Cross-Domain Forensic Shoeprint Matching

Bailey Kong[1]
bhkong@ics.uci.edu
James Supancic, III[1]
jsupanci@uci.edu
Deva Ramanan[2]
deva@andrew.cmu.edu
Charless Fowlkes[1]
fowlkes@ics.uci.edu

[1] Dept. of Computer Science
University of California, Irvine
United States of America
[2] Robotics Institute
Carnegie Mellon University
United States of America

Abstract

We investigate the problem of automatically determining what type of shoe left an impression found at a crime scene. This recognition problem is made difficult by the variability in types of crime scene evidence (ranging from traces of dust or oil on hard surfaces to impressions made in soil) and the lack of comprehensive databases of shoe outsole tread patterns. We find that mid-level features extracted by pre-trained convolutional neural nets are surprisingly effective descriptors for these specialized domains. However, the choice of similarity measure for matching exemplars to a query image is essential to good performance. For matching multi-channel deep features, we propose the use of *multi-channel normalized cross-correlation* and analyze its effectiveness. Finally, we introduce a discriminatively trained variant and fine-tune our system end-to-end, obtaining state-of-the-art performance.

1 Introduction

We investigate the problem of automatically determining what type (brand/model/size) of shoe left an impression found at a crime scene. In the forensics literature [1], this fine-grained category-level recognition problem is known as determining the *class characteristics* of a tread impression. This is distinct from the instance-level recognition problem of matching *acquired characteristics* such as cuts or scratches which can provide stronger evidence that a specific shoe left a specific mark.

Analysis of shoe tread impressions is made difficult by the variability in types of crime scene evidence (ranging from traces of dust or oil on hard surfaces to impressions made in soil) and the lack of comprehensive datasets of shoe outsole tread patterns (see Fig. 1). Solving this problem requires developing models that can handle *cross-domain* matching of tread features between photos of clean test impressions (or images of shoe outsoles) and photos of crime scene evidence. We face the additional challenge that we would like to use extracted image features for matching a given crime scene impression to a large open-ended database of exemplar tread patterns.

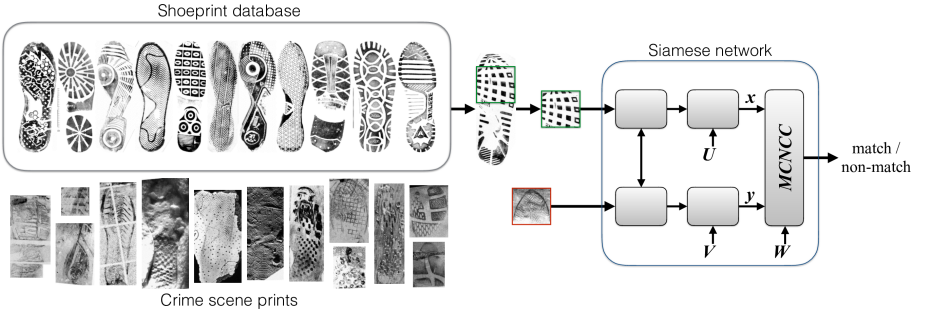

Figure 1: We would like to match crime scene prints to a database of test impressions despite significant cross-domain differences in appearance. We utilize a Siamese network to perform matching using a multi-channel normalized cross correlation. We find that per-exemplar, per-channel normalization of CNN feature maps significantly improves matching performance. Here U and V are the linear projection parameters for laboratory test impression and crime scene photo domains respectively. W is the per-channel importance weights. And x and y are the projected features of each domain used for matching.

Deep convolutional neural net (CNN) features hierarchies have proven incredibly effective at a wide range of recognition tasks. Generic feature extractors trained for general-purpose image categorization often perform surprising well for novel categorization tasks without performing any fine-tuning beyond training a linear classifier [20]. This is often explained by appealing to the notion that these learned representations extract image features with invariances that are, in some sense, generic. We might hope that these same invariances would prove useful in our setting (*e.g.*, encoding the shape of a tread element in a way that is insensitive to shading, contrast reversals, etc.). However, our problem differs in that we need to formulate a cross-domain similarity metric rather than simply training a k-way classifier.

We tackle this problem using similarity measures that are derived from normalized cross-correlation (NCC), a classic approach for matching gray-scale templates. For CNN feature maps, it is necessary to extend this to handle multiple channels. Our contribution is to propose a multi-channel variant of NCC which performs normalization on a per-channel basis (rather than per-feature volume). We find this performs substantially better than related similarity measures such as the cosine distance. We explain this finding in terms of the statistics of CNN feature maps. Finally, we use this multi-channel NCC as a building block for a Siamese network model which can be trained end-to-end to optimize matching performance.

2 Related Work

Shoeprint recognition: The widespread success of automatic fingerprint identification systems (AFIS) [11] has inspired many attempts to similarly automate shoeprint recognition. Much early work in this area focused on developing feature sets which were rotation and translation invariant. Examples include, phase only correlation [6], edge histogram DFT magnitudes [27], power spectral densities [2, 3], and the Fourier-Mellin transform [6]. Some other approaches pre-align the query and database image using the Radon transform [17] while still others sidestep global alignment entirely by computing only relative features between keypoints pairs [18, 21]. Finally, alignment can be implicitly computed by matching

rotationally invariant keypoint descriptors between the query and database images [18, 22]. In contrast to these previous works, we handle global invariance by explicitly matching templates using dense search over translations and rotations.

One shot learning: While we must match our crime scene evidence against a large database of candidate shoes, our database contains very few examples per-class. As such, we must learn to recognize each shoe category with as little as one training example. This is a one-shot learning problem [12]. Prior work has explored one-shot object recognition with only a single training example, or "exemplar" [13]. Specifically in the domain of shoeprints, Kortylewski *et al.* [9] fit a compositional active basis model to an exemplar which could then be evaluated against other images. Alternatively, standardized or whitened off-the-shelf HOG features have proven very effective for exemplar recognition [7]. We take a similar approach. Specifically, we examine the performance of using generic deep features which have proven surprisingly robust for a huge range of recognition tasks [20].

