A Project in Image Processing and Analysis

Conducted at the GIP and the ISL Labs, CS, Technion

Pedestrian Re-Identification Across Two Cameras

Ilya Gurvich

Ilya.gurvich@gmail.com

Under the supervision of Dr. Tammy Avraham

tammya@cs.technion.ac.il

September 2012

Table of Contents

1 Abstract

This project describes a novel method for pedestrian re-identification. The method implicitly models a domain of transfer functions. It simultaneously learns to distinguish between changes that are camera and location dependent and those that depend on the person's identity. As opposed to most previous work, which deals with re-identifying a person at any new location, we focus on a solution for the natural setup of surveillance systems in which the cameras are specific and stationary, and exploit the fact that for a pair of specific cameras the transfer domain is limited. We argue that the transformation function is a multi-valued mapping and address this in the proposed method. We show that the key to the ability to learn the transfer associated with two cameras from rather small sets of inter-camera example pairs lies in utilizing the more abundant negative examples. We experiment with the VIPeR dataset and with a video provided by Rafael. A new state-of-the-art performance is achieved on VIPeR.

2 Introduction

The re-identification problem has received increasing attention in the last five to six years, especially due to its important role in surveillance systems. It is desirable that computer vision systems will be able to keep track of people after they have left the field of view of one camera and entered the field of view of the next, even when these fields of view do not overlap.

We make the distinction between the general re-identification problem, in which the goal is to re-identify a person in any new location, and the camera-specific re-identification problem, in which the goal is to provide a solution for a specific site. In this work we tackle the second goal. Given a pair of stationary cameras, *A* and *B*, capturing two non-overlapping regions, and a training set of annotated people captured by those two cameras, our objective is to recognize correspondence between the appearance of a never-before-seen person in camera *A* and his appearance in camera *B*. As can be seen in the examples in Figure 1, learning the domain of the camera-specific transformations may be very informative. Each camera is associated with a limited variety of backgrounds, illumination conditions, and sometimes human poses.

(a) VIPeR examples (b) Examples from the video from Rafael

Figure 1: Examples from the VIPeR dataset and the video from Rafael: five people captured by one camera (top row) and another camera (second row). We see that the background, illumination, resolution and sometimes pose are camera dependent.

We propose an algorithm that exploits these properties and that is based on the observation that the transfer between two cameras is a multi-valued mapping which can be estimated using implicit function learning. The algorithm models the implicit transfer function by training a binary classifier with concatenations of pairs of vectors, the first describing an instance associated with camera *A*, and the second describing an instance associated with camera *B*. The objective is to be able to distinguish between positive pairs – pairs of instances capturing the same person with two different cameras, and negative pairs – pairs of instances whose members are associated with two different people and two different cameras. We consider the optimal number of negative examples to use for training and show that the key to the ability to learn the transfer associated with two cameras from rather small sets of inter-camera example pairs lies in utilizing the more abundant negative examples.

We denote our algorithm *ICT*, short for *Implicit Camera Transfer*. We experiment using the VIPeR [**1**] dataset and using a video provided by Rafael as a part of our cooperation with the Magnet-Vulcan Consortium[1]. A new state-of-the-art performance is achieved on VIPeR.

2.1 Related Work

Object re-identification is a challenge that has been receiving increasing attention. Person, or pedestrian, re-identification is a special focus of recent research, mainly due to its important role in surveillance systems.

Some recent methods focus on learning characteristics of the similarity between feature vectors describing two appearances of the same person against that of two vectors describing instances of different people. These methods usually use the absolute distance as the characteristic to be learned. The ELF method [**2**] models the distribution of the feature-wise difference between the instances using Ada-boost for feature selection and classification. In [**3**] it is observed that what matters is not the similarity itself, but the relative similarity: *positive pairs should be ranked higher than negative pairs.* Therefore, the goal is to weigh the features in a way that will maximize the difference between absolute differences of negative pairs and absolute differences of positive pairs. In our method, as opposed to the similarity-based methods, we do not make the assumption that greater similarity implies `same'.

