# Towards Zero-Shot Alignment and Retrieval for Forensic Detection

**Vidhi Jain**[*]
vidhij@andrew.cmu.edu

**Adithya Pratapa**[*]
vpratapa@andrew.cmu.edu

**Amrith Setlur**[*]
asetlur@andrew.cmu.edu

## Abstract

In a crime scene, shoeprints are considered as an important evidence. The probe shoeprint (query) needs to be matched against reference images from a large database of shoe impressions. This is a computationally challenging task as the probe image often contains partial patterns and is heavily induced with noise. In this paper, we investigate approaches for image retrieval with very limited labeled data. We propose our baselines and discuss various approaches including synthetic data generation for data augmentation. Particularly, we describe and compare different methods that (1) leverage class hierarchies (2) learn a distance metric (3) adaptively perform canonical correlation analysis (CCA) (4) learn to ignore noisy patterns through supervised alignments.

## 1 Introduction

Imagine you are at a crime scene and trying to match the shoeprint of the criminal to that of the suspects. Given a large database of outsole tread patterns, how shall one figure out which reference shoeprint it aligns the most to? Crime scene probe images are often partial and filled with unwanted textures of soil, blood, snow, etc.

With few (or zero) labeled samples of these probe images, we aim to match and align the probe shoeprint to its reference shoe tread. Our contributions are as follows (1) we propose both supervised and unsupervised baselines for alignment techniques, (2) we utilize off-the-shelf image features [8] for few-shot learning methods, (3) we identify structural priors in the label space and explore its usage in learning more robust classification boundaries, (4) we evaluate and analyze various metric learning approaches [16], (5) we propose an EM based algorithm for canonical correlation analysis (CCA), (6) and finally we propose a Noise Removal method that fine-tunes the learnt image features for the probe image features so as to capture the shoeprint texture instead of the noisy background patterns.

### 1.1 Dataset

Footwear Impression Database (FID-300) [12] contains 300 probe images and 1175 reference impressions. These limited probe images correspond to just 130 unique reference classes out of 1175. We split the probe dataset between train and test by 130 reference class labels. We randomly pick 25 reference classes ($\approx 20\%$ of 130) which turned out to correspond 76 probe images out of 300. We evaluate the top-k accuracies on a test set of size 224 corresponding to remaining 104 reference classes. For the remaining 104 classes in probe set, we have never seen a probe image and we are aiming to address the problem in a zero-shot manner.

[*] Equal Contribution

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

### 1.2 Synthetic Data Generation

We investigate ways to generate the probe-like synthetic data generation, by using standard data augmentation techniques and also texture blending with Describable Textures Dataset (DTD) dataset [1]. For robust classification and clustering, we generate synthetic data using standard data augmentation techniques like random crop, flip, scale, shear, etc. For synthetic probe-looking images, we blend the reference images with textures from DTD dataset. We also superimpose the same shoeprint more than once to produce a smudged texture (Refer to Appendix A for samples).

## 2 Our Baselines

### 2.1 Unsupervised approach: Basis Projection

The feature maps of the shoeprint images obtained from ResNet-34 are high dimensional vectors but we hypothesize that the images lie on a lower dimensional manifold. To substantiate this, we analyzed the singular values obtained from the SVD decomposition of the reference features $\mathbf{X}_R$. From Fig. 1 we see that the magnitude of the singular values, that corresponds to variance of different dimensions of $X_R$, drops very quickly. Hence, we aim to project these images on to a subspace with a smaller basis set comprised of a few “important" basis vectors that capture significant variance.

Let $M \in \mathbb{R}^{k\times p}$ be the projection matrix obtained through dimensionality reduction for $\mathbf{X}_R \in \mathrm{R}^{n\times k}$ onto a basis of size $p$, i.e. they $\mathbf{M} \in \mathrm{R}^{k\times p}$ consists of the first $p$ eigen vectors obtained from the eigen value decomposition of the scaled empirical covariance $X_R^T X_R = UDU^T$. The projected matrix $\hat{\mathbf{X}}_R$ is given by $\hat{\mathbf{X}}_R = \mathbf{X}_R\mathbf{M}$. At inference, for a probe image $\mathbf{x}_P \in \mathrm{R}^k$ we obtain the projected features $\hat{\mathbf{x}}_P = \mathbf{x}_P^T\mathbf{M}$. We then use the cosine similarity (dot product) distance metric to find its closest reference images in the lower dimensional sub-space with the basis given by $\mathbf{M}$. Finally, we compute the rank of each reference (for a given probe) and plot the cumulative distribution (CDF) plot of the ranks of the true references for all probe images.

