Individual Project:

Multi-Camera Person Searching

Thanh Vu

CS420: Artificial Intelligence

Lafayette College

December 9, 2017

— Submitted to

Prof. Chun Wai Liew

A. Introduction

In the spring of 2017, Lafayette College started to carry out its expansion plan, aiming to double the number of students, from over 2000 to over 4000. As a part of this plan, the school wants to strengthen its security to better project the students. Despite the school being located in the peaceful College Hill neighborhood of Easton, Pa, on-campus criminal activities such as robberies or burglaries might still occur from time to time. In such events, it is crucial to be able to identify and keep track of the suspects using different cameras available on campus.

We have been contacted by Public Safety and Information Technology Services (ITS) to develop a software that can be used to assist and control these incidents. In this project, we designed a system that, given an image of a person X and access to all on-campus cameras, can automatically search for X across different camera views, i.e., identifying the same person at a different time and/or a different place. It is worth to note that this technology is meant to be used as an assistance to post-incident tasks, including immediate search for the suspect or a person of interest, not to predict or identify if a criminal incident or abnormal event might happen.

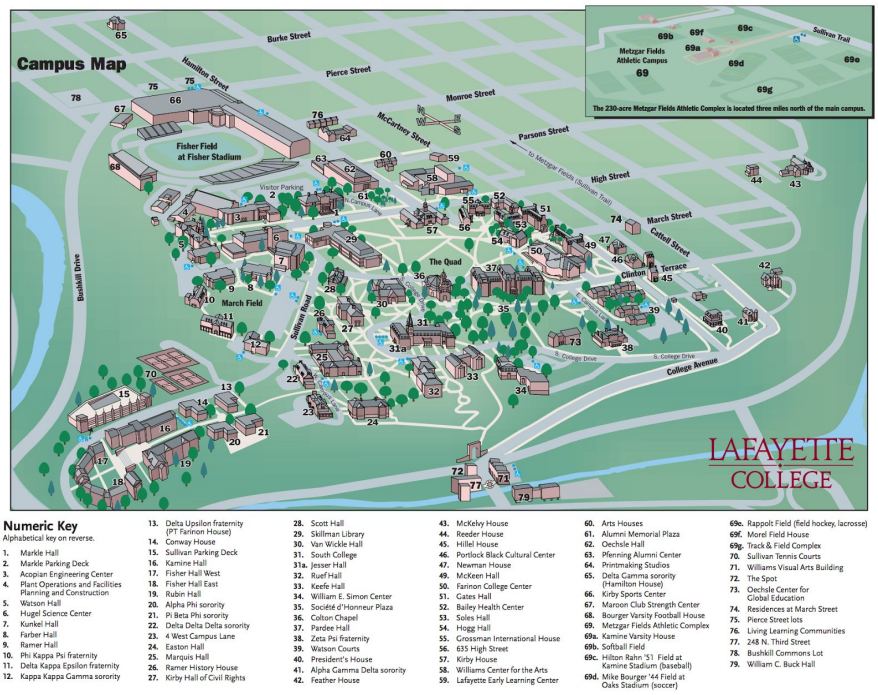


Figure 1a: Campus Map of Lafayette College.

Figure 1b: Google Satellite view of Lafayette College.

Motivation	• *Lafayette College wants to strengthen its security to account for on-campus criminal incidents such as robberies*
Project goal	• *given an image of a person X* • *automatically search for X across multiple camera views*

Listing 1: Project's motivation and goal

B. Parameters

I. Input

This technology will be used directly by Public Safety, with the assistance of Information Technology Services. At the time of operation, the system should have access to a set of video frames from available on-campus cameras. These images should be frames retrieved directly

from on-campus cameras in real-time Each camera frame should have metadata specifying what camera it comes from, where the camera is, what is the source video/clip and the index of the frame in that video, etc. The image set will be used as the search pool to detect the person of interest and return appropriate information about location and time.

The second input needed is an image of the targeted person to query. This input is vital because the system will extract visual clues from this source image and use them to identify the same individual in the image search pool. In order to get better detection, the input image of the target should be a full-body image with the background cropped out. Note that it does not need to be of high resolution, but recognizable. A good intuition for recognizable images is that if you were to put two full-body images of the same person, wearing the same outfit, but with different body poses or backgrounds, together, a human judge should be able to recognize that they depicted the same person. Examples of possible query images include cropped images from the system's camera frames (Figure 2a) or from pictures captured by pedestrians (Figure 2b).

(a) input images from security camera (b) input images captured by pedestrians

Figure 2: Examples of input query images

Input

- *A gallery of image G sequences of camera frames, with each frame's metatdata includes*
  - *Frame index*
  - *Camera index*
  - *Camera location*
  - *Other metadata*
- *An image of person X*

Listing 2: Inputs

II. Knowledge

As mentioned in Input section, the images (camera frames) of the search pool will be retrieved from on-campus cameras. As our system searches for the targeted individual using solely visual clues, other than the query image and access to camera frames, no other data or external knowledge is required from the user at the time operation. However, in order for the model to learn to identify and analyze these visual clues, a large number of images are required for training it. Fortunately, several big datasets have been made publicly available by researchers over the years. These include datasets such as PRW (Person Re-identification in the Wild) [1], with almost 12,000 images and 900 IDs (unique pedestrians) and LSPS (Large-Scale Person Search) [2], with over 18,000 video frames and 8,400 IDs. We utilize these datasets as training examples for our model to learn and improve on the task of person search.

Dataset	LSPS	PRW
#frames	18,184	11,816
#ID	8,432	932
#annotated bbox	99,809	34,304
#box per ID	11.8	36.8
#gallery box	50-200k	50-200k
#camera	-	6
Evaluation	CMC&mAP	CMC&mAP

Figure 3: Large datasets available for training the system

III. Output

At the low level, the outputs of the search are different images of the same person as the task that we are trying to solve is essentially retrieving images (camera frames) or the same person depicted in the given image. At a higher level, the outputs that users see are these result images along relevant identification information (metadata) of that image, such as frame index, camera index, location, time, etc. Thus, the end outputs are potential occurrences of the targeted individual at a different time and/or location, captured by different cameras on campus.