Similarity metric learning: While off-the-shelf deep features work well [20], they can be improved. In particular, for a paired comparison tasks, so-called "Siamese" architectures integrate feature extraction and comparison in a single differentiable model that can be trained end-to-end. Past work has demonstrated that Siamese networks learn good features for person re-identification, face recognition, and stereo matching [16, 23, 26]; deep pseudo-Siamese architectures can even learn to embed two dissimilar domains into a common co-domain [25]. For shoe class recognition, we learn to embed two types of images: (1) crime scene photos and (2) laboratory test impressions.

3 Multi-variate Cross Correlation

In order to compare two corresponding image patches, we extend the approach of normalized cross-correlation (often used for matching gray-scale images) to work with multi-channel CNN features. Interestingly, there is not an immediately obvious extension of NCC to multiple channels, as evidenced by multiple approaches proposed in the literature [4, 5, 15, 19]. To motivate our approach, we appeal to a statistical perspective.

Normalized correlation: Let x, y be two scalar random variables. A standard measure of correlation between two variables is given by their *Pearson's correlation coefficient* [15]:

$$\rho(x,y) = E[\tilde{x}\tilde{y}], \qquad \text{where} \qquad \tilde{x} = \frac{x-\mu_x}{\sqrt{\sigma_{xx}}}, \quad \sigma_{xx} = E[(x-\mu_x)^2], \quad \mu_x = E[x] \tag{1}$$

$$= \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}}\sqrt{\sigma_{yy}}} \qquad \text{where} \qquad \sigma_{xy} = E[(x-\mu_x)(y-\mu_y)] \tag{2}$$

and similar definitions hold for y. Intuitively, the above corresponds to the correlation between two transformed random variables that are "whitened" to have zero-mean and unit variance. The normalization ensures that correlation coefficient will lie between -1 and 1.

Normalized cross-correlation: Let us model pixels x from an image patch X as corrupted by some i.i.d. noise process and similarly pixels another patch Y (of identical size) as y. The *sample* estimate of the Pearson's coefficient for variables x, y is equivalent to the

normalized cross-correlation (NCC) between patches X, Y:

$$NCC(X,Y) = \frac{1}{|P|}\sum_{i\in P}\frac{(x[i]-\mu_x)}{\sqrt{\sigma_{xx}}}\frac{(y[i]-\mu_y)}{\sqrt{\sigma_{yy}}} \tag{3}$$

where P refers to the set of pixel positions in a patch and means and standard deviations are replaced by their sample estimates.

From the perspective of detection theory, normalization is motivated by the need to compare correlation coefficients across different pairs of samples with non-stationary statistics (*e.g.*, determining which patches $\{Y^1, Y^2, \ldots\}$ are the same as a given template patch X where statistics vary from one Y to the next). Estimating first and second-order statistics per-patch provides a convenient way to handle sources of "noise" that are approximately i.i.d. conditioned on the choice of patch P but not independent of patch location.

Multivariate extension: Let us extend the above formulation for random *vectors* $\mathbf{x}, \mathbf{y} \in R^N$. Intuitively, N corresponds to the multiple channels of values at each pixel (*e.g.*, $N = 3$ for a RGB image). The scalar correlation is now replaced by a $N \times N$ correlation *matrix*. To produce a final score capturing the overall correlation, we propose to use the *trace* of this matrix, which is equivalent to the sum of its eigenvalues. As before, we add invariance by computing correlations on transformed variables $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ that are "whitened" to have a zero-mean and identity covariance matrix:

$$\rho_{multi}(\mathbf{x},\mathbf{y}) = \frac{1}{N}Tr(E[\tilde{\mathbf{x}}\tilde{\mathbf{y}}^T]) = \frac{1}{N}Tr(\Sigma_{\mathbf{xx}}^{-\frac{1}{2}}\Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-\frac{1}{2}}) \tag{4}$$

$$\text{where} \quad \tilde{\mathbf{x}} = \Sigma_{\mathbf{xx}}^{-\frac{1}{2}}(\mathbf{x}-\mu_{\mathbf{x}}), \quad \Sigma_{\mathbf{xx}} = E[(\mathbf{x}-\mu_{\mathbf{x}})(\mathbf{x}-\mu_{\mathbf{x}})^T], \quad \Sigma_{\mathbf{xy}} = E[(\mathbf{x}-\mu_{\mathbf{x}})(\mathbf{y}-\mu_{\mathbf{y}})^T]$$

The above multivariate generalization of the Pearson's coefficient is arguably rather natural, and indeed, is similar to previous formulations that also make use of a trace operator on a correlation matrix [15, 19]. However, one crucial distinction from such past work is that our generalization (4) reduces to (1) for $N = 1$. In particular, [15, 19] propose multivariate extensions that are restricted to return a nonnegative coefficient. It is straightforward to show that our multivariate coefficient will lie between -1 and 1.

Decorrelated channel statistics: The above formulation can be computationally cumbersome for large N, since it requires obtaining sample estimates of matrices of size N^2. Suppose we make the strong assumption that all N channels are *uncorrelated* with each other. This greatly simplifies the above expression, since the covariance matrices are then diagonal matrices:

$$\Sigma_{\mathbf{xy}} = diag(\{\sigma_{x_c y_c}\}), \quad \Sigma_{\mathbf{xx}} = diag(\{\sigma_{x_c x_c}\}), \quad \Sigma_{\mathbf{yy}} = diag(\{\sigma_{y_c y_c}\}) \tag{5}$$

Plugging this assumption into (4) yields the simplified expression for multivariate correlation

$$\rho_{multi}(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{c=1}^{N}\frac{\sigma_{x_c y_c}}{\sqrt{\sigma_{x_c x_c}}\sqrt{\sigma_{y_c y_c}}} \tag{6}$$

where the diagonal multivariate statistic is simply the average of N per-channel correlation coefficients. It is easy to see that this sum must lie between -1 and 1.

