The Phenomenon of Correlated Representations in Contrastive Learning

1st Simon Klüttermann
TU Dortmund University
Dortmund, Germany
Simon.Kluettermann@cs.tu-dortmund.de

2nd Jérôme Rutinowski
TU Dortmund University
Dortmund, Germany
jerome.rutinowski@tu-dortmund.de

3rd Emmanuel Müller
TU Dortmund University
Dortmund, Germany
emmanuel.mueller@cs.tu-dortmund.de

***Abstract*—Contrastive learning is widely considered to be an important domain of machine learning. Its main premise is to use contrasting samples of data in order to learn common features that allow for accurate data clustering in a representation space. The generation of such a representation or embedding space can be of great value, for instance for re-identification tasks. While working on one such task, we noticed that, paradoxically, decreasing training data resolution led to a considerably higher re-identification accuracy. Upon further analysis, we discovered that this occurs since during training, highly correlated features are learned that are ultimately redundant and limit the model's performance. This effect is exaggerated with the use of high-resolution data, therefore eventually decreasing the obtained re-identification accuracy. In this contribution, we characterize this phenomenon, which we believe to be a novel problem in contrastive learning. We study the effects of various changes to a common neural network architecture on this phenomenon, linking it to the concept of bias, and propose a set of solutions to mitigate its effect.**

***Index Terms*—Re-identification, Contrastive Learning, Ensembles**

I. Introduction

Contrastive learning is a machine learning paradigm, concerned with the accurate representation of data by, e.g., successfully extracting the relevant features of an image. Due to this goal, contrastive learning can be considered a form of representation learning. Traditionally, this task is performed by minimizing a contrastive loss function, like the one first proposed by [1]. More precisely, when given a set of input samples the aim is to encode these samples into an embedding vector. The encoding ought to take place in such a way that samples from the same class are clustered close to one another, making them distinguishable from samples of other classes.

Besides the eponymous contrastive loss, triplet loss [2] is a commonly used loss function, first presented in combination with FaceNet. It differs from contrastive loss by virtue of using not only a positive and negative sample but also an anchor for the learning process.

One application of contrastive learning is the task of re-identification [4]. Often concerned with humans as subjects [4], [5], it can be described as the task of identifying a previously recorded individual at a different point in time (and place), by using the individual's idiosyncrasies to distinguish them from others. Thus, samples of the same subject are supposed to be grouped close to one another in the embedding space, making the subject distinguishable from other individuals.

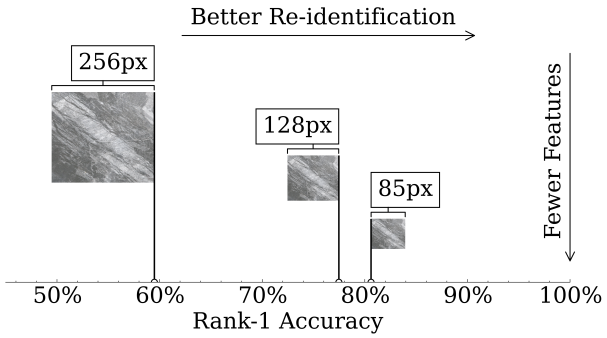

Fig. 1: Re-identification accuracy in relation to image resolution. Counter-intuitively, a decrease in resolution (the original resolution of 256px being divided by 2 (128px) and 3 (85px)) leads to an increase in accuracy. The images seen in this figure represent galvanized metal surfaces and are taken from [3].

A recent paper that is also concerned with the task of re-identification in the context of logistics [6], also using triplet loss, demonstrates an unexpected phenomenon. In contrast with the findings of other publications [7]–[10] and what could by now be regarded as common knowledge, this publication demonstrated that using data containing less information, i.e., lower resolution images, yielded a higher re-identification performance. While previous works [11] have demonstrated that it is possible to reduce dimensionality while still providing solid results, outperforming those results through the removal of information is a novel phenomenon and counter-intuitive.

A minimal example that encapsulates this behavior is shown in Fig. 1, in which the increase in accuracy based on a reduction in resolution can be observed. This phenomenon can be considered paradoxical, because neural networks are understood to be able to approximate functions to whatever information they are fed, and thus minimizing a triplet loss function should produce as good a representation as possible [2], [12]. And since on a higher resolution, we can learn additional representations, these should work at least as well.

We believe this phenomenon to be linked to a novel problem in contrastive learning, that occurs due to a high degree of correlation between extracted features. Therefore, in this

contribution, we will study the effects that various changes to a common re-identification model have on this phenomenon. We will apply said model to three datasets and propose a first set of solutions to mitigate the effect of this phenomenon. We will study and characterize this phenomenon by linking it to what we call the occurrence of *correlated representations*, which we believe to be its precursor. By mitigating the occurrence of correlated representations, that are neither dependent on a specific dataset nor a specific model architecture, we believe to be able to provide the research community with a way of significantly improving the performance of most re-identification approaches.

