sensors

MDPI

Article

Comparing Class-Aware and Pairwise Loss Functions for Deep Metric Learning in Wildlife Re-Identification [†]

Nkosikhona Dlamini [1,*,‡] and Terence L. van Zyl [2,*,§]

[1] Faculty of Science, Braamfontein Campus, School of Computer Science and Applied Mathematics, University of Witwatersrand, Johannesburg 2000, South Africa
[2] Auckland Park Campus, Institute for Intelligent Systems, University of Johannesburg, Johannesburg 2006, South Africa
[*] Correspondence: dlamininkd@gmail.com (N.D.); tvanzyl@gmail.com (T.L.v.Z.)

[†] This paper is an extended version of our paper published in "Nkosikhona Dlamini and Terence L van Zyl, Automated Identification of Individuals in Wildlife Population Using Siamese Neural Networks" in Proceedings of ISCMI, Stockholm, Sweden, 20 November 2020.

[‡] Current Address: 1 Jan Smuts Avenue, Braamfontein, Johannesburg 2000, South Africa.
[§] Current Address: Kingsway Avenue & University Road, Auckland Park, Johannesburg 2006, South Africa.

Abstract: Similarity learning using deep convolutional neural networks has been applied extensively in solving computer vision problems. This attraction is supported by its success in one-shot and zero-shot classification applications. The advances in similarity learning are essential for smaller datasets or datasets in which few class labels exist per class such as wildlife re-identification. Improving the performance of similarity learning models comes with developing new sampling techniques and designing loss functions better suited to training similarity in neural networks. However, the impact of these advances is tested on larger datasets, with limited attention given to smaller imbalanced datasets such as those found in unique wildlife re-identification. To this end, we test the advances in loss functions for similarity learning on several animal re-identification tasks. We add two new public datasets, Nyala and Lions, to the challenge of animal re-identification. Our results are state of the art on all public datasets tested except Pandas. The achieved Top-1 Recall is 94.8% on the Zebra dataset, 72.3% on the Nyala dataset, 79.7% on the Chimps dataset and, on the Tiger dataset, it is 88.9%. For the Lion dataset, we set a new benchmark at 94.8%. We find that the best performing loss function across all datasets is generally the triplet loss; however, there is only a marginal improvement compared to the performance achieved by Proxy-NCA models. We demonstrate that no single neural network architecture combined with a loss function is best suited for all datasets, although VGG-11 may be the most robust first choice. Our results highlight the need for broader experimentation and exploration of loss functions and neural network architecture for the more challenging task, over classical benchmarks, of wildlife re-identification.

Keywords: Proxy-NCA; similarity learning; triplet-loss; semi-hard negative mining

check for updates

Citation: Dlamini, N.; van Zyl, T.L. Comparing Class-Aware and Pairwise Loss Functions for Deep Metric Learning in Wildlife Re-Identification. *Sensors* **2021**, *21*, 6109. https://doi.org/10.3390/s21186109

Academic Editors: Paul Fergus, Carl Chalmers and Serge Wich

Received: 16 June 2021
Accepted: 27 July 2021
Published: 12 September 2021

1. Introduction

Wildlife monitoring and re-identification have evolved from capture-recapture, radio frequency identification to image-based automated re-identification using animal biometric features. Capture-and-recapture was dominant in the early to late 1990s when population estimation experiments extended over a year; first, animals were captured and marked then recaptured periodically to obtain a count [1]. Capture-recapture poses challenges where animal/human confrontation is unnecessary when the studies involve animals such as Lions. Ariff and Ismail [2] state that the RFIDs technology emerged as an alternative since using RFIDs removes the need to recapture. Individuals can, as a result, be identified from a distance. However, Schacter and Jones [3] discuss that implanting RFID tags alters the physiology and behaviour of the host animal. These behavioural changes can lead to ill-health and other effects that negatively impact the conservation of wild animals. Further, RFIDs have high costs associated with the deployment and maintenance of these systems [4]. Recent studies aim to replace RFID technologies with image-based biometric systems [5].

Animal biometric re-identification using images originated in domesticated animal applications and has gradually transitioned towards wild animal species [6]. Research in image-based biometric re-identification of animals was prompted by recent advances in human biometric projects using deep neural networks [7]. However, often images of wild animals are occluded by tree branches, contain background or are intentionally camouflaged [8]. The work of Verma and Gupta [9] shows that the deep neural networks can detect the animal existence in an image in what is termed animal background verification.

The era of image-based biometric re-identification is dominated by a search for better neural network architectures, better parameters, and better approaches ranging from deep neural networks for classification to deep neural networks for similarly learning, with the latter being the main focus of recent research. Deep neural networks for similarity learning involve selecting pair/triplet sampling techniques, ranking loss functions, and model evaluation metrics. However, these approaches are often evaluated on larger datasets with thousands of data points and tens of instances per individuals/class, all relatively well-balanced [10,11]. As a result, there remains a gap in knowledge concerning the efficacy of the increased complexity of recently proposed methods on smaller unbalanced datasets with fewer instances per class. The survey by Kaya and Bilge [12] depicts a summary of these advances in similarity learning and discusses varying model performances on different datasets.

The current research investigates how different design choices in deep similarity learning affect model performance on fine-grained biometric re-identification tasks in wild animals. Specifically, the current research focuses on comparing recent advances in loss functions and neural network architecture. The models are evaluated using Recall@1 and mean average precision at R (MAP@R) [13]. Our main contribution is to demonstrate that there is no single neural network architecture combined with a loss function that is best suited for animal re-identification. In particular, we:

- highlight the need for increased research into loss functions and neural network architecture specifically for wildlife re-identification;
- improve on the state-of-the-art results in numerous animal re-identification tasks;
- contribute two new benchmark datasets with results;
- provide minor support for a choice of triplet loss with a VGG-11 backbone as an initial architecture and loss.

2. Related Work

2.1. Animal Biometrics and Computer Vision

Wildlife biometric identifiers are distinctive and measurable characteristics used to label and describe individual animals within the population. Animal biometrics is the field that studies visual and behavioural features that discriminate species and individuals within species in the animal kingdom [14]. Identifying species is a coarse-grained task using mutual features across individuals, whereas individual re-identification is a fine-grained task reliant on distinct features among individuals. Fine-grained re-identification requires expert knowledge to distinguish between individuals in a species who might share many mutual features [15]. According to Kühl and Burghardt [16], the features used varies among species since a specific set of criteria needs to be met before features become a biometric feature candidate, namely [17]:

Universality: all the individuals in the population must have such a feature;
Uniqueness: two or more individuals should have a different form of the same feature.

The use of biometric re-identification through images to identify animals became popular because there are few disturbances in animals' natural habitat as the animal image data are collected via camera traps. Computer vision experts researched different areas in the systems developed for re-identification through images [18]. The broader areas of research included improving model deployment flexibility and overall experiment design, from the sampling of training data points up to more robust ways of evaluating model performance.