Multi-channel NCC: The sample estimate of (6) yields a multi-channel extension of NCC which is adapted to the patch:

$$MCNCC(X,Y) = \frac{1}{|P|N}\sum_{c=1}^{N}\sum_{i\in P}\frac{(x_c[i]-\mu_{x_c})}{\sqrt{\sigma_{x_c x_c}}}\frac{(y_c[i]-\mu_{y_c})}{\sqrt{\sigma_{y_c y_c}}} \tag{7}$$

The above multi-channel extension is similar to the final formulation in [4], but is derived from a statistical assumption on the channel correlation.

Cross-domain covariates and whitening: Assuming a diagonal covariance makes strong assumptions about cross-channel correlations. When strong correlations exist, an alternative approach to reducing computational complexity is to assume that cross-channel correlations lie within a K dimensional subspace, where $K < N$. We can learn a projection matrix for reducing the dimensionality of features from both patch X and Y which decorrelates and scales the channels to have unit variance:

$$\hat{\mathbf{x}} = U(\mathbf{x} - \mu_x), \qquad U \in R^{K \times N}, E[\hat{\mathbf{x}}\hat{\mathbf{x}}^T] = I \tag{8}$$

$$\hat{\mathbf{y}} = V(\mathbf{y} - \mu_y), \qquad V \in R^{K \times N}, E[\hat{\mathbf{y}}\hat{\mathbf{y}}^T] = I \tag{9}$$

In general, the projection matrix could be different for different domains (in our case, crime scene versus test prints). One strategy for learning the projection matrices is applying principle component analysis (PCA) on each domain separately. Alternatively, when paired training examples are available, one could jointly learn the projections that maximize correlation across domains. This is achievable by canonical correlation analysis (CCA) [14]. An added benefit of using *orthogonalizing* transformations such as PCA/CCA is that transformed data satisfies the diagonal assumptions (globally) allowing us to estimate patch multivariate correlations in this projected space with diagonalized covariance matrices of size $K \times K$.

Global versus local whitening: There are two distinct aspects to whitening (or normalizing) variables in our problem setup: (1) assumptions on the structure of the sample mean and covariance matrix, and (2) the data over which the sample mean and covariance are estimated. In terms of (1), one could enforce an unrestricted covariance matrix, a low-rank covariance matrix (e.g., PCA), or a diagonal covariance matrix (e.g., estimating scalar means and variances). In terms of (2), one could estimate these parameters over individual patches (local whitening) or over the entire dataset (global whitening). Sec. 5 empirically explores various combinations of (1) and (2) that are computationally feasible (e.g., estimating a full-rank covariance matrix locally for each patch would be too expensive). We find a good tradeoff to be global whitening (to decorrelate features globally), followed by local whitening with a diagonal covariance assumption (e.g., MCNCC).

To understand the value of global and per-patch normalization, we examine the statistics of CNN feature channels across samples of our dataset. Fig. 2 and Fig. 3 show how these per-channel normalizing statistics vary substantially patches and across channels. Notably, for some channels, the normalizing statistics change wildly from patch to patch. This makes local significantly different from global normalization.

One common effect of both global and local whitening is to prevent feature channels which by chance happen to have large means and variances from dominating the correlation score. However, by the same merit this can have the undesirable effect of amplifying the influence of low-variance channels which may not be discriminative for matching. In the next section we generalize both PCA and CCA using a learning framework which can learn channel decorrelation and per-channel importance weighting based on optimizing a discriminative performance objective.

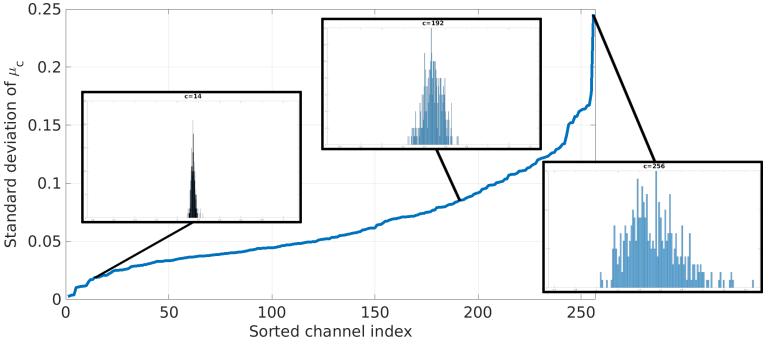

Figure 2: **Distribution of patch channel means:** For each query we match against the database, MCNCC normalizes the 'res2x' features by their individual mean and standard deviation (std). For a uniformly sampled patch, we indicate the normalizing mean for channel c using the random variable μ_c. For each channel, we plot the std of μ_c above. When a channel has the same mean for every channel (small std, left), we can learn global linear transform (*e.g.* PCA or CCA) to normalize it. But, for channels where patches have very different means (large std, right), normalizing by the local (patch) provides additional invariance.

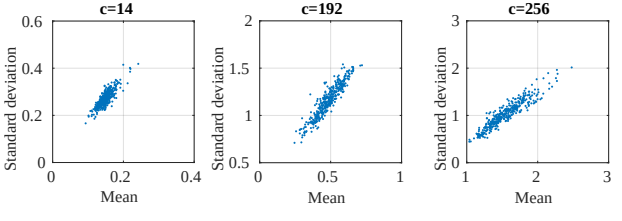

Figure 3: **Normalizing channel statistics:** As shown in the histograms of Fig. 2, for some feature channels, patches have wildly different means and standard deviations. For channel 14 (left), the statistics (and hence normalization) are similar from one patch to the next while for channel 256 (right), means and standard deviations vary substantially across patches. CNN channel activations are positive so means and standard deviations are strongly correlated.

