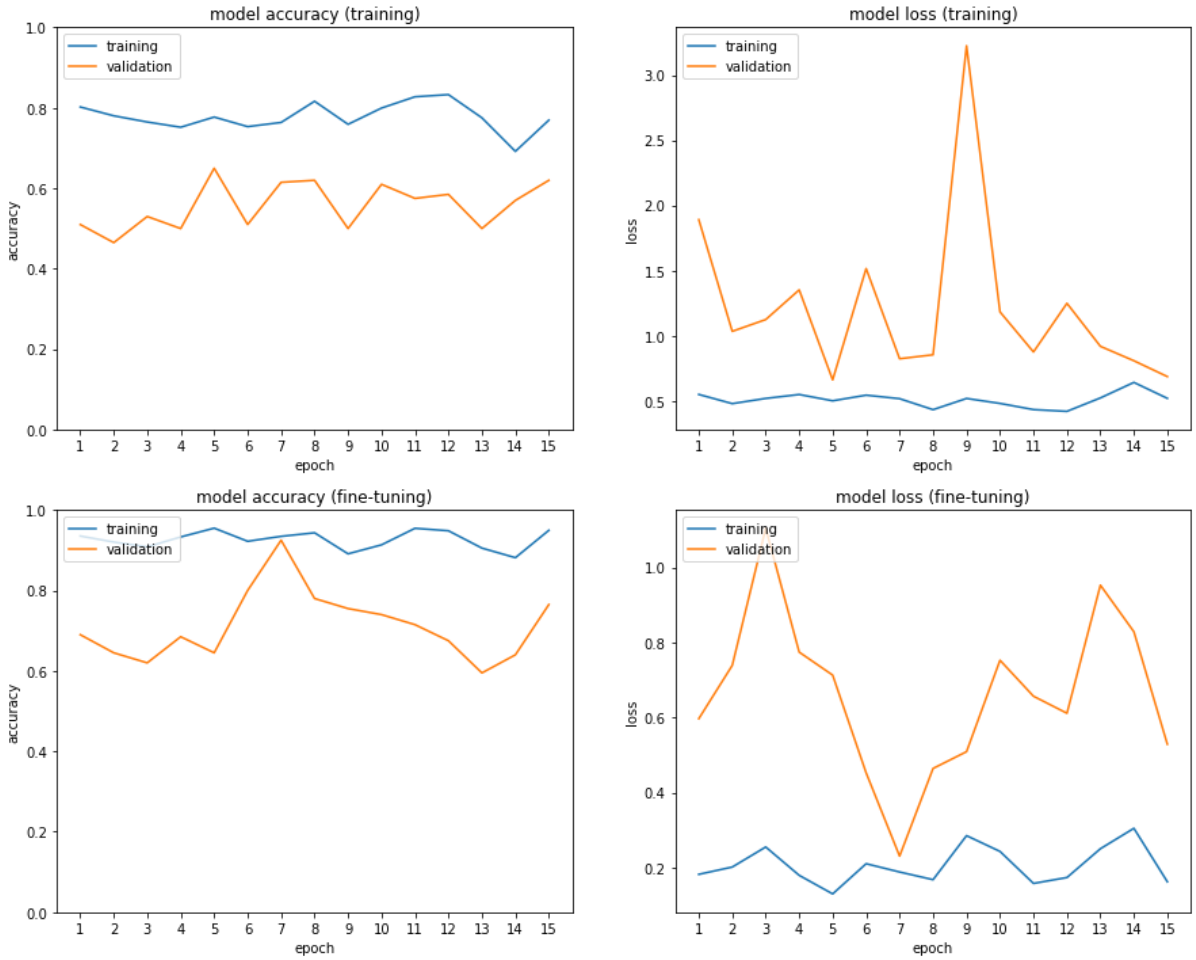


Fig 8. Training and validation graphs

The top graphs are from training the model with the base layers frozen. On the left, we can see the model accuracy for training was around 80%, whereas for validation data, it was hovering around the 60% mark with a slight upward trend. On the right, the training and validation loss have a slight downward trend, indicating that the model is not overfitting.