Some methods start with a pre-process for separating the people from the background and some also attempt to divide the person into a few semantic parts. These high-level processes may lead to mistakes that will then be dragged into the re-identification training and classification stages. In our work we use bounding boxes surrounding the people. This is possible as our algorithm is implicitly trained to filter out the background by recognizing the background associated with each camera as person-independent. As such, it allows items carried by the people (e.g., bags) to be used as cues without additional explicit analysis. Moreover, the semantic high level analysis requires processing time that is unlikely to be implemented for real-time performance, while the methods proposed here can be used for real-time re-identification.

[1] The website of the Magnet-Vulcan Consortium is located at http://www.vulcan.org.il/.

3 Implicitly Learning Inter-Camera Transfer

In this section we describe the ICT algorithm. Given that there are two differently located stationary cameras *A* and *B*, covering two non-overlapping regions of a site, our algorithm is trained to find correspondence between people captured by the two cameras. Let $V_{i,k}^{A}$ describe the *k*'th appearance of a person with identity i captured by camera A, and let $V_{j,l}^{B}$ describe the l'th appearance of a person with identity j captured by camera B. Given a pair $\left(V_{i,k}^{A}, V_{j,l}^{B}\right)$, the goal is to distinguish between *positive* pairs with the same identity $(i = j)$, and *negative* pairs $(i \neq j)$. Our algorithm trains a binary classifier using concatenations of such positive and negative pairs of vectors coming from training data. Then it classifies new such pairs by querying the classifier on their concatenations. A detailed description of the algorithm follows.

3.1 The ICT Algorithm

3.1.1 The Training Stage

The Input:

- A set $\left\{V_{i,k}^{A} \middle| i = 1,\ldots,n; k = 1,\ldots,m_i^A\right\}$ of vectors describing instances of n people captured by camera A.
- A set $\left\{V_{i,k}^{B} \middle| i = 1,\ldots n; k = 1,\ldots,m_i^B\right\}$ of vectors describing instances of the same n people captured by camera B.

That is, for each person and each camera we may be provided with a few descriptor vectors, each associated with his appearance in a different video frame. If tracking is not available, but only a single image per camera per person, then the sets above would be defined in a degenerate form with $m_i^A = m_i^B = 1$.

Let $[a||b] = (a_1,\ldots,a_n,b_1,\ldots,b_m)$ denote the concatenation of vectors $a = (a_1,\ldots,a_n)$ and $b = (b_1,\ldots,b_m)$. The training input for the binary classifier is:

- A set of positive examples $\{[V_{i,k}^{A}||V_{i,l}^{B}] \mid i \in \{1,\ldots,n\},\ k \in \{1,\ldots,m_i^A\},\ l \in \{1,\ldots,m_i^B\}\}$.
- A set of negative examples $\{[V_{i,k}^{A}||V_{j,l}^{B}] \mid i \neq j,\ i,j \in \{1,\ldots,n\},\ k \in \{1,\ldots,m_i^A\},\ l \in \{1,\ldots,m_j^B\}\}$.

For the type of descriptors used and for details about the classifier used in our experiments, see Section 4. Note that there are $\sum_{i=1}^{n} m_i^A m_i^B$ positive examples, while there is a quadratic number $\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} m_i^A m_j^B$ of negative examples. We do not use all the negative examples but show that even a fraction of them significantly contribute to the success of the algorithms. See Section 3.2.2 and Section 4.3.

3.1.2 The Classification/Decision Stage

The Input:

• A set $\{V_{I,k}^A | k = 1, \ldots, m_I^A\}$ of vectors describing a person's track as captured by camera *A*.

• A set $\{V_{J,l}^B | l = 1, \ldots, m_J^B\}$ of vectors describing a person's track as captured by camera *B*.

The Decision: Apply the trained classifier on each of the concatenations $[V_{I,k}^A || V_{J,l}^B], k = 1, \ldots, m_I^A, l = 1, \ldots, m_J^B$. In our experiments we use an SVM as the classifier and output the average of the decision values: let $y_{k,l},\ k = 1, \ldots, m_I^A, l = 1, \ldots, m_J^B$ be the decision values obtained from the classifier. The algorithm returns the mean $Y = \frac{1}{m_I^A m_J^B} \sum_{k=1}^{m_I^A} \sum_{l=1}^{m_J^B} y_{k,l}$. If tracking is not available, i.e. $m_I^A = m_J^B = 1$, the algorithm returns $Y = y$ where y is the decision value returned by the SVM classifier for the concatenation $[V_I^A || V_J^B]$.