2.2. Classification vs. Similarity Learning

According to Krizhevsky et al. [19], two main approaches employed in computer vision using deep convolutional neural networks (DCNN) are classification and similarity learning. Chen et al. [20] and Li et al. [21] demonstrate that (DCNN) produced state-of-the-art performance in solving ImageNet data tasks. Systems that emerged from classification-driven tasks were DeepID [22] which achieved an accuracy of 99.15% in human re-identification using face images. In classification tasks, the top layers of the DCNNs are responsible for assigning N-way class probabilities via an activation function, namely: soft-max or sigmoid. Models trained for N-way classification cannot be used if the number of classes in the dataset increases, retraining the model becomes necessary, this reduces the real-world deployment of these models as training deep neural networks may take several days [23]. As a result of this, there is an interest in finding ways of avoiding costly retraining, and research on similarity learning models has gained traction [24].

2.3. Similarity Learning

Similarity learning does not include an output layer that assigns class probability [25]. Instead, a similarity models' output layer produces an embedding of the input image into a representative feature space. A similarity learning model has two identical neural networks called Siamese neural networks. These networks are trained with back-propagation to optimise a ranking loss function. Each neural network produces an embedding through back-propagation, and the network's weights are updated to minimise the loss [26].

A similarity learning model is trained to learn a representative feature vector of an object from an image signal. The feature vector is generated so that dis/similar real-world objects are dis/similar in the embedding space. The application of similarity learning dates back to early 1990s where Bromley et al. [27] used similarity learning in automating signature verification [28]. Schroff et al. [29] discussed the successes of similarity learning in the re-identification of human beings using facial images. There are a significant number of loss functions adopted in experiments for training similarity learning models, namely contrastive loss, triplet loss, and Proxy-NCA [30–32].

2.4. Sampling Techniques for Pairwise Training

Schroff et al. [29] contended that for effective training of similarity learning models on triplet loss, it is essential to select meaningful pairs/triplets. Several research efforts produced the following pair mining schemes:

2.4.1. Hard Negative Mining

Hard negative mining is selecting image pairs; x_a and x_n that have most similar embedding vectors, yet they belong to different classes $C(x_a) \neq C(x_n)$:

$$x_s = \underset{d}{\operatorname{argmin}}\ d(f(x_a), f(x))x : C(x_a) \neq C(x), \tag{1}$$

where x_s is the candidate sample, f is a function that produces the embedding vectors for both x_a and x_n, and d is a distance measure [33]. The measure of how close two embedding vectors are is based on the hyper-parameter: margin or α. Where a triplet selection is needed, it is expected that the distance between an anchor image and a positive image be less than the distance between anchor image and negative image:

$$d(f((x_a, f(x_p))) + \alpha \ < d(f(x_a, f(x_n))). \tag{2}$$

2.4.2. Semi-Hard Negative Mining

Schroff et al. [29] argue that the Hard-negative pair mining technique can lead to bad generalisation and model collapse. An alternative to Hard-negative pair mining was proposed; known as Semi-hard negative, where the distance between the anchor and negative example images is allowed to lie within the margin α but still far from the distance between positive and anchor examples Equation (3):

$$d(f((x_a, f(x_p))) < d(f(x_a, f(x_n)). \tag{3}$$

The semi-hard negative triplet mining and triplet loss minimisation produced superior performance in human re-identification from face images [29]. Several additional mining techniques include easy positive mining, easy negative mining, and hard positive mining. Details on these techniques are discussed by Xuan et al. 2020 [33].

Previous works have shown that even with the best performing architecture, these mining techniques and loss functions are not consistent across all datasets. The DenseNet-201 neural network produced 75.5% for the chimpanzee dataset and 92.2% accuracy for the octopus dataset [34]. Another observation is that researchers decide on different metrics to measure their models' performance; some report accuracy, others Recall@*k*, and some mean average precision MAP, making it challenging to compare [12,35].

2.5. Challenges in Deep Metric Learning

Roth et al. [35] highlighted the challenges of comparing works in metric learning. These challenges are a result of the varying objectives the researchers are addressing in their work. Some researchers are seeking to improve the architecture setup [36], others are aiming to obtain better objective functions (loss functions), while others introduce new algorithms to solve problems. Kim et al. [37] proposed an efficient facial expression recognition algorithm. However, it is difficult to ascertain that the improvements come from the proposed algorithm or design choices, namely: feature extraction methods, choice of optimiser function or the selected network architecture. The concerns expressed by Musgrave et al. [13] are to the effect that experiments should be compared fairly by isolating the contributing factor to the observed performance improvements. One way to isolate the contributing factor is by applying the same algorithm to different neural network architectures across various animals to confirm that improvements come from the proposed factor.

Musgrave et al. [13] clarified that before metric learning projects claim superior performance over previous works, there should be a fair comparison. This comparison includes keeping all parameters the same to identify what contributes to the improvement. Comparing experiments where different neural network architectures and different model parameters are used is seen as problematic. Roth et al. [35] and Schroff et al. [29] agree that changing the embedding dimension size has an effect on model performance. Similar observations were made by Wang et al. [38] where the embedding sizes were tuned to find an optimum size that results in good performance while keeping all parameters of the model the same. This discussion highlights that a simple change of the output embedding size can create an unfair comparison if the objective was to compare, say, for instance, loss functions.

Another area that can create unfair comparisons is selecting loss optimiser functions and their hyper-parameters, such as the learning rate. It may not be incorrect to find a neural network architecture, an optimiser, and hyper-parameters that yield good performance in solving a task. However, a requirement should be transparency in the choices and omissions made by researchers. If future works make similar comparisons, the objectives need to be stated, and all other design choices must be kept consistent.

2.6. Animal Biometrics Using Image Features

Deep metric learning has received significant attention for solving image-based problems. The gradual use of neural networks on wild animals datasets is reflected in the survey conducted by Schneider et al. [39]. In the re-identification of animals, van Zyl et al. [8] used deep metric learning to identify individuals in Zebra and Nyala datasets. ResNet-50 pre-trained on ImageNet dataset was used as the DCNN backbone architecture, the contrastive loss approach was employed to output embedding vectors of 40 dimensions. The training pairs were randomly sampled. The datasets used in the experiments were comparatively smaller: 820 Zebra data points of 84 classes (individuals), and the Nyala was 1945 data points of 474 classes. Schneider et al. [34] presented a comparative study between AlexNet, VGG-19, DenseNet-201, ResNet-152 and Inception-V3 in re-identification of Chimpanzees, Whale, Fruitfly and Tiger. The mean average precision achieved by DenseNet-201 DCNN trained on triplet loss in the Chimpanzees dataset was 93.2%.

Qiao et al. [40] used the Inception-V3 convolutional neural network to extract discriminating features from images of farm cattle. A Long short term memory (LSTM) was used to extract temporal features from video sequences that capture variations in gait among other temporal features of an individual in the cattle dataset. The Inception-V3 was pre-trained on ImageNet. There were 8780 images of 41 individuals. The classification layer was reduced to 41-dimensions, representing the number of classes in the dataset. The loss function used during training was categorical cross-entropy loss.