The bottom graphs are from the fine-tuning process, where the base layers were made trainable and trained on our dataset. On the left, we can see that the training accuracy is over 90% and the validation accuracy has gone up from 60% to around 76%. On the right, the validation loss is around 66% lower than before and trending downwards.

Testing:

We used 20% of the training data to test the model. The binary accuracy achieved on our test dataset was 72.5%

Below are the training, validation, and testing accuracies of our model:

	Pre-tuning accuracy	Post-tuning accuracy
Train	76.9%	94.9%
Validation	64.5%	78%
Test	57.4%	72.5%

Table 3. Train, validation, test accuracies

Re-Identification accuracy:

We tested the re-identification model by taking 1 video from each of the 75 folders with varying number of video files. In these videos, we considered up to 10 frames per video, taking 1 frame per second. The accuracy obtained was 69.233%. We have evaluated this accuracy with the annotations provided object detection and our re-id model. We were unable to assess the accuracy of our object detection model against annotations because, annotation provided identities are different from our object detection model's identities.

6. POTENTIAL IMPROVEMENTS

We came up with 2 semi-novel theories for improving the performance and scalability of our model. They are listed below, along with some of the challenges and observations from our experimentation with these theories.

1. **Check with most recent frame containing a match**
   We theorized that instead of always checking with the first image of a person, if we check with the most recent frame that has been labeled as them, we could get **better accuracy**, as the new image is more likely to be similar to recent frame when compared to the first image.

   A challenge with this approach – toppling accuracy:
   Our re-id model's accuracy is 69.233%. When using the most recently re-identified frame, there is a chance that this image could be of a totally different person. In this scenario, the accuracy for the subsequent frame lowered to 58.731%.

Possible ways to remedy this:

There are 2 modifications we theorize could help counter this issue:

1. Adding a positional factor:

   Record the position of each detected bounding box, relative to the entire frame and then filter out the cases where the tracking window jumps from, say, (50, 50) to (600, 600) in the very next frame.

2. Apply a threshold on the similarity:

   Only considering matches with a similarity over a certain threshold (90%) as the 'most recent frame'

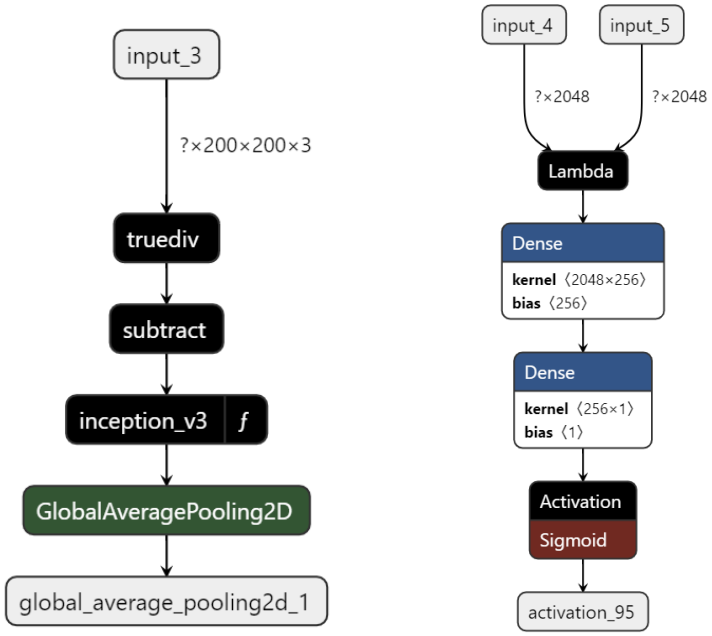
This is something we will consider implementing in the future.

2. Store feature vectors instead of entire images

To bring down the time taken to perform re-identification and reduce storage costs, we could store the feature vector of already re-identified people instead of their image, as they were already processed by the Siamese model. This feature vector could be sent for comparison to the Siamese model to predict similarity with a newly detected pedestrian's image.

