THESIS

Automated Individual Identification of Wildlife using Deep Neural Networks

Author:
Nkosikhona DLAMINI
(1938108)

Supervisor:
Prof. Terence VAN ZYL

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

submitted to
the Faculty of Science, in fulfilment of the requirements for the degree
of
Msc Computer Science Big Data Analytics

in the

Wits Institute of Data Science (WIDS)
School of Computer Science and Applied Mathematics

June 13, 2022

Declaration of Authorship

I, Nkosikhona DLAMINI (1938108), declare that this thesis titled, "Automated Individual Identification of Wildlife using Deep Neural Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 06/13/2022

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Abstract

Faculty of Science
School of Computer Science and Applied Mathematics

Msc Computer Science Big Data Analytics

Automated Individual Identification of Wildlife using Deep Neural Networks

by Nkosikhona DLAMINI (1938108)

Automated re-identification of individuals in endangered species has gained traction in nature conservation initiatives. Scholars in computer vision community have explored the use of the deep convolutional networks in the re-identification and classification of chimpanzees and gorillas from image signals. However, no work reports on the re-identification of lions and cheetahs. We provide an implementation of deep neural networks for individual re-identification of the big cats, lions, and envisage replicating the success obtained in classification and re-identification tasks reported in chimpanzee and human beings using face images. We present a comparison of the performance obtained from different deep neural network architectures on individual lion re-identification using face image features, and also search for the best loss function between pair-based loss function and class aware loss functions. This endeavor is aimed at assisting conservation initiatives to monitor, report and manage biodiversity on the population of lions with minimal manual labor.

Keywords-Deep convolutional neural networks, Triplet loss,Proxy-NCA,Individual animal identification,Pair-based loss functions

Acknowledgements

I like to thank my supervisor, Professor Terence van Zyl for the guidance and support he provided throughout this research. I have been lucky to have a supervisor who will spend time reading my write up and give clear insights and guiding me on how to compile the different sections of this work.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
SCNN	**S**iamese **C**onvolutional **N**eural **N**etwork
AP	**A**verage **P**recision
MAP	**M**ean **A**verage **P**recision
HOG	**H**istogram **O**riented **G**radients
LBP	**L**ocal **B**inary **P**atterns
RFID	**R**adio **f**requency **i**dentification

List of Publications

- Automated Identification of Individuals in Wildlife Population Using Siamese Neural Networks **IEEE Article 2020** [1]
- Comparing Class-Aware and Pairwise Loss Functions for Deep Metric Learning in Wildlife Re-Identification **MDPI Journal 2021** [2]

Chapter 1

Introduction

The identification of an individual animal in a population has attracted investments from custodians of animals. Rearing animals is for commercial, personal, and conservation purposes. The forms of identification employed in individual animals range from easily accessible to more sophisticated methods depending on the purpose of rearing the animals. The easily accessible forms of identification include numbering on plastic ear tags, collars, and tattoos on an animal's skin. The sophisticated forms are electronic identification methods and bio-metrics-based identification [3]. Electronic re-identification methods involve attaching a device to the animal's body or implanting the device under the animal's skin.

Focus has turned towards using biometric methods, where animals are re-identified by observing their unique biological features like coat patterns, gait, and hoof prints [4]. Human beings do these observations. However, for larger animal populations, human beings' observations can be error-prone and labor-intensive. In the 1990s Turk and Pentland [5] investigated incorporating computer algorithms that perform image analysis to extract unique features from animal face images, animal coat patterns, and animal iris images, thus removing the reliance on human beings to perform animal re-identification. The intersection of biometrics and image analysis to extract features that uniquely identify objects from images created a branch in computer vision that focuses on investigating newer and robust automated ways to extract biometric features from animal images. The computer vision algorithms for image analysis have evolved from using hand-engineered image features to automatically learned features [6]. The hand-engineered image features are the histogram of gradients (HOG) and local binary patterns (LBP). Wang *et al.* [7] demonstrated that histogram of oriented gradients (HOG) image features are suitable for estimating the shape of an object, and local binary patterns (LBP) are good features for texture analysis. Wang *et al.* [7] proposed a combination of these features in detecting human faces,

leveraging on the strength of each feature. Hand-engineered features imply that a labor-intensive image pre-processing step is required before training an automated image-identifying model. Researchers like Weinstein [6] show that the move towards automatically learned features resulted in faster and improved re-identification from image analysis. The algorithms that learn features automatically from images are called convolutional neural networks [8].

Schneider *et al.* [9] demonstrates the use of different convolutional neural networks for re-identifying individuals in endangered species; chimpanzees and gorillas. The work by Schneider *et al.* [9] shows that record keeping in chimpanzees is automated. Van Zyl *et al.* [10] worked on re-identification of individuals in nyala and zebra population using a pair of identical ResNet-152 convolutional neural networks; convolutional neural networks, while Li *et al.* [11] used three identical ResNet-50 neural networks for individual tiger re-identification. Extinction threatens other species like lions and cheetahs. However, the literature does not show an investigation directed towards automating individual re-identification in lions and cheetahs. Therefore, the record-keeping of these endangered species is still done manually and is prone to human error.

The current work aimed to investigate and implement automated re-identification of individuals in the lion population using deep neural networks. The breakthrough in automatic re-identifying individuals in animals like tigers, nyala, and zebra ignites interest to extend this to other species like lions. Additionally, searching for the best-suited neural network architecture in a specific dataset was particularly interesting. We observed that researchers chose different convolutional neural networks, and different configurations of these neural networks for experiments, there is no indication of what informs such choices: Van Zyl *et al.* [10] used two identical ResNet-152 (Siamese convolutional neural networks) for the re-identification of zebra and nyala, while Li *et al.* [11] used three identical ResNet-50 (triplet-loss convolutional neural networks) for the re-identification of individual tiger. Other neural networks and loss functions can be used to re-identify individual animals like the VGG [12] DenseNet [13], and class-aware loss functions.

An experimental research methodology was followed to implement automatic re-identification of individuals in a lion population. Our experiments were aimed to determine how deep convolutional neural networks can be used to automate individual re-identification from lion face images. Experiments in deep convolutional neural networks follow one of these approaches:

classification or similarity learning. Both approaches come with a choice of a loss function to minimize. We had two sets of experiments designed to answer the following questions:

- What performance is obtained by different deep convolutional neural network architectures: VGG, ResNet, and DenseNet in unique animal re-identification?
- How do class aware, and pair-based loss functions compare in unique animals re-identification?

In the first experiments, we searched for the best performing deep convolutional neural network architecture. In the second set of experiments, we searched for the loss function that resulted in better results. We considered the lion data and previously studied data sets like panda, zebra, nyala, and chimpanzee. The previously studied data sets formed a benchmark for our experiments. For all our experiments, we measured the model performance using the Recall@1 and mean average precision at R MAP@R.

For the lion experiments, the VGG-19 neural network architecture trained on the class aware loss function (Proxy-NCA) obtained better performance than other neural network architectures; ResNet-18, ResNet-152, and VGG-11 trained on the same loss function. VGG-19 obtained 71.3% ± 3 Recall@1, while the DenseNet-201 architecture trained on triplet loss obtained 70.1% ± 1 on the lion data set. In the chimpanzees' data set, we found that the DenseNet-201 neural network trained on triplet loss gave a better performance of 79.7% ± 2. In the panda and Tiger datasets, VGG-11 trained on triplet loss obtained the best Recall@1 of 91.2% ± 1,and 88.9% ± 1 respectively. The results show that the class-aware loss function does not always produce better results than the pair-based loss function. While VGG-11 appeared to perform better in most of the data sets studied in the current work, no single neural network architecture outperforms the other architectures in all data sets.

The contributions of the current work are as follows:

- The VGG-19 convolutional neural network architecture trained on class-aware Proxy-NCA is best suited for re-identifying individuals in the lion population.
- It is not always that class-aware loss functions (Proxy-NCA) outperform the pair-based triplet loss function. Other parameters may be contributing to the best performance.

- The current work sets new best Recall@1 in three of the previously studied datasets, namely tiger, zebra, and chimpanzees.

The rest of this thesis is organized as follows: **Chapter 2** presents literature that gives a detailed synthesis of previous works. Next, **Chapter 3** outlines the experimental setup and describes the datasets used in the experiments. Then, **Chapter 4** presents the results we obtained in our experiments alongside results we found in the literature. Finally, **Chapter 5** discusses the results and our interpretation, and **Chapter 6** concludes the thesis by highlighting objectives and briefly discussing the findings and conclusions drawn from the findings.

Chapter 2

Literature Review

2.1 Introduction

The current work intersects two fields; animal biometrics and computer vision. The first part of this chapter discusses the background of these fields. The second part elaborates on why the re-identification of individuals in endangered animals using computer vision is needed. The last two sections are about the approaches employed in computer vision for the re-identification of individuals and how the success of these approaches is measured.

2.2 Biometrics and Computer Vision

Zhang [14] defined biometrics as a field of study concerned with using unique characteristics that naturally discriminate an individual among others. These characteristics can be anatomy features, behavioral features, or chemical features. In the early 1990s, work on biometrics re-identification was already underway. Researchers like Clarke [15] already noted that the biometric features used for individuals in a population need to satisfy the following criteria:

- **Universality**; every individual should have the feature.
- **Uniqueness**; two or more individuals should have a different form of the same feature.
- **Permanence**; the feature should not change over time.
- **Collectability**; it must be easy to capture and measure the feature.
- **Acceptability**; the community of researchers in the field should agree that such a feature can be collected and measured without causing discomfort to the individuals.

Bolle *et al.* [16] supports the criteria for biometrics features but argues that it is difficult for a biometric feature to meet these criteria. As a result, no single feature is considered to be the best for individual re-identification. This observation validates why biometric re-identification is still an active research field. Researchers continuously build models and compare model performances when different biometric features are used for individual re-identification [17].