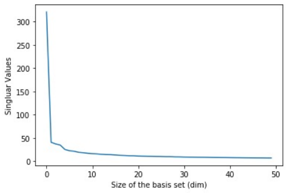

Figure 1: Few singular vectors for reference images captures most of the variance

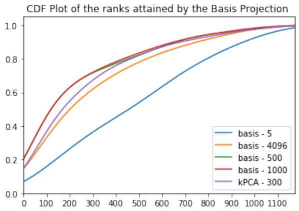

Figure 2: CDF of the ranks by Basis Projection

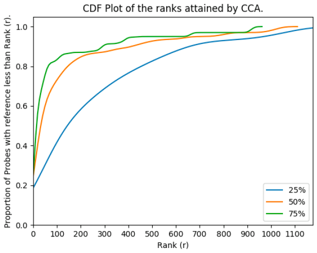

Figure 3: CDF of the ranks by CCA

From Fig. 2, we can note that having 5 basis vectors leads to a significant loss of information and for very high values like $4096$, the noise induced by the dimensions with low eigen values leads to a drop in performances. We also implemented the kernel-PCA [2] but didn't achieve any noticeable gains with it.

### 2.2 Supervised approach: Canonical Correlation Analysis

The reference and probe images are drawn from different distributions. The ResNet features for paired samples would most likely not correlate. Drawing from [11], we aim to find the a pair of orthonormal linear transformations $\mathbf{U}$ for $\mathbf{X}_R$ and $\mathbf{V}$ for $\mathbf{X}_P$ such that $\mathbf{UX}_R$ and $\mathbf{VX}_P$ would be have the maximum correlation.

$$\rho = \frac{Cov(\mathbf{UX}_R, \mathbf{VX}_P)}{\sqrt{Var(\mathbf{UX}_R)}\sqrt{Var(\mathbf{VX}_P)}} = \frac{(\mathbf{UX}_R)^T(\mathbf{VX}_P)}{\sqrt{(\mathbf{UX}_R)^T(\mathbf{UX}_R)}\cdot\sqrt{(\mathbf{VX}_P)^T(\mathbf{VX}_P)}} \tag{1}$$

Further, $\mathbf{UX}_R$ is fed as a feature matrix to the classification network. Here, $\mathbf{X}_R$ refers to all reference images including synthetic data. We hold out $50$ probe images out of $300$ to evaluate the alignment learnt by CCA . Fig 3 indicates the proportion of the probe data used for alignment to evaluate on

[2] rbf kernel, $\gamma = 0.001$

this held-out dataset. In this case, we see that for a good accuracy, we need to utilize almost all the alignments between the probe & reference images. Hence, this method is not very interesting for our problem and we only explore it to study the possibility of using a variation of CCA which would require fewer aligned samples.

# 3 Proposed Methods

## 3.1 Identifying structural prior in the reference class labels

The database has several reference images that look very similar. We can use this structure such that while learning to associate a probe image to a particular reference impression, we penalize the model less for a similar reference image.

**Hierarchical Clustering:** We investigated the hierarchical clusters formed by reference feature maps obtained from the pre-trained neural networks. On visual inspection, we found the clusters to capture meaningful patterns in the footprint images. Refer Fig 12 (in Appendix) to see that the shoeprints with dense empty circles get clustered together earlier than with prints with filled dark circles. We experimented with different linkage methods and found complete linkage with euclidean distance to be most informative. We also experimented with k-means++ but the hierarchical information is most relevant to our task.

**Hierarchical Softmax:** In order to use the latent hierarchy present in the reference classes instead of using one-hot vectors to represent the targets, we use the cluster tree for the target classes. The cluster tree is obtained using hierarchical clustering as described above and is constructed with the metric *euclidean* and linkage type as *complete*. The tree constructed from $N = 1175$ classes has $N - 1$ edges. Each node has at most two edges: *left* (0), *right* (1). Since we know the path from root to leaf for each reference, the classification *left vs right* is enforced for every internal node on thee path.