The classification stage is illustrated in Figure 2 below.

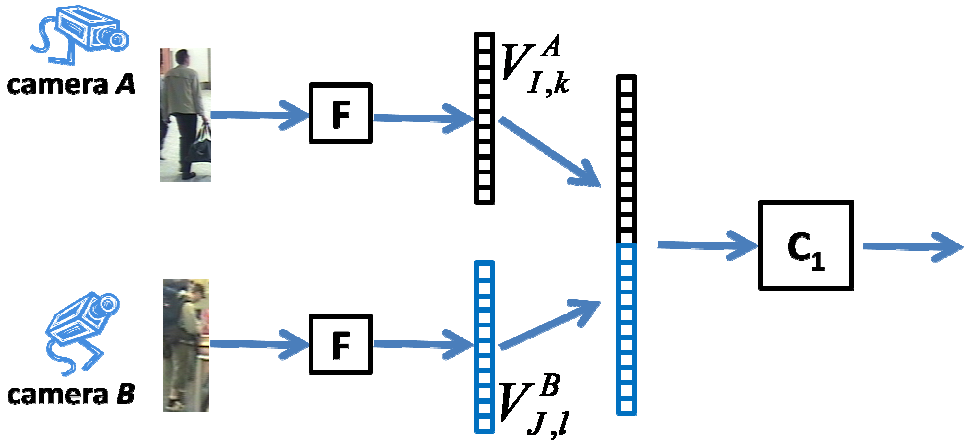

Figure 2: Illustration of the classification process used in the proposed algorithm. From each of the instances captured by cameras *A* and *B*, features are extracted (F). Then the concatenation of those two feature vectors, $V_{I,k}^A$ and $V_{J,l}^B$, is the input to the classifier C_1.

3.2 Algorithm Discussion

3.2.1 The Concatenation Technique as a Method for Modeling Binary Relations

If each person had one possible captured appearance for each camera, there would have be a function transforming a person's appearance in camera A to that same person's appearance in camera B, $f_1: R^d \to R^d$ where d is the descriptor vector's dimensions. If V_i^A was person i's one possible appearance in camera A and V_i^B was person i's only possible appearance in camera B, then $f_1(V_i^A) = V_i^B$. However, this is not the case, as each person has multiple possible appearances in each camera. Therefore, a single-value function cannot model the domain of all possible transformations. A multi-valued function, or a binary relation, $f_2: R^d \times R^d \to \{0,1\}$, is a more appropriate model:

$$f_2(V_{i,k}^A, V_{j,l}^B) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

This is actually what our vector concatenation based method does. The domain of transformations is represented as a mapping from concatenated examples of the transformation's input and output to a binary classification. We believe that this interpretation is fundamentally new and is an important contribution of this project.

3.2.2 The Role of Negative Examples

As mentioned above, the number of negative examples that can be used for training is quadratic in the number of positive examples. Using all the negative examples can lead to a strong bias and is computationally expensive. Do we need all the negative examples? Do we need negative examples at all? In our experiments (Section 4.3) we tested the contribution of the negative examples by checking the algorithm's performance as a function of the number of negative examples used for training. We also tested the possibility of ignoring all negative examples and using instead a one-class SVM trained only from positive examples. We learned that (a) not all negative examples are essential and training time can be saved by selecting only some of them; (b) the negative examples play an important role in compensating for the usually small number of positive examples, by helping in defining the borders of the "cloud" formed by the positive transformations.

4 Experiments

4.1 Implementation details

4.1.1 Features

We use a common and simple description of bounding boxes surrounding the people: each bounding box is divided into five horizontal stripes. Each stripe is described by a histogram with 10 bins for each of the color components H, S, and V. This results in feature vectors with 150 dimensions. We did not focus on finding optimal features. Any alternative descriptors can be easily used as well, and may further improve the algorithm's performance.

4.1.2 Classifiers

We use an RBF kernel binary SVM classifier in both algorithms as the classifier for the concatenated vectors. In one of our experiments below we test the use of a one-class-SVM also with an RBF kernel. For the above we used LibSVM [**4**]. Since many experiments had to be performed, we also implemented a custom classification routine which outperforms LibSVM. The routine was implemented in the C programming language, exporting a MATLAB MEX interface. Unlike LibSVM, our routine doesn't use sparse vectors, thus accelerating the classification stage.