4 Learning

In order to allow for additional flexibility of weighting the relevance of each channel we consider a channel-weighted variant of MCNCC parameterized by vector W

$$MCNCC_W(X,Y) = \sum_{c=1}^{N} W_c \left[\frac{1}{|P|} \sum_{i \in P} \frac{(x_c[i] - \mu_{x_c})}{\sqrt{\sigma_{x_c x_c}}} \frac{(y_c[i] - \mu_{y_c})}{\sqrt{\sigma_{y_c y_c}}} \right] \tag{10}$$

We note that the weighting can undo the effect of scaling by the standard deviation in order to re-weight channels by their informativeness. Furthermore, since the features x, y are themselves produced by a CNN model, we can consider those parameters as additional candidates for optimization. Specifically, PCA/CCA adds extra linear layers prior to the correlation calculation which can be initialized using PCA/CCA and then discriminatively tuned. The resulting "Siamese" architecture is illustrated in Fig. 1.

Siamese loss: To train the model, we minimize a paired regression loss:

$$\arg\min_{W,U,V} \sum_{s,t} \left| Z(X^s, Y^t) - MCNCC_W(\phi_U(X^s), \phi_V(Y^t)) \right|^2 \tag{11}$$

where we have made explicit the function ϕ which computes the deep features of two shoeprints X^s and Y^t, with W, U, and V representing the parameters for the per-channel importance weighting and the linear projections for the two domains respectively. For all pairs, we want the target, $Z(X^s, Y^t)$, to be $+1$ when X^s and Y^t come from the same category and -1 otherwise.

We implement ϕ using a deep architecture, which is trainable using standard backpropagation. Each channel contributes a term to the MCNCC which itself is just a single channel (NCC) term. The operation is symmetric in X and Y, and the gradient can be computed efficiently by reusing the NCC computation from the forward pass:

$$\frac{dNCC(X,Y)}{dx_c[j]} = \frac{1}{|P|\sqrt{\sigma_{x_c x_c}}} (\tilde{y}_c[j] + \tilde{x}_c[j] NCC(X,Y)).$$

We give a complete derivation in the supplement.

5 Experiments

We evaluate the performance of MCNCC on two datasets. The first contains 387 test impressions of shoes and 137 crime scene prints that was collected by the Israel National Police [24]. As this dataset is not publicly available, we use this dataset for the diagnostic analysis and when training and validating learned models. To compare our method to others, we test on the footwear identification dataset (FID-300) [10]. FID-300 contains 1175 test impressions and 300 crime scene prints.

In all of our experiments, we use the 256-channel 'res2bx' activations from a pre-trained ResNet-50 model (http://www.vlfeat.org/matconvnet/models/imagenet-resnet-50-dag.mat). We evaluated activations at other locations along the network, but found those to perform the best.

5.1 Partial Print Matching

In this section we compare MCNCC to two baseline methods: simple unnormalized cross-correlation and cross-correlation normalized by a single μ and σ estimated over the whole 3D feature volume. We note that the latter is equivalent to the cosine distance which is popular in many retrieval applications.

We evaluate these methods in a setup that mimics the occurrence of partial occlusions in shoeprint matching, but focus on a single modality of test impressions. We extract 512 query patches (random selected 97×97 pixel sub-windows) from test impressions that have two or more matching tread patterns in the database. The task is then to retrieve from the database the set of relevant prints. As the query patches are smaller than the test impressions, we search over spatial translations (with a stride of 1), using the maximizing correlation value to represent the test impression. We do not need to search over rotations as all test impressions were aligned to a canonical orientation. When querying the database, the original shoeprint the query was extracted from is removed (*i.e.*, the results do not include the self-match).

Global versus local normalization: The retrieval results showing the tradeoff of precision and recall are shown in Fig. 4. In the legend we denote different schemes in square brackets, where the first term indicates the centering operation and the second term indicates the normalization operation. A $\cdot$ indicates the absence of the operation. μ and σ indicate local (*i.e.*, statistics are estimated *per exemplar*) centering and normalization. μ_c and σ_c indicate local per-channel centering and normalization. $\bar{\mu}_c$ and $\bar{\sigma}_c$ indicate global (*i.e.*, statistics are estimated over the *whole dataset*) per-channel centering and normalization. Therefore, unnormalized cross-correlation is indicated as $[\cdot,\cdot]$, cosine distance is indicated as $[\mu,\sigma]$, and our proposed MCNCC measure is indicated as $[\mu_c,\sigma_c]$.

We can clearly see that using per-channel statistics estimated independently for each comparison gives substantial gains over the baseline methods. Performing global centering and scaling per channel is substantially better than the straight correlation or cosine distance. In general, removing the mean response is far more important than scaling by the standard deviation. Interestingly, in the case of cosine distance and global channel normalization, scaling by the variance actually hurts performance (*i.e.*, $[\mu,\sigma]$ versus $[\mu,\cdot]$ and $[\bar{\mu}_c,\bar{\sigma}_c]$ versus $[\bar{\mu}_c,\cdot]$ respectively). As normalization re-weights channels, we posit that this may be negatively effecting the scores by down-weighing important signals or boosting noisy signals.

Channel decorrelation: Recall that, for efficiency reasons, our multivariate estimate of correlation assumes that channels are largely decorrelated. We also explored decorrelating the channels globally using a full-dimension PCA (which also subtracts out the global mean $\bar{\mu}_c$). The second panel of Fig. 4 shows a comparison of these decorrelated feature channels (solid curves) relative to baseline ResNet channels (dashed curves). While the decorrelated features outperform baseline correlation (due to the mean subtraction) we found that full MCNCC on the raw features performed better than on globally decorrelated features. This may be explained in part due to the fact that decorrelated features show an even wider range of variation across different channels which may exacerbate some of the negative effects of scaling by σ_c.