## 3.2 Robust Embeddings

Our probe images contain high noise compared to the 1175 reference footprints. We first experimented with a simple model consisting of pre-trained ResNet-34 with a classifier layer on the top. We found this model to generalize poorly to the probe data owing to high noise. Therefore, we propose to modify the embedding space to deal better with noise by making use of our data augmentation techniques.

**Metric Learning:** In Section. 1.1, we presented various transformations that can help with data augmentation in our low-resource setting. We want our embedding space to capture the similarity between original reference images to their noisy versions, while differentiating between reference footprints. For this purpose, we perform similarity learning to maximize and minimize the intra-label and inter-label similarity respectively. We used an open-source metric learning library [3] for our experiments. In a recent work, [16] proposed Multi-similarity loss to improve over traditional losses like TripletMargin [10] and Contrastive [7] with a special weighting scheme. Their MS-loss is given below,

$$L = \frac{1}{m}\sum_{i}\left[\frac{1}{\alpha}log\Big[1 + \sum_{k\in P_i} e^{-\alpha(S_{ik}-\lambda)}\Big] + \frac{1}{\beta}log\Big[1 + \sum_{k\in N_i} e^{-\beta(S_{ik}-\lambda)}\Big]\right]$$

where, $P_i$ and $N_i$ denote the sets of positive and negative pairs for the anchor $i$, and $S_{ik}$ being the similarity between $i$ and $k$.

Our architecture consists of a trunk model, an embedder model and a classifier network. We used ResNet-34 as our trunk, with embedding layer being a two-layer fully connected network (4096 → 2048 → 512). We train our model with data augmented reference labels with scaling transforms and alpha blending. For every reference footprint, we use 10 randomly selected images for each of the above transforms. We experimented with standard metric losses like TripletMargin and Contrastive but found the best performance with Multi-similarity loss. Along with the pair weighting, the original paper also proposes a pair mining scheme but we didn't find mining particularly helpful in our task.

We jointly optimize for metric loss and classifier loss with weights being, 3:1. We fine-tuned for the hyper-parameters ($\alpha$, $\beta$) in multi-similarity loss and found the best performance on validation set

[3] `https://github.com/KevinMusgrave/pytorch_metric_learning`

with $\alpha$=10 and $\beta$=2. Along with embedding and classifer layers, we found fine-tuning the ResNet trunk to be helpful.

### 3.3 Adaptive CCA : Alignment with reduced sample complexity

In Sec. 2.2 we saw that if we were given a significantly large number of alignments for probe images and their true references it would be easy to use an alignment algorithm like CCA to learn transformations that maximally align the basis of the probe and reference subspaces. In this section we explore an E-M based approach to infer the alignments of probe images while estimating the transformation matrices simultaneously. Our algorithm infers the alignment labels and performs the CCA alignment in an iterative way. The transformation matrices are constantly adapting (M-step) to the inferred alignments (E-step). Hence we coin the name of our algorithm as "Adaptive CCA". Yger et al. [17] also propose an "Adaptive CCA" approach but they focus on estimating transformation matrices and satisfying certain matrix constraints iteratively. Instead in our formulation, we focus on estimating the alignment for the probes to their true references while updating the transformations.

Let $y_r$ be the reference class label & $\mathbf{x}_r = f_\theta(\mathcal{I}_r)$ represent the ResNet-34 features for reference image $\mathcal{I}_r$. Let $\hat{\mathcal{I}}_r$ sampled from the synthetic dataset be one of the probe-looking distortions for $\mathcal{I}_r$. For a probe image features $\mathbf{x}_p = f_\theta(\mathcal{I}_p)$, we aim to estimate its label probabilities (E-step) and then use it to minimize the classification and alignment objective (M-step). Finally, we use $\tilde{\mathbf{x}_i}$ to represent a lower dimensional projection of image features $\mathbf{x}_i$.

#### 3.3.1 E-step : Estimate posterior $q(z|\mathbf{x}_p)$

In order to infer the alignment for a probe image we compute the distribution $q(z|\mathbf{x}_p)$ over the reference images. For this we model the distribution $q(\mathbf{x}_p|z)$ using non-parametric kernel density estimation [6] and assume a uniform prior $q(z)$ over the reference classes. $q(z|\mathbf{x}_p)$ would then be given by the Bayes rule in Eq. 2.