Output

- *A list of potential occurrences of person X in gallery G identify the suspect at a different location and/or time*
- *Each occurrence includes*
    - *Video frame (image) that contains X*
    - *Frame index*
    - *Camera index*
    - *Camera location*
    - *Time captured*
    - *Other metadata*

Listing 3: Outputs

IV. Constraints

Several constraints, both inherent to the problem and imposed by either hardware or software limitation, must be considered to appropriately design a solution. The constraints on our system are as follows:

- There are complex visual variations between images capturing the same person at different time and location:
    - Camera viewpoints,
    - Body poses,

    - Illumination (lighting),
    - Occlusions,
    - Background clutter,
    - Image resolution,
    - Etc.
- Cameras often cannot capture clearly people's faces. Thus, the searching algorithm should focus more on the person's clothes and body shape instead

V. Assumptions

Like constraints, assumptions help narrow down and distinguish a specific problem from other versions of it. At the same time, they also reduce the complexity of the problem by reducing use cases and create a more homogenous setup. Assumptions about the environment, the subjects, and baseline capabilities of the system allow for the description of the best possible problem state. The assumptions that define our system and problem are as follows:

- At least one image of the targeted individual is available to query
- The query image is a full-body image of the targeted person, to provide visual clues for searching
- The suspect's appearance remains the same
- The system has access to real-time records of on-campus cameras
- Video frame are linked with necessary information about time and location
- Image frames are of high resolution enough that individuals captured are recognizable

VI. Use cases

The core of this program is a person searching and matching method based on visual features. It can be used to identify a given person of interest within a pool of images. In our context, the main targeted use case is to detect a given suspect of an on-campus criminal activities. Since visual clues are crucial to the algorithm, the program should be used within a short period of time after the incident, so that we can assume the suspect's outfit remains the same. For the same reason, the software can only be used to assist post-incident tasks such as

real-time search for the suspect. Usages such as predicting or identifying a potential criminal incident or abnormal event are out of the scope of the system.

VII. Evaluation Criteria

Evaluation criteria provide the system and context to evaluation the effectiveness of our system. We consider how a variety of factors, such as use cases, time, scalability, etc., are handled by our approach. With each factor, several relevant questions were composed to thoroughly assess the system as well as its the performance compared with other approaches. Our evaluation criteria are as follows:

- Use cases
    - Can the program be used to search for suspects of criminal incidents?
    - Is the program designed strictly for criminal settings? Can it be used for something else?
    - Can the system handle extreme use cases such as when the time range is large or when it is rainy?
- Time
    - Can the system work in real-time?
    - How long is the training time?
- Others
    - Is the system scalable and flexible?
- Overall
    - What are the method's strengths?
    - What are the method's limitations?

C. Algorithms

I. Problem Setup

With distinct characteristics discussed in the section A, including input, output, constraints, assumptions, etc., this problem that Public Safety and ITS are trying to tackle appears to be an instance of what is called *Person Re-Identification* problem, also known as person re-ID problem. In Computer Vision, a person re-ID problem takes the inputs of a gallery of images along with an image of the targeted person. In our case, the gallery is sequences of camera frames and the query image is of the suspect of the criminal incident. The desired outputs for person re-identification are also occurrences of the queried individual in the gallery, i.e., our task is essentially detecting/identifying the same person at different locations and time.

Moreover, this setup does not assume any physical mapping of cameras. Intuitively, one might think that using the correlation of cameras' physical locations could aid the system in narrowing down potential occurrences and reduce the search space. For example, knowing that the criminal incident happened at location L_1 and the query image was captured by camera C_1 near L_1, one might be able to predict that the suspect would likely be heading toward either location L_2 or location L_3, thus, we should check cameras C_2 and C_3 first. However, despite being potentially beneficial, the task of predicting a person's future movements and trajectories itself is very challenging [3, 4]. In our case, it may also require knowledge of criminological psychology and behaviors. Thus, to avoid adding more complexity to our problem, we decided that it is beneficial to assume no physical mapping of cameras and focus on only visual clues.

Similar to a general person re-identification problem, solving our problem requires handling two main tasks: pedestrian detection and identity matching. Unlike the queried image, in which the person of interest has already been cropped out, camera frames at a particular time may or may not include any person. Thus, search for the target, the system needs to first detect

different pedestrians in the scenes. Given these people, the matching algorithm should then be deployed to identify the person of interest among several, potentially hundreds of, detected pedestrians. The two tasks have been widely studied by computer vision researchers, commonly as separate problems, with identity matching itself being referred as person re-identification. Based on existing work, the following section discusses alternative approaches that we have considered to solve our criminal re-ID problem.

II. Alternative Approaches

1. Concatenating Pedestrian Detection and Identity Matching

As pedestrian detection and identity matching have been well-studied separately, an intuitive approach is to combine state-of-the-art methods in both fields to solve our version of the re-ID problem. In terms of pedestrian detection, Zhu and Peng proposed a multi-task model to handle different levels of occlusions [8] and a multi-resolution detection approach for decreasing resolution [9]. In [11], Mao et al. explores the effectiveness of different additional features and proposes HyperLearner architecture to jointly learn the extra features. Recently, proceeding the impressive success of YOLO [10], a state-of-the-art, real-time object detection system, Redmon and Ali proposed various further improvements to the model, making the new version, YOLO9000 [11], also a good candidate for pedestrian detection. For identity matching, which, by itself, have been traditionally referred to as person re-identification, existing techniques can be group into different types, such as image-based re-ID, video-based re-ID, hand-crafted systems, deeply-learned systems, etc. [12] Among these approaches, large-scale deeply-learned systems such as those proposed by [13], [14], and [15], appear to be the most suitable for our problem. However, despite the progress in both pedestrian detection and identity matching, training two types of models separately and then combining them suffer from two disadvantages: (1) having separated models for each subproblem avoids evaluation of the mutual effects such as how errors in detection/tracking affect matching result, and (2) the architecture makes it harder to optimize the system as a whole. To address these drawbacks, recent works have attempted to

solve the problem end-to-end manner - jointly handle both pedestrian detection and identity matching for person re-identification.

