Canonical Face Embeddings

David McNeely-White, Ben Sattelberg, Nathaniel Blanchard, and Ross Beveridge,

Abstract—We present evidence that many common convolutional neural networks (CNNs) trained for face verification learn functions that are nearly equivalent under rotation. More specifically, we demonstrate that one face verification model's embeddings (i.e. last–layer activations) can be compared directly to another model's embeddings after only a rotation or linear transformation, with little performance penalty. This finding is demonstrated using IJB-C 1:1 verification across the combinations of ten modern off-the-shelf CNN-based face verification models which vary in training dataset, CNN architecture, way of using angular loss, or some combination of the 3, and achieve a mean true accept rate of 0.96 at a false accept rate of 0.01. When instead evaluating embeddings generated from two CNNs, where one CNN's embeddings are mapped with a linear transformation, the mean true accept rate drops to 0.95 using the same verification paradigm. Restricting these linear maps to only perform rotation produces a mean true accept rate of 0.91. These mappings' existence suggests that a common representation is learned by models with variation in training or structure. A discovery such as this likely has broad implications, and we provide an application in which face embeddings can be de-anonymized using a limited number of samples.

Index Terms—Representational similarity, de-anonymization, facial recognition, feature space mapping, neural network equivalence

1 Introduction

The last decade of research into neural networks might be coarsely categorized into efforts toward: 1) advancing the state-of-the-art as expressed through accuracy, and 2) better understanding and analyzing what networks learn. Efforts toward understanding have been far ranging: from comparisons to human vision [1], [2], to investigations into the quality and content of learned features [3], to detailed breakdowns of what causes networks to fail [4]. Even research areas as distinct as style transfer [5], [6] and transfer learning [7] are all, from a certain vantage, centered around understanding the nature of what CNNs learn and, in turn, how what they learn separates and semantically organizes information. Indeed, such fundamental understanding is crucial in the pursuit of better performing and more predictable machine learning models.

Our work here focuses specifically on the embeddings generated by different CNNs. To avoid confusion based on a number of conflicting definitions for neural network "embeddings," we use the term embedding to refer to the feature vector generated by the pooled output of the final, typically convolutional, layer. In closed-set classification tasks, these embeddings are typically fed to a final linear and fully-connected classification layer with one output unit for each class label in the dataset. In open-set face recognition, these embeddings are typically used in a distance-based nearest-neighbors approach instead.

Face recognition is an excellent domain for measuring if embeddings from different networks are equivalent in the sense that they are related to each other through a simple linear mapping. As discussed in the following section, open-set verification (i.e. no people in common between training and testing) reduces the chance that equivalence is trivial, based only upon common labels or logit spaces. Additionally, the now-standard use of unit-normalized features compared by cosine distance ([8], [9], [10], [11]) provides a clear sense in which network outputs are the same or not. In essence, this allows us to ask questions of the kind 'Is this geometric object A the same as object B?". For rigid objects, if B is just A rotated, the common answer is they are the same. The work here takes this basic concept of "same" (or equivalent) and applies it to face embeddings from two CNNs.

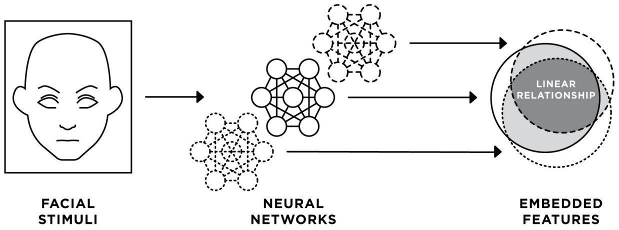

Fig. 1: CNN-based facial recognition systems produce embedding spaces which are typically used for distance-based face verification. The embedding spaces generated by different networks have similar geometry such that we are able to train a linear mapping to recover the embeddings of one network from another. This allows embeddings from two networks to be compared with little performance penalty. Rotation-only mappings can also be used with only a modest additional penalty in the cross-model face verification performance.

Using a handful of embedding pairs generated from two nets on the same faces, our approach computes mappings using least-squares regression in order to align, as much as is possible, the corresponding embedding vectors. These mappings are evaluated by comparing the embeddings of one CNN to the mapped embeddings of another using a standard face verification paradigm. Despite differences in training dataset, CNN architecture, and angular loss function, cross-CNN face verification using these mappings drops very little. These results demonstrate a fundamen-

- *All authors are with the Department of Computer Science, Colorado State University, Fort Collins, Colorado.*
E-mail: {david.white, ben.sattelberg, nathaniel.blanchard, ross.beveridge}@colostate.edu
- *Code to reproduce these findings will be released upon publication.*

Manuscript received XXXXXXXX.

tal underlying near equivalence between the embedding spaces produced by diverse models.

Our work complements a growing sense that different architectures applied to the same data are converging upon similar solutions. For example, some within the neural architecture search community have shifted their focus from improving the architecture itself to finding new data augmentation strategies [12], [13], [14]. Our discovery of an underlying canonical embedding space also strongly suggests that recovery of identities from embeddings is possible, and even arguably easy in some cases: for example using embeddings from the specific CNNs studied here.

Dataset	# individuals	# images/video frames
VGGFace2 [33]	9.1 K	3.3 M
CASIA-WebFace [34]	10.5 K	0.5 M
MS1M [35]	100 K	10 M
MS1MV2[1] [8]	85 K	5.8 M
Glint360K [36]	17 M	360 K
IJB-C [37]	**3,531**	**148.8 K**

TABLE 1: The datasets used in our experiments. The first five datasets were used to train networks used in our experiments. The sixth, **IJB-C**, is a test dataset used to test mappings between feature spaces.

2 Related Work

Many have studied neural networks for the purpose of understanding their hidden representations. Techniques range from visualization [15], [16], [17], to semantic interpretation [18], [19], to similarity metrics. Efforts using similarity metrics are most aligned with our goals

Li *et al.* [20] used matching algorithms to align individual units of ImageNet-trained deep CNNs to determine if networks of different initializations converge to similar representation. Others have argued that drawing semantic meaning from individual bases is problematic, suggesting that semantic meaning is contained within the entire feature space, rather than individual units [21].