In the proceeding section, we will elaborate on the related work that first needs to be discussed. Then, we will present the experimental setup, including model architecture and training procedure as well as datasets and evaluation measures used in this work. Subsequently, we will describe and characterize the correlated representations we observe and analyze the factors influencing their occurrence. Finally, we will suggest some potential solutions to mitigate them and draw a conclusion based on our experiments and observations.

To increase reproducibility, our code, as well as our supplementary material are available at https://github.com/psorus/cf

II. Related Work

This section will briefly discuss the fundamental concepts behind re-identification. Afterwards, we will describe phenomena that we believe to be related to the herein-described phenomenon of correlated representations.

A. Contrastive learning-based re-identification

Besides others, like natural language processing [13] and object detection [14], re-identification is one of the most common applications of contrastive learning. As hinted at in the introduction, re-identification deals with the task of identifying a previously recorded subject. For this, a sample image (called the query image) is compared to a set of previously recorded images (the gallery images). Generally, using triplet, classification or contrastive loss [12], [15] as a loss function, a neural network is trained to find a representation in which the gallery images are ranked according to their similarity to the respective query image. A ranked accuracy of the respective re-identification model is commonly gauged by testing it on a set of such query images [16].

For every novel application of re-identification, datasets have to be curated. This is due to the fact that common image datasets usually do not include more than one labeled sample of each individual subject. This means, for instance, that when the aim is the re-identification of a specific animal in a herd, multiple images of the very same animal have to be included in the dataset. While beyond the scope of this contribution, much more could be said about the task of re-identification and its wide field of applications, notably and primarily pedestrian surveillance [17] but also newly developing fields, like the previously mentioned identification of animals in a herd [18] or the identification of industrial entities [11], [19], [20], as seen in Fig. 1.

B. Related phenomena in contrastive learning

Just last year, researchers from Facebook AI [21] studied what they called *dimensional collapse*, occurring during contrastive learning tasks. This term describes the phenomenon of embedding vectors spanning a subspace that is of a lower dimensionality than the available embedding space. In their experiments, Jing et al. use a version of contrastive loss, and in contrast to mode collapse [22], observe only the collapse of parts of their embedding vector. Thus, while the vector does not collapse to a single mode or point in space, information is unnecessarily lost, by not using the entire embedding space dimensionality.

It is understood that contrastive loss is the mechanic at play, preventing complete collapse, by virtue of using its negative sample to prevent vectors representing different image types from collapsing around the same points. In order to remedy the phenomenon of dimensional collapse, Jing et al. propose the use of linear projectors, with which they have demonstrated promising results.

While also being a phenomenon occurring in (self-supervised) contrastive learning, dimensional collapse differs from the phenomenon we observe. We believe this to be the case, since what we observe is notably the increase in correlation between learned features, along with the simultaneous increase in data dimensionality and decrease in prediction accuracy. Both are not affected by small changes to our network architecture, as would be the case when employing a projector.

Another approach related to such phenomena is the *Barlow Twins* model [23]. Like other contributions [24], [25] it addresses the effect that representation learning often results in redundant features (instances in the vector representation that are linearly dependent or zero), and is a method to remove those features, resulting in increased performance for self-supervised learning or clustering tasks.

Another perspective on the phenomenon of correlation and redundancy is given by pruning [26], focusing on accelerating the learning process. While some of these contributions are somewhat related to the herein described phenomenon, we believe to study a novel phenomenon in contrastive learning, which we will elaborate upon in the proceeding sections.

III. Experimental Setup

This section will explain the experimental foundations of this contribution. We try to maintain a consistent setup for the experiments in this work. Thus, for simplicity's sake, we describe our approach here and will only highlight deviations from it further down the line.

A. Neural network structure

For this work, we train a neural network $f(x)$ to learn a representation of a given input image. To do so, we randomly generate $10,000$ combinations of an anchor image (x_a^i), an

image of the same entity (x_b^i), an image of another entity (x_c^j), and optimize them using the triplet loss function given in Eqn. 1 (using the *l2* distance $d(x, y) = \|f(x) - f(y)\|_2$)

$$L_{triplet} = \max(0, d(x_a^i, x_b^i) - d(x_a^i, x_c^j) + \alpha) \ , \ i \neq j \quad (1)$$

We select $\alpha = 1.0$ and for the function $f(x)$ to output a 100 dimensional representation. We build the function $f(x)$ through two convolutions with a depth of 16 features, a ReLU activation function and a filter size of 3. After these convolutions, we add a max pooling layer, which reduces each dimension by a factor two. This structure is repeated two more times, after which we flatten the output and use three fully connected layers of 128 nodes with ReLU activation functions and one fully connected layer of 100 nodes without any activation function to generate the representation. The architecture is visualized in Tab. 1:

TABLE I: Neural network architecture for our experiments.