4.1.3 Evaluation Methods

In the preliminary experiments described in Section 4.2 we used confusion tables and the Normalized Area Under Curve (nAUC) statistic of the Receiver Operating Characteristic (ROC) curve. An ROC curve is a graphical plot of the true positive rate versus the false positive rate for a binary classifier system as its discrimination threshold is varied. The machine learning community often uses the ROC nAUC statistic for model comparison.

In the experiments described in Section 4.3, and 4.4, we compute and compare average Cumulative Match Characteristic (CMC) curves. This is the most widely accepted way to evaluate re-identification algorithms. For each person in the test set, each algorithm ranks the matching of his or her appearance in camera *A* with the appearances of all the people in the test set in camera *B*. The CMC curve summarizes the statistics of the ranks of the true matches. That is, a point (r, p) on the CMC curve (r is located on the horizontal axis) means that in p percent of the re-identification attempts the correct person appeared within the best r results. For quantitative comparison we use the measure $rank(i)$, which denotes the percentage of true matches found within the first i ranked instances, the CMC-expectation measure, which is the mean rank of the true match, and the nAUC.

4.1.4 VIPeR dataset

In our first set of experiments we use the VIPeR dataset, the most commonly used dataset for evaluating re-identification methods. It contains 632 pedestrian image pairs. Each pair contains two images of the same individual seen from different viewpoints by two cameras. See examples in Figure 1(a).

4.2 Feasibility Tests

In this preliminary set of experiments we aimed at exploring whether concatenating feature vectors and using an SVM classifier could potentially yield competitive results.

In the first experiment we tested the performance of an SVM classifier to correctly identify positive and negative examples by using its binary output. I.e. Let us denote the real valued decision value of the classifier by *v*. Then, $sign(v) \in \{\pm 1\}$ was used to determine whether the classified pair belongs to a positive or to a negative example. Note that in this stage of the research we had not treated the problem as a ranking problem. In this experiment, both the disjoint training and the testing sets contained a single randomly-selected negative example per a positive example. We performed a 5-fold cross-validation on the VIPeR dataset, repeating this procedure for 100 times and averaging the results. The results of this experiment are shown in Table 1 below.

Truth \ Prediction	Negative(%)	Positive(%)
Negative	78.4304	21.5696
Positive	14.9699	85.0301

Table 1: The confusion table resulting from the first experiment performed on the VIPeR dataset.

We noticed that if two or more negative examples are given to the SVM per a positive example, the SVM classifies the *whole* training set as negative. This bias is due to the SVM learning the prior probability of an example to belong to a class.

To eliminate the aforementioned bias, we decided to use the probabilistic output of the SVM and treat the problem as a detection problem. We plotted the nAUC values as a function of the number of negative examples selected per a single positive example. On the same graph we plotted the time it takes to perform the whole experiment in hours. In this setup, the experiment time included both the training and the testing stages. We made 4 random selections of the negative examples and averaged the results. We decided to decrease the number of repetitions from 100 to 4 because of run-time considerations. We conducted 5-fold cross-validations. The number of negative examples was set for both the training and the testing sets.

The results of this experiment are shown in Figure 3 below.