Correlation-based methods have also been used for studying representational similarity, such as canonical correlation analysis (CCA), singular vector CCA (SVCCA), projection weighted CCA, or centered kernel alignment [22], [23], [24]. These studies have revealed a number of interesting deep neural network properties, including the tendency of layers to be learned in a bottom-up fashion [22], a correspondence between networks' size and relative similarity [23], and similarity both between and within networks trained on different datasets [24]. The latter work is particularly relevant, as it focuses on the degree of representational similarity between networks which differ in parameterization, much as our own does. While correlation-based techniques may be invariant to orthogonal transform, linear transform, or isotropic scaling, and thus revealing of types of relationships between feature representations, the metric they provide is ultimately subjective. Kornblith *et al.* address this by evaluating similarity indexes based on their ability to accurately identify whether two CNN representations belong to the same layer of architecturally identical networks trained from different random initializations [24]. In contrast, our work provides a *parameterization* of the relationship between feature representations (i.e., a mapping matrix), which can be evaluated using the same methods used to evaluate the models in question. Consequently, we can compare the performance of features which have been mapped against those which have not for a clearer and more objective baseline.

To our knowledge, only two other works have used linear transformations to compare hidden deep neural network representations. Lenc and Vedaldi [25] previously investigated affine mappings between layers of AlexNet [26], VGG-16 [27], and ResNet-v1-50 [28] on ILSVRC2012. However, they fit mappings between spatially-sensitive convolutional layers, and note "there is a correlation between the layers' resolution and their compatibility." Further, they fit mappings using the same classification loss used during training of the models they study, amounting to a more coarse-grained optimization target, and resembling the fixed-feature extraction used for transfer learning. Still, these efforts provide an exciting foundation for us to build upon. In our work, we map between the spatially-pooled outputs of the final convolutional layer of each model, requiring no lossy spatial interpolation, and use pairs of embeddings and a distance-based loss to fit maps.

Our own prior work, McNeely-White *et al.* [29] demonstrated linear similarity between feature representations of two ILSVRC2012-trained CNNs despite differences in architecture (Inception-v4 and ResNet-v2 152). We have since expanded this finding to cover 10 different ILSVRC2012 models [30], [31].

While clearly encouraging to us, efforts such as these (along with many of those mentioned earlier in this section) may be problematic. Namely, there is a degree to which similarity should be expected between embedding spaces for models trained on not only the same task, but the same *labels*. In essence, as long as a linear classifier is used to convert hidden representations into same set of labels, cross-CNN similarity is explicitly encouraged during training by softmax cross-entropy loss. This topic is described in greater detail in [31]. Further, Roeder *et al.*recently established a theoretical proof for the existence of linear correspondence between deep supervised classification models (among others) [32].

Consequently, face verification is an excellent task domain for facilitating linear correspondence studies such as this. Even though models may use a linear classifier with softmax cross-entropy loss during training, we include models which do not have overlapping training labels. What's more, we do not map between faces in the training set, instead mappings are computed using a testing dataset (IJB-C) which is disjoint from the data used to train networks. While linear correspondence between closed-set image classifiers is certainly compelling, by establishing linear correspondence between open-set image classifiers, we establish even stronger evidence that a fundamental similarity in learned representations exists between modern deep CNNs.

3 Methods

Ten independently trained face-recognition models of various pedigree and performance are selected for study. Selection was based upon: 1) could we replicate published performance, 2) a mix of architectures and training data across models, and 3) a mix of angular loss functions. Table 2 summarizes training datasets, architectures, loss functions, performance on IJB-C and GitHub source for the ten models. The training datasets used are described in Table 1. For more details on models and datasets, please refer to their respective publications and code sources.

3.1 Model Evaluation

All 10 models were downloaded pre-trained from their respective sources, and evaluated on IJB-C using the 1:1 verification protocol. Each model was evaluated using its own source repository's preprocessing steps including face detection and cropping. These may differ in crop size, aspect ratio, or similarity transform target, however, the same set of IJB-C images were used in all cases.

Each model is then passed preprocessed images to produce an associated set of embeddings. Embeddings were all evaluated in the same fashion, using evaluation code adapted from Jia Guo and Jiankang Deng's InsightFace project on GitHub [8], [39]. For all 10 networks here, the dimensionality of the feature space is 512. However, with the Probabilistic Face Embeddings (PFE) architecture (models M1-64S-PFE and C-64S-PFE) [46], the output of a second uncertainty module is concatenated to features yielding 1024 dimensions. This uncertainty module consists of a network with two fully-connected layers with input and output dimensions of 512, equal to the dimension of the output of the base CNN model.

IJB-C 1:1 verification consists of generating many templates, each corresponding to one or more images. In cases where video frames are present in a template, we first aggregate features belonging to the same video by simple vector average. Templates are then calculated as an L2-normalized sum of all image and video features. Template pairs are scored by the inner product (dot product), equivalent to cosine similarity since all templates are unit-length. Finally, a list of template pairs is used for ROC analysis to determine true acceptance rates at fixed false acceptance rates (TAR @ FAR). In the case of PFE, this differs from their mean likelihood score and treats σ simply as an additional feature, allowing us to maintain a uniform distance measure across all models for later cross-model comparisons. The performance of each model at 6 FARs is provided in Table 2.

3.2 Calculating Mappings

We are interested in calculating the extent to which a linear map converts between the features of two networks. Let X_E and X_V be the 11,856 enrollment and 457,519 verification images belonging to the IJB-C 1:1 Verification protocol. For a source network f_A and target network f_B, we fit a matrix $\mathbf{M}_{A\rightarrow B} \in \mathbb{R}^{d_A \times d_B}$ such that

$$f_B(X_E) \approx \mathbf{M}_{A\rightarrow B} f_A(X_E) \tag{1}$$

for all input images $x \in \mathbb{R}^{w\times h\times 3}$. Essentially, this approach seeks a mapping which minimizes the distance between pairs of points in feature space corresponding to the same image, up to differences in preprocessing. We also explicitly normalize the result of $\mathbf{M}_{A\rightarrow B} f_A(X_E)$ so that it corresponds to the output of the models we study.

We calculate mappings using two methods, both using pairs of embeddings generated from the IJB-C 1:1 verification enrollment set. To elaborate, these 11,856 images are passed to both models to generate 11,856 pairs of embeddings.