The other reasons biometrics is an active research field to date are the different challenges researchers seek to solve. For example, some works are looking at ways to increase biometrics re-identification systems' accuracy and efficiency. Improved accuracy scores are desired to ensure that the biometric re-identification systems produce few wrong re-identifications. On the other hand, better efficiency means that the system captures, extracts information from the biometric feature, and does re-identification faster to ensure practical usage is achieved [18].

Other researchers looked at how biometrics-based systems can be secured such that these systems are not susceptible to malicious attacks. These investigations are aimed at ensuring that deployed systems are robust to imposters [19].

The current work seeks to find out if lion face features can be used as a biometric feature that discriminates against individuals in the lion population. Attention will also be given to investigating ways that improve the accuracy of re-identification. In addition to re-identification of individual lions, we searched for the most suitable loss function by comparing types of loss functions used in computer vision experiments: class aware and pair-based loss function. Publicly available datasets will be used so that our results are compared with prior research outputs on these datasets.

2.2.1 Reasons for Automated Identification of Individuals

The broader reason why automated re-identification of individuals in human beings was the need to grant access to the right person. Hathaliya *et al.* [20] observed that systems in healthcare facilities are rapidly being deployed online, and the privacy of patients' records is essential. The use of biometric features to grant or deny access to patients' records was found to be faster.

Another reason why biometric re-identification is needed in human beings is for forensic investigations, where biometric features can reveal who was at a crime scene [21].

The reasons behind the use of biometric systems in re-identifying individual animals are slightly different from the reasons observed in human re-identification. Shanahan *et al.* [22] reflected on the importance of tracing beef products from the breeding farms up to the shelf where consumers buy the products. Tracking the food value chain ensures that the quality of food products can be known from production up to consumption. Also, if there are recalls, the appropriate steps are taken back from retail shelves to the farm where the individual animal was raised [23].

The aim of tracing individuals in wild animals is to keep records of an individual's whereabouts. This supports conservation initiatives where it is necessary to know and protect individuals, especially endangered species. If an individual disappears, the wildlife custodians may need to reliably investigate if the individual went astray, joined other populations, or is lost through poaching. Buk *et al.* [24] collects and reports on data about the cheetah population in South Africa and demonstrates that there are no details about an individual cheetah. More focus is on counting how many cheetahs exist in South Africa, and nothing is known about each individual in the cheetah population. The same is observed with lions.

Keeping track and conserving wild animals supports the tourism industry. In what Dou and Day [25] termed human-wildlife interaction tourism, people travel to wildlife destinations to see, touch and feed wild animals. Governments' conservation agencies charge a fee for tourists to participate in these activities, thus creating revenue streams that support economies. Preserving wild animals, therefore, becomes critical in supporting the tourism industry. Mancini [26] pointed out that wildlife tourism is on the rise, generates one out of 11 jobs in the world, and contributes 10% of gross domestic product (GDP). Therefore enhanced efforts aimed at preserving wildlife cannot be ignored.

Poaching and wildlife trafficking has created a constant danger to wild animals [27]. Governments across the world have invested in ways of curbing animal trafficking. While the sale of animals is permitted, there are some acts of criminality where endangered animals are sold on the black market. McMillan *et al.* [28] noted that there could be no guarantees that the trafficked animals' well-being will be taken care of by the illicit dealers. Poor animal care would result in the loss of these animals. Automated ways of tracking and re-identifying wild animals can contribute toward early detection and prevention of trafficking, thus preserving endangered species [29].

The inherent challenge of dealing with wild animals is that they live in

vast areas that may not be accessible to human beings. It is, therefore, a difficult task to manually count, track, and detect missing individuals in the wild. Another challenge is that encounters between humans and wild animals can be dangerous. Counting and tracking lions may not be safe. There have been several ways proposed to track wild animals. The subsection below discusses the proposed ways of tracking wild animals.

2.2.2 Methods Used in Identifying and Tracking Animals

There are three major categories of ways in which individuals in wild animals have been identified, tracked, and counted. These are:

- numbering systems using tags or tattoos [30],
- attaching electronic devices like the radio frequency identification devices (RFID) on animal body parts or embedding RFID in the animal skin [31],
- biometric features extracted from images [32].

RFID electronic devices have primarily replaced tattooing and numbering systems. Awad [32] claimed that tattoos and numbering systems are cheaper than RFID devices, but these can be removed, changed, or destroyed easily. The true identity of an individual may be compromised. Therefore the individual may not be tracked reliably. When dealing with wild animals, different challenges exist. Capturing a lion requires carefully planned interventions because such an encounter can be fatal. The temporal capture of an individual wild animal can cause distress to the individual and may need special care after the capture to relieve the distress [33]. Further, the number system for re-identification means the custodians of animals have to be in close contact with the individual to read the identity. For this reason, applying a numbering system in wild animal re-identification is not practical.

The RFID tags were introduced to limit the contact between humans and animals. The only time there is close contact is during tagging or implanting the device. After implanting the device, the individual's identity can be read several meters away from the animal [34]. Even though RFID technology brings about better and more reliable tracking means, it has challenges.

Gillenson *et al.* [35] presented work on problems associated with the use of RFIDs in animal re-identification. The first problem is that RFIDs are expensive. Deploying RFIDs to a large number of individual wild animals can be an expensive exercise. As a result, the benefits of this technology cannot be

realized. The high costs of the RFID systems are a result of the fact that these systems come with three components: the tag, tag readers, and computers that will display and interpret the tag reader data [36]. Additionally, there are operational costs that are incurred when implanting the tags on the animal and installation of tag readers in the animals' environment. Wildlife custodians would need to invest substantially to roll out RFID systems; otherwise, a search for more affordable ways of individual animal re-identification is inevitable.

Besides the cost associated with RFID systems, there are also health challenges imposed on the animals that have RFID devices implanted under the skin. These can cause an allergic reaction in the animal. Deploying these devices to wild animals that roam vast and inaccessible terrain may be undesirable. Close monitoring may be required after implantation to monitor the animal for adverse allergic reactions, and this further increases the overhead cost associated with RFIDs. There is a need to improve current RFIDs to reduce these effects or look for other ways of identification. For animals that live in the wild, if RFID implantation causes adverse effects and if the host animal is not monitored closely may increase death rates and not assist in the conservation of these species [37].

The emerging methods for individual animal identification seek to address the costs and health issues that are associated with RFID identification. Biometric features are used in emerging identification techniques. The use of biometrics in unique individual identification is not a new phenomenon. It has been used extensively in human beings' identification [38]. However, the use of biometric technologies in individual animal identification is a relatively new research field. The subsection below depicts the trends in animal biometric research papers in google scholar for three decades: 1990-2000, 2001 -2010, and 2011-2020.

Wildlife Identification Trends Decade: 1990-2000

Literature between 1990 and 2000 shows that scholars working on animal identification were using a technique referred to as capture-recapture or capture-re-sighting [39], [40]. The individuals were captured, marked, and released into the wild. Then, the animals were re-captured weekly for over a year to verify that those captured in the first experiment were still in the population [41]. The oversimplified metrics computed by these researchers are return rate, calculated to determine how many individuals returned versus

how many were captured in the first experiment, and annual survival rate, calculated to determine how many individuals survived in a year.

The capture and re-capture method has limited application in wild animal re-identification. Capturing a lion is not only an expensive task but also exposes researchers to undesired dangers. In the period 2001-2010, there is an observed transition to safer ways of wild animal re-identification using biometric features extracted from images.

Wildlife Identification Trends Decade: 2001-2010

In the era 2001-2010, images of animals were analyzed to get unique biometric features that discriminate individuals in wildlife populations. Capturing and analyzing images is safer than actual capturing of the animals for purposes of re-identification. Burghardt and Campbell [42] worked on automated re-identification of an individual penguin from panda image dataset. Classification models were trained on features generated from penguin chest spots. Barron *et al.* [43] used images from the retina of sheep to automatically identify individual. While Barron *et al.* [43] achieved 99% accuracy, using retina images for wild animals may not be feasible. The use of retina images may be attractive given the high accuracy achieved by Barron *et al.* [43]. However, collecting retina images from other wild animals like lions need close contact between human beings and animals. This may be an undesired dangerous encounter.

Emerging Image-based Biometric Datasets: Trends in Decade: 2011-2020

The research work in the decade 2011-2020 embraced the fact that image analysis presents a better way of re-identification, tracking, and counting individuals in a wildlife population. This is drawn from the observation that there is more work done to automatically collect images via deploying camera traps in the wild animals' habitat. Table 2.1 lists some wild mammals image datasets collected.

This list reflects the urgency needed for collecting image datasets of other endangered species. Only eight datasets are collected for individual re-identification in the species population. The current work will add the lion dataset to this list and present model performance achieved by different neural network architectures on individual lion re-identification.

TABLE 2.1: Wild animals image datasets

Datasets	
Dataset Name	Individuals/Species
Serengeti dataset [44]	Species
iNaturalist [45]	Species
Caltech Camera Traps [46]	Species
iWildCam [47]	Species
Lemur and Golden Monkey[48]	Individuals
Amur Tiger [11]	Individuals
Zebra dataset[49]	Individuals
Humpback Whale[50]	Individuals
Chimpanzee faces in the wild [51]	Individuals
Elephants dataset [52]	Individuals
Panda [53]	Individuals
Nyala [2]	Individuals

2.2.3 Challenges with Image-based Identification

The datasets above can be separated into two major categories. One category is about individuals, and the other category is about species. The individuals' dataset is geared towards identifying individuals in the population, and the species dataset is focused on identifying species in the animal kingdom. There are eight individual datasets: zebra, elephant, tiger, nyala, humpback whale, lemur, golden monkey, and chimpanzees.