Only storing the feature vectors from the base model, rather than the entire image itself should yield better **model performance** and lower resource utilization, since we would only have to run one image through the base model instead of two.

Model Architecture:



(a) Image to Feature Vector.CNN *(b) Image-Feature vector Siamese*

Fig 9. Image-Feature vector Siamese Model

The initial Siamese model was split into 2 parts. Fig 9(a) represents the first part which takes the image input of a newly identified pedestrian and returns the feature vector of this image. This new pedestrian feature vector and feature vectors from the list of already identified pedestrians are sent to the Siamese model in Fig 9(b), the second part, to find out the similarity.

We performed the re-id task on the first 5 video files using the Siamese re-id approach and by storing the feature vector of already re-identified pedestrians. The two approaches gave the following results

Time taken by image-image siamese network 315.0s
Time taken by image-feature vector siamese network 169.694s

From the above observations, we can see that the feature vector approach is 46.35% more efficient than the image-image Siamese re-id method. However, when combined with detection, the relative speed improvement gets less significant, since detection takes significantly more time than the re-identification model. In the case where detection is already done and only re-identification needs to be done, this technique would actually make a significant difference to the processing time.

6. TEST CONTAINER

We have added a test script as a part of our project deliverables which performs object detection, re-identification, re-identification evaluation and the semi-novelties on the first 10 frames of a video. The re-identification accuracy obtained on this test sample is 74.074%. The accuracy of re-identification using the most recent frame method drops to 59.259 as expected. The time taken to perform re-identification using the image-image vector method is 19.24 seconds while the image-feature vector method takes only 11.87 seconds. This dip in computation time confirms the better performance of the image-feature vector method.

7. CONCLUSION

We implemented person re-identification using Faster RCNN for object detection and the Siamese Network for re-identification. We obtained a test accuracy of 72.5% with our re-identification model. We implemented the two semi novelties we observed: (i) Store the most recent image of an already detected pedestrian instead of their first frame occurrence – We observed that this resulted in decrease in performance by ~ 11%, and also identified possible solutions to increase the accuracy while using this method; (ii) Store the feature vector of an already detected pedestrian instead of their image-resulted in increase in re-identification processing time (46.35% for 5 frames).

8. CONTRIBUTION

Name	Contribution	Time (hours)
Darshan Lakhankiya	Evaluation of models; Data generator; Project report	45
Sneha Seenuvasavarathan	Siamese model; Data generator; Object detection; Re-identification; Semi-novelties	52
Vedant Vohra	Siamese model, Fine-tuning and evaluation; Semi-Novelties, Containerization, Readme, Project Report	50
Total Time		147

Table 4. Team contribution

9. REFERENCES

[1] S.V. Aruna Kumar, Ehsan Yaghoubi, Abhijit Das, B.S. Harish and Hugo Proença. The P-DESTRE: A Fully Annotated Dataset for Pedestrian Detection, Tracking, Re-Identification and Search from Aerial Devices. IEEE Transactions on Information Forensics and Security,doi: 10.1109/TIFS.2020.3040881, 2020.

[2] Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett (eds.), NIPS (p./pp. 91-99),

[3] Siamese neural network – Wikipedia https://en.wikipedia.org/wiki/Siamese_neural_network

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[5] Figure 1, Dubrofsky, Elan. (2011). Using Reasoning to Aid With Tracking Objects in Video.

[6] Figure 2, Saboune, Jamal & Laganiere, Robert. (2009). People detection and tracking using the Explorative Particle Filtering. 1298-1305. 10.1109/ICCVW.2009.5457459.

[7] Figure 4, Object Detection II
https://www.crcv.ucf.edu/wp-content/uploads/2018/11/Lecture-14-Object-Detection-II.pdf
[8] Intersection over Union (IOU)
https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef
[9] Siamese networks with Keras, TensorFlow, and Deep Learning
https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/
[10] ImageNet Dataset
https://www.image-net.org/
[11] Convolutional neural network – Wikipedia
https://en.wikipedia.org/wiki/Convolutional_neural_network
[12] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.