Once we have $q(z|\mathbf{x}_p)$ we use it to compute the convex combination of the reference features $\mathbf{x}_r$, given by $g(\mathbf{x}_p)$. If we consider the case where we know the true alignment $y_p$ for the probe image $\mathcal{I}_p$, the discrete distribution $q(z|\mathbf{x}_p)$ would resolve to point mass with $q(y_r|\mathbf{x}_p) = 1.0$, in which case $g(\mathbf{x}_p) = \mathbf{x}_{y_p}$. And this would lead to a solution given by the closed form CCA. Hence the method we propose is generic and can be applied to a large class of alignment problems. Since non-parametric estimates suffer from the curse of dimensionality [6] we estimate the class conditional distribution $q(z|\mathbf{x}_p)$ using projected features $\hat{\mathbf{x}}$. In Eq. 3 $K_h$ is a Gaussian kernel of bandwidth $0.001$ and $z$ is the estimated soft label.

$$q(z|\mathbf{x}_p) = \frac{q(\mathbf{x}_p|z) \cdot q(z)}{q(\mathbf{x}_p)} \qquad q(z|\mathbf{x}_p) = \frac{q(\mathbf{x}_p|z)}{\sum\limits_{z'} q(\mathbf{x}_p|z')} \tag{2}$$

$$q(z|\mathbf{x}_p) = \frac{\sum_{i \in y_r} K_h(\tilde{\mathbf{x}_p} - \tilde{\mathbf{x}_\mathbf{i}})}{\sum_j \sum_{j \in y_r} K_h(\tilde{\mathbf{x}_p} - \tilde{\mathbf{x}_j})} \quad ; \quad g(\mathbf{x}_p) = \sum_r \mathbf{x_r} \cdot q(z = r|\mathbf{x}_p) \tag{3}$$

#### 3.3.2 M-step: Learn the alignment and classification parameters

$$\min_{\mathbf{U},\mathbf{V},\theta} -\log p_\theta(y_r|\mathbf{U}\mathbf{x}_r) - \log p_\Theta(y_r|\mathbf{V}\hat{\mathbf{x}}_r) + \textstyle\sum_p ||\mathbf{U}g(\mathbf{x}_p) - \mathbf{V}\mathbf{x}_p||_2^2 \tag{4}$$

$$\text{st.} \;\; \mathbf{U}^T\mathbf{U} = \mathbf{I} \;\; \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{5}$$

In Eq.4 $\theta$ are the parameters involved in feature representation and classification, $\mathbf{U}$, $\mathbf{V}$ are the matrices learnt for alignment. The orthonormal constraints are enforced using Lagrange multipliers, i.e. as an auxiliary loss.

In the E-step we had sampled a batch of reference and probe images. For the probe image $\mathbf{x}_p$ we had estimated the posterior $q(z|\mathbf{x}_p)$. All the parameters of the classifier and ResNet (feature extractor) were frozen. Since we begin training with a few known alignments (for the 76 probe images in our training data) we had to only infer the alignment for the rest of the (224) probe samples. In the M-step step, with the knowledge of the true alignments (for the 76 images) and the inferred alignments ($g(\mathbf{x}_p)$) for the rest, we compute the classification and alignment loss. Finally, we use Gradient Descent to update the model parameters before the E-step of the next iteration.

### 3.4 Noise Removal Network (NRN): learning to capture the shoeprint texture amidst structured noise

The reference images are clean aligned imprints while the probe images have complex distortions and occlusions. As image features are biased towards textures [4], we consider separate networks to represent both types of images.

We obtain reasonable features for the reference images as their images have one consistent texture throughout. This is evident with basis projection too in sec 2.1. As the probed images capture more than one texture: the shoeprint and structured noise textures. Therefore, we aim to fine-tune this network to be biased towards representing shoeprints more than the noise textures.

We learn the noise removal network (NRN) using a few probe images and probe-like synthetic data. We obtain cosine similarity between the projected probe image features with the all reference class projected features, and expect to have maximum similarity value aligned to its respective groundtruth reference.