Other researchers looked at the re-identification of individuals from Panda face images. Hou et al. [41] collected and experimented on 25 individual Pandas each with about 4300 images. The total number of data points was 65,000. VGGNet was trained with the top fully connected layer of 25-dimension, a soft-max activation function was used to assign class probabilities. The reported metric is the accuracy of 95% for detecting individuals from the test dataset and 78.7% on detecting unknown individuals in the dataset.

Korschens and Denzler [15] investigated the use of ResNet-50 neural network architecture, pre-trained on ImageNet, in the re-identification of individuals in an elephant dataset. Earlier layers of ResNet-50 were fine-tuned instead of fine-tuning the top layer. The features generated by ResNet-50 were used to fit an SVM classifier model. The elephant dataset had just over 2000 data points split into 75% training and 25% test set. The reported top-10 accuracy was 79%. Korschens and Denzler [15] did not investigate shallower variants of the ResNet architecture or alternative convolutional neural network architectures.

Nepovinnykh et al. [42] adapted a Deeplab [20] model that was trained on triplet loss to re-identify individuals in a ringed seals population. Deeplab was pre-trained on Pascal VOC dataset [43], no pair sampling technique was discussed in the Deeplab system. The top-5 accuracy reported was 87%. The accuracy reported was on a hold-out test data of 2000 points, with 46 individual seals.

The current work expands on previous wildlife re-identification by using a similarity learning approach coupled with searching for optimal loss functions, sampling techniques, and convolutional neural networks architecture. The specific species considered are individuals in a lion population and individuals in a Nyala population. The only prior study to the authors' knowledge working on a Lion dataset focused on annotating lion images to depict the lion's activity [44].

Further, we note the threat to the world population of Lions demands that conservation efforts be enhanced [45]. Norouzzadeh et al. [46] stated that conservationists will leverage automated counting and tracking animals to report on biodiversity reliably, with less human effort and fewer disruptions on the habitat of wild animals.

2.7. Loss Functions

2.7.1. Pairwise Loss

Contrastive loss is the original pairwise loss function that takes a pair of image samples, and a class indicator function. The training follows a feed-forward with back propagation, learning embedding vectors that maximise the embedding distance if the images are from a different class, and minimises the distance if the images belong to the same class [30].

Triplet loss, an improvement on contrastive loss shown in Figure 1, takes three images: an anchor image x_i^a, a positive image x_i^p and a negative image x_i^n. The network learns to minimise the Euclidean distance between the embedding of anchor and positive image, and increases the distance between the anchor and negative image simultaneously by minimising:

$$\sum_{i}^{N}\left\lceil \| f(x_i^a) - f\left(x_i^p\right) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha, 0\right\rceil, \tag{4}$$

where α is a hyper-parameter forcing a minimum distance between images of different class. A graphical view of the triplet loss training is given in Figure 1.

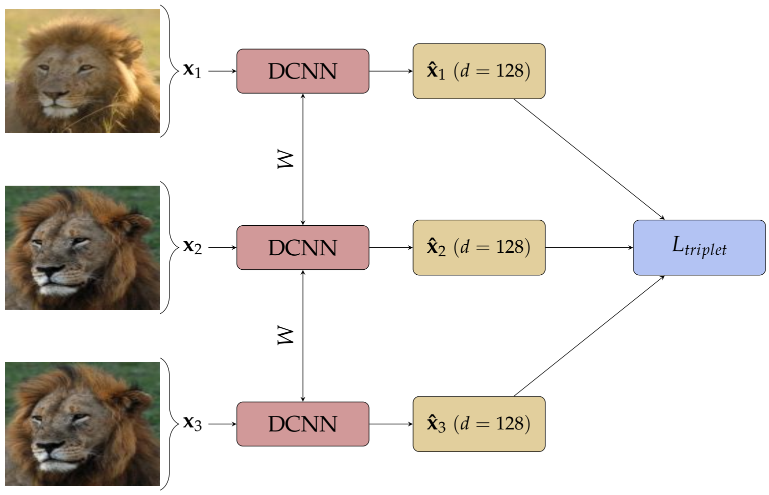
$\mathbf{x}_1$
DCNN
$\hat{\mathbf{x}}_1$ $(d = 128)$
W
$\mathbf{x}_2$
DCNN
$\hat{\mathbf{x}}_2$ $(d = 128)$
$L_{triplet}$
W
$\mathbf{x}_3$
DCNN
$\hat{\mathbf{x}}_3$ $(d = 128)$

Figure 1. Triplet Network with d the dimension of the embedding.

The triplet loss and contrastive loss functions are regarded as pairwise losses since they are computed considering the similarities of pairs. The triplet loss approach, coupled with semi-hard pair sampling, resulted in state-of-the-art performance in human face re-identification in experiments [29]. As reflected in Section 2.6, triplet loss and contrastive loss have been applied in animal re-identification tasks by Nepovinnykh et al. [42] and van Zyl et al. [8]. The pairwise loss functions demand that meaningful pairs are found during the training phase. Due to the inefficiencies of pair-mining, researchers explored ways to avoid mining training pairs by developing class-aware loss functions [47].

2.7.2. Class Distribution Based Loss

Wang et al. [48] and Chen et al. [49] also acknowledged the limitations of pairwise loss functions such as contrastive loss; however, instead of formulating a completely new loss functions, they proposed a distribution aware variant of the contrastive loss function. Rippel et al. [50] proposed a deviation from pairwise losses to what Wang et al. [38] termed class aware losses. The proposed loss function, Magnet loss, considers class clusters that are updated during training. These clusters capture intra-class variations but also penalise inter-class overlaps. This approach ensures that clusters of the same class attract and clusters of different classes repulse. This approach is different from the pairwise loss functions that do not consider clusters but individual pairs. Rippel et al. [50] note that the Magnet loss function demands that partially pre-trained models be used; through fine-tuning deeper layers as opposed to fine-tuning top layers.

A similar approach of deviating from pairwise losses to class-aware losses is discussed by Movshovitz-Attias et al. [51]. An adaptation of NCA [52] is presented called Proxy-NCA, with a deliberately constructed small set of data points P referred to as proxies to a point x. The assumption is that a point in P is sufficiently close to x in terms of a distance d and as such can be a substitute for x. This point called the proxy of x is given by:

$$p(x) = \underset{p}{\text{argmin}}\ d(x, p), \tag{5}$$

where $p \in P$. ϵ is the proxy approximation error and is given by the maximum error using all data points in P:

$$\epsilon = \max\ d(x, p(x)). \tag{6}$$

In the case where class labels are available, the proxies are selected using class labels. The ranking loss is minimised amongst a data point anchor x with two proxies $p(y), p(z)$ where $p(y)$ is a proxy of x with data points of the same label, and $p(Z)$ is a proxy of the data point with a different label from the anchor. In each training iteration, sample P containing (x, y, Z) is selected from the training dataset, and

$$l = -\log\left(\frac{\exp(-d(x, p(y)))}{\sum_{p(z)\in p(Z)} \exp(-d(x, p(z)))}\right) \tag{7}$$

is minimised. This approach removes the need to mine training pairs. Proxy-NCA was used with inception network architecture in image retrieval tasks for datasets: Cars196 [53] and Stanford Products dataset, where better performance is observed with a margin improvement of 21.7% when compared with the same model trained on triplet loss and semi-hard mining technique [51]. Teh et al. [47] adapted the Proxy-NCA to Proxy-NCA++ which incorporates maximising a proxy probability instead of minimising a proxy distance. Increasing the proxy probability has the same effect of attracting a data point to its positive data points proxy-set and repels data points of different proxies. Teh et al. [47] further proposed a scaling factor to be applied in the proxy loss function in order to scale the distribution of probability across classes. We do not consider Proxy-NCA++ further in this research.