The image dataset, in part, solved the problems of invasive and risky re-identification methods like RFID systems. However, new challenges emerged. The amount of data grew faster due to image collection initiatives like the Serengeti project. Over a million images were collected in Serengeti project [44]. When training supervised machine learning algorithms for animal re-identification, the data points in the dataset need to be labeled. Labeling the species dataset may need limited expert knowledge. Figure 2.1 depicts features that distinguish giraffes from lions and are easy to identify. However, when dealing with individual animal datasets, where the individuals may be very similar, then more fine-grained identifying features need to be considered during labeling, and this may require expert knowledge. Figure 2.2 depicts two giraffes, and distinguishing a giraffe from another

giraffe may not be an easy task [52]. Körschens and Denzler [52] also observed that the image capture in wild animals is not controlled; as a result, different views of an individual can be captured under differing lighting conditions. In other images, the individual may be occluded by tree branches' or shades. Models built for purposes of animal re-identification from images collected in this fashion should be robust to background noise. The last problem with automated image data collection is that class imbalances are a common phenomenon. Table 2.2 shows the class imbalances in individuals' datasets. Johnson and Khoshgoftaar [54] argues that models tend to be biased towards the class with many data points; this bias is undesirable because models should be robust in the re-identification of minority classes as well. Johnson and Khoshgoftaar [54] gave reasons why the models tend to be biased towards the majority class: one of the reasons is that the model weights will be updated mainly by the features generated from the majority class. The current work employed transfer-learning to address the problem of class imbalance. In transfer-learning, the model weights are updated by a larger dataset; only the last layer of the models we trained was fine-tuned with a specific dataset, thus removing the bias towards the majority class in model layers that extract features.

TABLE 2.2: Individual mammal wildlife datasets

Individuals Datasets				
Species	Total data	Individuals	Min-max Individual	Average
Amur Tiger	8076	96	[> 1, ≈10]	-
Chimpanzee	5559	90	[3, 315]	63
Elephants	2078	276	[1, 22]	10
Giraffe	1200	600	[1,≈3]	2
Golden-Monkeys	1450	49	[2, 120]	30
Lemur	3000	129	[7, 42]	23
Nyala	1945	474	[1, 16]	6
Pandas	6441	218	[1, 70]	30
Whale	4542	427	[1, 45]	20
Zebra	820	84	[1, 29]	12

(A) Lion

(B) Giraffe

FIGURE 2.1: Lion vs Giraffe

(A) Giraffe *a*

(B) Giraffe *b*

FIGURE 2.2: Giraffe vs Giraffe

2.2.4 The Search for Solutions to Image-based Identification Challenges

The rise of wild animal image datasets and the inherent challenges presented by such datasets has ignited wide research interest. Researchers have worked on the various components of machine learning for animal biometrics to find solutions. The components are feature engineering, training models for classification tasks, training similarity learning methods, and the model evaluation criteria. These components are discussed in detail in section 2.3.

2.3 Components of Machine Learning Models in Computer Vision

2.3.1 Feature Extraction

The existence of large image datasets sparked investigations in feature extraction algorithms that increase the accuracy of identification. Parallel to increasing the accuracy of identification, other studies were aimed at finding computationally efficient feature extraction algorithms. The following lists are the different features extraction algorithms discussed in the animal biometric survey conducted by Kumar and Singh [55]:

- Hand-engineered features like; scale invariant feature transform (SIFT) [56], local binary patterns (LBP) [57], histogram of oriented gradients (HOG) and Speeded Up Robust Features (SURF) [58].
- Model extracted features: Convolutional layers generated features [59]

Looking to improve the re-identification accuracy a combination of the hand-engineered features and dimensionality reduction algorithms like principal component analysis (PCA) were considered in research work by Finn *et al.* [60]. Once the features are extracted from the images, they are used to train a classification model. Such a model would be used for the automated re-identification of a query image. Several classification models are used to this end, including Random Forest (RF), Support Vector Machines (SVM), and Nearest neighbor K-NN classifiers, among others.

In the decade 2010-2020, alternative classification models and feature extraction techniques like deep convolutional neural networks were introduced in computer vision tasks [61]. Deep learning techniques were adopted because of their robustness to learning from large amounts of data that have high feature dimensions. Also, the availability of powerful computing hardware in the form of graphic processing units enhanced the shift from traditional hand-engineered features to convolutional features and deep learning neural networks [61]. Weinstein [6] notes that active research in computer vision using deep neural networks was on finding the optimal size of training sets and model parameters; as a result, several deep neural networks architectures emerged. These neural network architectures have different model parameters, depth sizes, and computational speeds. A list of popular architectures is provided below.

- ResNet [62]

- VGG [12]
- AlexNet [63]
- DenseNet [13]
- SqueezeNet [64]
- GoogleNet [65]

The architectures above each have variants. For instance, the VGG architecture has these variants: VGG-11, VGG-16, and VGG-19.

2.3.2 Description of Convolutional Neural Networks used in the Current Work

The VGG network architecture, came as an improvement to a neural network architecture developed by Krizhevsky *et al.* [66] in 2012. Simonyan and Zisserman [12] made the VGG to have a stack of convolutional layers, that use smaller filters of 3×3 size as opposed to the previous neural networks that used filters of varying: sizes $11 \times 11 \times 3$ for the first convolutional layer filter, the second filter is; $5 \times 5 \times 48$ and the third and fourth layers were filtered with $3 \times 3 \times 256$ and 3×3 by 192 respectively. In the VGG network, the filters are left to be small sizes of 3×3, and 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 1000 dimension 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 depicted in Figure 2.5 comes in two commonly used variants, 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.* [62] proposed a different convolutional neural network setup (ResNet), because increasing the depth of neural networks may not be suited for some tasks as a result of vanishing gradients. Therefore He *et al.* [62] suggested a mechanism in which deep neural networks do not suffer from the vanishing gradient problem; instead of having stacked layers, some deeper layers are skipped, as shown in Figure 2.3. The motivation of the 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

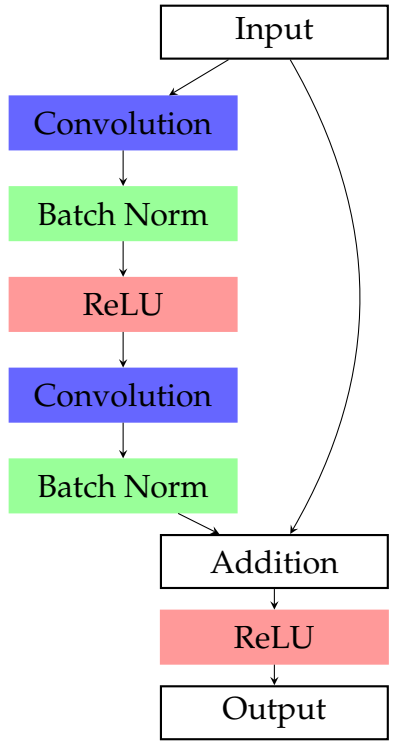


FIGURE 2.3: ResNet Network architecture [67]

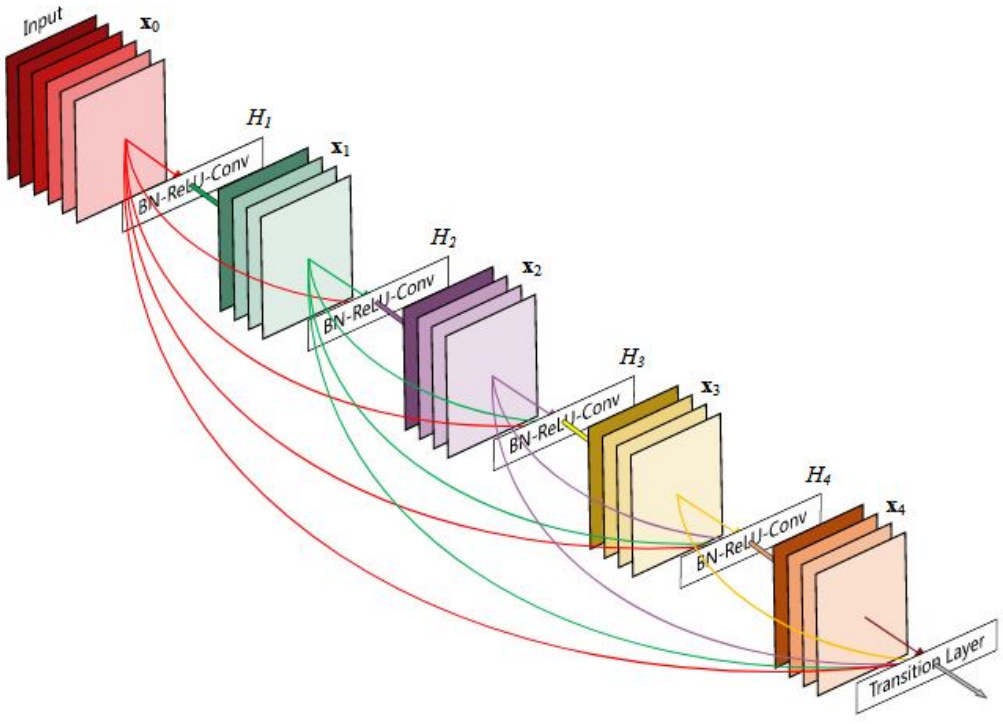


FIGURE 2.4: Dense Network architecture [13]

output of the earlier layers is used to learn a residual mapping, and this mapping is added to deeper layers preserving the information learned before a possible vanishing gradient occurs. The ResNet architecture comes in variations namely: ResNet-34, ResNet-50, ResNet-101 and ResNet-152, where the number indicates total layers in the residual neural network.