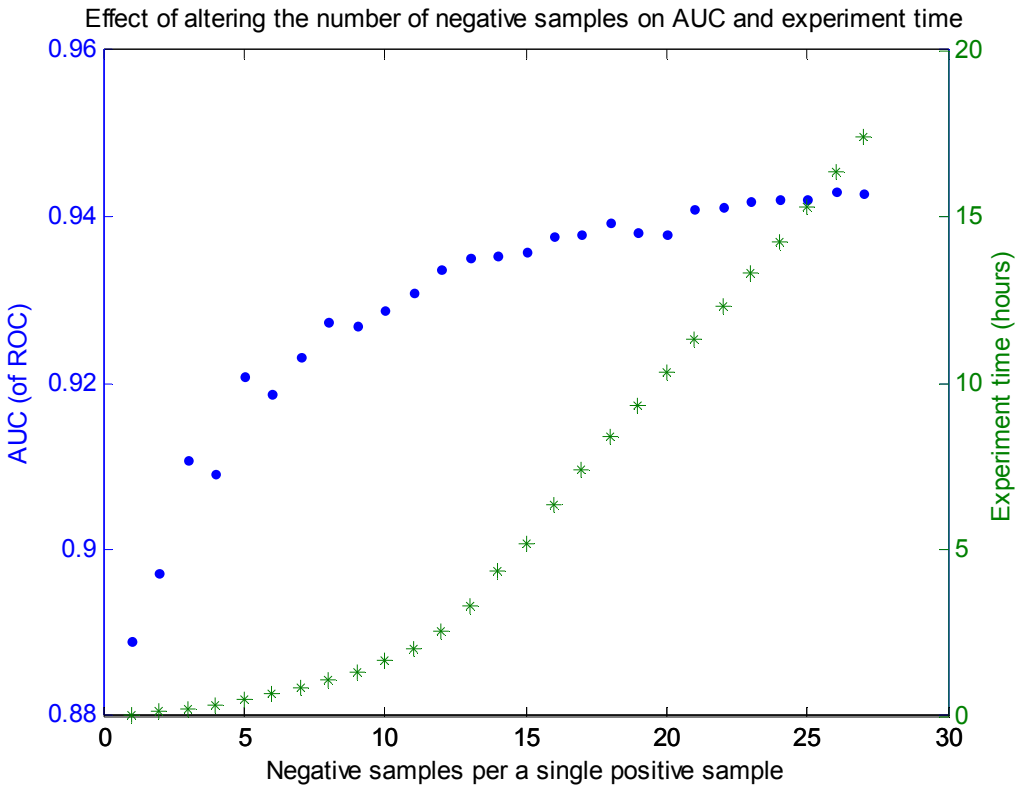

Figure 3: The effect of altering the number of negative samples on AUC and experiment time.

Figure 3 shows that increasing the number of negative examples doesn't yield a significant increase in the nAUC from a certain stage. The experiment time does keep rising.

4.3 How Many Negative Examples to Use?

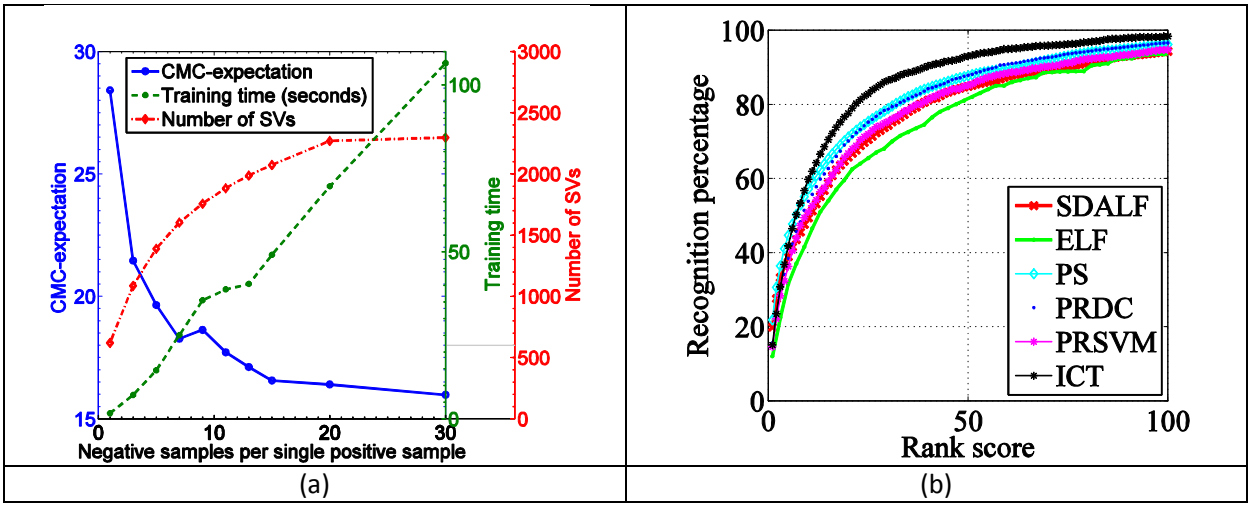

Figure 4: (a) ICT's performance on the VIPeR dataset as a function of κ, the number of negative examples per single positive example, measured by the CMC expectation, the training time, and the support-vectors used by the SVM. (b) CMC curves comparing ICT's results on VIPeR with recent state-of-the-art reported in [2], [3], [5], [6], and [7].