Linear mappings are computed by solving the ordinary least squares regression problem over image pairs:

$$\underset{\tilde{\mathbf{M}}_{A\rightarrow B}}{\text{minimize}} \sum_{i=1}^{m} ||\tilde{\mathbf{M}}_{A\rightarrow B} f_A(x) - f_B(x)||_2. \tag{2}$$

Rotation mappings are computed using the methods developed by Wahba and Kabsch for finding the optimal rotation for minimizing the distances between two sets of points [48], [49]. Simply put, this algorithm consists of computing the singular value decomposition of the cross-covariance matrix of two sets of points $f_A(X_E)$ and $f_B(X_E)$, followed by recomposition with all singular values set to 1. To ensure no flips or mirroring, the last singular value (corresponding dimension of least variance) is optionally set to -1. In other terms, we calculate rotation mappings as:

$$\begin{aligned} f_A(X_E)^T f_B(X_E) &= U\Sigma V_h \\ M_{A\rightarrow B} &= UI'V_h \end{aligned} \tag{3}$$

where

$$I' = \operatorname{diag}\left(\begin{bmatrix}1 & 1 & \ldots & 1 & \det(U) * \det(\mathrm{V_h})\end{bmatrix}\right).$$

This produces a linear mapping matrix with the additional constraint of being orthogonal and having determinant 1, a rotation. Note that $f_A(X_E)$ and $f_B(X_E)$ would typically be centered first to find the optimal rotation axes, but we leave points in their original translation (on the unit hypersphere), since we intend to rotate about the origin.

3.3 Evaluating Mappings

A natural method for mapping evaluation is to measure impact on performance using a validation dataset of faces unseen during training of any models. Essentially, we produce mapped features

				IJB-C (TAR @ FAR)						
Short Name	**Training Dataset**	**CNN Architecture**	**Angular Loss Function**	**1e-1**	**1e-2**	**1e-3**	**1e-4**	**1e-5**	**1e-6**	**Source**
M2-R100-A	MS1MV2	ResNet100 [38]	ArcFace [8]	0.991	0.984	0.975	0.963	0.945	0.898	[39]
V-R50-A	VGGFace2	ResNet50 [38]	ArcFace [8]	0.994	0.984	0.963	0.928	0.875	0.744	[39]
G-R100-P1.0	Glint360k	ResNet100 [38]	PartialFC (r=1.0) [36]	0.993	0.988	0.981	0.973	0.960	0.912	[39]
G-R100-P0.1	Glint360k	ResNet100 [38]	PartialFC (r=0.1) [36]	0.992	0.987	0.981	0.974	0.961	0.872	[39]
M1-R50-A	MS1M	ResNet50 [38]	ArcFace [8]	0.979	0.954	0.918	0.861	0.782	0.701	[40]*
M1-MB2-A	MS1M	MobileNetV2 [41]	ArcFace [8]	0.981	0.940	0.869	0.766	0.629	0.503	[40]*
V-IR1-C	VGGFace2	InceptionResNetV1 [42]	Center Loss [43]	0.990	0.967	0.908	0.808	0.681	0.518	[44]*
C-IR1-C	CASIA-WebFace	InceptionResNetV1 [42]	Center Loss [43]	0.981	0.929	0.832	0.697	0.534	0.408	[44]*
M1-64S-PFE	MS1M	64-CNN+PFE [45], [46]	AM-Softmax [9]	0.985	0.970	0.942	0.872	0.757	0.610	[47]
C-64S-PFE	CASIA-WebFace	64-CNN+PFE [45], [46]	AM-Softmax [9]	0.982	0.949	0.889	0.798	0.678	0.530	[47]

TABLE 2: Configuration and accuracy of each model. A shortened name is provided for later reference. Note that these accuracy values are calculated by our internal verification and may differ slightly from the stated values for each model's source publication (when available).
*Sources not associated with original publication.

from the mapping's source network, and evaluate them against features generated by the mapping's target network. As in Section 3.1, templates are generated from collections of embeddings, except each template in a pair is generated by a different network.

To be precise, target model templates are calculated from embeddings in the verification set, $f_B(X_V)$, and source model templates are calculated from mapped embeddings generated from the same images $\mathbf{M}_{A\to B} f_A(X_V)$. As in the previous section, template match scores are computed as the inner product, equivalent to cosine similarity when templates are unit-length. ROC analysis is performed to produce true accept rates at various false accept rates (TAR @ FAR), which may be compared to unmapped model performance.

4 Cross-CNN Mapping Results

Mapping evaluation results are summarized in Figure 2. In each grid, elements along the diagonal (bolded values) show the same TARs listed in Table 2. Off-diagonal elements contain TARs produced by cross-CNN evaluation as described in Section 3.3, with the row label indicating the source model, and the column label indicating the target model. These labels correspond to each model's ”Short Name” in Table 2. The three grids along the top of the figure are produced using linear mappings, and the bottom three produced using rotation mappings. The two leftmost grids contain TARs at a FAR of 1e-1, the middle two at 1e-3, and the rightmost two at 1e-5.

All mappings have a high TAR at a FAR of 0.1. Even when constrained to only rotate embedding spaces, cross-CNN performance is still at or near single-CNN performance. While this FAR is very weak, these results provide a demonstration that mapping between face verification CNNs is possible using linear or rotation maps. Interestingly, some mappings exceed the performance of their target or source model (but not both). Compared with linear maps, rotation produces a higher penalty in all cases, and the penalty is (nearly) symmetric due to the structure of finding an orthonormal solution. Still, the extra constraint of rotation maps further reveals the nature of cross-CNN relationships.

As the FAR is decreased to 1e-3, we maintain high TAR for the most successful networks, but networks with less representational complexity have less success in representing the more complex networks. As the FAR is decreased further to 1e-5, clear failure cases emerge. For models which already perform well, though, mappings still preserve performance. For a handful of cases, however, mappings produce near-random performance. This indicates an exciting direction for future work, as discussed in Section 6.