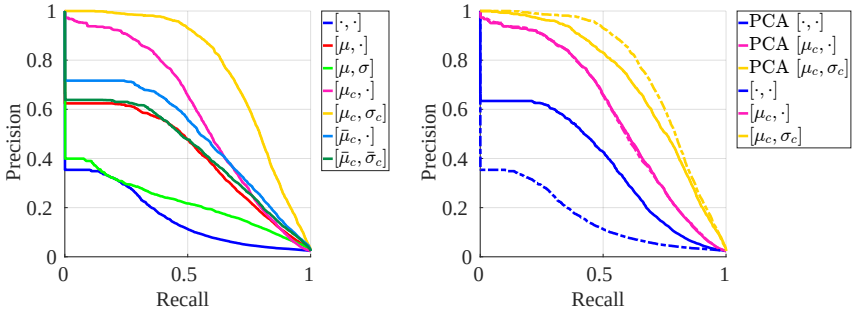

Figure 4: Comparing MCNCC to baselines for image retrieval within the same domain. The methods are denoted by two operations in square brackets: mean subtraction and normalization, respectively. μ and σ denote computing the statistics across all channels, μ_c and σ_c denote computing per-channel statistics, and $\cdot$ denotes the absence of the operation (*e.g.*, MCNCC is denoted as $[\mu_c, \sigma_c]$, whereas cross-correlation is denoted as $[\cdot, \cdot]$. Finally, $\bar{\mu}_c$ and $\bar{\sigma}_c$ denote computing the average per-channel statistics across the dataset. The left panel shows the performance on the raw features, whereas the right panel compares globally whitened features using PCA (solid lines) against their corresponding raw features (dot-dash lines). (Best viewed in color.)

5.2 Cross-Domain Matching

In this section, we evaluate our proposed system in a setting that closely resembles the real world task of retrieving relevant shoeprint test impressions for crime scene prints. The task here is similar to that of the previous section, but now matching is done across domains. Additionally, as the crime scene prints are not aligned to a canonical orientation, we search over both translations (with a stride of 2) and rotations (from -20° to +20° with a stride of 4°). We also compute the local statistics only over the valid support region of the test impression from a predetermined mask associated with each image. The correlation score is similarly computed over the same valid support region P.

As mentioned in Sec. 4, we can learn both the linear projections of the features and the importance of each channel for the retrieval task. We demonstrate that learning is feasible and can significantly improve performance. We use a 50/50 split of the crime scene prints of the Israeli dataset for training and testing, and determine hyperparameters settings using 10-fold cross-validation. In the left panel of Fig. 5 we compare the performance of three different models of varying degrees of learning. The model with no learning is denoted as $[\mu_c, \sigma_c]$, with learned per-channel weights is denoted as $[\mu_c, \sigma_c \cdot W_c]$, and with end-to-end learning of both projection and per-channel weights is denoted as "Siamese." $[\mu_c, \sigma_c \cdot W_c]$ was learned using a regularization value of 0.5 and W_c has 257 parameters (a scalar for each channel and a single bias term). Our Siamese network is learned using a regularization value of 0.05 and has 66K parameters (256 centering, 256^2 projection, 256 channel importance, and 1 bias). We initialize our Siamese network with identity weights for the linear projection and $[\mu_c, \sigma_c \cdot W_c]$ weights for the per-channel importance. As seen in the left panel of Fig. 5, performance increases with more learning. Both $[\mu_c, \sigma_c \cdot W_c]$ and Siamese outperform $[\mu_c, \sigma_c]$ by a large margin, and Siamese outperforms $[\mu_c, \sigma_c \cdot W_c]$ by a modest amount.

We subsequently take $[\mu_c, \sigma_c]$, $[\mu_c, \sigma_c \cdot W_c]$, and our Siamese network (both without retraining), and evaluate them on FID-300 (shown in the right panel of Fig. 5). On FID-300,

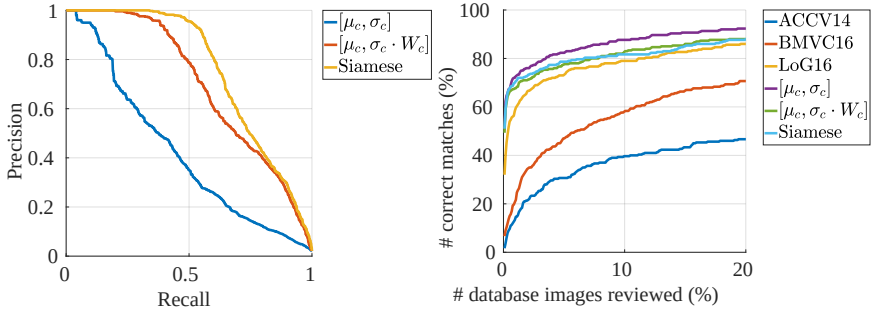

Figure 5: Comparing MCNCC with uniform (denoted as $[\mu_c, \sigma_c]$), learned per-channel weights (denoted as $[\mu_c, \sigma_c \cdot W_c]$), and learned projection and per-channel weights (denoted as Siamese) for retrieving relevant shoeprint test impressions for crime scene prints. The left panel shows our three methods on the Israeli dataset. The right panel compares our proposed system against the current state-of-the-art, as published in: ACCV14 [10], BMVC16 [9] and LoG16 [8] using cumulative match characteristic (CMC).

both MCNCC models with learned weights, performs slightly worse than MCNCC with uniform weights, indicating some overfitting. However, all three variants outperform the current state-of-the-art, once again demonstrating the power of centering and normalizing using the per-channel local statistics.