	Layer	Nodes	Kernel	Activation
3x{	Convolution	16	3x3	ReLU
	Convolution	16	3x3	ReLU
	Max pooling	-	2x2	-
	Flatten	-	-	-
3x{	Fully connected	128	-	ReLU
	Fully connected	100	-	-

To train the model, we use Adam as an optimizer with a learning rate of 0.001 for at most 100 epochs in batches of 128. We also employ early stopping [27] with a patience of ten epochs. For more information, please refer to our code. All our experiments are conducted using an NVIDIA A100 graphics card with 40GB of VRAM.

B. Datasets

To apply our model, we use three different re-identification datasets, which are shown in Fig. 2. Two of these datasets are datasets related to industrial applications, while the third one is a well-known person re-identification dataset. We intend to thereby demonstrate that the phenomenon and suggested solutions in this work are not restricted to a single dataset.

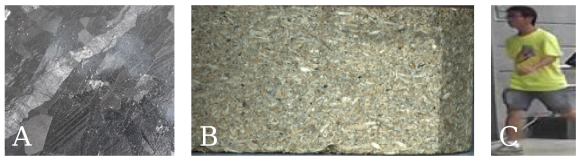

Fig. 2: Example images of the three datasets used in this study (dataset A, *galvanized-636* on the left-hand side, dataset B, *pallet-block-502* in the center, and dataset C, *Market-1501*, on the right-hand side).

The first dataset, *galvanized-636* [3], contains images of 636 plates of galvanized metal, of which four images per plate face were taken from different perspectives. This amounts to a dataset of 5,088 images in total. The dataset was curated in order to facilitate the identification of industrial entities made of this type of metal, which due to the galvanization coating process possesses a unique surface structure. The dataset was previously used in [6].

The second dataset, *pallet-block-502* [28], consists of 5,020 images of 502 chipwood pallet blocks, of which pictures were taken from ten different perspectives. A pallet block is the (commonly chipwood, but sometimes solid wood) block between the planks of an industrial pallet, in this case Euro-pallets. Again, this dataset was curated for industrial re-identification purposes, so that pallets can be tracked over the span of their life cycle. The unique surface structure that is being exploited in this case is the chipwood or solid wood surface, that again yields a unique pallet for each individual pallet block. The dataset was previously used and extensively described in [11], [19], [29]

Finally, the third dataset, *Market-1501* [30], is a widely used pedestrian re-identification dataset. It consists of 32,668 images of 1,501 individuals that are recorded by six cameras. Since this dataset records a public space, the amount of images per individual are not equally distributed, unlike for the previous datasets. For this dataset, each individual is, on average, recorded 3.6 times per viewpoint.

Going forward, for simplicity's sake, we will call *galvanized-636* dataset A, *pallet-block-502* dataset B, and *Market-1501* dataset C.

C. Validation

The performance of the models used in this work is gauged using standard ranked accuracy, as is common in re-identification publications [11]. To increase the reliability of our results, we employ cross-validation. The datasets *A* and *B* are split into six and five folds, respectively. We stick to the splits in [6] to ensure the comparability of our results. These splits further ensure that there is no overlap between the training and test datasets. Finally, we provide uncertainties for our results by calculating the standard deviation across the various folds (and dividing by the square root of the number of folds).

As cross-validation is not common for dataset C, we employ the dataset split proposed by its creators, guaranteeing comparability.

In order to quantify the correlations that we want to study, we define what we call *mean Absolute Correlation* (mAC) in Def. 3.1.

Definition 3.1 (mean Absolute Correlation (mAC)): Given the representation r_i^μ of the feature μ of sample i, we define mAC as:

$$mAC = \frac{2}{D \cdot (D-1)} \sum_{\nu}^{D} \sum_{\mu>\nu}^{D} \cdot \|corr^{\mu\nu}\| \quad (2)$$

where $corr^{\mu\nu}$ represents the Pearson correlation between μ and ν

$$corr^{\mu\nu} = \frac{1}{N} \cdot \sum_{i}^{N} \frac{(r_i^\nu - \bar{r^\nu}) \cdot (r_i^\mu - \bar{r^\mu})}{std_{r^\nu} \cdot std_{r^\mu}} \quad (3)$$

We suggest to study mAC since we believe there to be an interdependence between its value and the achieved accuracy of a re-identification model. Additionally, we also define a metric to study dimensional collapse with Def. 3.2.

Definition 3.2 (Correlated Feature): Given the representation r_i^μ of the feature $\mu \in [1, D]$ of sample i, we define the amount of correlated features as:

$$CF = D - rank(r) \tag{4}$$

Here $rank(r)$ represents the matrix rank of r.

IV. Correlated Representations

A data representation of a higher dimensionality allows learning a more complicated distance or loss function. As a representation including correlations or redundancies does not utilize all of the dimensions it has access to, it stands to reason that it is also not the most efficient representation. To examine this topic further, we herein propose Def. 4.1.

Definition 4.1 (Correlated Representation): A representation (in the sense of a learned feature) is considered redundant if it does not improve a model's accuracy due to it being correlated to other features.