The ability for one network's features to robustly replicate features coming from a different model demonstrates a near-linear equivalence of their feature spaces. Although some information does not transfer, there is clearly a great deal of similarity in the structure of embedding spaces produced by seemingly disjoint networks. This finding is in contrast to previous results that methods such as feature fusion increase the accuracy of the networks [50], [51]. However, the information not encoded by this linear transform may result in the increases in accuracy revealed in those experiments.

To confirm this behavior requires a linear mapping, and that feature spaces are not directly equivalent, we also compare features directly, without mappings. As discussed in [20], [21], direct interchangeability between feature spaces is unlikely to exist. By using an identity mapping as $\mathbf{M}_{A\to B}$ we achieve near random accuracy (i.e. the TAR nearly equals the FAR), confirming that a mapping of some kind is necessary to convert between features. The results of these direct comparisons, are displayed in Figure 4 along with both other methods at all 6 FARs.

To summarize, the networks we consider have multiple resolutions of accuracy. As additional representational complexity allows the highest-performing networks to achieve good solutions, they appear to converge to solutions that differ largely in ways that are explainable by rotation. If the mapping is less constrained and allowed to be any linear transformation, larger groupings of networks are similar.

4.1 Sensitivity to number of images

To better understand the complexity of mappings between embedding spaces, we also fit and evaluate mappings using a variable number of paired examples. Specifically, we select a random subset of the 11,856 images in the IJB-C 1:1 verification enrollment set, and fit mappings as described before, using a smaller set of images. Then, mappings are evaluated as before, by comparing the mapped embeddings of one network to the unmapped embeddings of another on the IJB-C 1:1 verification set, after both sets of embeddings have been aggregated into templates. The distribution of mapping TARs at a FAR of 0.01 using various numbers of samples to fit said mappings is illustrated in Figure 3.

Notably, while rotation-only mappings perform worse using mappings fit with 1024 or greater images, their performance degrades more slowly using fewer than 1024. That solving for a rotation seems to require fewer paired samples makes sense given that the degrees of freedom in the space of rotation matrices is smaller than for a general linear mapping. However, it is also revealing of the nature of the relationship between these face verification models, since the added rotation-only constraint ”guides” the mapping towards a better solution given fewer examples. Since these methods produce similar performance using many examples, this suggests that the bulk of the relationship between face verification models can be described simply as a rotation.

5 Application: Embedding Re-identification

We have demonstrated the existence of simple underlying relationships between face verification models which vary in some combination of training dataset, architecture, and angular loss function. Provided with knowledge of these relationships, we also present a method of attack for face recognition systems using CNN-based models.

For the purposes of this application, we assume that an attacker has access to a limited set images and corresponding embeddings from an otherwise unknown model. If our attacker obtains/trains their own face recognition model (preferably multiple), and generates embeddings for this set of images, a mapping can be calculated as in Section 3.2. Then, given new embeddings from the unknown model, the attacker can generate mapped embeddings in the model(s) they have access to. Using their own database of face images, the attacker can create a database of templates by which to compare mapped embeddings from the unknown model. Provided the attacker has an example image of the face embedded by the unknown model, they can determine the likely face which produced the embedding.

Of course, the size and diversity of the initial set of images and embeddings determines the quality of the resulting mapping, as studied in Section 4.1. It may be that a minimal set of images