6 Conclusion

In this work, we proposed an extension to normalized cross-correlation that performs normalization on a per-channel basis. We explain per-channel normalization in terms of local data whitening and compare it to global data whitening using PCA. We use MCNCC as a building block for a Siamese network model that can be trained end-to-end. We show that this performs substantially better than other similarity measures for shoeprint image retrieval within and across domains. We expect our findings here will be applicable to a wide variety of single-shot and exemplar matching tasks using CNN features.

7 Acknowledgement

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through NIST Cooperative Agreement #70NANB15H176.

References

[1] William J Bodziak. *Footwear impression evidence: detection, recovery and examination*. CRC Press, 1999.

[2] Francesca Dardi, Federico Cervelli, and Sergio Carrato. A texture based shoe retrieval system for shoe marks of real crime scenes. *Image Analysis and Processing–ICIAP 2009*, pages 384–393, 2009.

[3] Philip De Chazal, John Flynn, and Richard B Reilly. Automated processing of shoeprint images based on the fourier transform for use in forensic science. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):341–350, 2005.

[4] Robert B Fisher and Peter Oliver. Multi-variate cross-correlation and image matching. In *Proc. British Machine Vision Conference (BMVC)*, 1995.

[5] S Geiss, J Einax, and K Danzer. Multivariate correlation analysis and its application in environmental analysis. *Analytica chimica acta*, 242:5–9, 1991.

[6] Mourad Gueham, Ahmed Bouridane, and Danny Crookes. Automatic recognition of partial shoeprints using a correlation filter classifier. In *Machine Vision and Image Processing Conference, 2008. IMVIP'08. International*, pages 37–42. IEEE, 2008.

[7] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. *Computer Vision–ECCV 2012*, pages 459–472, 2012.

[8] Adam Kortylewski. private communication, 2017.

[9] Adam Kortylewski and Thomas Vetter. Probabilistic compositional active basis models for robust pattern recognition. In *British Machine Vision Conference*, 2016.

[10] Adam Kortylewski, Thomas Albrecht, and Thomas Vetter. Unsupervised footwear impression analysis and retrieval from crime scene data. In *Asian Conference on Computer Vision*, pages 644–658. Springer, 2014.

[11] Henry C Lee, Robert Ramotowski, and RE Gaensslen. *Advances in fingerprint technology*. CRC press, 2001.

[12] Fei-Fei Li, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[13] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.

[14] Kantilal V Mardia, John T Kent, and John M Bibby. *Multivariate analysis (probability and mathematical statistics)*. Academic Press London, 1980.

[15] Nick Martin and Hermine Maes. *Multivariate analysis*. Academic press, 1979.

[16] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[17] Pradeep M Patil and Jayant V Kulkarni. Rotation and intensity invariant shoeprint matching using gabor transform with application to forensic science. *Pattern Recognition*, 42(7):1308–1317, 2009.

[18] Maria Pavlou and Nigel Allinson. Automatic extraction and classification of footwear patterns. *Intelligent Data Engineering and Automated Learning–IDEAL 2006*, pages 721–728, 2006.

[19] Juliet Popper Shaffer and Martin W Gillo. A multivariate extension of the correlation ratio. *Educational and Psychological Measurement*, 34(3):521–524, 1974.

[20] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[21] Yi Tang, Sargur N Srihari, Harish Kasiviswanathan, and Jason J Corso. Footwear print retrieval system for real crime scene marks. In *International Workshop on Computational Forensics*, pages 88–100. Springer, 2010.

[22] Chia-Hung Wei and Chih-Ying Gwo. Alignment of core point for shoeprint analysis and retrieval. In *Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on*, volume 2, pages 1069–1072. IEEE, 2014.

[23] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1249–1258, 2016.

[24] Yoram Yekutieli, Yaron Shor, Sarena Wiesner, and Tsadok Tsach. Expert assisting computerized system for evaluating the degree of certainty in 2d shoeprints. Technical report, Technical Report, TP-3211, National Institute of Justice, 2012.

[25] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[26] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015.

[27] Lin Zhang and Nigel Allinson. Automatic shoeprint retrieval system for use in forensic investigations. In *UK Workshop On Computational Intelligence*, 2005.

Supplementary Material: Cross-Domain Forensic Shoeprint Matching

Bailey Kong[1]
bhkong@ics.uci.edu
James Supancic, III[1]
jsupanci@uci.edu
Deva Ramanan[2]
deva@andrew.cmu.edu
Charless Fowlkes[1]
fowlkes@ics.uci.edu

[1] Dept. of Computer Science
University of California, Irvine
United States of America
[2] Robotics Institute
Carnegie Mellon University
United States of America

1 Backpropagation for NCC

Here we derive $\frac{dNCC(x,y)}{dx}$. NCC is symmetric with respect to x and y so $\frac{dNCC(x,y)}{dy}$ is the same. NCC is a sum of terms over individual pixels i. Defining

$$NCC(x,y) = \frac{1}{|P|}\sum_i r_i(x,y)$$

we derive $\frac{dNCC(x,y)}{dx}$ by taking the total derivative:

$$\frac{dNCC(x,y)}{dx[j]} = \frac{1}{|P|}\sum_{i\in P}\frac{dr_i(x,y)}{dx[j]} \tag{1}$$

$$\frac{dr_i(x,y)}{dx[j]} = \tilde{y}[i]\frac{d\tilde{x}[i]}{dx[j]} \tag{2}$$

$$= \tilde{y}[i]\left(\frac{\partial\tilde{x}[i]}{\partial x[j]} + \frac{\partial\tilde{x}[i]}{\partial\mu_x}\frac{\partial\mu_x}{\partial x[j]} + \frac{\partial\tilde{x}[i]}{\partial\sigma_{xx}}\frac{\partial\sigma_{xx}}{\partial x[j]}\right) \tag{3}$$