Following the feasibility tests described above, we conducted a rigorous study using standardized metrics to demonstrate the effect of altering the number of negative examples.

In this set of experiments we use the VIPeR dataset. We perform a 2-fold cross-validation, dividing the 632 pedestrians into equal-size training and test sets. We repeat this process four times with different random choices for the sets. The number of positive examples available for training is $P = 316$ (one concatenated pair for each person). We test the performance of ICT for different numbers of negative examples $N = \kappa P$, where $\kappa = 1,3,5,7,9,11,13,15,20,30$. That is, per a positive example associated with person i, κ of the $N - 1$ negative examples involving person i's appearance in camera A, are randomly selected. Each training involves a parameter learning stage: we learn the optimal C and γ parameters for the RBF SVM by a 4-fold cross-validation inside the training set, searching for the parameters that result in the lowest CMC-expectation.

See Figure 4(a) for ICT's performance as a function of κ. It reports the CMC expectation, the training time, and the number of support vectors found by the SVM. We see that the expectation drops as κ increases, at a high slope for small κ's and at an almost zero slope for $\kappa > 15$. We also see a similar convergence in the number of support vectors, which means that adding more than a certain number of negative examples does not add information to the model. Note that the computation time for training grows linearly with κ. Thus in our experiments we set $\kappa = 30$, which gives CMC-expectation of 15.9 and $rank(1)$ of 15.1.

We also tested a variation of the algorithm that learns only from positive examples using one-class-SVM (i.e, $\kappa = 0$). The one-class SVM test, which followed a similar procedure, yields a CMC expectation value

of 45.6 and $rank(1)$ of 5.9. The conclusion is, obviously, that the negative examples play an important role in modeling the domain of the positive transfers.

4.4 Comparing to State-of-the-Art on VIPeR

See Figure 4(b) for a comparison of ICT's performance on the VIPeR dataset with the results of recent work. The results of the ELF [**2**] and the SDALF [**6**] algorithms were kindly provided by the authors of [**6**]. The results of PRDC were kindly provided by the authors of [**7**]. The results of the PS based algorithm were kindly provided by the authors of [**5**]. The results of PRSVM are those presented in [**3**]. See Table 2 for a comparison of the CMC expectation, $rank(1)$, $rank(10)$, $rank(20)$, and nAUC of the different methods. The CMC-expectation and the nAUC are much better for ICT than for all previous methods. ICT does not achieve the best $rank(1)$ performance, but performs best for all ranks 8 and up. Note that the first ranks may not be so relevant in the surveillance context. Most scenes include much less than 316 people.

Method	Expectation	$rank(1)$	$rank(10)$	$rank(20)$	nAUC
SDALF	25.5	19.9	49.4	65.7	92.2
ELF	28.9	12	44	61	91.2
PS	21.2	**21.8**	57.2	71.2	93.6
PRDC	21.5	15.7	53.9	70.1	93.5
PRSVM	27.9	14.6	50.9	66.8	91.4
ICT	**15.9**	15.1	**59.8**	**77.6**	**95.3**

Table 2: Results of ICT on the VIPeR dataset compared to the models in [2], [3], [5], [6], and [7].

4.5 Experiments on the video from Rafael

As a part of our cooperation with the Magnet-Vulcan consortium we were provided with two annotated video files that were taken by two cameras positioned outdoors by Rafael. There are 5 people that appear in these videos. Every person has 12-250 annotated instances in every video. We used a simple automatic heuristic to filter partially occluded instances. We included only instances whose bounding boxes were larger than 65x14 pixels. See examples of frames in Figure 5 and examples of bounding boxes containing people in Figure 1(b).

Camera *A* Camera *B*

Figure 5: Examples of frames taken from the two video files from Rafael.

Three people are selected as the training set and the other two as the test set. We evaluated all 10 possible options for such a partition.