2. End-to-end Person Re-ID in the Wild

End-to-end systems are those that take raw video frames and a query image and perform both human detection and identification. To this end, along with their large-scale dataset PRW for person search, Zheng et al. also proposed a cascaded fine-tuning strategy to train the detection model and then the classification model [1]. The authors also presented a method called Confidence Weighted Similarity (CWS) metric to incorporate detection confidence scores in the similarity measurement for identification. Regardless of the promising experimental results and an architecture that concerns with both detection and identification, PRW approach still handles the two tasks separately in a two-phase manner. Instead, the final algorithm that we selected jointly handles both tasks in a single network [0]. Additionally, according to Zheng's experimental results, utilizing annotated pedestrian boxes without IDs as weakly labeled data can help improve re-identification performance. This something that the selected algorithm is able to take advantage of.

III. Final System

Among different end-to-end systems that incorporates both pedestrian detection and identity matching, we select Joint Detection and Identification algorithm [0] because it's unified network architecture jointly handles both problems. Moreover, the algorithm is scalable, was tested on a large dataset, and is able to exploit weakly labeled data to improve its re-id performance. In this paper, we refer to Joint Detection and Identification algorithm as DIID algorithm.

1. Overview of the Model

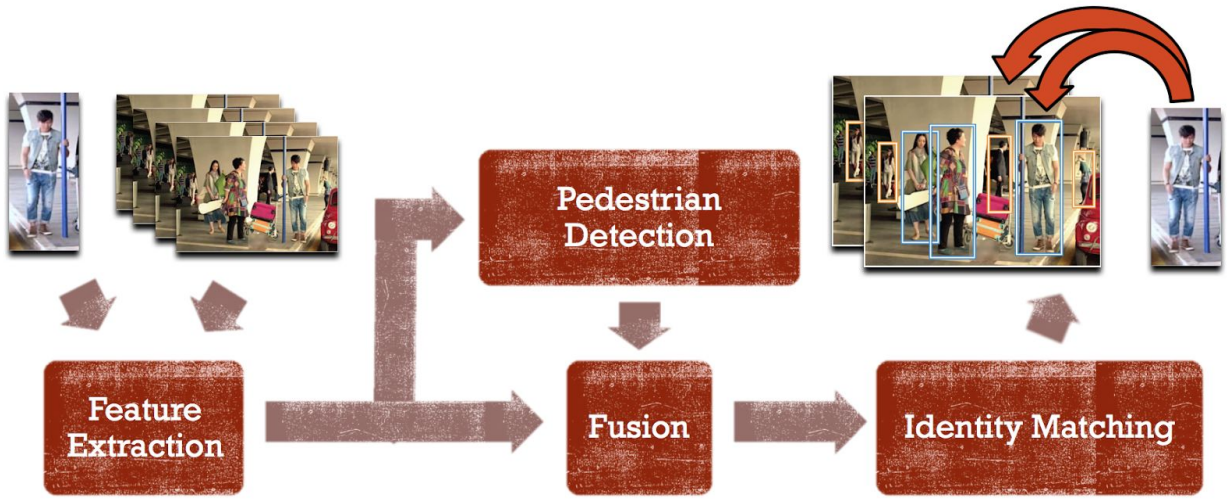


Figure 4a: Overview of the system.

DIID is essentially one big convolution neural network consisting of several components (Figure 4a). Input images, including both query image and raw camera frames, first go through a Feature Extraction module that converts them into feature maps, which are vectors of image features. Given these feature vectors, the second module, Pedestrian Detection, predicts and proposes locations of pedestrians in the scenes. The third component is simply a Fusion layer that combines both the feature maps with pedestrian information to create unified, comprehensive feature vectors for the detected pedestrians. Finally, pedestrian features and the query features are used by Identity Matching module to identify the person of interest (our suspect) if appeared in the scenes. Together, the four components create the full model for person re-identification. We provide visualization for each module in Appendix section.

The underlying implementation is constructed from two deep neural networks (Figure 4b), namely Residual Network (ResNet-50) [16] and Faster R-CNN (RPN + Fast R-CNN) [17], both were developed by Microsoft researchers. In our system, ResNet-50 is divided into two parts. The first few layers of it are used to extract shared image features for both detection and identification. The rest of the network, on the other hand is used for identity matching task. This

division allows the incorporation of Pedestrian Proposal Module, which helps creating the single, unified network that jointly handles pedestrian detection and identification. In the following sections, we discuss the basic of deep convolutional neural networks (CNNs), which is the foundation for both ResNet-50 and Faster R-CNN, and thus, the architecture of DIID itself. Then we provide details on the different components of the system.