2.8. Motivation

The emerging loss functions that are aimed at replacing the pairwise loss are tested on larger datasets such as Stanford Cars [54] and CUB-200 [55]. The effectiveness of the proposed class-aware loss functions remains unknown for smaller, highly unbalanced datasets. We compare models trained on class-aware loss function with models trained on traditional pairwise loss functions. To provide fair comparisons, we conducted experiments taking into account the challenges identified in Section 2. We widen the scope of our experiments by investigating the effect of changing the loss function across six datasets, applied to five different neural network architectures.

3. Materials and Methods

The structure of experiments followed in deep neural networks projects is depicted in Figure 2. The boundaries indicate areas where design choices are made that affect the resulting models' performance. The sampling section is only relevant when pairwise loss functions are used, such as triplet loss. We ran each experiment 10 times for the train and test set. Each training experiment consists of 30 epochs, and one epoch uses 5-fold cross-validation to update the model weights. Our experiments are compared using two loss functions: Triplet-loss and Proxy-NCA, on six animal datasets discussed further below.

3.1. Data

We carry out experiments using Lion face data collected from the Mara Masia project in Kenya (http://livingwithlions.org/mara/browse/all/all/) (accessed on 27 October 2019). We also trained our models on the Nyala dataset collected from South African nature reserves. Both of these datasets, Nyala and Lions have relatively few (7) examples per an individual. In addition, we conducted experiments with publicly available datasets, namely Tai Zoo Chimpanzee [56], Zebra [57], Panda [58] and Tiger [59] datasets to compare our approach with previous works. Table 1 shows the total data points each dataset has, total individuals in the dataset, and average samples per individual. The neural network architectures we trained were VGG, ResNet, and DenseNet. We followed a zero-shot train-test split, such that all classes in the test set did not form part of the training set.

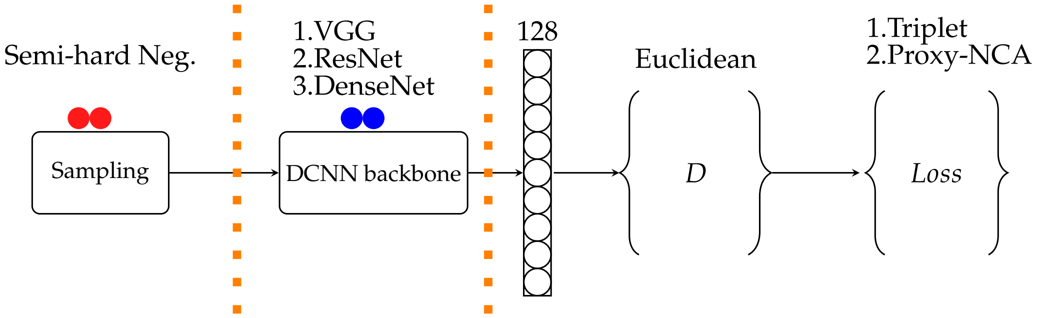

Figure 2. Experiment setup starts with training pairs selection; then, DCNN is the network architecture chosen for a particular experiment, the output of DCNN is an embedding vector of dimension *D* and distance metric measured with ranking loss function being minimized.

Table 1. Sample Size (N), Individuals (I) and Mean Examples (#) per Individual in Full and Train/Test Sets.

Animal	N	I	$\mathbb{E}[\#/I]$	**Split**	N	I
Lion	750	98	7.7 ± 4.0	**Train**	594	79
				Test	156	19
Nyala	1934	274	7.1 ± 5.1	**Train**	1213	179
				Test	729	95
Zebra	2460	45	54.7 ± 7.3	**Train**	1989	36
				Test	471	9
Chimp	5078	78	65.1 ± 17.0	**Train**	3908	62
				Test	1170	16
Panda	6462	218	29.64 ± 8.0	**Train**	5546	174
				Test	916	44
Tiger	3651	182	20.1 ± 15.0	**Train**	1887	107
				Test	1764	75

3.2. Neural Network Architectures

The VGG network architecture came as an improvement to a neural network architecture developed by Krizhevsky et al. [19]. Simonyan and Zisserman [60] constructed VGG as a stack of convolutional layers that use smaller filters of $[3 \times 3]$ size as opposed to the previous neural networks that used filters of varying sizes. In the VGG network, all hidden layers have a ReLU activation function to induce non-linearity. Three fully connected layers, two of these layers have a dimension of 4096, and the last layer with dimension 1000 to capture the classes found in the ImageNet dataset. The Soft-max activation function is applied to the last layer to capture class probabilities. The VGG comes in two commonly used variants, and one variant has 11 layers (VGG-11), eight convolutional layers, and three fully connected layers. VGG-19, on the other hand, is 19 layers deep (16 convolutional layers and three fully connected layers).

He et al. [61] proposed a different convolutional neural network setup (ResNet) to overcome vanishing gradients. He et al. [61] suggested that instead of having stacked layers, some deeper layers are skipped. The motivation for skipping is based on the assumption that if the deeper layers suffer from a vanishing gradient, then earlier layers should have learned an optimal mapping. The output of the earlier layers is used to learn a residual mapping, and this mapping is added on deeper layers preserving the information learned before a possible vanishing gradient occurs. The ResNet architecture comes in numerous variations, namely: ResNet-34, ResNet-50, ResNet-101 and ResNet-152, where the number indicates total layers in the residual neural network.

Huang et al. [62] suggested DenseNet, which is an alternative of both VGG and ResNet architectures. VGG and ResNet are based on the premise that a prior layer's output is fed into the next layer as input. DenseNet, on the other hand, suggests a densely connected architecture where the next layer is fed with output from all other preceding layers. The positive effect of having densely connected neural layers is that the dense connections have a regularisation effect, a good thing for smaller datasets where over-fitting is prevalent. DenseNet also comes in several variants, such as DenseNet-121, DenseNet-201. It is worth noting that the number of layers is not fixed to 121 and 201. Some DenseNet can have N layers and are referred to as DenseNet-N.

3.3. Training

We split our data into 80% training and 20% test set. Table 1 contains the total data points in both the train and hold-out test set. The 80% train set was split into 5-folds in each training epoch. All the backbone architectures used were pre-trained on ImageNet, and these were used as feature extractors as opposed to fine-tuning all layers. We replaced the 1000-class output layer from ImageNet with a 128-dimensional embedding layer for one set of our experiments. On another set, we searched for an optimal output layer size.