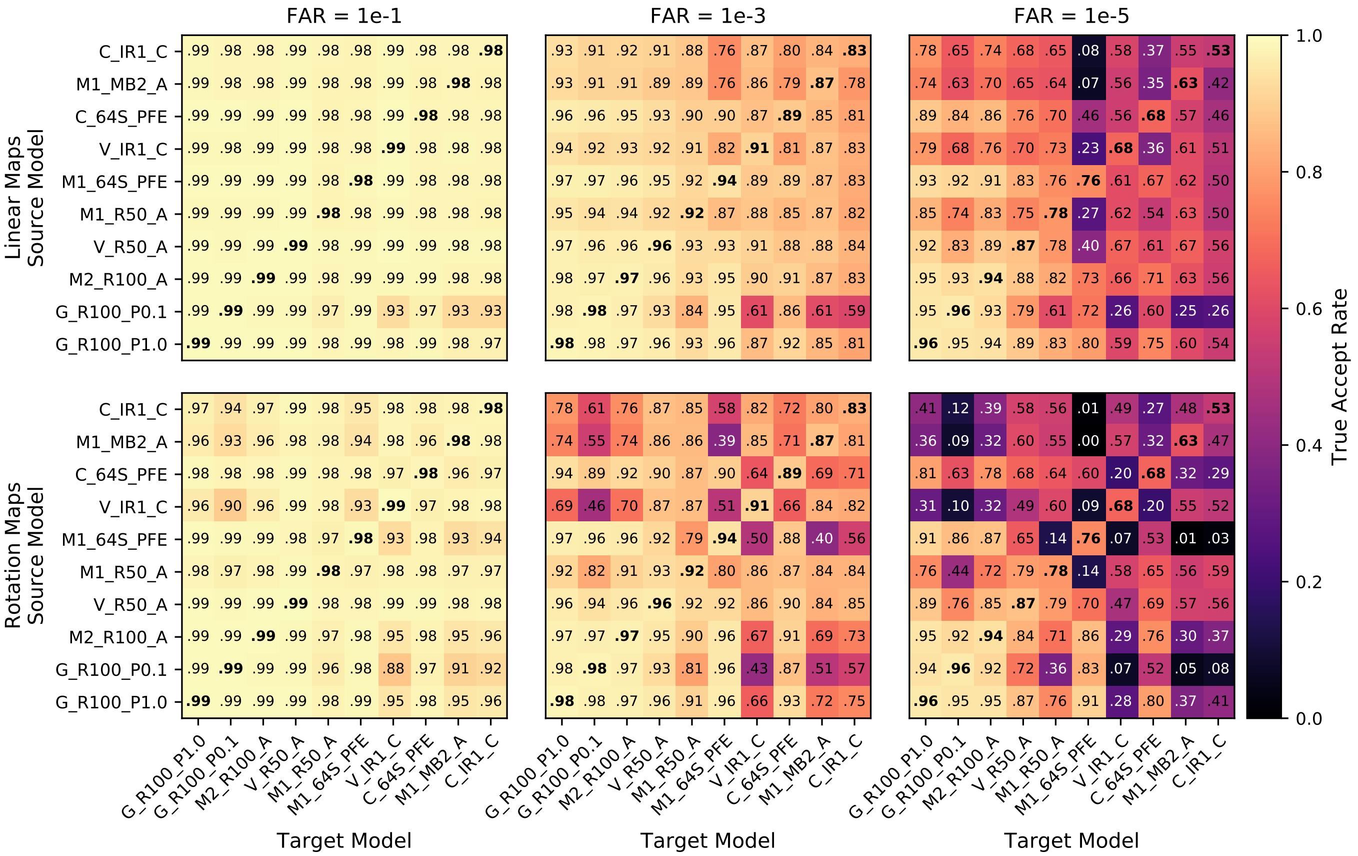

Fig. 2: Linear maps reveal consistent overlap between feature spaces of distinct CNNs. Whether constrained to rotation or not, linear mappings enable cross-CNN comparison of face feature embeddings on IJB-C, despite variation in training dataset, architecture, or angular loss function. Elements along the diagonal (bottom-left to top-right) correspond to the unmodified accuracy of each model (see Table 2). Off-diagonal elements correspond to the accuracy obtained when comparing features across networks, with the "Source" model's features mapped by linear transformation to approximate the "Target" model's features. Rows and columns are sorted according to each model's mean TAR across all FARs, and labeled using the model's "Short Name" from Table 2. Best viewed in color.

exists (or could be generated) which provide a maximal amount of information for fitting linear or rotation maps. Along these lines, additional work is required to find which images in particular are best at constraining these mappings. However, given a sufficient limited set, a fundamental link between any CNN resembling those studied here can be established. While this does not explicitly provide reversibility of embeddings, the identities used to generate embeddings can be determined so long as an example of their face can be obtained.

This method was recently used in a multi-institution federally-funded research program to reveal the concealed identity of individuals of interest, and should challenge any notion that face embeddings anonymize individuals. As part of this research program, we receive face embeddings corresponding to potentially unknown individuals from multiple sources using different neural networks to construct the embeddings. As before, we fit a mapping using a collection of paired embeddings. In this case, the source image, face, and model are all unknown–all we know is which embeddings likely correspond to the same individual. Once a mapping is constructed, we can then predict when embeddings produced by any source correspond to the same individual, even when that individual is not identified by any source model. Ultimately, this technique allowed us to reveal more information than the sum of otherwise independent upstream sources.

6 Potential Limitations and Future Work

We are confident that the results presented here are strong evidence for a fundamental similarity between common CNN-based face recognition systems. While we may have shed some light on finer details, more experiments are required to further characterize the nature and boundaries of this effect. For example, take G_R100_P0.1 (ResNet100 trained on Glint360k using PartialFC loss) which performs comparatively well on its own, and in many cross-CNN mappings. This model is already somewhat distinct in that it uses a random subset of 10% of identities for computing softmax loss during training [36]. Take note, however, of the target models which produce poor mapping performance, V_IR1_C, C_IR1_C, and M1_MB2_A. While these models perform worse already, they seem to have greater incompatibility with G_R100_P0.1 than others. Roughly speaking, if model A is compatible with model B, and model B is compatible with model C, then why isn't A compatible with C? We hypothesize that, despite evidence for broad task-driven similarity between CNNs, certain design features of these CNNs do impact finer aspects of their representation. In other words, it may be that subsampling identities during training as with PartialFC biases the resulting model away from a representation found using InceptionResNet or MobileNet architectures. If true, design features could be

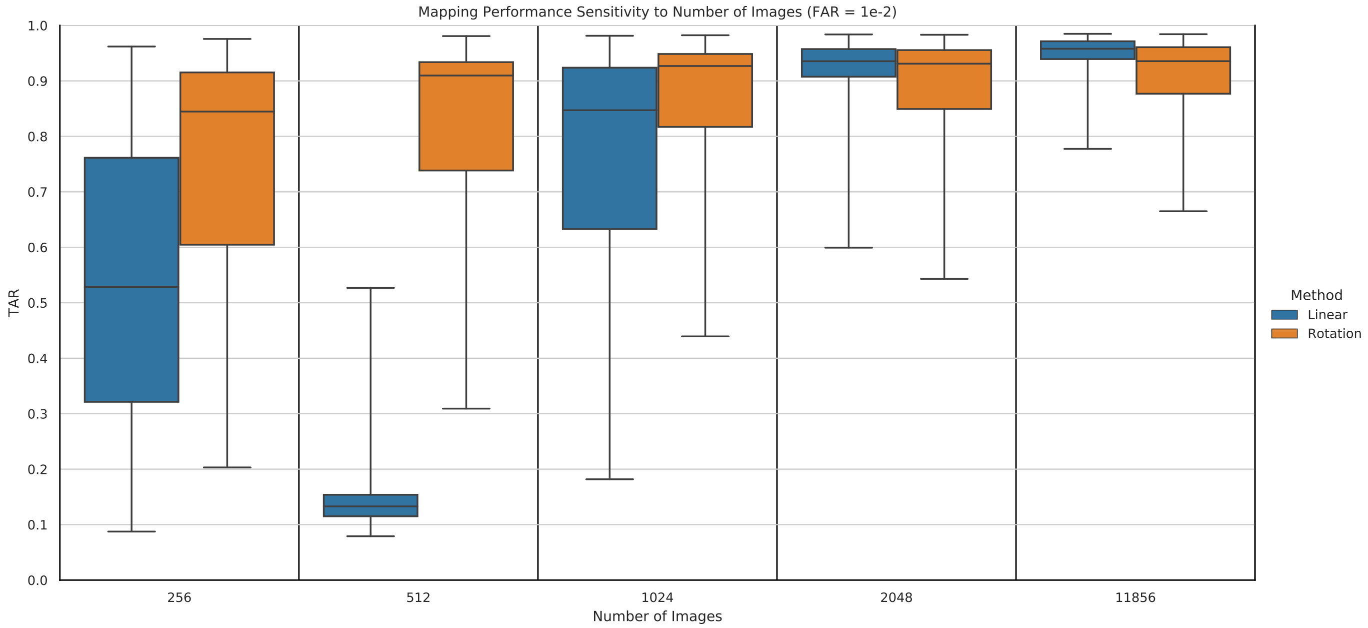

Fig. 3: Mappings can be fit using a reduced number of images (i.e. embedding pairs). The general robustness of both linear and rotation-only maps to fewer examples may be supported by the inherently low dimensionality of image features described by [52]. Rotation-only maps require fewer embedding pairs to obtain the same performance, likely due to the constrained space of rotation-only mappings compared to the space of linear mappings. All performances represented here are generated from mapped features (i.e. no single-model performance is included). Each point is the mean performance of 3 mappings fit using new random subsets. Whiskers are extended to include all points, i.e. no points are treated as outliers.

characterized by their effects on representation, guiding future designs.