The projection matrix is used to initialize the parameters of the final layer in the NRN, while rest of the model is initialized with pre-trained ResNet-34 features. This ensures that we obtain results at least as in basis projection baseline (Section 2.1). We improve over this baseline's performance by fine-tune the NRN parameters with cross entropy loss.

Otherwise, we observe a local minima setting where the loss function decreases rapidly without much change in the top-k accuracies.

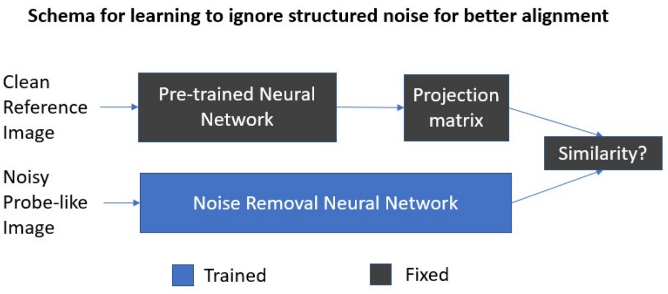
Schema for learning to ignore structured noise for better alignment
Clean Reference Image
Pre-trained Neural Network
Projection matrix
Similarity?
Noisy Probe-like Image
Noise Removal Neural Network
Trained
Fixed

Figure 4: Schema for noise removal

## 4 Background and Related Works

MCNCC [11] use normalized cross-correlation for CNN features, and perform normalization on a per-channel and per-exemplar basis. As PCA/CCA can be seen as a linear layer in neural network, it is used to initialize the weights of the CNN features. These weights are later learned by Siamese loss on the calculated correlation of the image features. MCNCC used Israeli shoeprint dataset which provided good alignment while we obtain similar results with a smaller dataset part: 75% of 250 probe data, evaluated on the 50 held out set.

CABM [12] use probabilistic learning process to learn the even and odd Gabor wavelets as basis filters, a linear additive model can be applied on ensemble of image patches in a greedy EM-type clustering. However, these model work on the assumption of fixed pose of the object.

Although handcrafted features worked well for rotation and translation invariance like in [2], their assumptions came out too restricted for real-world images. The data-driven CNN feature maps seems to be robust to these situation, provided sufficient training data. The idea of metric learning has been extensively studied in the literature [9; 13; 14] including one the prior works on footprint recognition, MCNCC work [11]. A survey [15] shows several previous work in shoeprint identification task that study the distortions of noise, rotation and translation primarily. Along with these standard data augmentation techniques, we also explore synthetic data creation with texture blending reference prints with their same shoeprint to capture the smudging effect and amidst different textures in DTD dataset by [1].

# 5 Results

Table 1 shows the top-k accuracy of the various techniques described in Sec. 3. We compute accuracy over 1175 reference images (or class labels) for all 224 probe images in our test set. The rest 76 out of the 300 probe images were either used for fine-tuning or validating the methods. Results from prior works [12; 11] are not directly comparable with the accuracies for any of our methods since they train their modules on a much higher number of labeled samples from FID-300. Additionally, they also have labeled samples of shoeprints from external sources[4].

Table 1: Top-k Accuracies for Shoeprint Image Matching. † with large private dataset * with top 100 eigenvectors.

| Method | Top-1 | Top-5 | Top-50 | Top-100 | Top-500 |
|---|---|---|---|---|---|
| Prior Work | | | | | |
| MCNCC† | 0.45 | 0.49 | 0.78 | 0.85 | - |
| CABM | 0.07 | 0.10 | 0.42 | 0.58 | - |
| Our Baselines | | | | | |
| Basis Projection* | 0.08 | 0.15 | 0.35 | 0.47 | 0.78 |
| Our Methods | | | | | |
| Adaptive CCA | 0.03 | 0.07 | 0.45 | **0.69** | **1.00** |
| Hierarchical S-max | 0.05 | 0.14 | 0.27 | 0.39 | 0.75 |
| Metric Learning | **0.10** | **0.24** | 0.46 | 0.57 | 0.84 |
| Noise Removal Network* | **0.12** | **0.26** | **0.56** | **0.71** | **0.94** |

The Noise Removal Network has higher accuracies in general since we refrain from making strong assumptions while modelling it. The data augmentation from our curated synthetic dataset aids in the end-to-end training of the NRN. Additionally, since we compute the similarity of the probe and reference features in a lower dimensional space the network requires fewer samples/epochs to reach a local optima [5].