We trained for 30 epochs minimising the Proxy-NCA loss. The Proxy-NCA loss parameters need an optimiser; we used the Adam optimiser with default parameters $\beta_1 = 0.9$, $\epsilon = 10^{-3}$ and $\beta_2 = 0.999$. The learning rate was set to 0.0001. The same parameters were used in triplet loss experiments. However, the additional parameter in triplet loss, the margin was kept at 0.2. A semi-hard negative sampling technique was used on batch examples selected randomly from the training set. We ran ten train-test iterations and the results reported are the average performance metrics of the ten test runs with confidence intervals as done in [13].

3.4. Metrics Measured

A recently developed benchmark PyTorch library [63] was used to measure all the metrics reported in our experiments. During the model evaluation, we used hold-out test data containing classes that were not seen during the training phase. To compare with the previous works, we computed the Recall@1. We also measured the mean average precision at R (MAP@R) given in Equation (8) by:

$$\text{MAP@R} = \frac{1}{R}\sum_{i=1}^{R} P(i), \tag{8}$$

where $P(i)$ is precision at i if the ith neighbour is a true positive otherwise $P(i)$ is 0. MAP@R is the mean average precision computed over R nearest neighbours of the query image retrieved by the model of Musgrave et al. [13]. We did not find papers that reported on MAP@R for the datasets we used in our experiments.

3.5. Embedding Dimension Size

For each dataset, we verified if using 128-dimension embedding for the output layer is justified. We tested three embedding dimension sizes at 64, 128 and 512. We kept all other network parameters constant, from sample size up to the loss function parameters and only altered the dimension of the layer before the output layer. The results for each dataset with varying embedding layer dimensions are shown in Table 2 and justify the use of 128 across all experiments.

Table 2. DenseNet-201 Proxy-NCA loss, performance search for optimal output embedding dimension D.

		MAP@R%	
Dataset	**D-64**	**D-128**	**D-512**
Chimp	8.4 ± 0	9.1 ± 1	9.1 ± 0
Nyala	38.0 ± 1	38.6 ± 1	38.5 ± 1
Zebra	29.8 ± 2	29.6 ± 1	30.6 ± 2
Lion	48.8 ± 2	50.6 ± 1	50.5 ± 2
Tiger	21.6 ± 3	23.2 ± 2	23.0 ± 3
Panda	27.5 ± 1	28.4 ± 2	28.1 ± 1

4. Results

As reflected in Table 2, for each dataset, we run preliminary experiments where we searched for the best size of the output embedding vector that results in improved model performance. All the results we present were generated using optimal embedding size for each dataset as found from the preliminary investigation. We present our results for the neural network architectures and datasets used in the experiments. We run a pair of experiments for each neural network backbone: triplet loss vs. Proxy-NCA loss across all datasets. The aim is to observe the effect of each loss function when the dataset, network backbone, and parameters are kept constant.

Table 3 presents Recall@1 and Table 4 presents mean average precision at R (MAP@R). In our tables, we further show what body parts the images of the individual animals contained. The Lions, Panda, and Chimpanzees datasets have faces, while Nyala, Tiger, and Zebra datasets contained the body side (flanks). We report the average of ten train-test runs with statistical significance obtained by doing a two-sided t-test at 95% confidence level. The results in bold in Table 3, and Table 4 are the best performing method. In the same tables, the results in grey are those that are not statically significantly different from the best. Table 5 is extracted from Table 4 where we highlight the best-performing methods concerning both the mean and standard deviations.

Table 3. Recall@1: triplet loss semi-hard mining vs. Proxy-NCA for training classes. Bold indicates the best performing method, and grey highlights results that are not statistically significantly different from the best.

			Top-1/Recall@1				
			Faces			**Flanks**	
Architecture	**Loss**	**Lions**	**Chimps**	**Pandas**	**Nyala**	**Zebra**	**Tiger**
VGG-11	Triplet	66.5 ± 2	79.0 ± 1	**91.2 ± 1**	68.7 ± 2	94.6 ± 0	**88.9 ± 1**
	P-NCA	68.2 ± 3	78.9 ± 1	89.3 ± 2	68.4 ± 2	93.8 ± 2	87.0 ± 1
VGG-19	Triplet	70.2 ± 2	70.6 ± 0	86.3 ± 2	**72.3 ± 0**	82.8 ± 1	86.3 ± 2
	P-NCA	**71.3 ± 3**	66.3 ± 0	90.9 ± 0	69.2 ± 3	82.7 ± 0	84.4 ± 1
ResNet-18	Triplet	67.8 ± 1	79.2 ± 2	90.0 ± 0	64.9 ± 2	**94.8 ± 1**	87.1 ± 1
	P-NCA	66.8 ± 3	77.9 ± 0	90.1 ± 1	64.1 ± 0	93.6 ± 2	84.8 ± 1
ResNet-152	Triplet	63.2 ± 2	71.2 ± 1	87.6 ± 3	61.0 ± 3	80.7 ± 0	76.5 ± 2
	P-NCA	61.0 ± 1	69.5 ± 1	83.4 ± 0	59.7 ± 0	79.1 ± 3	75.5 ± 2
DenseNet-201	Triplet	70.1 ± 1	**79.7 ± 2**	89.6 ± 1	67.1 ± 2	89.1 ± 0	85.0 ± 1
	P-NCA	69.5 ± 3	78.2 ± 2	90.7 ± 1	66.3 ± 1	87.5 ± 0	85.6 ± 1
Prior Research	-	-	77.5 ± 0	92.1 ± –	72.1 ± 0	72.6 ± 0	86.3 ± 0

Table 4. MAP@*R*: triplet loss semi-hard mining vs. Proxy-NCA on training classes. Bold indicates the best performing method, and grey highlights results that are not statistically significantly different from the best.

			MAP@R				
			Faces			**Flanks**	
Architecture	**Loss**	**Lions**	**Chimps**	**Pandas**	**Nyala**	**Zebra**	**Tiger**
VGG-11	Triplet	16.5 ± 2	12.9 ± 2	**32.0 ± 2**	**11.2 ± 0**	16.8 ± 1	22.8 ± 1
	P-NCA	17.7 ± 1	**13.8 ± 3**	31.8 ± 1	11.0 ± 1	16.5 ± 0	22.9 ± 2
VGG-19	Triplet	18.0 ± 2	11.7 ± 1	25.0 ± 0	10.8 ± 1	16.7 ± 2	21.8 ± 1
	P-NCA	17.7 ± 0	12.0 ± 2	28.7 ± 0	9.7 ± 3	16.4 ± 3	20.0 ± 1
ResNet-18	Triplet	18.5 ± 0	11.2 ± 2	26.3 ± 1	9.9 ± 2	**19.0 ± 0**	**24.6 ± 4**
	P-NCA	19.0 ± 1	11.5 ± 1	24.9 ± 0	9.5 ± 1	18.2 ± 1	21.7 ± 2
ResNet-152	Triplet	17.3 ± 2	10.1 ± 0	26.9 ± 1	8.2 ± 0	12.1 ± 3	12.5 ± 3
	P-NCA	17.1 ± 0	9.4 ± 3	20.3 ± 1	9.0 ± 2	11.9 ± 2	11.0 ± 1
DenseNet-201	Triplet	**20.8 ± 1**	9.9 ± 2	31.1 ± 1	11.0 ± 2	15.9 ± 2	22.3 ± 1
	P-NCA	20.2 ± 2	11.6 ± 3	28.4 ± 2	10.4 ± 1	16.0 ± 1	23.2 ± 3

Table 5. Model performance Feature Extractor loss optimal-dimension top layer. Bold indicates the best performing method, and grey highlights results that are not statistically significantly different from the best.