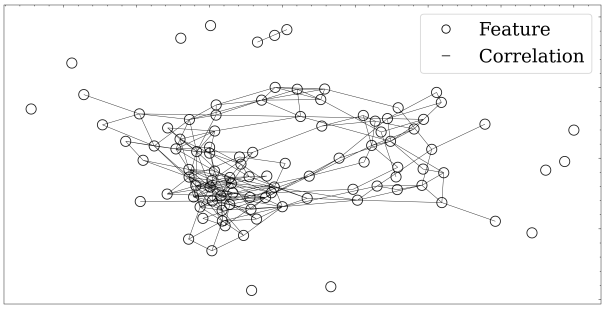

Fig. 3: Visualization of the correlations between learned features for dataset B when using 256px images for training. Each feature is represented as a node, while the edges between them represent correlations of either $> 75\%$ or $< -75\%$.

An example of such correlations can be visualized by using a graph representation of the mAC defined in the previous section (see Fig. 3). In such a visualization, a link, represented as edges, between two features, represented as nodes, means that these features hold a correlation of at least 75%. 89% of features represented in this figure are connected to at least one other feature and thus are at least partially redundant. We suggest using this amount of interconnections as an indicator of correlated representations.

As previously mentioned, we find there to be an interdependence between mAC and the accuracy of a re-identification model. We demonstrate this relationship in Fig. 4. The figure shows the achieved accuracy depending on the image resolution used during training in relation to mAC. A trend that closely follows a linear dependence fit can be seen, i.e., a decrease in mAC seems to lead to an increase in accuracy.

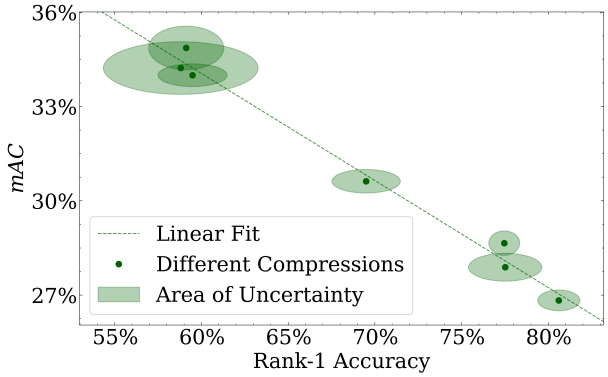

Fig. 4: Accuracy and mAC for different networks on different compression levels on dataset A. These range from 256px in each direction to $\frac{256}{7} \approx 37$px. Notice the evident correlation (-99.4%) between the accuracy and mac value. The specific values of the different compressions, as well as further information on the figures of this work, can be found in the supplementary material.

This is related to the effect of dimensional collapse, as has been studied before in [21]. In this case, as measured by Eqn. 4, the features become linearly dependent on each other and thus the effective dimensionality of the learned representation is smaller than its amount of features and thus CF increases. This effect also occurs in almost all our experiments, but with mAC we also measure an additional effect. This can be seen, as minimizing the amount of correlated representations without affecting the mAC is possible, especially since it is easy to control them using specific hyperparameter choices.

While a prominent pattern is visible for the relationship of mAC and the achieved accuracy of a model, the amount of non-correlated representations paints a different picture. As can be seen in Fig. 5, some relationship between the amount of non-correlated features and the achieved accuracy could be present. However, this relationship does not seem to be as linear and obvious as the one shown in Fig. 4. Nonetheless, with a decrease in correlated features, an increase in accuracy also might occur.

We believe correlated representations to occur since our neural networks contain bias. While neural networks are often considered to have a high variance and a low bias [31], they can still only approximate arbitrarily complex functions for the impractical case of an infinite amount of nodes [32]. This implies that realistic neural networks still contain bias. And while the neural networks used in this paper are still able to approximate a fairly complex function, this is not what we aim to achieve in contrastive learning. Instead, we strive to learn multiple complex functions, effectively dividing the number of available neurons per function and thus also their possible complexity.

This forces our networks to converge towards an optimum between the complexity of each learned feature and the amount of them it can learn. Notice that in both cases, the learned representation is not optimal. This also explains the observa-

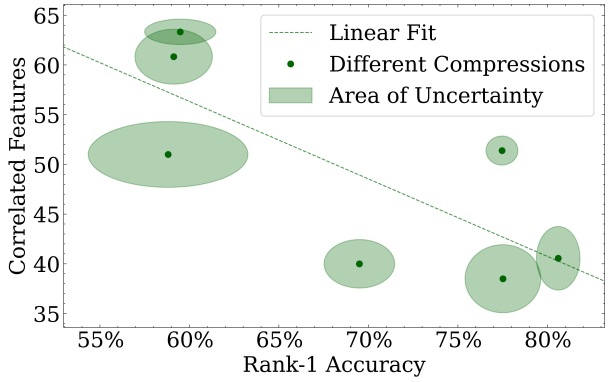

Fig. 5: Accuracy and amount of correlated features (calculated as the difference between the dimension of the prediction matrix and the number of features) for models on different compression levels on dataset A. While there is still correlation (-75%), it is less evident than in Fig. 4.