The partial derivative $\frac{\partial\tilde{x}[i]}{\partial x[j]} = \frac{1}{\sqrt{\sigma_{xx}}}$, if and only if $i = j$ and is zero otherwise. The remaining partials derive as follows:

$$\frac{\partial\tilde{x}[i]}{\partial\mu_x} = -\frac{1}{\sqrt{\sigma_{xx}}} \qquad \frac{\partial\mu_x}{\partial x[j]} = \frac{1}{|P|} \tag{4}$$

$$\frac{\partial\tilde{x}[i]}{\partial\sigma_{xx}} = \frac{1}{2\sigma_{xx}^{3/2}}(x[i]-\mu_x) \qquad \frac{\partial\sigma_{xx}}{\partial x[j]} = \frac{2\,(x[j]-\mu_x)}{|P|} \tag{5}$$

Pulling everything together, we arrive at a final expression:

$$\frac{dNCC(x,y)}{dx[j]} = \frac{\tilde{y}[j]}{|P|\sqrt{\sigma_{xx}}} + \frac{1}{|P|}\sum_{i\in P}\tilde{y}[i]\left(\frac{-1}{|P|\sqrt{\sigma_{xx}}} + \frac{2\,(x[i]-\mu_x)\,(x[j]-\mu_x)}{2|P|\sigma_{xx}^{3/2}}\right) \tag{6}$$

$$= \frac{1}{|P|\sqrt{\sigma_{xx}}}\left(\tilde{y}[j] + \frac{1}{|P|}\sum_{i\in P}\tilde{y}[i]\left(-1 + \frac{(x[i]-\mu_x)\,(x[j]-\mu_x)}{\sigma_{xx}}\right)\right) \tag{7}$$

$$= \frac{1}{|P|\sqrt{\sigma_{xx}}}\left(\tilde{y}[j] - \frac{1}{|P|}\sum_{i\in P}\tilde{y}[i] + \frac{1}{|P|}\sum_{i\in P}\tilde{y}[i]\tilde{x}[i]\tilde{x}[j]\right) \tag{8}$$

$$= \frac{1}{|P|\sqrt{\sigma_{xx}}}(\tilde{y}[j] + \tilde{x}[j]NCC(x,y)) \tag{9}$$

where we have made use of the fact that $\tilde{y}$ is zero-mean.

2 Retrieval Results for Cross-Domain Matching

In this section we take a deeper look at our proposed system for the cross-domain matching problem, first looking at some qualitative results and second looking at its performance w.r.t. the size of crime scene prints.

In Fig. 2 and Fig. 3 we show the top-10 retrieved test impressions for a subset of crime scene prints from FID-300. These results correspond to $[\mu_c, \sigma_c]$ and $[\mu_c, \sigma_c \cdot W_c]$ of the right panel of Fig. 5 in the main paper.

Next we shows the performance of our two methods when evaluating crime scene prints of certain sizes. The size can give us an estimate of how much information is in the print—as crime scene prints and test impressions in FID-300 were scaled to a canonical size (10 pixels is 1 cm). For FID-300, the prints all fell into one of two categorizes which we call "full size" and "quarter size". "Full size" prints are crime scene prints whose pixel area is at least 90% of the corresponding test impression, whereas "quarter size" prints are crime scene prints whose pixel area is at most 25% of the corresponding test impression. 76 of the 300 crime scene prints were full size, while 224 of the 300 crime scene prints were quarter size.

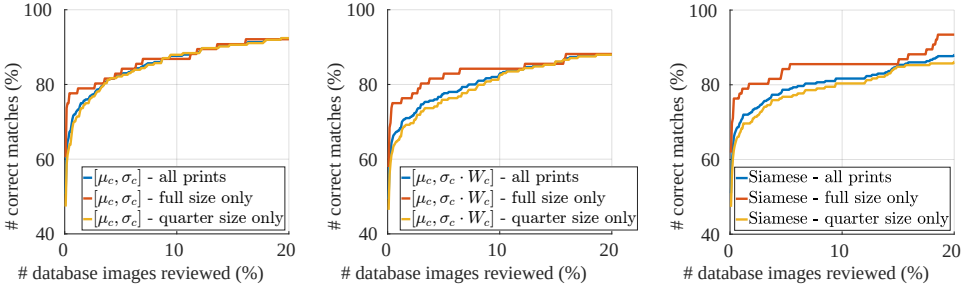

Figure 1: A study on the performance of our proposed methods on different sizes of crime scene prints in FID-300. The left panel shows the performance of MCNCC with uniform weights, the middle panel shows the performance of MCNCC with learned per-channel weights, and the right panel shows the performance of our Siamese network.

Fig. 1 shows us that, unsurprisingly, our system performs better on full size prints than on quarter size prints. This fits our intuition as larger prints will almost surely have more

information to discriminate with. We also see our proposed system with learned per-channel weights performs worse than uniform weights on both categories. On the other hand, our Siamese network performs slightly better on full size prints but much worse on quarter size prints compared to uniform weights. These results reaffirm our suspicion that there is some overfitting in the models where we learn weights.

Figure 2: FID-300 retrieval results for $[\mu_c, \sigma_c]$. The left column shows the query crime scene prints. Green boxes indicate the corresponding ground truth test impression.