			MAP@R			
		Faces			**Flanks**	
Architecture	**Lions**	**Chimps**	**Panda**	**Nyala**	**Zebra**	**Tiger**
VGG-11	17.7 ± 1	**13.8 ± 3**	**32.0 ± 2**	**11.2 ± 0**	16.7 ± 2	22.8 ± 1
ResNet-18	19.0 ± 1	11.2 ± 2	26.3 ± 1	9.9 ± 2	**19.0 ± 0**	**24.6 ± 4**
DenseNet-201	**20.8 ± 1**	11.6 ± 3	31.1 ± 1	11.0 ± 2	16.0 ± 1	23.2 ± 3

5. Discussion

5.1. Class Aware vs. Pairwise Loss

The observations made from our results are that the pairwise loss function performs better than the class aware loss function. However, the difference in performance between the loss functions is marginal. This trend is observed across all the datasets and neural networks we investigated. To illustrate this point, DenseNet-201 trained on triplet loss achieved the best performance of 79.7% on the Chimpanzees dataset, while DenseNet-201 trained on Proxy-NCA on the Chimpanzees dataset obtained 78.2%. The second trend is that the best performance is not always achieved by the same neural network architecture and loss function across all datasets. For the Tiger and Panda datasets, VGG-11 trained on triplet loss achieved 88.9% and 91.2%, respectively. In the Lion dataset, VGG-19 Proxy-NCA obtained the best performance of 71.3%. Musgrave et al. [13] who studied three datasets Cars-196, CUB-200 and Stanford online products made the same observations. While we cannot directly compare our work with the results presented by Musgrave et al. [13], we also found that it is not always that Proxy-NCA will outperform the pairwise loss function.

A study carried out by Chen et al. [64] on the Panda dataset obtained Recall@1 of 92.1% on unseen classes. However, a major preprocessing step involved hand annotation locating the Panda face in the images, so that background and occlusions are removed from each image before using the image for training. While the proposed algorithm yielded good results, we demonstrate that VGG-11, trained on triplet loss function, compares well with the method proposed by Chen et al. [64]. VGG-11 triplet loss obtained 91.2 ± 1% on raw Panda images depicted in Figure 3, without a need to apply extensive preprocessing steps. We observe that removing background may create challenges when identifying Pandas in their natural habitat. On the Tiger dataset, Schneider et al. [34] found that DenseNet-201 triplet loss achieved a Recall@1 of 86.3%, we replicated these results, our DenseNet-201 obtained 85.0% ± 1; however, we found that VGG-11 trained on triplet loss performs better at 88.9%. A sample of retrieved images from animal flanks is shown in Figure 4.

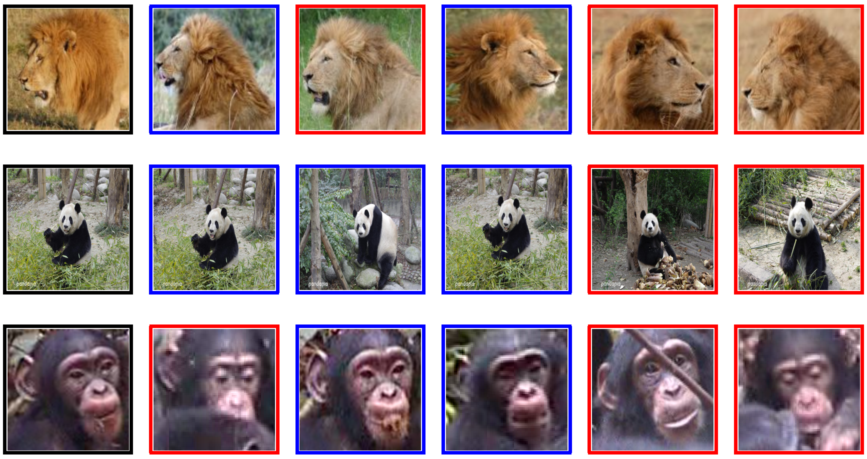

Figure 3. Faces: The first image is the query, with five nearest neighbours: blue border is true positive and red border is false positive image.

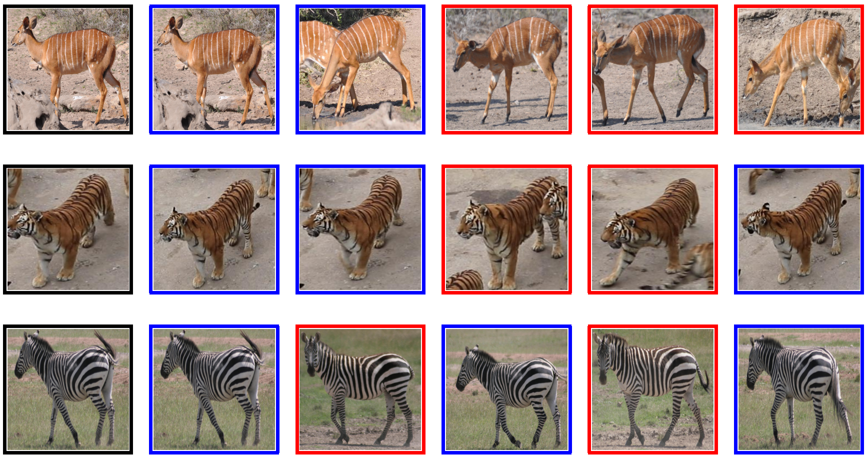

Figure 4. Flanks: The first image is the query, with five nearest neighbours: blue border is true positive and red border is false positive image.

5.2. Backbone Architectures

Investigating the best performing backbone architectures was not the primary objective of the research. Still, it is interesting to note from Table 4 that if one were to select a single backbone, then VGG-11 with Triplet loss would give one the most consistently robust results followed by DenseNet-201 also with triplet loss.

5.3. Flanks vs. Faces

We test out various architectures in combination with different loss functions and different embedding sizes to see if some generalised lessons exist for animal faces versus flanks. We find no obvious commonality, which leads us to hypothesise that perhaps pre-training on ImageNet was sufficient. To this end, we explored fine-tuning the models using datasets on flanks for flanks and faces for faces but found no benefit for transfer learning. Further, we find that using more advanced/complex or suited towards faces Neural Network architectures leads to no statistically significant improvements.