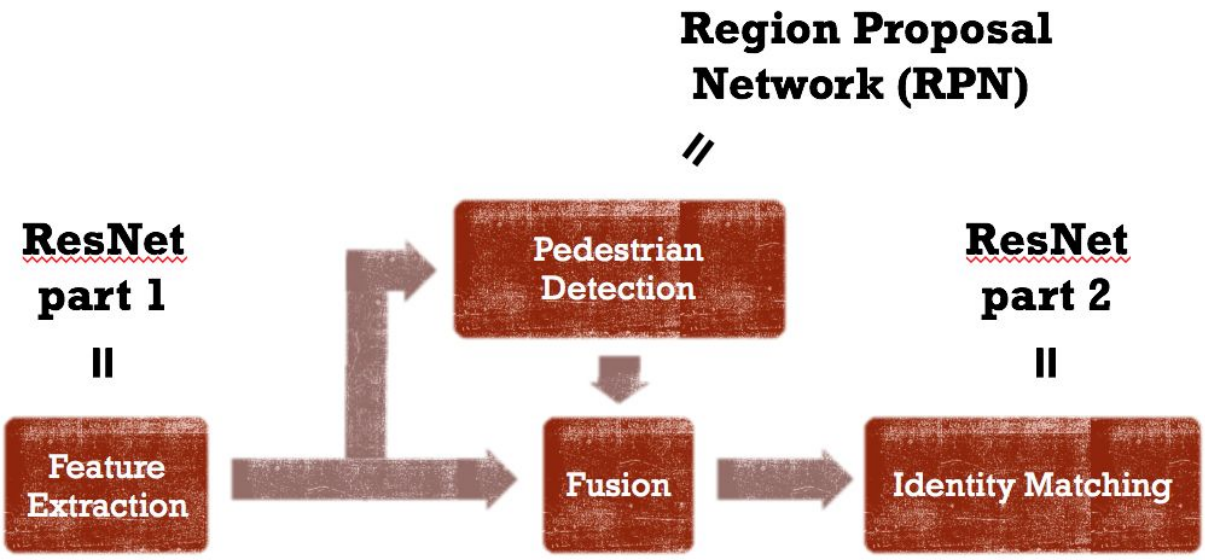


Figure 4b: CNNs underlying different components of the systems.

2. Background Knowledge

Similar to most currently prevalent algorithms for person re-ID, DIID is deep-learning-based. In this section, we present some basic knowledge of how deep neural networks work with Computer Vision problems. A good example to aid the explanation is the task of image classification: given a picture of an object, the network needs to classify this object into the correct class, e.g. “dog,” based on visual features. As simple as this sounds, the task is very challenging. Indeed, recognizing that Figure 5a is of a dog is easy for the average humans. However, for a computer, it is simply a matrix of pixel values as illustrated by Figure 5b. The challenge is enhanced by variations in lighting, perspective, body pose, image resolution, etc. Thus, in order to recognize that there is a dog, the machine needs to cumulatively identify visual features, from low-level components like edges, curves, or corners to higher-level concepts such

as paws or 4 legs. These feature recognition tasks are done by different layers of the neural network, as illustrated by Figure 6.

(a) (b)

Figure 5: Comparison of how the average humans perceive digital images (a) and how computers "see" them.

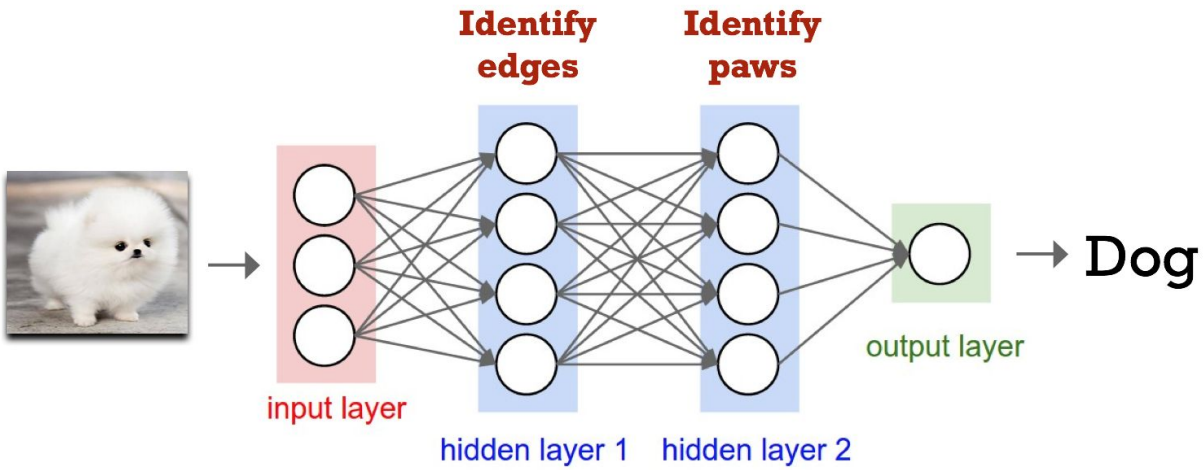


Figure 6: Illustration of different layers of a neural network. This network has an input layer, an output layer, and 2 hidden layers. The dots inside each layer represents different neurons. Nowadays, real-world networks such as ResNet [16] are much deeper, with many more layers.

Different layers in the network recognize different types of features with increasing levels of complexity, from lines, curves, or corners to paws or ears. Particularly, to do this, each neuron

in a layer would look for a variation of the feature, such as short horizontal line, long horizontal line, vertical line, etc. (Figure 7) These neurons can be thought of as filters, which allows only certain features to "go through" to the next layer. When features are found, the corresponding neurons (or filters) would "activated." In Figure 7, we see that first layer's neurons can be activated by edges and stripes; those in the second layer pick up eyes and noses; while neurons in the top layer look at the whole face.