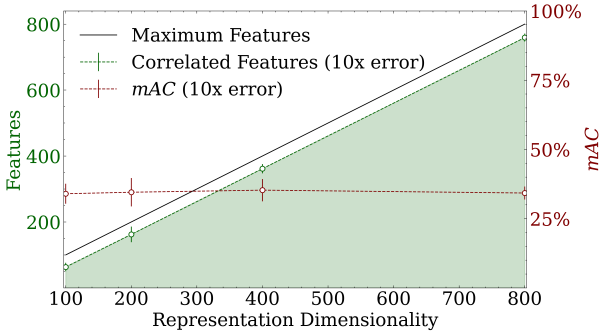

Fig. 6: Amount of correlated features and mAC as a function of the representation dimensionality on dataset A. Notice that while the amount of correlated features grows with the representation dimensionality, this is only because the amount of features increases and the amount of non-correlated features (and mAC) stay approximately the same. We multiply the error by ten to make it more visible.

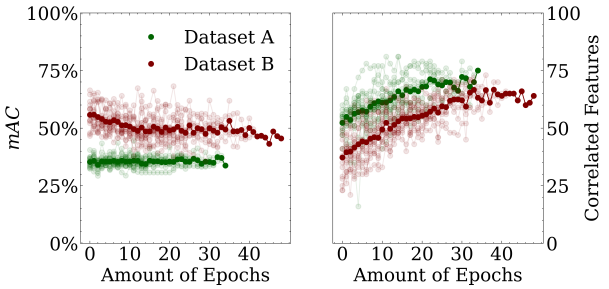

Fig. 7: Amount of correlated features and mAC during training on datasets A and B. We show three repetitions times 5 and 6 folds, respectively, and average the results. Notice how mAC remains relatively constant while the amount of correlated features grows continuously.

tion in Fig. 1. While there is more information available in higher-resolution images, accessing each piece of information requires more complicated functions, thus reducing their available amount and creating a worse representation.

V. Factors Influencing Correlation

To test this assumption and suggest possible solutions in the following sections, we test three factors which we assume to impact the extent to which correlated representations appear.

A. Representation dimensionality

If the complexity given by a neural network is limited, increasing the amount of output features should not increase the quality of the resulting representation. While when there is another reason for the high amount of correlations between learned features, like a deficiency of the loss function, increasing their amount might increase the amount of information stored in them, thus improving prediction results. To test this, we present mAC and the number of correlated features for various representation dimensions in Fig. 6.

We see the expected increase in correlated features as well as an almost constant mAC and amount of useful features. And while not shown here, the accuracy also stays almost constant. When adding more features to the learned representation, these will become redundant and added to the correlated features set, confirming our assumption.

B. Training behavior

In this section, we studied how the amount of correlated features and mAC change in relation to the training duration, i.e., the amount of epochs used. This study is presented in Fig. 7 for datasets A and B, respectively.

For this experiment, mAC seems to stay nearly constant. This might explain the higher correlation with the resulting accuracy, as the value is not affected by changes in the training process.

Importantly, the amount of correlated features tends to start as a smaller overall percentage and increases continuously and nearly linearly during training. This again matches our assumption: There might be enough hidden nodes to generate a high amount of basic features. However, when more complex features are learned during training, more nodes are needed to generate the former. Thus, only a lower amount of more complex features can be created, and the remaining features are then just linear combinations of those.

C. Hyperparameters

Finally, we test the influence that hyperparameter changes have on the degree to which correlated representations occur. But, since testing all possible hyperparameters and their combinations would reach beyond the scope of this publication, we choose to focus on changes in the learning rate and activation function. We do so since both tend to greatly affect neural networks and especially re-identification models.

In Fig. 8, we present the changes in accuracy, mAC, and the amount of correlated features depending on learning rate and activation function changes. We vary the learning rate