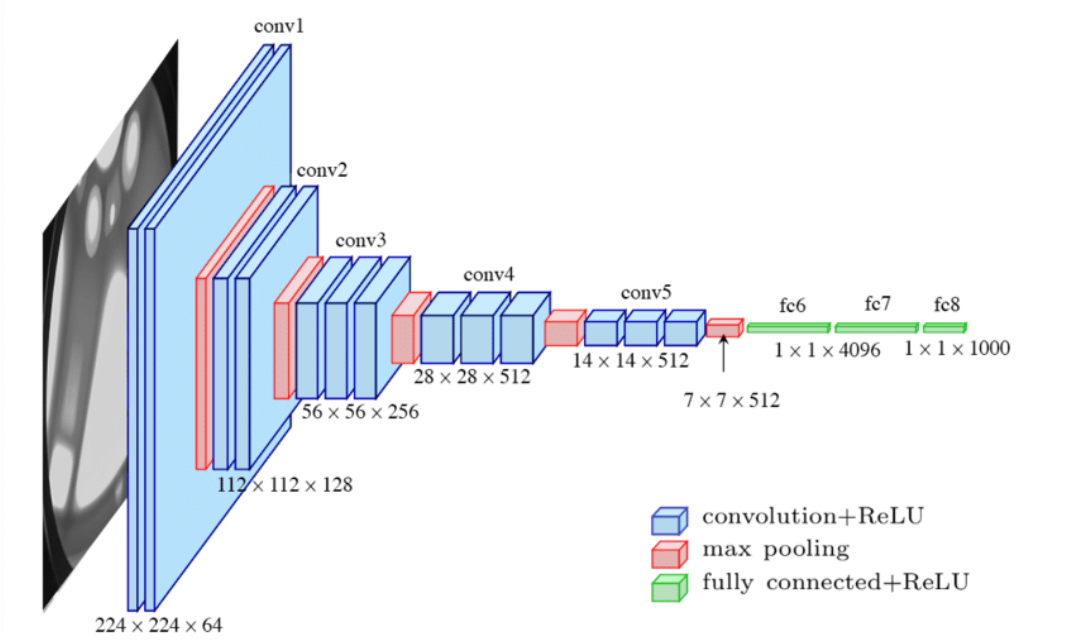


FIGURE 2.5: VGG Network architecture [12]

Huang *et al.* [13] created DenseNet, which is an opposite 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 Figure 2.4. The positive of densely connected neural layers is that the dense connections have a regularization effect, a good thing for smaller datasets where over-fitting is prevalent. DenseNet also comes in several variants, like DenseNet-121, DenseNet-201. It is worth noting that the number of layers is not fixed.

2.3.3 Classification vs Similarity Learning

Classification

Deep learning in image retrieval solves problems in one of two ways, namely: classification and similarity learning. In the classification space, the models are trained to learn a function that maps features to a particular class. The

class should have been seen during training. In an N-class problem, the classifier model predicts which class in one of the N-classes a query data point belongs to [63].

In the deep convolutional neural networks classification approach, the top layer of the network is fully connected and has an activation function that produces a probability of the query image belonging to a class. Commonly used activation functions at the top layer are softmax and sigmoid. A classifier model trained this way cannot predict an unknown class. If a new class is introduced in the dataset, the classifier model needs to be re-trained. Re-training means limited flexibility in deploying such a model in environments where chances of encountering new classes are high [68]. You *et al.* [69] claimed that some training can take up to 14 days. You *et al.* [69] proposed an algorithm to efficiently reduce the training time. However, not only the training time is a challenge here, but also computer resources are reserved for the training process. As noted by Schroff *et al.* [68], re-training models is inefficient and advocated for the use of similarity learning.

Similarity Learning

Similarity learning differs from classification in that the top layer of the deep convolutional neural network does not have a softmax or sigmoid function. There is no class probability produced at the top layer, but a representation is referred to as an embedding vector. The neural networks learn to transform input images into embedding vectors such that images from different classes have embedding vectors that are far apart in terms of some metric measure like euclidean distance. This is achieved through minimizing a ranking loss function and backpropagation during training. When evaluating the performance of a similarity learning model, the distance between a query image embedding and other embedding is computed, if the distance is sufficiently small, based on a certain threshold, the images are predicted to be from the same class [70].

Similarity learning has been applied successfully in a wide range of fields. In the sports and entertainment industry Manack and Van Zyl [71] used deep similarity learning to rank soccer teams and Burns and Zyl [72] used similarity learning for a music recommender application. Variawa *et al.* [73] used similarity learning to automate the re-identification of galaxy patterns and cosmic representation.

Commonly used setups in similarity learning are: a pair of neural networks that share the same weights called Siamese neural networks [74]–[76]

and a triplet of neural networks that share the same weights referred to as Triplet network [77], [78]. Other researchers have looked at quadruplet networks [79]. The goal of these setups is the same: to reduce the distance between embedding vectors for the same class data points and increase the distance between embedding vectors of different class data points. However, this goal is achieved in slightly different ways in terms of training data points selection and ranking loss functions used.

Pair Networks; the Siamese Neural Networks

Kaya and Bilge [80] conducted a survey showing how Siamese and triplet networks are set up and trained. Siamese neural networks depicted in Figure 2.6, are trained by selecting a pair of images from a batch to minimize the distance D_W between the resulting embedding vectors $G_W()$ of input X_1 and X_2, using a contrastive loss function;

$$L_{Contrastive} = (1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, \alpha - D_W)\}^2, \tag{2.1}$$

where D_W is given by $D_W(X_1, X_2) = \|G_W(X_1) - G_W(X_2)\|$ and Y takes value 1 if X_1 and X_2 are from the same class, and 0 if X_1 and X_2 from different classes and m is a margin to force a certain distance between embedding vectors of different classes.

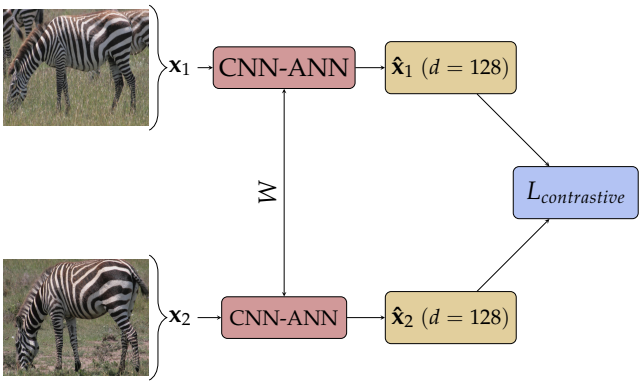


FIGURE 2.6: Pair Networks (SNN), W (Shared weights) and d is the size of the last layer embedding [10].

Triplet Networks

The triplet network, consists of three convolutional neural networks, and is trained to minimize the triplet loss function given by:

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]. \tag{2.2}$$

The input images anchor x_i^a, and positive image x_i^p belong to same class while the negative image x_i^n belongs to a different class. The triplet network learns to simultaneously increase the distance between embedding vectors of the anchor and negative image while minimizing the distance between embedding vectors of anchor and positive image [77]. Figure 2.7 depicts the structure of a triplet network adapted from [77].

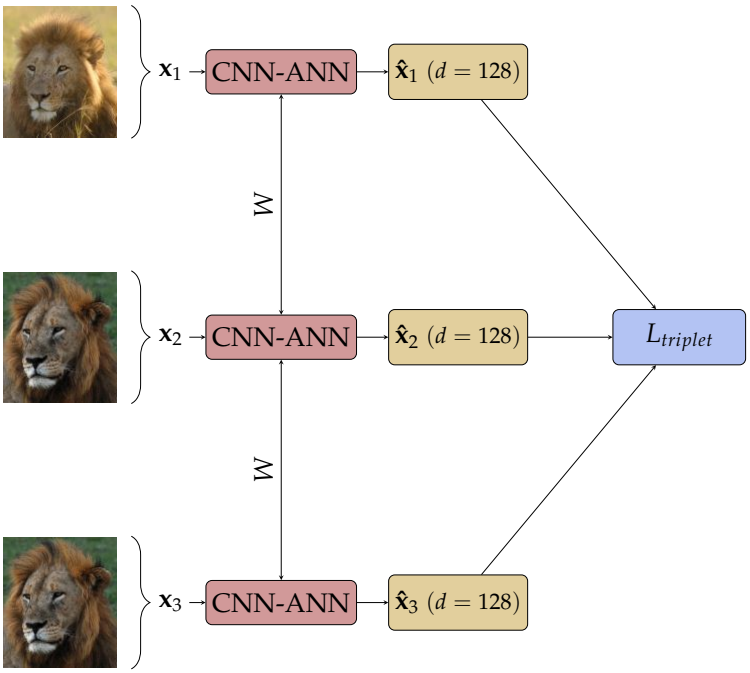


FIGURE 2.7: Triplet-loss Network with (W) the shared weights and d the dimension size of the last layer embedding.

Sampling Pairs

Schroff *et al.* [68] realized that triplet network training could be enhanced by using some criteria to select training triplet pairs. Semi-hard negative triplet sampling was proposed for human face image re-identification. According to Xuan *et al.* [81] in hard negative mining, the training pairs as selected such that the negative samples are similar to the anchor samples presented

in Equation 2.3. The distance $d(f(x))$ between the hard negative pairs must not lie within the margin α in Equation 2.2.

$$x_{hardnegative} = argmin\ d(f(x_a), f(x_n)). \tag{2.3}$$

Semi-Hard negative sampling on the other hand selects training pairs such that the anchor image x_a and the negative image x_n are closer to each other and are allowed to lie with the margin α. Put differently, the distance $d(f(x_a), f(x_n))$ can be greater than $d(f(x_a, f(x_p))$. Other sampling strategies are Easy negative mining, a pair sampling scheme that selects the most different anchor x_a and negative x_n image pairs illustrated by:

$$x_{easynegative} = argmax\ d(f(x_a), f(x_n)). \tag{2.4}$$

Xuan *et al.* [81] claimed that easy negative does not result in better performing model, and proposed the use of easy positive mining: where the most similar image pairs, anchor and positive images are selected during training presented in:

$$x_{easypositive} = argmin\ d(f(x_a), f(x_p)). \tag{2.5}$$

Research on varying sampling techniques is continuing. There are emerging techniques proposed in recent studies like the bayesian updating triplet mining [82] and SoftTriple that does not employ any sampling technique [83]. There are loss functions that do not require a sampling technique. The following subsection gives more details about some of the class-aware loss functions.

Common Distance Measures of Similarity

At the core of similarity learning methods is the function used to measure the distance between embedding vectors. A smaller distance measure is preferred between embedding vectors of data points belonging to the same class. The different distance measures used in similarity learning are discussed in the following paragraphs.