6. Conclusions

Recent works in metric learning have focused on loss function design, intending to substitute the pairwise loss functions. Comparing these loss functions across datasets in individual re-identification of wild animals; Lions, Chimpanzees, Zebra, and Nyala, we find that the performance gains are marginal. More often, the triplet loss models produced better results.

Our work shows the proposed improvements in the design of deep neural networks for similarity learning experiments. From how training samples are selected to how loss functions are optimised, there cannot be one solution that is best suited for all datasets. Each dataset presents a unique challenge to the model, and hence, there is a need to look for the best possible design for each dataset continuously. The models trained on the triplet loss function achieved better Recall@1 across datasets. It is worth pointing out that the Proxy-NCA loss removes the need to search for an optimal sampling technique. This removal cannot be overlooked as it simplifies experimental design. In the future, it may be necessary to investigate a combination of model parameters and a specific loss function to improve models' information retrieval capabilities.

Author Contributions: Conceptualization, N.D. and T.L.v.Z.; methodology section was written by N.D.; code and experiments were done by N.D. writing—original draft preparation, N.D.; writing—review and editing, T.L.v.Z.; supervision, T.L.v.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Zebra data publicly available at: https://code.google.com/archive/search?q=stripes%20dataset (accessed on 12 November 2019); Chimpanzees data publicly available at http://www.inf-cv.uni-jena.de/chimpanzee_faces.html (accessed on 12 November 2019); Tiger data publicly available at https://cvwc2019.github.io/challenge.html (accessed on 11 June 2021); Panda data available on request from https://researchdata.ntu.edu.sg/ (accessed on 16 May 2021); Lion data publicly available at https://github.com/tvanzyl/wildlife_reidentification (accessed on 29 October 2019); Nyala data publicly available at https://github.com/tvanzyl/wildlife_reidentification (accessed on 24 October 2019).

Acknowledgments: We would like to thank the Nedbank Research Chair for the support provided to us while we undertook this research. We also extend our appreciation to Sara Blackburn for allowing us to collect and pre-process the Lion dataset from the living with lions website.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DCNN	Deep Convolutional Neural Network
P-NCA	Proxy Neighbourhood Component Analysis
Proxy-NCA	Proxy Neighbourhood Component Analysis
MAP	Mean Average Precision

References

1. Borchers, D.L.; Zucchini, W.; Fewster, R.M. Mark-recapture models for line transect surveys. *Biometrics* **1998**, *54*, 1207–1220. [CrossRef]
2. Ariff, M.; Ismail, I. Livestock information system using Android Smartphone. In Proceedings of the 2013 IEEE Conference on Systems, Process & Control (ICSPC), Kuala Lumpur, Malaysia, 13–15 December 2013; pp. 154–158.
3. Schacter, C.R.; Jones, I.L. Effects of geolocation tracking devices on behavior, reproductive success, and return rate of Aethia auklets: An evaluation of tag mass guidelines. *Wilson J. Ornithol.* **2017**, *129*, 459–468. [CrossRef]
4. Wright, D.W.; Stien, L.H.; Dempster, T.; Oppedal, F. Differential effects of internal tagging depending on depth treatment in Atlantic salmon: A cautionary tale for aquatic animal tag use. *Curr. Zool.* **2019**, *65*, 665–673. [CrossRef] [PubMed]
5. Awad, A.I. From classical methods to animal biometrics: A review on cattle identification and tracking. *Comput. Electron. Agric.* **2016**, *123*, 423–435. [CrossRef]
6. Nguyen, H.; Maclagan, S.J.; Nguyen, T.D.; Nguyen, T.; Flemons, P.; Andrews, K.; Ritchie, E.G.; Phung, D. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 40–49
7. Chen, G.; Han, T.X.; He, Z.; Kays, R.; Forrester, T. Deep convolutional neural network based species recognition for wild animal monitoring. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 858–862
8. van Zyl, T.L.; Woolway, M.; Engelbrecht, B. Unique Animal Identification using Deep Transfer Learning For Data Fusion in Siamese Networks. In Proceedings of the 2020 23rd International Conference on Information Fusion (FUSION 2020), Rustenburg, South Africa, 6–9 July 2020
9. Verma, G.K.; Gupta, P. Wild animal detection using deep convolutional neural network. In *Proceedings of 2nd International Conference on Computer Vision & Image Processing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 327–338.
10. Burns, J.; van Zyl, T.L. Automated Music Recommendations Using Similarity Learning. In Proceedings of the SACAIR 2020, Muldersdrift, Africa, 22–26 February 2021; p. 288.
11. Manack, H.; Van Zyl, T.L. Deep Similarity Learning for Soccer Team Ranking. In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 6–9 July 2020; pp. 1–7
12. Kaya, M.; Bilge, H.Ş. Deep metric learning: A survey. *Symmetry* **2019**, *11*, 1066. [CrossRef]
13. Musgrave, K.; Belongie, S.; Lim, S.N. A metric learning reality check. In *European Conference on Computer Vision*; Springer: Glasgow, UK, 2020; pp. 681–699.
14. Jain, A.K.; Flynn, P.; Ross, A.A. *Handbook of Biometrics*; Springer: Berlin/Heidelberg, Germany, 2007.
15. Korschens, M.; Denzler, J. Elpephants: A fine-grained dataset for elephant re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
16. Kühl, H.S.; Burghardt, T. Animal biometrics: Quantifying and detecting phenotypic appearance. *Trends Ecol. Evol.* **2013**, *28*, 432–441. [CrossRef] [PubMed]
17. Clarke, R. Human identification in information systems. *Inf. Technol. People* **1994**, *7*, 6–37 [CrossRef]
18. Rowcliffe, J.M.; Field, J.; Turvey, S.T.; Carbone, C. Estimating animal density using camera traps without the need for individual recognition. *J. Appl. Ecol.* **2008**, 1228–1236. [CrossRef]
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
20. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
21. Li, J.; Lin, D.; Wang, Y.; Xu, G.; Zhang, Y.; Ding, C.; Zhou, Y. Deep discriminative representation learning with attention map for scene classification. *Remote Sens.* **2020**, *12*, 1366. [CrossRef]
22. Sun, Y.; Chen, Y.; Wang, X.; Tang, X. Deep learning face representation by joint identification-verification. In Proceedings of the 27 International Conference on Neural Information Processing Systems, Montereal, QC, Canada, 8–13 December 2014; pp. 1988–1996.
23. Meyer, B.J.; Drummond, T. The importance of metric learning for robotic vision: Open set recognition and active learning. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2924–2931.