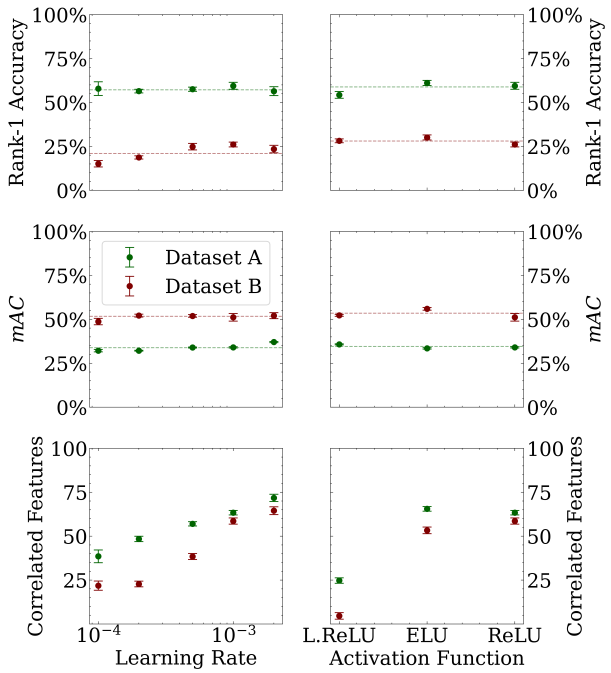

Fig. 8: Relation between accuracy, mAC, and the amount of correlated features depending on learning rate and activation function changes for datasets A and B. While both accuracy and mAC remain nearly constant when varying the learning rate and the activation function, the amount of correlated features changes drastically. To highlight this, the average accuracy and mAC are indicated by a dotted line.

between 0.0001 and 0.02 and use ReLU, ELU and Leaky ReLU as activation functions. We chose to experiment with ReLU and its derivatives, as activation functions with multiple almost constant regions (like sigmoid or tanh) tend to provide drastically worse results.

While the accuracy and especially mAC remain almost constant, both the activation function as well as the learning rate show a noticeable influence on the amount of correlated features. Especially Leaky ReLU [33] leads to the lowest amount of correlated features. We believe this to be the case due to the observation that correlated representations, in the way we gauge them, are closely related to the phenomenon of dead neurons (neurons that always output a constant value regardless of input) [34]. When half of the neurons in a layer are dead, at least half of the outputs of the next layer will be correlated (ignoring effects caused by the activation function). Dead neurons are more likely to appear when the activation function possesses regions in which only minor changes occur. Thus, changing the activation so that it does not contain these regions (as Leaky ReLU does) also decreases the amount of correlated features.

Still this does not solve the fundamental problem of limited complexity, as the almost constant mAC and accuracy indicate.

VI. Proposed Solutions

After studying the effects of several changes on the phenomenon of correlated representations, we present first suggestions to mitigate this effect, in order to potentially improve the performance of re-identification models. As seen in the preceding sections, a neural network at some point reaches a complexity limit of representations learned. Thus, to reduce the occurrence of correlated representations, we will focus on methods to increase this limit by reducing the bias of our networks.

A. Neural network parameters

One way in which the bias of a neural network can be decreased is to increase the amount of features of each layer. Instead of the amount of features described in the *Experimental Setup*, we increase this number using varying multipliers. The resulting effects are shown in Fig. 9.

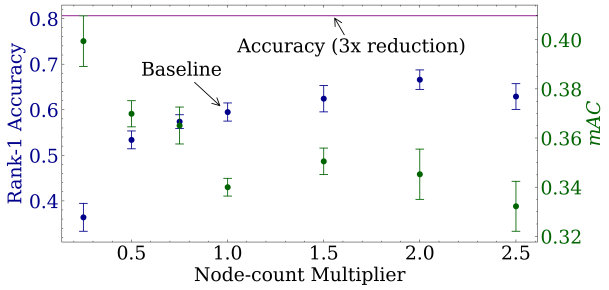

Fig. 9: Accuracy and mAC on dataset A when multiplying the amount of features after each hidden layer. While this can improve both values, its effects remain less significant than the effect stemming from using low-dimensional inputs (depicted as the red line at the top of the diagram).

While this decreases mAC and increases accuracy, there is a limit to how much this affects the results. In addition, the expenses in terms of both memory and time are a quadratic function of the multiplication factor, making it difficult to bring this approach to scale. On our graphics card, we cannot scale up our network by a factor of 3 or more.

B. Ensembling

So instead, we suggest decreasing the bias of a model by replacing it with an ensemble of models [35], [36]. To study this, we divide the 100 dimensional representation learned by a neural network $f(x)$ into k parts and let each part be learned by an independent network of the same architecture. Afterwards, each sub-representation is concatenated into a similar 100 dimensional representation (for the case of $k = 3$, the dimensionality is 99). This allows us to compare the generated features, as shown in Fig. 10.

We notice that ensembling is more than enough to outperform the benefits a lower dimensional representation offers (our best results through lower dimensional representations are shown as horizontal lines in the figure). The main benefit of ensembling seems to be that the non-correlated representations

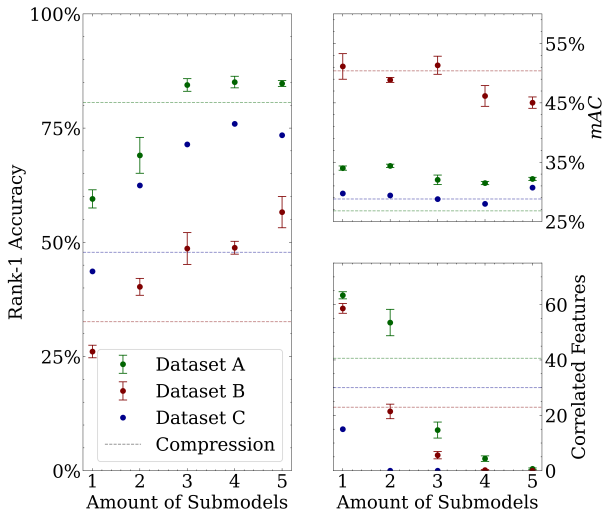

Fig. 10: Accuracy, mAC, and amount of correlated features, when dividing the 100 dimensional representation into k independent models contributing $\lfloor \frac{100}{k} \rfloor$ parameters to the representation. All datasets are used for this experiment.