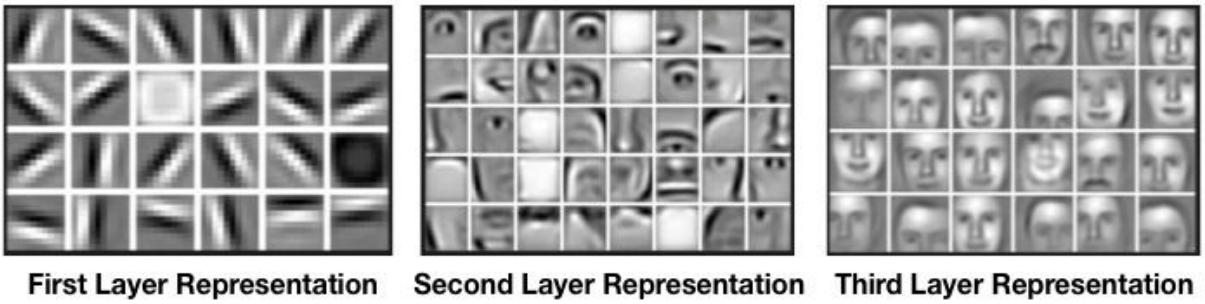


Figure 7: Visualization of feature maps at different layers of a neural network [18].

For neuron to know what types of edges or curves to look for if it's trying to recognize a specific object such as a dog, the deep neural network need to be trained. The idea is to provide the network thousands of example images of different classes such as dog, cat, car, human, etc. For each image, the network would try to recognize different features and make a prediction based on what it finds. Depending on how close (or how far) the prediction is compared to the actual answer known as "ground truth," the network would adjust its neurons to better match the expected outcome. The performance evaluation step is done using a loss function - a function that quantify how bad the network is doing. Thus, the ultimate goal is to adjust the neurons in a way that the overall loss minimized. The adjustment step is known as backpropagation (Figure 8) Backpropagation is done for mini-batches of training images to estimate the true network loss while reduce computational needs. Essentially, evaluating loss function and doing backpropagation are the learning parts of a convolutional neural network.

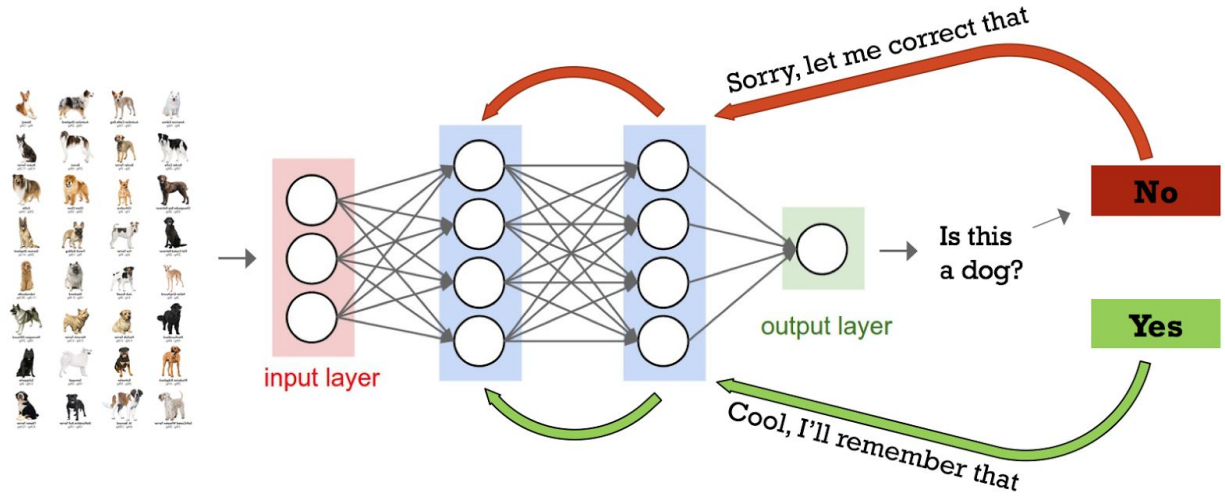


Figure 8: Visualization of training process with backpropagation.

3. ResNet

Microsoft ResNet [16] is an architecture for ultra-deep convolutional neural network - the largest version of it has 152 layers, much deeper than some other well-known networks such as AlexNet (8 layers) [19] or VGG-16 (16 layers) [20]. With ResNet, He et al. achieved 3.6% error rate on the ImageNet test set, surpassing humans at 5-10% error rate, and won ILSVRC 2015 classification task. The key component that separates ResNet from other architecture is the *residual block* (Figure 9) The idea is when training images are fed through convolutional layers, instead of learning a transformation $F(x)$ from x_i to $F(x_i)$, the network would compute a function $D(x)$ of the difference so that $x_i + D(x_i) = F(x_i)$. This way, we preserve the information x_i from previous layers. The authors believe that optimizing the residual mapping is easier than optimizing the traditional, unreferenced mapping. Currently, ResNet is one of the best CNNs architecture available. We follow the design choice of Xiao et al. to adopt ResNet-50 (50 layers) as out system's base CNN model.