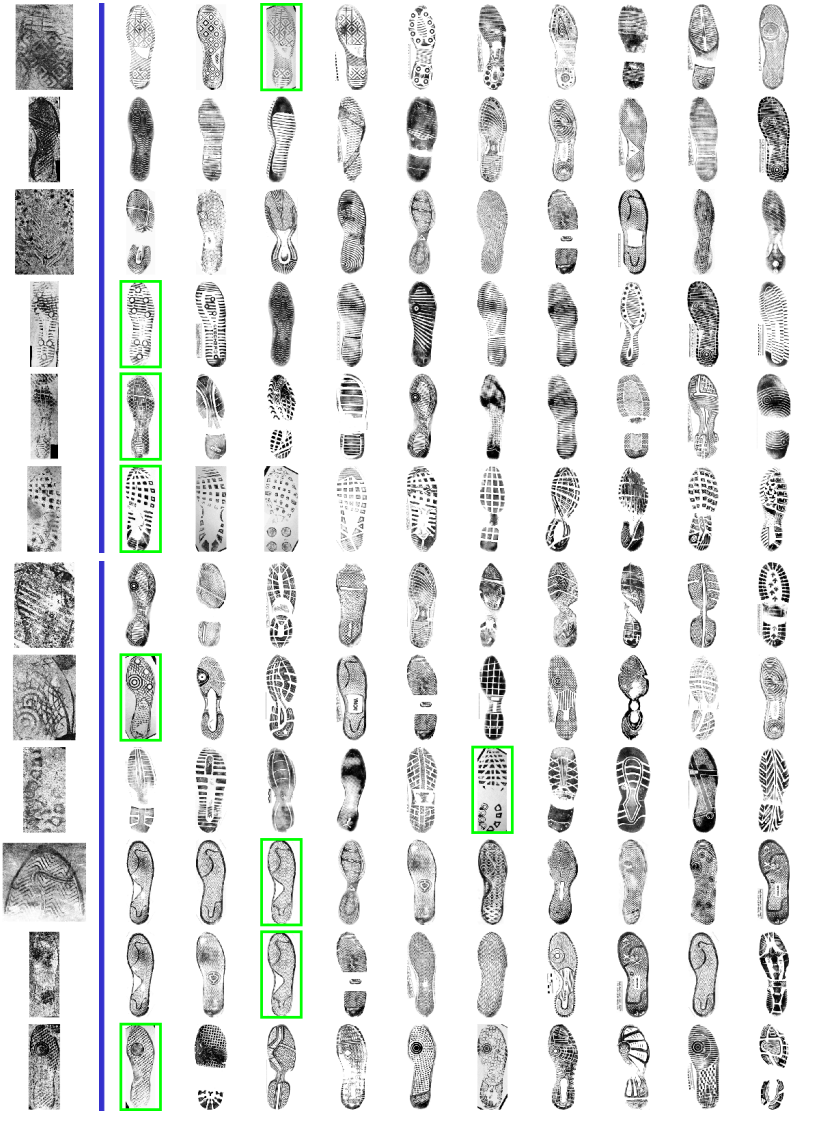

Figure 3: FID-300 retrieval results for $[\mu_c, \sigma_c \cdot W_c]$. The left column shows the query crime scene prints. Green boxes indicate the corresponding ground truth test impression.

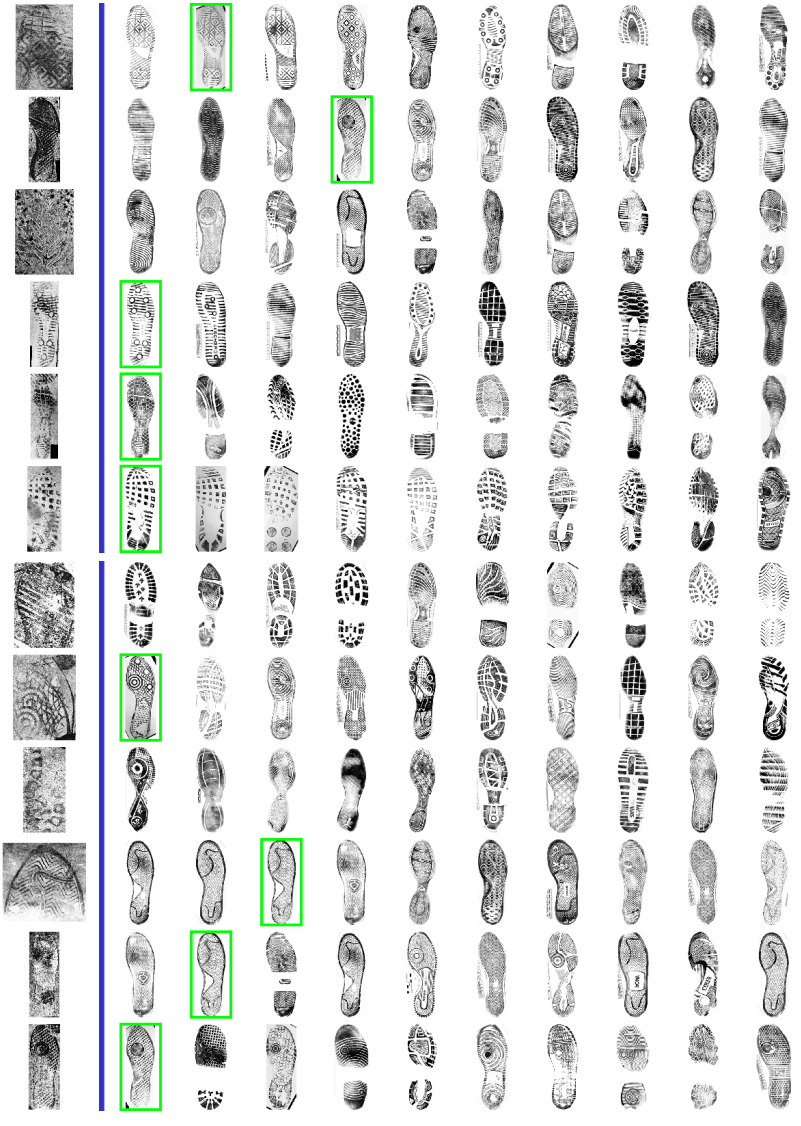

Figure 4: FID-300 retrieval results for "Siamese." The left column shows the query crime scene prints. Green boxes indicate the corresponding ground truth test impression.