of one model are not the same as those of the next one. Notice that the amount of parameters is approximately the same for an ensemble of five submodels (8.3 million on dataset A) as for a model with 2.5 times more nodes in every hidden layer (10.3 million). Yet, the ensemble significantly increases performance, proving that ensembles are a more viable solution than increasing the model complexity.

We argue that this is the case since when the outputs of the hidden layers are not the same, it is less likely that two features learn to represent the very same aspect of an image. Additionally, the ensemble can easily be parallelized, further decreasing training durations.

Interestingly, while mAC seems to decrease thanks to ensembling, it is not always able to reach the same mAC as was reached through compression. This implies that further optimization steps are possible, by employing other methods to decrease mAC specifically.

The fixed representation dimensionality still limits the results shown in Fig. 10. At some point, the benefit of more submodels is hindered by fewer non-correlated features per submodel. To improve our results further, we test ensembles of more submodels with the same representation dimensionality. We use submodels with a representation size of 25/100, as this should allow us to capture a majority of non-correlated features without unnecessarily bloating the representation dimensionality. These ensembles are shown in Fig. 11.

An increase in representation dimensionality helps the ensemble drastically, improving the already competitive performance in Fig. 10 of 84.7% (dataset A) and 56.6% (dataset B) to 94.0% (dataset A) and 82.3% (dataset B). Interestingly enough, ensembles on all datasets converge at different amounts of submodel used. While dataset A reaches its peak performance at approximately 14 submodels, dataset

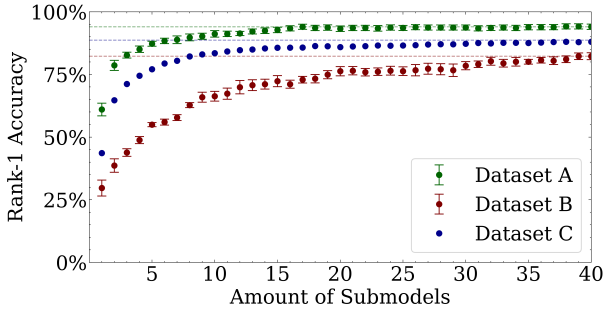

Fig. 11: Accuracy of an ensemble of 25 dimensional submodels (for dataset A and B, but 100 dimensions for dataset C because of the lower amount of correlated features) as a function of the amount of models.

B requires more than twice the amount to converge (approximately 35). This might be due to a more heterogeneous set of data in dataset B, displaying chipwood that might hold more individual features than the images of galvanized metal plates in dataset A. Similarly, the high amount of features in pedestrian re-identification tasks means that dataset C does not yet converge completely in the shown range. Finally, while the achieved performance of 88.7% is only vaguely comparable to other modern solutions, it shows that the same phenomenon of correlated representations also appears on a state-of-the-art dataset for a common pedestrian re-identification tasks.

VII. Conclusion

In this work, we presented, characterized, and studied the phenomenon we call correlated representations. We conducted some first experiments in which we attempted to demonstrate the occurrence of this newly defined phenomenon when performing a common contrastive learning task. We applied our models to multiple re-identification datasets, showing that the phenomenon occurs in contrastive learning-based re-identification tasks in general. We delved deeper into the effects that correlated representations have on re-identification accuracy, explained it through the bias of our neural networks and finally proposed some preliminary solutions. By doing so, we were able to mitigate the accuracy-decreasing effect of the herein presented phenomenon and increase the performance further.

Nonetheless, our results are preliminary and invite future research, e.g., by using more datasets and models. Notably, while the resulting accuracy is promising, the improvement of one of our metrics, mAC, is not yet as substantial as we would have expected, potentially allowing us to improve the performance even further. Additionally, it would be interesting to carefully optimize our hyperparameters to see if our approach can outperform state-of-the-art models.

VIII. Acknowledgement

This work was supported by the Lamarr-Institute for ML and AI, the Research Center Trustworthy Data Science and Security, the Federal Ministry of Education and Research of

Germany and the German federal state of NRW. The Linux HPC cluster at TU Dortmund University, a project of the German Research Foundation, provided the computing power.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 539–546, 2005.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[3] J. Rutinowski, J. Endendyk, C. Reining, and M. Roidl, "galvanized-636 – A galvanized steel re-identification dataset," *Zenodo DOI: 10.5281/zenodo.7386956*, Dec. 2022.