For Adaptive CCA, although we observe good accuracies ($\geq 0.7$) for higher values of $K$, the training is unstable and on several occasions a posterior collapse is observed. Also, since we don't enforce the algorithm to output posterior distributions with low entropy, during inference the retrieval mechanism finds multiple reference images (for the same probe) with similar confidence values. This leads to low accuracy values for $K \leq 10$.

Hierarchical S-max certainly converges faster and has higher top-K accuracy as compared to traditional softmax but it struggles to beat our unsupervised baseline Basis Projection. The retrieval for this method is most interpretable since we can clearly see which path (in the hierarchical clustering tree) is traversed. So it is easier to observe if there was an error made near a leaf node or close to the root node itself.

Adding a metric-loss objective to our model, we are able to achieve competitive results for all k reinforcing the need for capturing similarity among noisy synthetic data. In metric learning, we are aiming to make our embedding space to be robust to noise but we are not accounting for structural similarity between reference footprints (Sec. 3.1). One interesting future direction would be to take advantage of this structural prior. This can potentially done by using the hierarchical structure in the classifier like in hierarchical softmax.

# 6 Discussion

## 6.1 Analysis of Metric Learning

We have evaluated our various model and parameter choices on held-out validation probe set and we provide a detailed analysis below and compare it to our best model of joint optimization of metric loss and classifier loss, with multi-similarity loss being the metric loss and scaling, blending as our transforms and fine-tuning ResNet-34. Traditionally, similarity learning methods are evaluated through k-NN on the global embedding space. Since we are using multi synthetic images for each

[4] https://forensicstats.org/shoeoutsoleimpressionstudy/

original reference, this evaluation would be less informative for our ranking task. Therefore, we evaluated our methods via the output logits. To make sure our learnt embedding space maps the different noisy versions of our reference set close-by, we performed the nearest neighbor evaluation on original reference footprints (not directly used in train) and as we expect, we get near perfect precision @ $k = 20$.

**Joint Training:** We experiment with three different optimization functions, only metric loss, only classifier loss and joint metric, classifier loss. Given the nature of our task, we always use a classifier loss. Table. 2 shows the performance of the two variants. The overall mean rank was 201 for cross-entropy loss but 175 with joint loss training, showing a significant increase in overall performance.

Table 2: Model Selection in Metric Learning (numbers of validation set)

| | **Top-1** | **Top-5** | **Top-50** | **Top-100** |
|---|---|---|---|---|
| Best model | 0.09 | **0.29** | **0.53** | **0.66** |
| w/o metric loss | | | | |
| only cross-entropy loss | **0.16** | 0.25 | 0.47 | 0.54 |
| other metric losses | | | | |
| w. contrastive loss | 0.08 | 0.21 | 0.37 | 0.45 |
| other transforms | | | | |
| $scale(\text{im})$ | 0.05 | 0.09 | 0.29 | 0.36 |
| $crop(\text{im})$ | 0.00 | 0.01 | 0.21 | 0.29 |
| $blend(\text{im})$ | 0.08 | 0.21 | 0.45 | 0.57 |
| $crop(scale(blend(\text{im})))$ | 0.04 | 0.07 | 0.28 | 0.37 |
| freezing ResNet | | | | |
| Frozen | 0.04 | 0.08 | 0.26 | 0.37 |

**Choice of Metric Loss:** Along with the multi-similarity loss, we experimented with triplet margin and contrastive losses. Table. 2 presents the metrics when we use contrastive loss instead of multi-similarity loss (as in Best). We found triplet-margin loss to be poor. We think the crucial aspect for a metric loss function in our setting is appropriate weighting of positive and negative pairs, therefore, contrastive and multi-similarity losses are more suitable to our task.

**Data Augmentation:** The three types of image transformation proposed in Section. 1.1 can have different impact on the final performance. To this end, we created a fixed number of synthetic images (10) for each of the scale, crop and alpha-blend transformations. We trained our metric learning framework on this data and evaluated it on held-out validation probe set to see the individual impact of each. Using $scale(\text{im}) \cup blend(\text{im})$ gives the best performance as reported in Best.