The Minkowski distance measurement is a parent of two distance functions namely Euclidean and Manhattan [84]. The Minkowski distance is

$$d_{Minkowski} = \sqrt[z]{\sum_{i=1}^{n}(X_i - X_j)^z}, \tag{2.6}$$

where X_i and X_j are vectors in n-dimension space and z is positive real number greater than one. The Euclidean distance is a special instance of Minkowski where $z = 2$

$$d_{Euclidean} = \sqrt{\sum_{i=1}^{n}(X_i - X_j)^2}. \tag{2.7}$$

There are variants of the Euclidean distance: average distance and weighted euclidean distance. These variants came about because the Euclidean distance suffers when the largest-scaled feature dominates others. In the average distance, the resulting distance measure is scaled down with the features' total number of dimensions. Equation 2.8 shows the average distance:

$$d_{ave} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(X_i - X_j)^2}. \tag{2.8}$$

In some datasets the features may not have the same weight, instead of applying the same scaling factor to all features as done in the average distance, the weighted Euclidean distance applies a variable weight at each feature. The updated weighted distance becomes:

$$d_{Euc} = \sqrt{\sum_{i=1}^{n} w_i(X_i - X_j)^2}, \tag{2.9}$$

where w_i is the weight given to the ith feature. Similar to the Euclidean distance, the Manhattan distance is also a modified Minkowski distance where $z = 1$:

$$d_{Man} = \sum_{i=1}^{n}(X_i - X_j). \tag{2.10}$$

Other distance measures not related to the Minkowski family are Mahalanobis and Cosine similarity measures. The Minkowski distance is said to be dependent on the dataset being studied; as we see in the weighted Euclidean, the weight will be dependent on each feature in the data set.

De Maesschalck *et al.* [85] discusses how the Mahalanobis distance measures similarity in multivariate chemometrical. Mahalanobis is preferred because it considers the correlation in the dataset by calculating the variance-covariance matrix C:

$$d_{mah} = \sqrt{(X_i - X_j)C^{-1}(X_i - X_j)^T}. \tag{2.11}$$

The variance-covariance matrix C is computed as:

$$C_x = \frac{1}{n-1}(X_c)^T(X_c), \tag{2.12}$$

where X is the matrix of n objects, X_c is the column-centered data matrix (X - $\overline{X}$). Figure 2.8 adopted from De Maesschalck *et al.* [85] illustrates how clusters are formed by Euclidean sub-plot a distance function sub-plot *a* and Mahalanobis function sub-plot *b*. The Cosine similarity is the measure of the

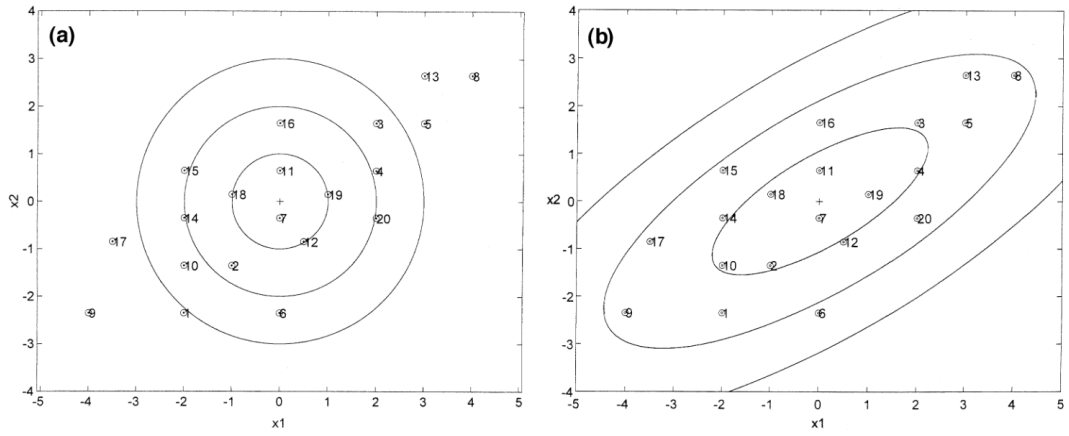


FIGURE 2.8: A plot of simulated data showing a comparison of clusters generated by (a) Euclidean distance and Mahanalobis distance (b) [85]

angle between two feature vectors [86]. If the angle calculated using Cosine similarity measure is sufficiently small the vectors are said to be belonging to the same class

$$d_{cosine} = \frac{\sum_{i=1}^{n} x_i x_j}{\|x_i\|_2 \|x_j\|_2}, \tag{2.13}$$

where $\|x_i\|_2$ is the magnitude calculated using:

$$\sqrt{x_1^2 + x_2^2 + ... + x_n^2}. \tag{2.14}$$

The discussion of these similarity measures is relevant because Cha [84] and Kim *et al.* [87] stated that for different dataset sets the choice of distance measure affects the clustering performance of the models.

Class Distribution Based Loss

Wang and Isola [88] also acknowledged the limitations of pairwise loss functions such as contrastive loss; however, instead of formulating completely

new loss functions, they proposed a distribution-aware variant of the contrastive loss function. Rippel *et al.* [88] proposed a deviation from pairwise losses to what Wang *et al.* [90] termed class aware losses. The proposed loss function, Magnet loss, consider class clusters that are updated during training. These clusters capture intra-class variations but also penalize 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.* [89] note that the Magnet loss function demands that partially pretrained 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.* [91]. An adaptation of NCA [92] 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:

$$p\left(x\right) = \underset{p}{\operatorname{argmin}}\ d\left(x, p\right), \tag{2.15}$$

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

$$\epsilon = \max\ d\left(x, p\left(x\right)\right). \tag{2.16}$$

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

$$l = -\log\left(\frac{\exp\left(-d\left(x, p\left(y\right)\right)\right)}{\sum_{p(z)\in p(z)}\exp\left(-d\left(x, p\left(z\right)\right)\right)}\right) \tag{2.17}$$

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 [93] 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 [91]. Teh *et al.* [94] 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.* [94] 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. Proxy-NCA++ belong to the class-aware loss functions category, the current work investigates if there is reason to move from pair-based loss functions to class-aware loss function in general. The loss functions compared are the the basic Proxy-NCA and triplet-loss

2.3.4 Model Evaluation: Metrics Measured

The commonly used metrics used to determine information retrieval capabilities of deep neural networks models include accuracy top-k, recall top-k, mean average precision (MAP), and mean reciprocal ranking (MMR). The k is a number deliberately chosen in experiments. Calculating the accuracy at top-k is calculating how many true positives (TP) were predicted by the model over the k returned predictions. Precision is computed using a top-k scan, where k is incremented by one starting from one up to k. At each top-k scan, the precision for a single query image will be computed as $P@k$. Average precision is the sum of all $P@k$ divided by the total true positive as:

$$AP = \frac{1}{N} \sum_{k}^{n} P@k \times rel@k, \tag{2.18}$$

where N is the total number of true positives in the test set of n images, $P@k$ is the precision at k and rel@k is 1 if the image at k is a true positive otherwise rel@k is 0. The mean average precision (MAP) is the average of AP for all query images Q in the dataset:

$$MAP = \frac{1}{Q} \sum_{i=1}^{N} AP_i. \tag{2.19}$$

Authors report on one of these metrics; Cakir *et al.* [95] reported on Recall-k. The survey by Kaya and Bilge [80] lists research works in computer vision and the varying metrics measured for models in the research works.

2.4 Transfer Learning

Deep neural networks need a large amount of data to train, and this poses a challenge with individual wild animal datasets because they still suffer from imbalance and small datasets. Transfer learning has been investigated where models trained with larger data are used to learn to solve new problems that have smaller datasets [96].

Girshick *et al.* [97] and He *et al.* [98] confirm that models pre-trained on a larger dataset does improve accuracy on small dataset tasks. However, He *et al.* [98] states that if there is sufficient training data (greater than 10000 images) for the new task transfer learning model may not yield better performance. Pan and Yang [99] termed the condition where transfer learning yields poor results as negative learning.

2.5 Zero-shot Learning

When there is few training classes, model performance suffers. Methods like zero-shot learning were studied to improve classification model performance under data-scarce applications [100].

Wang *et al.* [101] states that zero-shot learning can be thought of as a subset of transfer learning. Zero-shot learning came about in scarce data application of machine learning for object detection and classification. In data-scarce environments, there are few classes that are available to train a machine learning model.

In supervised machine learning, models are trained with class examples. Some examples in each class are left out to be used during the testing phase. In zero-shot is learning. However, the classes that are in the test set were not seen during training. The model is trained to use knowledge learned from seen classes to classify unseen classes. The use of prior knowledge for the classifying of unseen data made Wang *et al.* [101] believe that zero-shot learning is a subset of transfer learning. Rezaei and Shahidi [102] applied zero-shot learning in image classification; diagnosing COVID-19 using chest X-ray images. Since there were few examples of COVID-19 chest X-rays at the time, the models were pre-trained using images from other chest x-rays from Asthma patients.

Applying zero-shot learning in individual animal re-identification, as undertaken in the current work, is relevant because the number of captured individuals in animal populations is small, and it is highly likely that an unseen

individual may be observed at the deployment of the models for individual re-identification.

2.6 Few-shot Learning

Unlike zero-shot learning, few-shot learning is based on training models to perform better using few examples. The classes in the training set are not disjoint from classes in the test set. A few examples in each class are left to be used in the testing phase [103]. Bateni *et al.* [104] proposed a simple few-shot learning method. The proposed method is considered simple because it removed the constraint presented in the widely cited work of Snell *et al.* [105] that the training set must have balanced and uniform classes. The current work investigates both few-shot learning and zero-shot learning methods.

2.7 Challenges in Deep Metric Learning

Roth *et al.* [106] 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 [107], others are aiming to obtain better objective functions (loss functions), while others introduce new algorithms to solve problems. Kim *et al.* [108] proposed an efficient facial expression recognition algorithm. However, it is difficult to ascertain whether the improvements come from the proposed algorithm or design choices: feature extraction methods, choice of optimizer function, or the selected network architecture. The concerns expressed by Musgrave *et al.* [109] 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 algorithm.