[4] Z. Ming, M. Zhu, X. Wang, J. Zhu, J. Cheng, C. Gao, Y. Yang, and X. Wei, "Deep learning-based person re-identification methods: A survey and outlook of recent works," *Image and Vision Computing*, vol. 119, 2022.

[5] A. Zahra, N. Perwaiz, M. Shahzad, and M. M. Fraz, "Person re-identification: A retrospective on domain specific open challenges and future trends," *Pattern Recognition*, 2023.

[6] S. Klüttermann, J. Rutinowski, A. Nguyen, B. Grimme, and E. Müller, "On the effectiveness of heterogeneous ensemble methods for the re-identification of industrial entities," *under review: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2024.

[7] V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa, and M. A. Riegler, "Impact of image resolution on deep learning performance in endoscopy image classification: an experimental study using a large dataset of endoscopic images," *Diagnostics*, vol. 11, no. 12, 2021.

[8] C. F. Sabottke and B. M. Spieler, "The effect of image resolution on deep learning in radiography," *Radiology: Artificial Intelligence*, vol. 2, no. 1, 2020.

[9] M. Koziarski and B. Cyganek, "Impact of low resolution on image recognition with deep neural networks: An experimental study," *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 4, pp. 735–744, 2018.

[10] O. Ciga, T. Xu, and A. L. Martel, "Self supervised contrastive learning for digital histopathology," *Machine Learning with Applications*, vol. 7, pp. 100–198, 2022.

[11] S. Klüttermann, J. Rutinowski, C. Reining, M. Roidl, and E. Müller, "Towards graph representation based re-identification of chipwood pallet blocks," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1543–1550, 2022.

[12] A. Hermans, L. Beye, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv preprint: 1703.07737*, 2017.

[13] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[14] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo, "Detco: Unsupervised contrastive learning for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8392–8401, 2021.

[15] Y. Zhai, X. Guo, Y. Lu, and H. Li, "In Defense of the Classification Loss for Person Re-Identification," *arXiv preprint: 1809.05864*, 2018.

[16] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1318–1327, 2017.

[17] K. Islam, "Deep learning for video-based person re-identification: A survey," *arXiv preprint: 2303.11332*, 2023.

[18] B. V. Li, S. Alibhai, Z. Jewell, D. Li, and H. Zhang, "Using Footprints to Identify and Sex Giant Pandas," *Biological Conservation*, vol. 218, pp. 83–90, 2018.

[19] J. Rutinowski, T. Chilla, C. Pionzewski, C. Reining, and M. ten Hompel, "Towards Re-Identification for Warehousing Entities – A Work-in-Progress Study," in *Proceedings of the IEEE Conference on Emerging Technologies In Factory Automation (ETFA)*, pp. 501–504, 2021.

[20] J. Rutinowski, C. Pionzewski, T. Chilla, C. Reining, and M. T. Hompel, "Deep learning based re-identification of wooden euro-pallets," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 113–117, 2022.

[21] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," *arXiv preprint: 2110.09348*, 2022.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[23] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proceedings of the International Conference on Machine Learning*, pp. 12310–12320, PMLR, 2021.

[24] L. Miklautz, D. Mautz, M. C. Altinigneli, C. Böhm, and C. Plant, "Deep embedded non-redundant clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5174–5181, 2020.

[25] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao, "On feature decorrelation in self-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

[26] B. O. Ayinde, T. Inanc, and J. M. Zurada, "Redundant feature pruning for accelerated inference in deep neural networks," *Neural Networks*, vol. 118, pp. 148–158, 2019.

[27] L. Prechelt, *Early Stopping — But When?*, pp. 53–67. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[28] J. Rutinowski, T. Chilla, C. Pionzewski, C. Reining, and M. ten Hompel, "pallet-block-502 – A chipwood re-identification dataset," *Zenodo DOI: 10.5281/zenodo.6353714*, Sept. 2021.

[29] J. Rutinowski, B. Vankayalapati, N. Schwenzfeier, M. Acosta, and C. Reining, "On the applicability of synthetic data for re-identification," *AAAI Workshop on AI to Accelerate Science and Engineering (AI2ASE). arXiv preprint: 2212.10105*, 2022.

[30] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[31] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15849–15854, 2019.

[32] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, Dec 1989.

[33] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky relu," in *2020 IEEE Symposium on Computers and communications (ISCC)*, pp. 1–7, 2020.

[34] L. Lu, Y. Shin, Y. Su, and G. Karniadakis, "Dying relu and initialization: Theory and numerical examples," *Communications in Computational Physics*, vol. 28, no. 5, pp. 1671–1706, 2020.

[35] Y. Bian and H. Chen, "When does diversity help generalization in classification ensembles?," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9059–9075, 2022.

[36] N. Gupta, J. Smith, B. Adlam, and Z. E. Mariet, "Ensembles of classifiers: a bias-variance perspective," *Transactions on Machine Learning Research*, 2022.