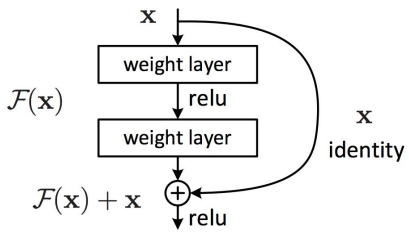


Figure 9: . Residual block - a building block for residual learning

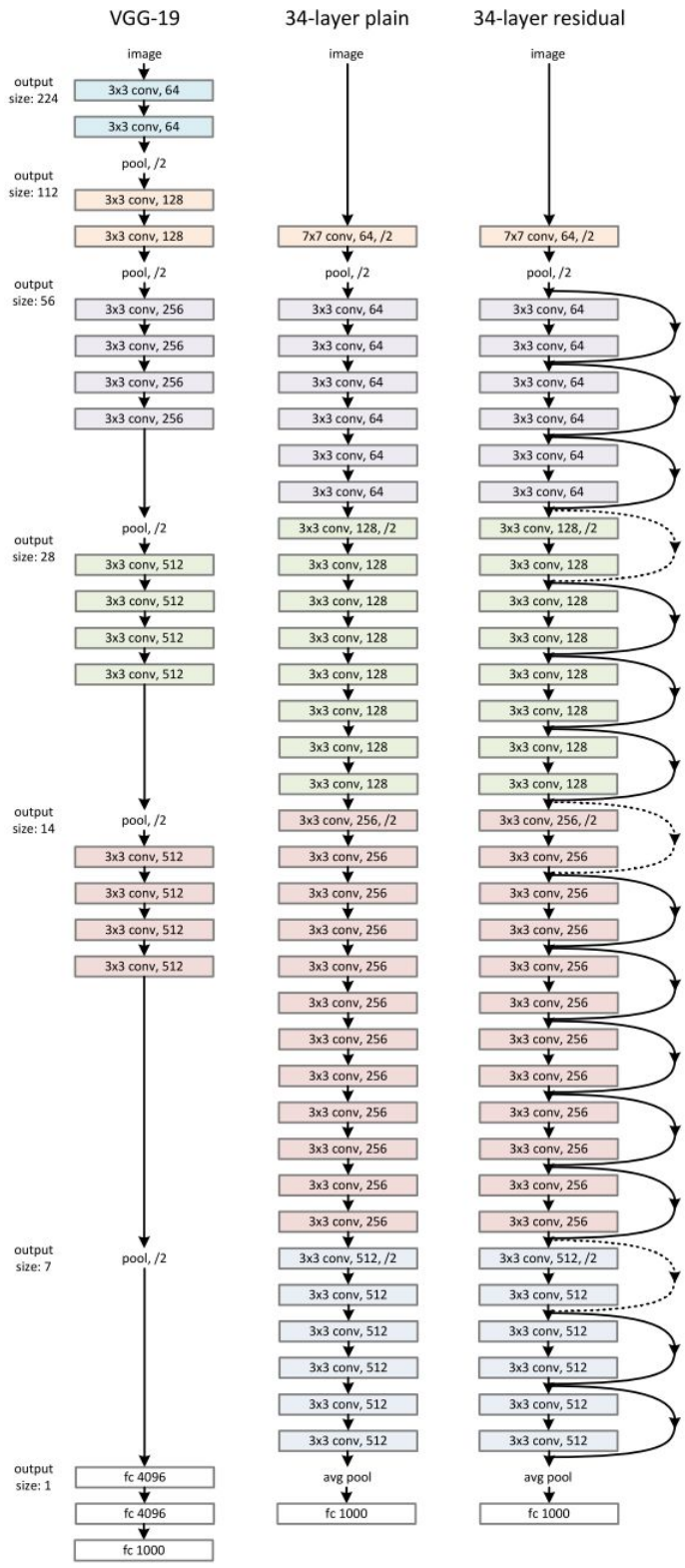


Figure 10: Comparison of network architectures for ImageNet. Left: the VGG-19 model [20]. Middle: a 34-parameter-layer non-residual network. Right: ResNet-34.

4. RPN/Faster R-CNN

DIID's Pedestrian Proposal Module follows [17] to take advantage of Region Proposal Network (RPN). RPN is a fast and robust region proposal algorithm developed by Microsoft researchers. Given an input image, an RPN simultaneously predicts where objects are in the image (object bounds) and how confident it is that there are objects at those locations (objectness scores). The object bound and objectness score together define a region proposal. Proposal generated by RPN are then used by Fast R-CNN [21], a state-of-the-art object detection network. The strength of RPN architecture lies in the fact that it "shares full-image convolutional features" with Fast R-CNN, and thus offers nearly cost-free region proposals. [17] took a step further and merged RPN and Fast R-CNN into a single network called Faster R-CNN. We follow this design of Faster R-CNN to construct the Pedestrian Proposal Module.

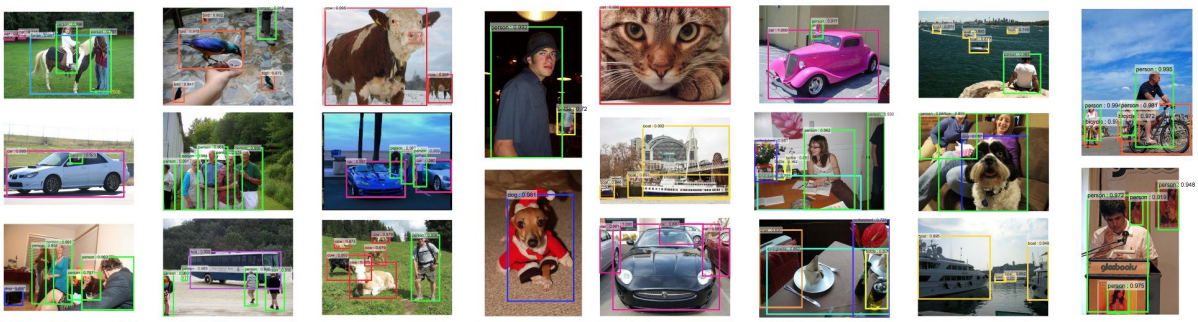


Figure 11: Examples of Faster R-CNN's object detection results.

5. Additional improvement

Inspired by the approach of Shi et al. in [22], we want to leverage the power of transfer learning to improve the current DIID system. As a large portion of visual clues are drawn from the targeted person's outfit, the can take advantage existing large-scale dataset in fashion domain. In 2016, Liu et al. published one of such datasets, namely DeepFashion [23]. DeepFashion contains over richly annotated 800,000 images taken under different scenarios such as in-store, street view, and consumer. It is currently the largest fashion dataset publicly available and is much larger than those used in person re-ID. Thus, we can exploit this resource to both

further fine-tune DIID system for visual-based person search, as well as to integrate semantic attributes learning for description-based person search.

Figure 12: DeepFashion's sample data.

IV. Algorithm evaluation

Criteria	Our solution
Use cases	
Can the program be used to search for suspects of criminal incidents?	- Yes, given a query image of the suspect and access to the on-campus cameras (or their frames).
Is the program designed strictly for criminal settings? Can it be used for something else?	- No, the system discussed in this paper is designed for general person re-identification purpose. Thus, it can be used to search for any person of interest if satisfies the constraints and assumptions stated, and is not restricted to only criminal settings.