Musgrave *et al.* [109] 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 of experiments where different neural network architectures and different model parameters are used

is seen as problematic. Roth *et al.* [106] and Schroff *et al.* [68] agree that changing the embedding dimension size has an effect on model performance. Similar observations were made by Wang *et al.* [90] 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 optimizer functions and their hyperparameters, such as the learning rate. It may not be incorrect to find a neural network architecture, optimizer, 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 need to be kept consistent.

2.8 A Summary of Works in Wild Animal Biometrics and the State-of-the-art

2.8.1 A Summary of Works in Wild Animal Biometrics

Literature shows that earlier efforts in individual animal re-identification used hand-engineered features, and these performed remarkably well. Bolger *et al.* [110] used SIFT features to train a giraffe re-identification model. The giraffe dataset was easy to obtain, and the authors were able to capture just the right side of the body of an individual giraffe. Giraffes and human beings encounters are not so dangerous, compared to human encounters with animals like lions, as a result, Bolger *et al.* [110] was able to control how the images are collected. This means that there will be no feature occlusion via shade or tree branches. Controlled image capture makes the problem of re-identification easier, and most algorithms can achieve good performance. The controlled image capture seen with the giraffe dataset may be difficult to emulate in other animal species where a good number of images collected may have feature occlusion, different postures, and lighting conditions.

A similar identification task was done using cheetah images. Gray-scale intensities were extracted and used to compare the two images. If two images are from the same individual, the gray-scale intensities are expected to have a maximum correlation. This method was not robust to changes in quality of the images [111].

Schofield *et al.* [112] adopted deep convolutional neural networks to identify chimpanzees from video images. The ResNet-50 and VGG-16 architectures were used in their experiments, achieving an average of 92.8% individual recognition accuracy. The features were automatically learned through the convolutional layers instead of using hand-engineered features. The top layer of the network was a fully connected layer with a soft-max activation function used to predict 25 classes. The chimpanzees' dataset used in training the models had a total of 15,274 face-image data points of 23 individuals. These face images varied in posture and lighting conditions. Schofield *et al.* [112] noted that recognition accuracy increased to 95.05% when they only used frontal face images. This observation shed some light on the fact that controlled image capture makes models perform better. Limiting the top layer to predict 25 individuals means that if a new individual chimpanzee is introduced in the dataset, then the model needs to be re-trained, which may not be ideal in real-time deployment. In the wild, the chances of encountering a new chimpanzee that was not in the training dataset are high.

A recent study by Van Zyl *et al.* [10] used a variant of ResNet CNN architecture (ResNet-152) and looked at the re-identification of individuals in a nyala population and zebra population. The zebra dataset had 820 total samples of 84 individuals, and the nyala dataset had 1945 samples with 474 individuals. Van Zyl *et al.* [10] employed the concept of similarity learning and the model was pre-trained on the Serengeti dataset. The results obtained vary across the dataset used. On the zebra dataset, a top-1 accuracy of 72.6% was achieved, whereas for the nyala dataset, a top-100 accuracy of 62% was reported. Since all parameters were kept constant and the network architecture used is the same, the varying performance can be attributed to a dataset under investigation. During training, a contrastive loss function was used with a random sampling of training pairs. The use of similarity learning removes the burden of re-training the model each time a new individual is introduced, and this simplifies the deployment of the model in a wild setting. However, Van Zyl *et al.* [10] only tested one network architecture: (ResNet-152). There is a possibility that a different neural network architecture or a different loss function can yield better performance.

2.8.2 State-of-the-art in Computer Vision Problems

The state-of-the-art in computer vision problems was achieved by convolutional neural networks. Computer vision problems are generally categorized into the following: image classification, human pose detection, object detection, and activity recognition. The search for better performance in these problems involved convolutional neural networks. Guo *et al.* [113] summarizes the convolutional neural networks that produced state-of-the-art performance in the computer vision problems. The survey discusses different versions of the ResNet architecture and found that some variants of ResNet are more suitable for image classification on the ImageNet [114] dataset. Other projects have demonstrated the best performance of convolutional neural networks in the re-identification of human beings based on face images. A survey conducted by Li and Deng [115] shows the progression of research in human face re-identification and the best performing models from the year 2014 up to the year 2020. FaceNet [68] and SphereFace [116] were stated as best performing models.

Based on the successes of convolutional neural networks on image datasets is not surprising to observe that the convolutional neural networks have been adopted on wildlife biometrics problems. Authors like Deb *et al.* [48] modified SphereFace model, designed for human re-identification, to be suitable for re-identification of wild animals; the primates faces dataset.

Some researchers have taken a new direction in computer vision experiments. Early indications of superior performance support the deviation from convolutional neural networks to vision transformers. Dosovitskiy *et al.* [117] supports the move from convolutional neural networks to vision transformers. However, he argues that adopting vision transformers comes at a cost because they are computer resource intensive compared to convolutional neural networks. Park and Kim [118] explains how visual transformers combined with convolutional neural networks can achieve better performance than convolutional neural networks. There is, however, a need to find hardware that will make the execution of visual transformers algorithm less computation-intensive. Research has begun to find efficient ways in which visual transformers can be deployed Sun *et al.* [119].

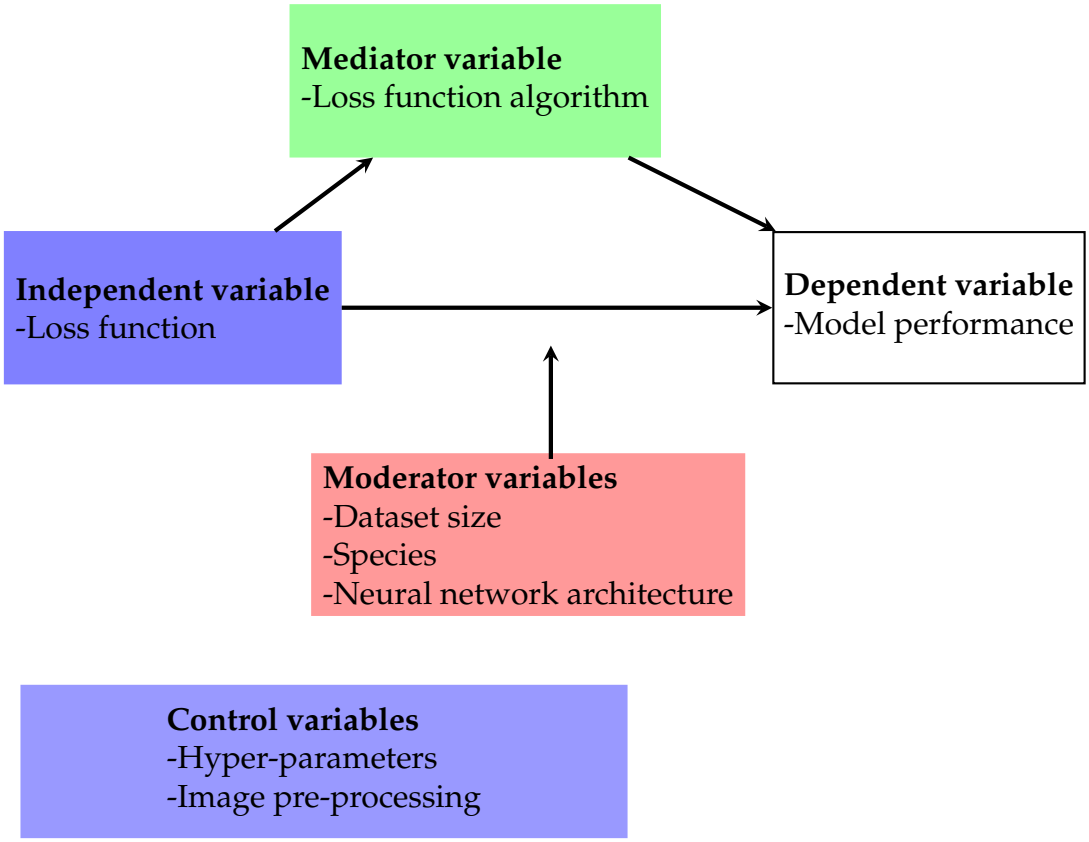


FIGURE 2.9: Conceptual framework

2.9 Conceptual Framework

The conceptual framework in Figure 2.9 illustrates the variables that are under investigation in the current work. The class-aware loss functions like Proxy-NCA were developed as improvements to pair-based loss functions [94]. The loss function, therefore, was the independent variable in experiments conducted in the current work, the loss function algorithm being the mediator variable. Shahinfar *et al.* [120] argues that the dataset size affects model performance, Van Zyl *et al.* [10] indicated that re-identification of individuals in certain species like nyala may be a difficult task compared to the re-identification of individuals in zebra species. These variables, together with the choice of neural network architecture, are regarded as moderator variables in the conceptual framework. Other variables that can affect model performance are hyper-parameters and image pre-processing. These were kept constant in all experiments undertaken in the current study.

2.10 Conclusion

From reviewing the literature, firstly, we found that a project dealing with lion identification only tracked the lions using face images. Such a study was only concerned with identifying the behavior of a lion through tracking the position of the face [121]. The Panthera leo (African lion) is on the red list of endangered mammals [122]. However, there is no work done on the re-identification of individual lions. There is no dataset that contains records of existing individual lions. Therefore, individual lion re-identification, monitoring, and conservation are still done at the species level.

Secondly, the reasons why computer vision in collaboration with biodiversity experts can lead to better ways of tracking, counting, and protecting not just species but individuals in the species population. Thus avoiding the use of invasive, unsafe RFIDs and capture-recapture techniques is unavoidable.

Lastly, even though there are state-of-the-art network architectures in computer vision problems, the performance varies across datasets. The metrics used to measure the performances are a choice of the researcher. Some measure only the mean average precision (MAP), others the Recall@k, and some mean reciprocal ranking (MRR). For bench-marking new experimental designs, researchers will need to measure similar metrics to prior research.