Further, our method may be explicitly lossy when features are represented with different numbers of dimensions, producing rectangular mapping matrices (i.e. non-zero nullity per the rank–nullity theorem). The work of Gong *et al.*provide insight along these lines by providing evidence for a far reduced ”intrinsic dimensionality” produced by common face verification CNNs [52]. This suggests that while rectangular linear transformations indeed project information into fewer dimensions, mapping between two sets of face embeddings may require far fewer dimensions than the maximum rank of a rectangular matrix.

It would also be of interest to determine the minimum number of known identities and associated embeddings that are necessary to construct an effective mapping. The algorithm presented here for embedding re-identification still requires in the order of thousands of samples. While this requirement is modest compared to how networks are initially trained, it still implies that around a thousand paired samples processed by two different CNNs must be identified before a mapping can be established.

7 Conclusion

We believe the evidence presented here is compelling and convincing that a fundamental relationship exists between common-task CNNs (i.e. the existence of a canonical embedding space). Still, much more work is necessary to further characterize this relationship. For example, while we demonstrate results using 90 pairs of CNNs, it's unclear how small variations in preprocessing or model implementation may contribute to mapping performance. While we include many architectures and results for a large face dataset, finer characterization of these relationships requires carefully controlling for architecture and training details. Such analysis will likely come at great resource cost, so we leave a systematic expansion of experimental granularity and controls to future works.

The existence of performance-preserving linear mappings between face recognition CNNs which vary in training and construction suggests this phenomenon depends primarily–if not solely–upon the modeling task. If so, then the desired feature space is not unique. Instead, embeddings produced by CNN models trained for the same task approximate a **canonical** space, which each model appears to converge upon. The existence of task-dependent canonical embedding spaces suggests there is a high degree of redundancy in the training procedure of bespoke CNN-based models for a common task, as the bulk of the learned representation is unchanged. Individual networks can and do contain information that is not contained within other networks, but this appears to come in the form of more nuance and a finer grained approximation, rather than meaningfully different behavior.

Appendix A
Full Figures

Acknowledgments

This work was supported by a subaward from DARPA FA8750-18-2-0016-AIDA – RAMFIS: Representations of vectors and Abstract Meanings for Information Synthesis. This work does not express the opinions of the sponsor. Thank you to Michelle Wern for designing Figure 1.

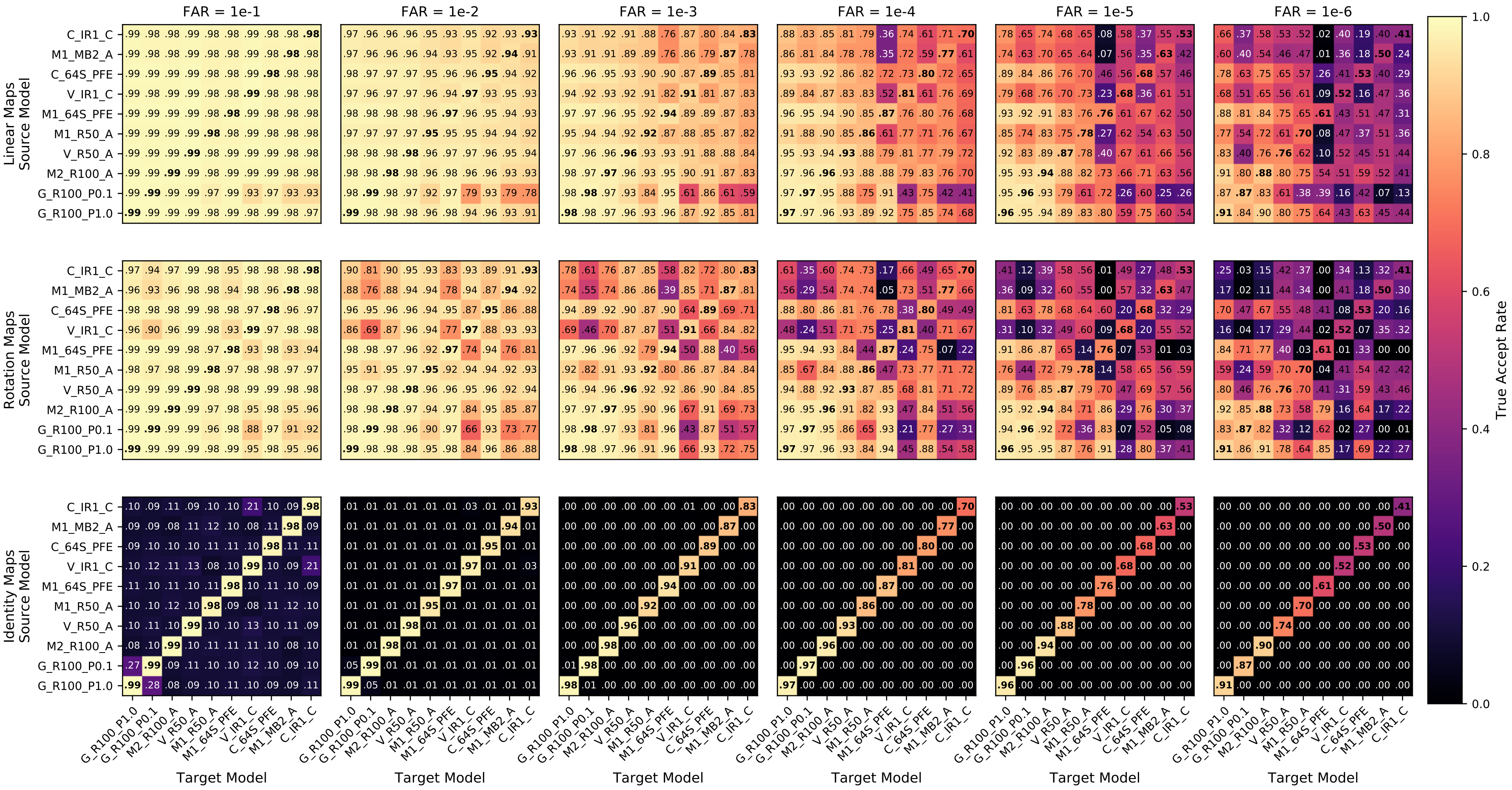

Fig. 4: Full mapping performance results, including the same data presented in Figure 2 and following the same format. Identity mappings are also included to illustrate the lack of direct compatibility between feature representations.