Can the system handle extreme use cases such as when the time range is large or when it is rainy?	- As long as the assumption about appearance of the targeted person is satisfied, the program can perform its job. If not, there is not guarantee that the algorithm and detection any visual clues to perform matching. - The software might not work well in extreme cases where the suspect is severely occluded or when the camera's field of view is limited. - Identification can still be deployed in moderate weather. This might not be true to extreme weather situations that can compromise the views such as heavy rains or snowstorms.
Time	
Can the system work in real-time?	- Yes, thanks to RPN's robustness, fast R-CNN architecture, and the use of ResNet-50 instead of ResNet-152.
How long is the training time?	- Training time can take a long time, possible up to one or two weeks. Although this should not happen frequently, only when new training data are available, one way to account for this is to have 2 instances of the network: one that is always online, ready for criminal incidents, while the other are trained offline. After the former finished training process, the online network can then update its weights to the new version.
Others	
Is the system scalable and flexible?	- Yes, according to [0], sub-sampling the labeled and unlabeled identities would help with scalability,
What are the strengths of the system?	- DIID jointly handles both pedestrian detection and identity matching in a single network, which allows for directly inference of the mutual influence of the two tasks. This unified architecture also

	help simplify future optimization. - DIID is scalable and was tested on one of the large datasets available for person re-identification.
What are the limitations of the system?	- Since the method relies on visual clues, several assumptions need to be satisfied for it to work. - Similar to any deep-learning-based system, DIID also requires lots of training data.

V. Future work

Below is some directions and ideas for follow up works to improve the system.

- Several techniques for improving re-ID performance, such as using multi-loss classification or attribute-based learning, have been proposed by researchers over the years. Experimenting with these could potentially enhances the performance of the DIID system.
- To ensure the system's robustness, the bottleneck of training data should be tackle. One approach is to cooperate with different departments at Lafayette to collectively gather more training data through the school's camera system. Another approach is to outsource more datasets from other related domain such as fashion and perform transfer learning.

D. Conclusion

Public security and surveillance have attracted much attention as technology evolves. On one hand, many people are afraid that better software may be used to invade their privacy. While on the other hand, we need it for better protection for the community. In this project, we present a person detecting and tracking system that can be used for criminal suspect search using Joint Detection and Identification algorithm. Although in this project, the system discussed is not meant to be used to predict or avoid on-campus criminal incidents, an effective suspect

identification may indirectly reduce the rate of such events - people are likely to think twice if they know it is easier for them to get caught.

References

[0] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. InThe IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[1] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In The IEEE Conference on ComputerVision and Pattern Recognition (CVPR), July 2017.

[2] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3376–3385, 2017.

[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[4] Mathew Monfort , Anqi Liu , Brian D. Ziebart, Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, p.3672-3678, January 25-30, 2015, Austin, Texas

[5] Person Re-Identification by Deep Joint Learning of Multi-Loss Classification. Wei Li et al. IJCAI. 2017.

[6] Large scale similarity learning using similar pairs for person verification. Yang Yang et al. AAAI. 2016.

[7] Fast Person Re-Identification via Cross-Camera Semantic Binary Transformation. Jiaxin Chen et al. CVPR. 2017.

[8] Chao Zhu and Yuxin Peng. Group cost-sensitive boosting for multi-resolution pedestrian detection. InProceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, pages 3676–3682. AAAI Press, 2016.

[9] Chao Zhu and Yuxin Peng. A boosted multi-task model for pedestrian detection with occlusion handling. InProceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pages 3878–3884. AAAI Press, 2015

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once:Unified, real-time object detection. InThe IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[11] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[12] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. Person re-identification: Past,present and future.CoRR, abs/1610.02984, 2016

[13] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. In IJCAI, 2017

[14] J. Chen, Y. Wang, J. Qin, L. Liu, and L. Shao. Fast person re-identification via cross-camera semantic binary transformation. In 2017 IEEE Conference on ComputerVision and Pattern Recognition (CVPR), pages 5330–5339, July 2017.

[15] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Z. Li. Large scale similarity learning using similar pairs for person verification. InProceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, pages 3655–3661. AAAI Press, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. InThe IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal.Mach. Intell., 39(6):1137–1149, June 2017.

[18] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. Commun. ACM, 54(10):95–103, October 2011

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. InProceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012.

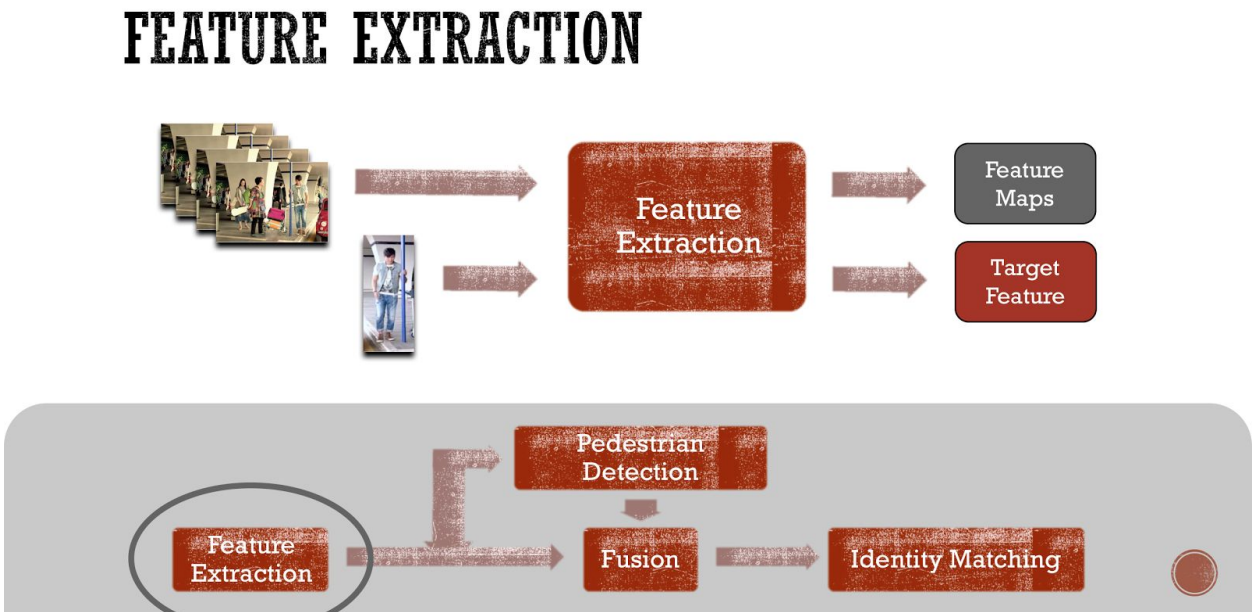
[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition.CoRR, abs/1409.1556, 2014.