Chapter 3

Research Methodology

3.1 Introduction

This chapter discusses the research design chosen for the current work and lists research instruments used in the collection and labeling of data. A Summary of investigations and model performances achieved by prior research work that tackled individual animal re-identification is given. The previous works form a benchmark to which the current research outputs are compared. The last part of this chapter provides a guideline on what analyses were conducted to describe the data and measurements calculated to evaluate the performance of our model.

3.2 Research Design

Confirmatory experimental research was adopted for this work. Experimental research, in general, allows the researcher to investigate how changing a dependent variable while keeping other variables constant affects outcomes. Confirmatory experimental research extends experimental research by suggesting that researchers must state up-front the phenomenon investigated and describe statistical methods that will be used to arrive at the findings before the data is seen. The phenomena to be investigated should be stated before the data is seen to avoid the temptation of fine-tuning data, removing data points to fit the expected outcomes of the research [123]. The experimental research design was found appropriate for the current work because we would like to investigate how deep neural networks, similarity learning neural networks, assist in the re-identification of individuals in animal populations using animal biometric features. In the current work, the depth of the network was altered, and observations were made on how this alteration affects the performance of the model. Different architectures of convolutional

neural networks were investigated, and the performance was measured for each architecture on the same dataset. And lastly, different loss functions were investigated to determine the most suitable loss function for animal re-identification tasks. Other researchers have conducted research on the re-identification of individual chimpanzees, tigers, zebras, and nyalas. We extended their work to include the re-identification of the lions. Most of the prior work looked at a single neural network and a single loss function. We extend prior work by investigating other neural network models and also compare loss functions. The comparison was necessary because Genç and Ekenel [124] found that different neural network models give varying performance on different data sets. There was a need, therefore for the current work to investigate several neural network models to find the best-suited model for a dataset.

3.3 Methodology

3.3.1 Research Instruments

Data Collection Tools

A web scraper developed in python [1] was used to collect data from websites that have labeled lion images.

Train-test Split

A 10-fold cross-validation protocol was followed in the training, validation, and testing of the model performance. Kohavi *et al.* [125] explained that 10-fold cross-validation splits a data-set into ten equal parts. The training set was split: 80% - 20%, 80% was used as training, and 20% was used for testing model performance. The test dataset was a hold-out set used to measure the mean average precision at R and Recall@1. Ten training and test runs were employed so as to report on the average of the metrics and the confidence intervals.

Baseline Comparisons

Table 3.1 shows a summary of recent works investigating the individual re-identification of chimpanzee, panda, tiger, zebra, and nyala datasets. The

[1] https://github.com/NkosikhonaD/DataCollection

associated features used by prior research and test accuracy achieved by the authors are depicted. These works provided a benchmark on which the current research outputs were compared.

TABLE 3.1: Summary of Performance per Species and Model

Prior Works per Dataset				
Dataset	Features	Total data	Individuals	Accuracy %
Chimpanzee	(SCNN) [126]	5,599	90	77.5
Chimpanzee	(CNN) [127]	5,599	90	59.9
Nyala	(SCNN) [10]	1934	274	72.1
Panda	(CNN) [53]	6462	218	92.1
Tiger	(CNN) [126]	3651	182	86.3
Zebra	(SCNN) [10]	2460	45	72.6

Machine Learning Libraries and Hardware

PyTorch [128] version 1.9 library was used for tensor processing and to build deep convolutional neural networks. Pytorch was chosen because it is open-source software, and it supports the use of hardware accelerators like graphic processing units (GPUs) that are essential in deep neural network experiments that analyze image data.

Google Colaboratory [129]: a cloud service gave us access to open-source graphic processing unit hardware. This hardware was essential in speeding up the training and testing of image analysis-based deep learning experiments that we did in the current work.

Metric learning library: this library extends the Pytorch API; it was designed by Musgrave *et al.* [109] to bring commonly used model performance metrics computation in one place to promote faster benchmarking of new implementations with previous works.

Transfer learning and neural networks architectures: Transfer learning was employed in all the experiments undertaken in this study. The following deep convolutional neural network architectures were selected for our experiments: VGG-11, VGG-19, ResNet-18, ResNet-152 and DenseNet-201. For the VGG and ResNet family, we choose shallow and deeper variants so that we can observe the effect of network depth on the resulting performance for each dataset. These neural network architectures were pre-trained on ImageNet data [114].

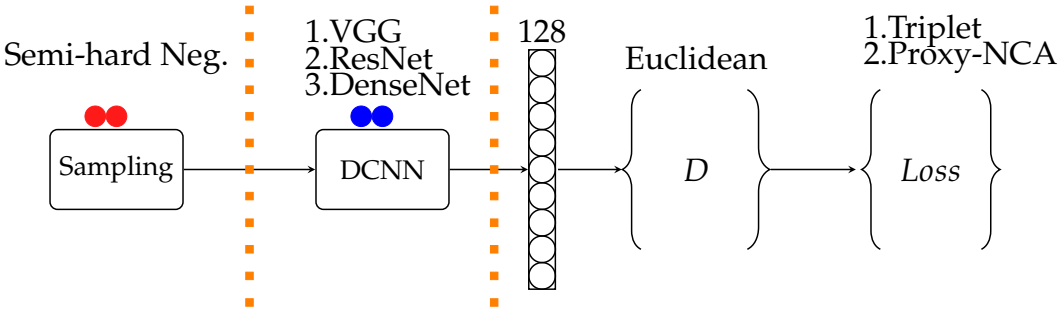


FIGURE 3.1: Experiment setup; Sampling: Selections of training examples; DCNN; neural network backbone, the last layer of the neural network is an embedding vector of size 128 and (D) the distance function used with ranking loss function [2].

Experiment Set-up

Figure 3.1 depicts the general design of our experiments. The first part of the experiment is selection of training pairs, the second part is the network backbone to be trained, the third part is the distance function that measures dis/similarity, and lastly, the loss function that is being minimized through model weights that are updated by back-propagation.

We used the random pair sampling technique for the Proxy-NCA loss function, and semi-hard triplet pair sampling was employed for the triplet loss function. The Proxy-NCA loss function does not depend on pair selection techniques to converge at a local minimum during training, whereas the triplet loss function depends on pair selection techniques to converge at a local minimum [68]. A link to our GitHub repository [2].

Model hyper-parameters

In the triplet loss experiments, we set the margin to 0.2. We used the adaptive moment estimation algorithm (Adam) as an optimizer. The learning rate for the Adam optimizer was set to 0.01, and we left the $\beta 1$ and $\beta 2$ to default values of 0.9 and 0.999, respectively. The Proxy-NCA loss function requires the number of classes to be specified and a softmax-scale. The number of classes varied depending on the dataset under consideration, and the softmax-scale was left to the default value of 1.

We ran two experiments for each dataset, and neural network backbone, each of the experiments shared the same model parameters and data, except for the loss function being minimized. This setup allowed us to determine if one loss function is better suited for individual wild animal re-identification.

[2] https://github.com/NkosikhonaD/WildIdentifcationDeepWild

3.3.2 Data

Table 3.2 summarises the datasets used in our experiments. Total data points in each dataset (N), individuals/classes (I) found in each animal species, and the 80% train and 20% test split. As opposed to Dlamini and Zyl [1], we split our datasets in a zero-shot learning fashion. All the classes that formed part of the test data were classes that were not seen during the training phase. The data splits were done in a zero-shot fashion because the prior works we benchmarked the current research employed similar data split technique.

Mara Predator Conservation Project Data

Dataset from the Mara predator project in Kenya [3] will be automatically collected using a Web scrapper, and labels will be obtained from images metadata. The Lion data set is a result of a carnivore preservation drive by Frank [130] in Kenya. The community of Kenya Masia was given incentives to report sightings of the Lions. Individual Lions were given names, and their images were saved online. Frank [130] observed that there was a remarkable decline in the population of Lions in Kenya due to hunting practices and the urge to protect livestock.

Publicly Available Datasets

Four publicly available datasets will be used to compare the current work with benchmark model performances set for each of the datasets. These datasets are: Amur tiger dataset [11], the giant panda dataset [53], zebra dataset [49], and Tai chimpanzees dataset [51].

Dataset Annotation

The datasets were annotated using the following criteria:

- *name*: identity of the animal
- *body part* : face, flank
- *species*

[3] http://www.livingwithlions.org/mara/

TABLE 3.2: *N*: Dataset Size; *I* Total individuals in dataset; and #: Average data points per Individual for **zero-shot learning experiments**

Dataset Splits Zero-Shot learning						
Dataset	*N*	*I*	#	Split	*S*	#
Lion	750	98	7.7±4.0	**Train** **Test**	594 156	79 19
Nyala	1,934	274	7.1±5.1	**Train** **Test**	1,213 729	179 95
Zebra	2,460	45	54.7±7.3	**Train** **Test**	1,989 471	36 9
Chimp	5,078	78	65.1±17.0	**Train** **Test**	3,908 1,170	62 16
Panda	6,462	218	29.64±8.0	**Train** **Test**	5,546 916	174 44
Tiger	3,651	182	20.1±15.0	**Train** **Test**	1,887 1,764	107 75

Data Pre-processing

Our image data had varying sizes. The models pre-trained on ImageNet requires that the image input be at least $H \times W$ of 224×224, and the channel should be 3. We used the Pytorch transforms library to resize all the input images to 224×224 dimensions. This pre-processing step was kept the same for all data sets to achieve uniformity across our experiments.

3.3.3 Analysis

Descriptive Measures

The metrics used to measure the model performance were computed over ten train-test runs, and the mean of the measured metrics was reported. The standard deviation was used to compute confidence intervals of the metrics measured based on a 95% confidence level.