24. Huo, J.; van Zyl, T.L. Comparative Analysis of Catastrophic Forgetting in Metric Learning. In Proceedings of the 2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI), Stockholm, Sweden, 14–15 November 2020; pp. 68–72.
25. Weinberger, K.Q.; Saul, L.K. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
26. El-Naqa, I.; Yang, Y.; Galatsanos, N.P.; Nishikawa, R.M.; Wernick, M.N. A similarity learning approach to content-based image retrieval: Application to digital mammography. *IEEE Trans. Med. Imaging* **2004**, *23*, 1233–1244. [CrossRef] [PubMed]
27. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "siamese" time delay neural network. In Proceedings of the 6th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November–2 December 1994; pp. 737–744.
28. Dlamini, N.; van Zyl, T.L. Author Identification from Handwritten Characters using Siamese CNN. In Proceedings of the 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 21–22 November 2019; pp. 1–6.
29. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
30. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 539–546.
31. Yu, B.; Liu, T.; Gong, M.; Ding, C.; Tao, D. Correcting the triplet selection bias for triplet loss. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 71–87.
32. Cui, Z.; Charoenphakdee, N.; Sato, I.; Sugiyama, M. Classification from Triplet Comparison Data. *Neural Comput.* **2020**, *32*, 659–681. [CrossRef] [PubMed]
33. Xuan, H.; Stylianou, A.; Pless, R. Improved embeddings with easy positive triplet mining. In Proceedings of the The IEEE Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 2474–2482.
34. Schneider, S.; Taylor, G.W.; Kremer, S.C. Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops, Snowmass, CO, USA, 1–5 March 2020; pp. 44–52.
35. Roth, K.; Milbich, T.; Sinha, S.; Gupta, P.; Ommer, B.; Cohen, J.P. Revisiting training strategies and generalization performance in deep metric learning. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 13–18 July 2020; pp. 8242–8252.
36. Deng, J.; Guo, J.; Liu, T.; Gong, M.; Zafeiriou, S. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*; Springer: Glasgow, UK, 2020; pp. 741–757.
37. Kim, J.H.; Kim, B.G.; Roy, P.P.; Jeong, D.M. Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. *IEEE Access* **2019**, *7*, 41273–41285. [CrossRef]
38. Wang, X.; Han, X.; Huang, W.; Dong, D.; Scott, M.R. Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5022–5030.
39. Schneider, S.; Taylor, G.W.; Linquist, S.; Kremer, S.C. Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods Ecol. Evol.* **2019**, *10*, 461–470. [CrossRef]
40. Qiao, Y.; Su, D.; Kong, H.; Sukkarieh, S.; Lomax, S.; Clark, C. Individual cattle identification using a deep learning based framework. *IFAC-PapersOnLine* **2019**, *52*, 318–323. [CrossRef]
41. Hou, J.; He, Y.; Yang, H.; Connor, T.; Gao, J.; Wang, Y.; Zeng, Y.; Zhang, J.; Huang, J.; Zheng, B.; et al. Identification of animal individuals using deep learning: A case study of giant panda. *Biol. Conserv.* **2020**, *242*, 108414. [CrossRef]
42. Nepovinnykh, E.; Eerola, T.; Kalviainen, H. Siamese network based pelage pattern matching for ringed seal re-identification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, Snowmass, CO, USA, 1–5 March 2020; pp. 25–34.
43. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
44. Burghardt, T.; Calic, J.; Thomas, B.T. *Tracking Animals in Wildlife Videos Using Face Detection*; EWIMT: London, UK, 2004.
45. Henschel, P.; Coad, L.; Burton, C.; Chataigner, B.; Dunn, A.; MacDonald, D.; Saidu, Y.; Hunter, L.T. The lion in West Africa is critically endangered. *PLoS ONE* **2014**, *9*, e83500. [CrossRef] [PubMed]
46. Norouzzadeh, M.S.; Nguyen, A.; Kosmala, M.; Swanson, A.; Palmer, M.S.; Packer, C.; Clune, J. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, E5716–E5725. [CrossRef] [PubMed]
47. Teh, E.W.; DeVries, T.; Taylor, G.W. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*; Springer: Glasgow, UK, 2020.
48. Wang, T.; Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Proceedings of the International Conference on Machine Learning, PMLR, Cambridge, MA, USA, 13–18 July 2020; pp. 9929–9939.

49. Chen, T.; Li, L. Intriguing Properties of Contrastive Losses. *arXiv* **2020**, arXiv:2011.02803.
50. Rippel, O.; Paluri, M.; Dollar, P.; Bourdev, L. Metric learning with adaptive density discrimination. *arXiv* **2015**, arXiv:1511.05939.
51. Movshovitz-Attias, Y.; Toshev, A.; Leung, T.K.; Ioffe, S.; Singh, S. No fuss distance metric learning using proxies. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 360–368.
52. Goldberger, J.; Hinton, G.E.; Roweis, S.; Salakhutdinov, R.R. Neighbourhood components analysis. *Adv. Neural Inf. Process. Syst.* **2004**, *17*, 513–520.
53. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, NSW, Australia, 2–8 December 2013.
54. Yang, L.; Luo, P.; Change Loy, C.; Tang, X. A large-scale car dataset for fine-grained categorization and verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3973–3981.
55. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*; Technical Report CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.
56. Freytag, A.; Rodner, E.; Simon, M.; Loos, A.; Kühl, H.S.; Denzler, J. Chimpanzee faces in the wild: Log-euclidean CNNs for predicting identities and attributes of primates. In *German Conference on Pattern Recognition*; Springer: Hannover, Germany, 2016; pp. 51–63.
57. Lahiri, M.; Tantipathananandh, C.; Warungu, R.; Rubenstein, D.I.; Berger-Wolf, T.Y. Biometric animal databases from field photographs: Identification of individual zebra in the wild. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, Trento, Italy, 18–20 April 2011; pp. 1–8.
58. Matkowski, W.M.; Kong, A.W.K.; Su, H.; Chen, P.; Hou, R.; Zhang, Z. Giant panda face recognition using small dataset. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1680–1684.
59. Li, S.; Li, J.; Tang, H.; Qian, R.; Lin, W. ATRW: A benchmark for Amur tiger re-identification in the wild. *arXiv* **2019**, arXiv:1906.05586.
60. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
61. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
62. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
63. Musgrave, K.; Belongie, S.; Lim, S.N. PyTorch Metric Learning. *arXiv* **2020**, arXiv:2008.09164.
64. Chen, P.; Swarup, P.; Matkowski, W.M.; Kong, A.W.K.; Han, S.; Zhang, Z.; Rong, H. A study on giant panda recognition based on images of a large proportion of captive pandas. *Ecol. Evol.* **2020**, *10*, 3561–3573. [CrossRef] [PubMed]

Short Biography of Authors

Nkosikhona Dlamini is a student at the University of Witwatersrand studying towards a Master of Science in Computer science. Completed honours degree in Computer science at the University of Pretoria. Currently employed at the Tshwane university of Technology. He spent four years working at the CSIR as a research and development technologist focusing on NLP for South African languages.

Terence L. van Zyl holds the Nedbank Research and Innovation Chair at the University of Johannesburg where he is a Professor in the Institute for Intelligent Systems. He is an NRF rated scientist who received his PhD and MSc in Computer Science from the University of Johannesburg for his thesis on agent-based complex adaptive systems. He has over 15 years of experience researching and innovating large scale streaming analytics systems for government and industry. His research interests include data-driven science and engineering, prescriptive analytics, machine learning, meta-heuristic optimisation, complex adaptive systems, high-performance computing, and artificial intelligence.