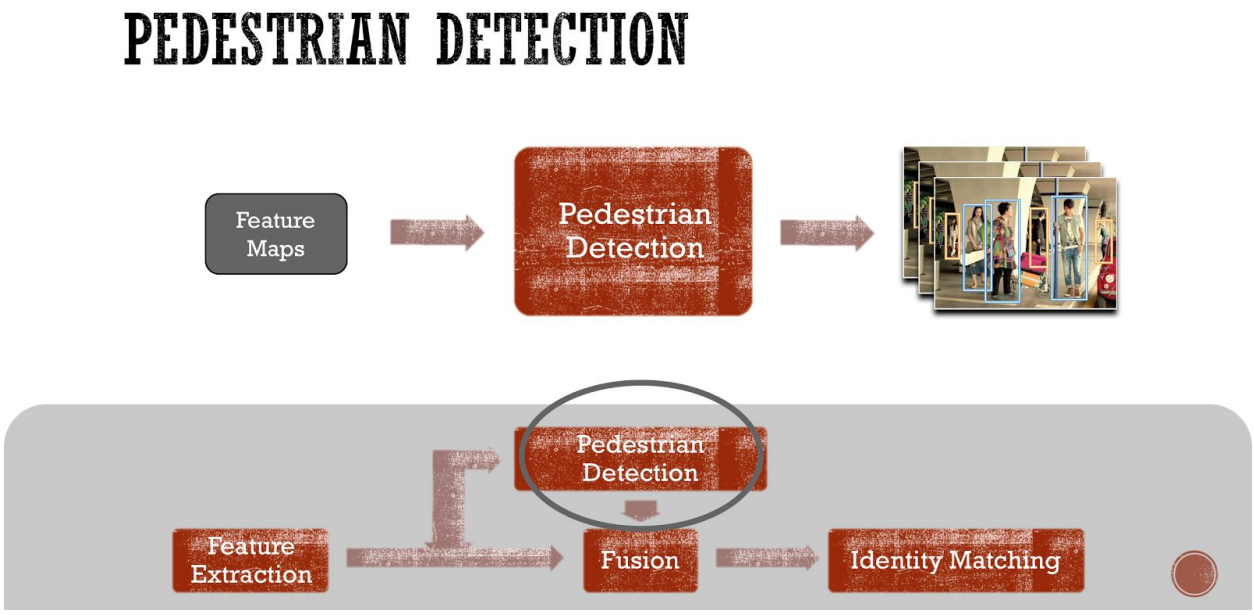
[21] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015.

[22] Zhiyuan Shi, Timothy M. Hospedales, and Tao Xiang. Transferring a semantic representation for person re-identification and search. InThe IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

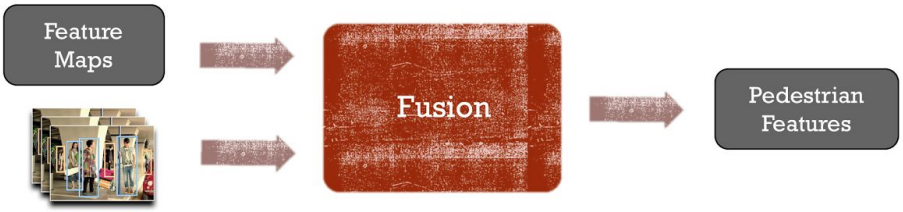
[23] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion:Powering robust clothes recognition and retrieval with rich annotations. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016

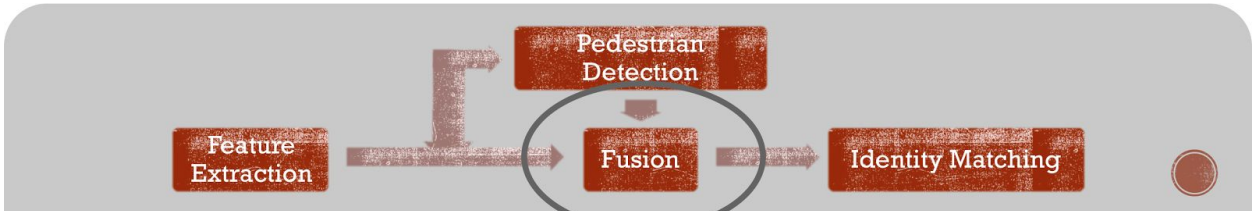
Appendix


FEATURE EXTRACTION
Feature Extraction
Feature Maps
Target Feature
Pedestrian Detection
Feature Extraction
Fusion
Identity Matching


PEDESTRIAN DETECTION
Feature Maps
Pedestrian Detection
Pedestrian Detection
Feature Extraction
Fusion
Identity Matching

FUSION LAYER





IDENTITY MATCHING