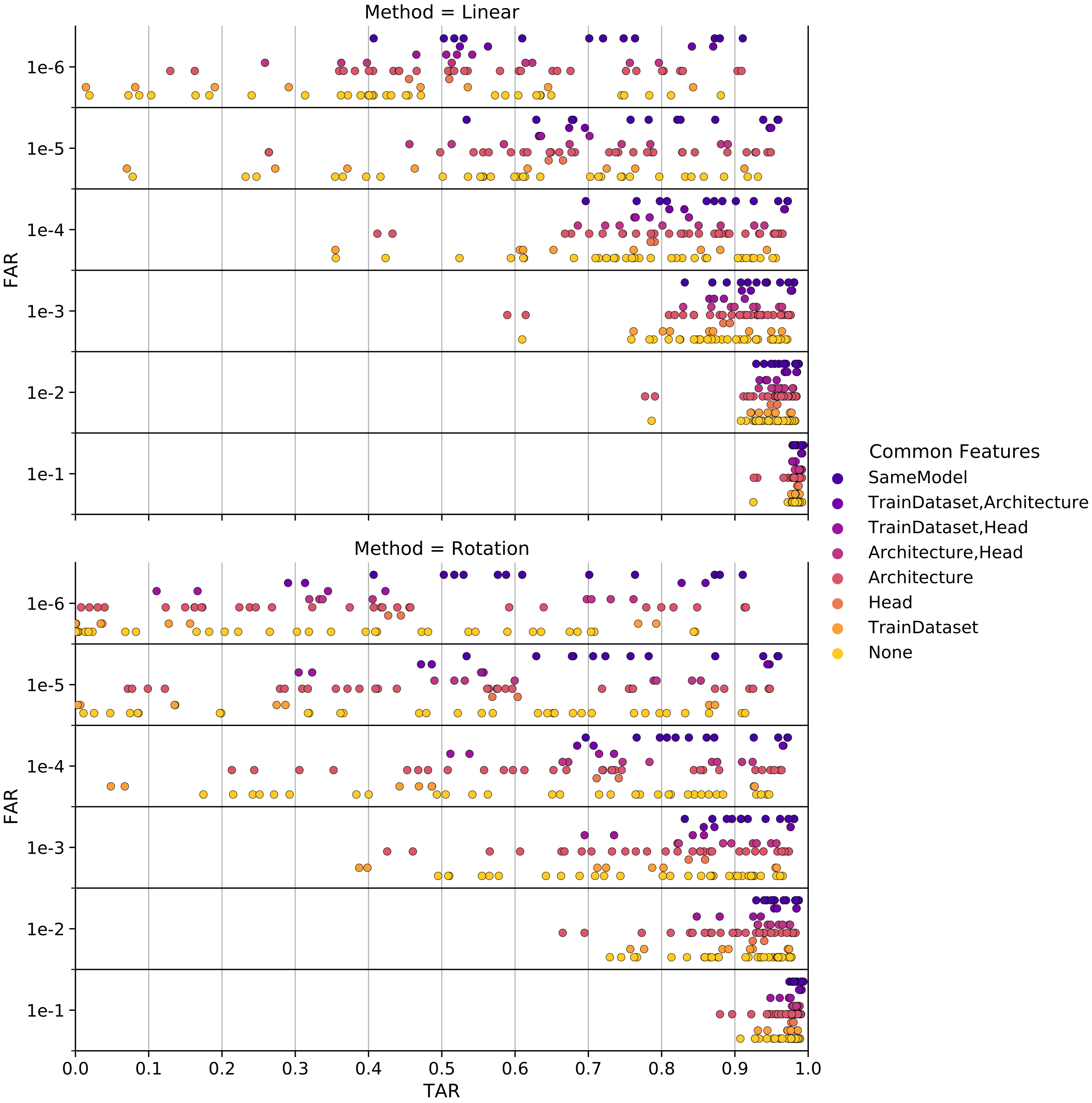

Fig. 5: While our sample size is limited and model distinctions subjective, mapping performance is largely independent of overlap in model pedigree (training dataset, architecture, and angular loss function). Note: all ResNet models are marked as having the same architecture; MS1M and MS1MV2 are marked as the same dataset.