4.5.1 The Training Stage

For the purpose of building positive examples, we choose 20 instances of every person from camera *A*, and 20 instances from camera *B*. If not enough instances are available, we choose with repetitions. Positive examples are built by random coupling that creates 20 pairs.

Negative examples are chosen by randomly selecting $7 \cdot 20$ instances for every person that are concatenated with $7 \cdot 20$ instances of each of the other people in training set.

4.5.2 The Classification/Decision Stage

For every person in the training set, 20 randomly selected instances are chosen per camera. Then $400 = 20^2$ concatenations are created to represent correct matches (positive examples) for person 1 appearing on camera A and the same person 1 appearing on camera B. We shall denote this set of concatenations S_{11}. Similarly, a set of incorrect matches S_{12} (person 1 on camera *A* and person 2 on camera *B*), incorrect matches S_{21}, and correct matches S_{22} are created. Each set contains 400 concatenations.

The we apply the algorithm described in Section 3.1.2 on every set S_{11}, S_{12}, S_{21}, S_{22}, and the outputs of the algorithms are Y_{11}, Y_{12}, Y_{21}, Y_{22}, respectively. A match is considered successful if $Y_{11} \cdot Y_{22} > Y_{12} \cdot Y_{21}$ holds. The meaning of such a success is that if two people (1 and 2) appeared on camera A and after a while appeared on camera B, the algorithm would know how to re-identify them correctly. Meaning that person 1 would still be identified as person 1 and person 2 would still be identified as person 2.

4.5.3 Results

The algorithm succeeded for all 10 possible partitioning options.

5 Future Work – Transitive Re-Identification

We believe that the method presented in this project can be extended to work in a transitive setup. Namely, in a setup with three cameras – A, B and C, where we are given a set of examples of individuals appearing on cameras A and B, and another set of such examples of individuals appearing on cameras B and C. No such examples need to be available for cameras A and C. We believe that it would be possible to learn implicit transfer functions from A to B and from B to C and to be able to use them to model the transfer from A to C.

A further extension would include re-identification in a network of cameras. Given an undirected connected graph of cameras $G = (V, E)$, where the vertices V represent cameras and the edges E represent the availability of examples between the vertices of an edge, we will be able to model the transfer from any vertex u_i to any other vertex u_j on the graph. Note that the edge $(u_i - u_j)$ need not exist as long as there is a path $P = (u_i - \cdots - u_j)$ available. This would be extremely useful because inter-camera examples are hard to acquire, but a connected graph formed by these examples would allow re-identification in the whole network.

5.1 The First Proposed Method - Marginalization

At the classification stage we should have two binary SVM models available. One for transferring from A to B, and another for transferring from B to C. These models should contain probability information. Thus, it would be possible to compute the posteriors $P(Y_{AB}|X_A, X_B)$ and $P(Y_{BC}|X_B, X_C)$. Where X_A, X_B, X_C are the feature vectors describing individuals on the cameras, and $Y_{AB}, Y_{BC} \in \{+1, -1\}$ are the events of X_A, X_B (and X_B, X_C) being the same/not-same individuals.

5.1.1 Modeling the Distribution of Feature Vectors on Camera B

We will model the distribution of the features vectors on camera B by training a one-class SVM using feature vectors of images acquired from this camera only. Note that concatenated feature vectors are *not* used here. The SVM should be trained with *single* feature vectors, each representing a person. After training the SVM using the training set, we will classify the *same* training set by the one-class SVM to build a histogram of decision values. We believe that the decision values will be distributed *normally*. This is based on [**8**], where this phenomenon is observed, supported by empirical evidence and derived using central limit theorems. At this stage we will be able to compute the decision values' mean and standard deviation to build a parametric (normal) model of their distribution.

More formally, let $v(x)$ denote the decision value output of the SVM which is defined by

$$v(x) = \Sigma_i y_i \alpha_i k(x_i, x)$$

where x is the feature vector of an individual on camera B, x_i are the support vectors, $y_i$are the labels of the support vectors and α_i are the coefficients of the support vectors. Our assumption is when treating $v(x)$ as a random variable, $v(x) \sim N(\mu, \sigma^2)$ and that the parameters of the distribution can be found by performing maximum likelihood estimation (MLE) on the training set.