**Fine-tuning ResNet:** We also experimented with fine-tuning our pre-trained trunk model, ResNet-34 on the noise augmented training data. Table. 2 compares the metrics for best and frozen model, and we clearly see significant gain in performance with fine-tuning. ResNet-34 is originally trained on generic and clean image corpus, and we believe it is therefore necessary to fine-tune the weights to adapt to our noisy domain.

### 6.2 Analysis of Adaptive CCA

To infer the predicted class values for the probe images in the test set, we run the Adaptive CCA EM algorithm for 150 epochs. Figures 7, 8, 9 gives the posterior distribution $q(z|\mathbf{x}_p)$ for epochs 0, 30 and 80 respectively. The given progression is for noisy image from reference class 320. We observe that this distribution over reference labels is almost uniform in the beginning. As the iterations of the EM algorithm progress we can see that the distribution becomes peakier. With a certain probability, after 150 epochs the latent class $z$ converges in distribution to the point mass at value 320. The maximization step at this stage would be equivalent to that of the supervised CCA. This is substantiated by the comparable performance given by the Adaptive CCA model which is trained on fully labeled data.

### 6.3 Analysis of Noise Removal Method

The basis projection is unsupervised selection of a ordered set of basis along maximum variance in the data. This set may or may not help in the downstream task of matching shoeprints.

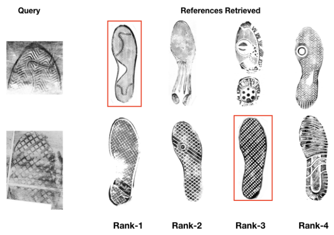

Figure 5: Visual image retrieval results by Adaptive CCA. The red boxes indicate the true references for a query.

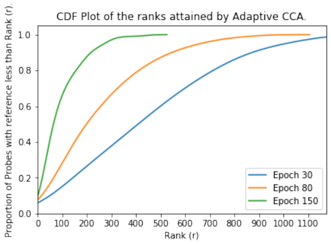

Figure 6: CDF Plot of the ranks of the true reference classes for every probe image in the test set (across EM epochs).

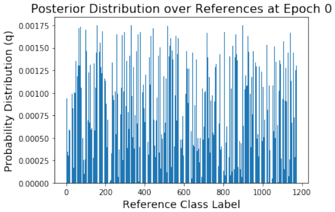

Figure 7: $q(z|\mathbf{x}_p)$ at epoch 0.

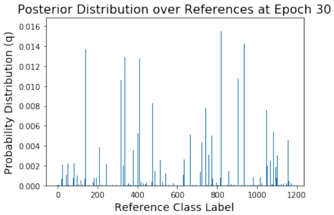

Figure 8: $q(z|\mathbf{x}_p)$ at epoch 30.

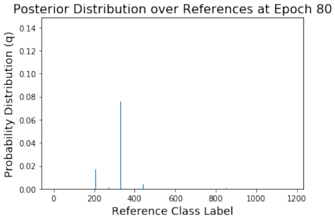

Figure 9: $q(z|\mathbf{x}_p)$ at epoch 80.

In the ablation study in Table 3, we observe only a slight difference with and without basis projection to improve top-k accuracies.

More important difference is due to presence of actual probe labeled dataset, as we obtain slightly lower top-k accuracies without it. The heuristic function for synthetic data still seems to give reasonable accuracies as compared to other methods in Table 1.

Table 3: Ablation results for Noise Removal Method

| Method | Top-1 | Top-5 | Top-50 | Top-100 | Top-500 |
|---|---|---|---|---|---|
| Without Labelled Probe Images | 0.09 | 0.23 | 0.45 | 0.58 | 0.90 |
| Without Basis Projection | 0.11 | 0.33 | 0.51 | 0.64 | 0.91 |
| Top-1000 basis | 0.10 | 0.28 | 0.58 | 0.736 | 0.959 |
| 4096 (Full) basis | 0.09 | 0.29 | 0.58 | 0.741 | 0.959 |

## 7 Conclusion and Future Work

We evaluated baselines based on identifying important “directions" (Sec. 2.1, 2.2) in the reference/probe feature spaces and carried out an analysis of the hierarchical structure in the label space (Sec. 3.1). We also did an exhaustive analysis of metric learning approaches and proposed an EM type algorithm “Adaptive CCA" to find maximally correlated directions with limited number of aligned samples. Finally, we proposed a simple noise removal pipeline which works well when paired with data augmentation and basis projection.