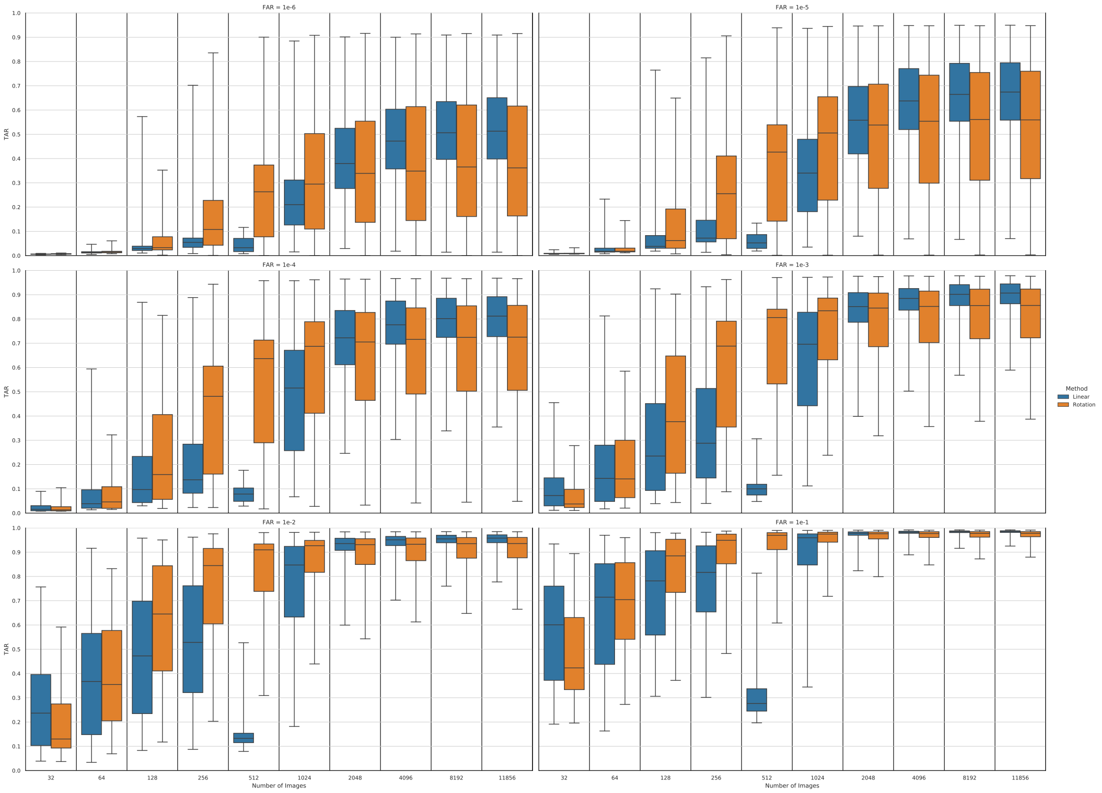

Fig. 6: Full mapping sensitivity results, including the same data presented in Figure 3 and following the same format. All performances represented here are generated from mapped features (i.e. no single-model performance is included).

References

[1] N. Blanchard, J. Kinnison, B. RichardWebster, P. Bashivan, and W. J. Scheirer, "A neurobiological evaluation metric for neural network model search," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 5404–5413. 1

[2] K. R. Storrs, T. C. Kietzmann, A. Walther, J. Mehrer, and N. Kriegeskorte, "Diverse deep neural networks all predict human it well, after training and fitting," *bioRxiv*, 2020. 1

[3] K. Hermann, T. Chen, and S. Kornblith, "The origins and prevalence of texture bias in convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020. 1

[4] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016, pp. 1–6. 1

[5] D. Hart, B. Morse, and J. Greenland, "Style transfer for light field photography," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 99–108. 1

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423. 1

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724. 1

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699. 1, 2, 3

[9] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018. 1, 3

[10] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049. 1

[11] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274. 1

[12] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 113–123. 2

[13] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," *arXiv preprint arXiv:1906.11172*, 2019. 2

[14] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703. 2

[15] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009. 2

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833. 2

[17] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization. 2

[18] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," *arXiv preprint arXiv:1711.11279*, 2017. 2

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929. 2

[20] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. E. Hopcroft, "Convergent learning: Do different neural networks learn the same representations?" in *FE@ NIPS*, 2015, pp. 196–212. 2, 4

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013. 2, 4

[22] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, 2017, pp. 6076–6085. 2

[23] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational

similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems*, 2018, pp. 5727–5736. 2

[24] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," *arXiv preprint arXiv:1905.00414*, 2019. 2

[25] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 456–476, May 2019. [Online]. Available: https://doi.org/10.1007/s11263-018-1098-y 2

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 2

[27] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013. 2

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 2

[29] D. McNeely-White, J. Beveridge, and B. Draper, "Inception and resnet features are (almost) equivalent," *Cognitive Systems Research*, vol. 59, 10 2019. 2

[30] D. G. McNeely-White, "Same data, same features: Modern imagenet-trained convolutional neural networks learn the same thing," Master's thesis, Colorado State University, 2020. 2

[31] D. McNeely-White, B. Sattelberg, N. Blanchard, and R. Beveridge, "Exploring the interchangeability of CNN embedding spaces," *arXiv preprint arXiv:2010.02323v4*, 2020. 2

[32] G. Roeder, L. Metz, and D. P. Kingma, "On linear identifiability of learned representations," *arXiv preprint arXiv:2007.00810*, 2020. 2

[33] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 67–74. 2

[34] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014. 2

[35] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European conference on computer vision*. Springer, 2016, pp. 87–102. 2

[36] X. An, X. Zhu, Y. Xiao, L. Wu, M. Zhang, Y. Gao, B. Qin, D. Zhang, and F. Ying, "Partial fc: Training 10 million identities on a single machine," in *Arxiv 2010.05222*, 2020. 2, 3, 5

[37] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 158–165. 2

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645. 3

[39] J. Guo and J. Deng, "Insightface: 2d and 3d face analysis project," Apr. 2021. [Online]. Available: https://github.com/deepinsight/insightface 3

[40] K.-Y. Huang, "Arcface unofficial implemented in tensorflow 2.0+ (resnet50, mobilenetv2)." Jun. 2020. [Online]. Available: https://github.com/peteryuX/arcface-tf2 3

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520. 3

[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 3

[43] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515. 3

[44] D. Sandberg, "Face recognition using tensorflow," Apr. 2018. [Online]. Available: https://github.com/davidsandberg/facenet 3

[45] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220. 3

[46] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6902–6911. 3

[47] Y. Shi, "Probabilistic face embeddings," Aug. 2019. [Online]. Available: https://github.com/seasonSH/Probabilistic-Face-Embeddings 3

[48] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965. 3

[49] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976. 3

[50] A. Bansal, R. Ranjan, C. D. Castillo, and R. Chellappa, "Deep features for recognizing disguised faces in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 10–16. 4

[51] N. Bodla, J. Zheng, H. Xu, J.-C. Chen, C. Castillo, and R. Chellappa, "Deep heterogeneous feature fusion for template-based face recognition," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 586–595. 4

[52] S. Gong, V. N. Boddeti, and A. K. Jain, "On the intrinsic dimensionality of image representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3987–3996. 6