5.1.2 The Classification Stage

Given the feature vectors X_A, X_C calculated from images taken by cameras A and C respectively, we would like to compute $P(same|X_A, X_C)$ which is the probability that both feature vectors represent the same individual. We can also treat this as a ranking problem by comparing the probabilities of two candidates X_{C_1} X_{C_2}. E.g.

$$P(Y_{AC} = 1 | X_A = x_A, X_C = x_{c_1}) >^{?} P(Y_{AC} = 1 | X_A = x_A, X_C = x_{c_2})$$

We can use marginalization to compute:

$$P(Y_{AC} = 1 | X_A = x_A, X_C = x_C) = \int_{x_B} P(Y_{AB} = 1, Y_{BC} = 1 | X_A = x_A, X_B = x_B, X_C = x_C) \cdot f_{X_B}(v(x_B)) dx_B$$

Where $f_{X_B}(\cdot)$ is the normal probability *density* function (PDF), estimated previously. Note that this function's argument is the decision value of the one-class SVM. The events Y_{AB}, Y_{BC} are independent, and we get:

$$= \int_{x_B} P(Y_{AB} = 1 | X_A = x_A, X_B = x_B) \cdot P(Y_{BC} = 1 | X_B = x_B, X_C = x_C) \cdot f_{X_B}(v(x_B)) dx_B$$

The two first terms of the integrand are the posteriors computed by the binary SVM, as described in [**9**]. These posteriors are sigmoids as functions of the decision values of the binary SVMs. This integral cannot be solved analytically. Thus, we should turn to numerical methods such as Metropolis-Hastings or other Markov-Chain-Monte-Carlo (MCMC) algorithms to compute it.

5.2 The Second Proposed Method – Global Search

Another possible method to re-identify is to search for the feature vector x_B that yields the maximal value of the product of the first two terms in the integrand that appears in the previous section. More formally,

$$\max_{x_B} P(Y_{AB} = 1 | X_A = x_A, X_B = x_B) \cdot P(Y_{BC} = 1 | X_B = x_B, X_C = x_C)$$

$$s.t. \left| x_{B(H)} \right| = 1, \left| x_{B(S)} \right| = 1, \left| x_{B(V)} \right| = 1$$

The constraints restrict the feature vector to be a valid *normalized* HSV color histogram.

The objective function is not convex, thus global numeric search methods (such as simulated-annealing, combined with gradient-descent) should be used.

6 Summary

This project considers the re-identification task and proposes three main contributions: 1. The observation that the transfer between two cameras is a multi-valued mapping which can be estimated using implicit function learning. 2. A re-identification algorithm which models the transformation between the cameras. We show that incorporating a large number of negative examples can compensate for the usually low number of manually annotated positive examples. The algorithm yields an extremely fast classifier. We present new state-of-the-art re-identification performance. 3. We propose a new direction for research to enable transitive re-identification and re-identification in a network of cameras, based on the proposed algorithm.

7 References

1 Gray, D, Brennan, S, and Tao, H. Evaluating Appearance Models for Recognition, Reacquisition, and Tracking. *PETS Workshop in conjunction with ICCV* (2007).

2 Gray, D and Tao, H. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. *ECCV* (2008).

3 Prosser, B, Zheng, W, Shaogang, G, and Xiang, T. Person Re-Identification by Support Vector Ranking. *BMVC* (2010).

4 Chang, C and Lin, C. *LIBSVM: a library for support vector machines*. 2001.

5 Cheng, D S, Cristani, M, Stoppa, M, Bazzani, L, and Murino, V. Custom Pictorial Structures for Re-identification. *BMVC* (2011).

6 Farenzena, M, Bazzani, L, Perina, A, Murino, V, and Cristani, M. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. *CVPR* (2010).

7 Zheng, W, Gong, S, and Xiang, T. Person Re-identification by Probabilistic Relative Distance Comparison. *CVPR* (2011).

8 Balasubramanian, Krishnakumar, Donmez, Pinar, and Lebanon, Guy. Unsupervised Supervised Learning II: Margin-Based Classification Without Labels. *J. Mach. Learn. Res.*, 12 (nov 2011), 3119--3145.

9 Platt, John C. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. (1999), MIT Press, 61--74.