Lack of public benchmarks like Imagenet [3] makes it difficult to propose and evaluate new techniques as there is no standard way to compare them against private datasets used by others. The limited training data is a logistic restriction and data-driven models can be better compared by collecting a sufficient collection of shoeprints in different textured surfaces to train and evaluate upon. Shoe marks may vary based on depth i.e. they may 3-dimensional information (like at the beach) or just 2-dimensional (like on a floor) [2]. The latter corresponds to the reference images in FID-300 while the actual probe ones are of the former type. Future work in this area can be focused on harnessing by putting efforts into the collection of reference impressions on snow, soil etc.

## References

[1] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[2] P. de Chazal, J. Flynn, and R. 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, March 2005. ISSN 1939-3539. doi: 10.1109/TPAMI.2005.48.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bygh9j09KX`.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[6] L Gyorfi. Recent results on nonparametric regression estimate and multiple classification. *Problems of Control and Information Theory*, 10(1):43–52, 1981.

[7] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[10] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[11] Bailey Kong, James Steven Supancic III, Deva Ramanan, and Charless C. Fowlkes. Cross-domain image matching with deep feature maps. *CoRR*, abs/1804.02367, 2018. URL `http://arxiv.org/abs/1804.02367`.

[12] Adam Kortylewski and Thomas Vetter. Probabilistic compositional active basismodels for robust pattern recognition. *BMVC*, 2016. URL `https://fid.dmi.unibas.ch/BMVC16.pdf`.

[13] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

[14] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition.

[15] Imad Rida, Sambit Bakshi, Xiaojun Chang, and Hugo Proença. Forensic shoe-print identification: A brief survey. *CoRR*, abs/1901.01431, 2019. URL `http://arxiv.org/abs/1901.01431`.

[16] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[17] Florian Yger, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Adaptive canonical correlation analysis based on matrix manifolds. *arXiv preprint arXiv:1206.6453*, 2012.

# A Synthetic Data with Texture Blending

The synthetic data can be generated using the reference images. The objective of generating synthetic data images are:

- **Use as a slightly perturbed version of reference image:** Update the pre-trained network's parameters and evaluate if the slightly perturbed image is close to its reference image (or its cluster)
- **Use as probe image:** Learn a model to convert the probe image into its corresponding reference image. We hope that this generated reference image should lie close to its actual reference image (or its cluster) for shoeprint matching

We can use the following ways to transform on the reference image into its slightly perturbed version:

- Crop
- Flip Horizontally
- Brightness and Contrast
- Rotation and shear
- translation and scale distortions (Rida et al, Forensic Survey)

In addition to the above transforms, we can use the following to generate probe-like images:

- Texture blending
- Smudge resulting in same shoeprint more than once
- Occlusion (in part of shoeprint and at image corners)

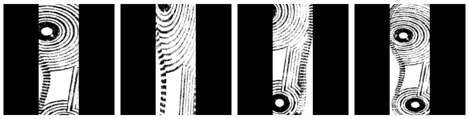

Figure 10: Standard data augmentation

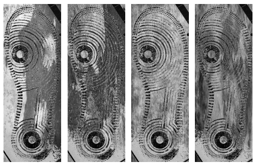

Figure 11: Texture blending with DTD Dataset

## B Hierarchical Clustering

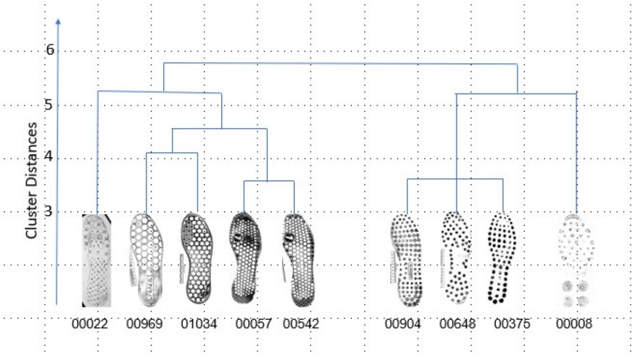

Figure 12: Sample hierarchical clusters in Shoeprints