Model Performance Metrics

Mean Average Precision (MAP*)*@R and Recall@1 was computed to assess the performance of the models in our experiments. The MAP@R given in

TABLE 3.3: N: Dataset Size; I Total individuals in dataset; and #: Average data points per Individual for **few-shot learning**

Dataset Splits Few-shot Learning						
Dataset	N	I	$\mathbb{E}[\#/I]$	Split	S	#
Lion	750	98	7.7±4.0	**Train** **Test**	750 736	8 3
Nyala	1,934	274	7.1±5.1	**Train** **Test**	1,213 1,189	10 2
Zebra	2,460	45	54.7±7.3	**Train** **Test**	1,989 1,771	49 12
Chimp	5,078	78	65.1±17.0	**Train** **Test**	3,908 3,801	50 13

Equation 3.1 combines r-precision [131] and mean average precision Equation 2.19.

$$\text{MAP@R} = \frac{1}{R}\sum_{i=1}^{R} P(i), \tag{3.1}$$

where $P(i)$ is precision at i if the ith neighbour is a true positive otherwise $P(i)$ is 0. $MAP@R$ is mean average precision computed over R nearest neighbours of the query image retrieved by the model Musgrave *et al.* [109]. Precision measures the ratio of true positive retrieved data points to the total retrieved data points. Table 3.4 adopted from Musgrave *et al.* [109] give details on how MAP@R is less noisy compared to recall@1 and r-precision.

TABLE 3.4: MAP@R Robustness explained compared with recall@1 and r-precision when R =10 [109].

Illustrating MAP@R			
Retrieved images	Recall@1 %	r-precision %	MAP@R %
a) 10 retrieved images, only the 1st one is correct	100	10	10
b) 10 retrieved images, 1st and 10th are correct	100	20	12
c) 10 retrieved images, 1st and 2nd are correct	100	20	20
d) 10 retrieved, all 10 are correct	100	100	100

When the 1st and the 10th image retrieved are correct, for R = 10:

$$Recall@1 = \frac{1}{1} \;\; which\ evaluates\ to\ 100\%,$$

$$r - precision = (\frac{1}{1} + \frac{2}{10}),\ which\ evaluates\ to\ 20\%,$$

and

$$MAP@R = \frac{1}{10}(\frac{1}{1} + \frac{2}{10}) \textit{ evaluates to } 12\%.$$

We measured Recall@1 as the top-1 accuracy achieved by the models. The Recall@1 measurements were used to compare our results with the benchmark results set by previous research that was conducted using some of the public datasets we adopted in the current study.

3.4 Limitations

The results drawn from experiments conducted in the current work are limited to the data used and experimental design choices made. These results cannot be generalized to other endangered animals and plant species in the world. In order to draw conclusions on other species, there would be a need to extend the current work to include experiments for all endangered species. This is beyond the scope of this work.

3.5 Ethical Considerations

The use of animal data was only for this research; data was not shared or published to make profits. Permissions were given for the use of some data sets, and all conditions made under the permissions granted were observed. The permissions required that where we refer to the dataset, a citation and reference of the custodian be included. The animals used in the study were not subjected to any physical harm or physical contact because the images were collected from online sources.

3.6 Conclusion

This chapter discusses the research design chosen for the current work, which we followed, to arrive at the outputs of the current work. The experimental research design and relevant components are discussed, and the role these components played in each stage of the current work, from data collection, analysis, and performance measurements.

Chapter 4

Results

4.1 Introduction

This chapter presents our results. The first set of results we present is for the few-shot learning followed by zero-shot learning. We only trained triplet loss models for the few-shot learning, using nyala, lion, and chimpanzees datasets. For the zero-shot learning, we did preliminary experiments where we searched for a dimension of an output vector that results in better model performance. Wang *et al.* [90] demonstrated that changing the output vector dimension affects model performance. We then made comparisons between a model trained on triplet loss and the same model trained on Proxy-NCA loss function. We also present the best neural network in each dataset by comparing the performance achieved by the different models per dataset. Where there is prior work on a dataset, we included the result as "prior," and these results are compared with the performance from our experiments.

We did additional classification experiments using only ten classes. Classification experiments required that we have a fixed number of classes for the training set, and from these classes were randomly selected 30% examples from each class were set aside, which we used for testing model performance. We plotted clusters in our training set using PCA dimensionality reduction. We replaced the last layer of our VGG-11 neural network with a softmax activation function so that we could get class probabilities; we plotted the confusion matrix and computed the f1-score from our test set. We conclude the chapter by summarizing trends observed in the results.

4.2 Few-shot Learning Results

Table 4.1 presents the MAP results for few-shot learning models. We did not pursue the few-shot learning approach further because previous works we

compared the current results with did zero-shot learning.

TABLE 4.1: Few-shot results for 128-D triplet loss [1]

MAP %				
Model	Lion	Nyala	Chimp	Zebra
VGG-11	67.7	64.3	61.2	70.4
VGG-19	57.6	61.8	59.8	84.9
ResNet-18	72.1	60.0	70.9	66.8
DenseNet-201	60.6	59.8	91.9	64.3

4.3 Zero-shot learning Results

4.3.1 Search for Optimal Dimension Size

The results presented in Table 4.2 contains mean average precision at R for output dimension vector sizes: 64, 128 and 512.

TABLE 4.2: 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

We found that the best choice of output-vector dimension is 128 for all the datasets we studied except for the zebra dataset, where we chose the output vector dimension of 64. This is because the size 64 resulted in better MAP@R than the output-vector dimension of 128 [2]. We show the returned results from query images of the different datasets in Figure 4.3 and Figure 4.2.

4.3.2 Comparing Triplet Loss and Proxy-NCA

For all the datasets, we used output vector dimension 128 except for the zebra experiments, the results that we present are based on an output vector dimension of 64. Table 4.3 shows the Recall@1 and Table 4.4 shows the MAP@R for each model per dataset. One row is for the triplet loss result, and the other

row is for the Proxy-NCA loss result. We bold the best result in each experiment and highlighted with gray the other results that are not significantly different from the best.

TABLE 4.3: Recall@1: triplet loss semi-hard mining vs. Proxy-NCA for training classes. Bold indicates the best performing method, and gray highlights results that are not statistically significantly different from the best.

Recall@1 %							
Architecture	Loss	Lion	Chimp	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 ± 0	72.1 ± 0	72.6 ± 0	86.3 ± 0

4.3.3 Comparing Neural Networks Performance per Dataset

We summarize the performance achieved by the different neural networks in each dataset. The results achieved by the different neural network architectures per dataset are presented in Figure 4.1 [2]. From these results, we can see the best-suited neural network architecture in a specific dataset. In the same table, we show in the red bar the benchmark results from previous work.

4.4 Additional Experiments: The Classification Approach

We conducted additional experiments where we did a classification instead of a similarity learning. We selected a maximum of ten classes from each dataset. Selecting ten classes enabled us to view the performance measures

TABLE 4.4: MAP@*R*: triplet loss semi-hard mining vs. Proxy-NCA. Bold indicates the best performing method, and grey highlights results that are not statistically significantly different from the best.

MAP@R %							
Architecture	Loss	Lion	Chimp	Pandas	Nyala	Zebra	Tiger
VGG-11	Triplet	16.5 ± 2	12.9 ± 2	$\mathbf{32.0 \pm 2}$	$\mathbf{11.2 \pm 0}$	16.8 ± 1	22.8 ± 1
	P-NCA	17.7 ± 1	$\mathbf{13.8 \pm 3}$	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	$\mathbf{19.0 \pm 0}$	$\mathbf{24.6 \pm 4}$
	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	$\mathbf{20.8 \pm 1}$	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

like the confusion matrix more clearer. Viewing a 10×10 confusion matrix is more clearer than viewing a 98×98 confusion matrix. All our datasets have more than 90 classes.

We split the dataset into training and testing sets: For the training set, we performed a PCA so as to view the clusters found in our training set and to see how separable are the data points amongst the individuals. We plotted the clusters for all training samples in all the datasets we used in our experiments. The lion and chimpanzee datasets have real names as labels to the different individuals. However, in the rest of the datasets, the individuals do not have real names but are represented by numbers. This is depicted in the cluster plots and confusion matrix.

From the ten classes selected randomly for the classification experiments, we trained a VGG-11 network backbone, with the final layer having a dimension size of 10; we applied the softmax activation function on the final layer so that we get class probabilities. Our similarity learning experiment results suggested that VGG-11 is generally the most suitable neural network backbone for our datasets. During the testing phase, we calculated the f1-scores, depicted in Table 4.5 and the confusion matrix for each dataset.

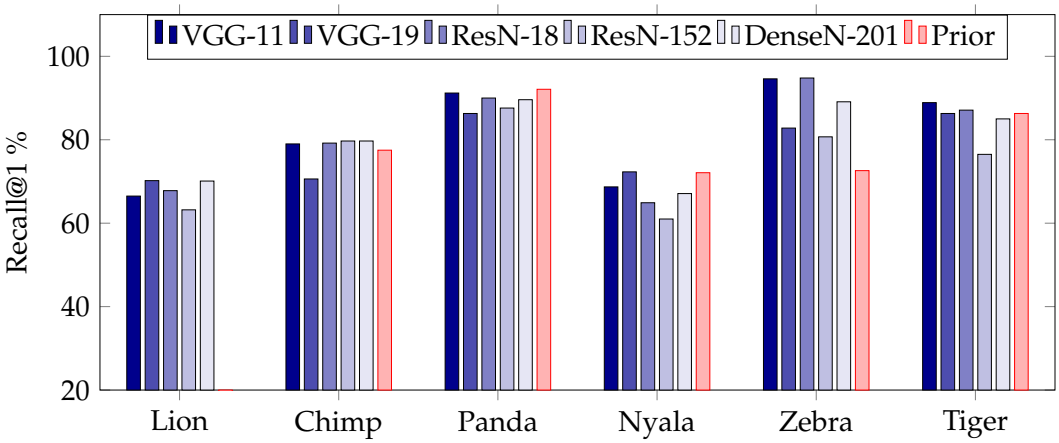


FIGURE 4.1: Comparing the Recall@1 achieved by each neural network in a dataset. The red bar represents the best performance achieved by other researchers we found in the literature.

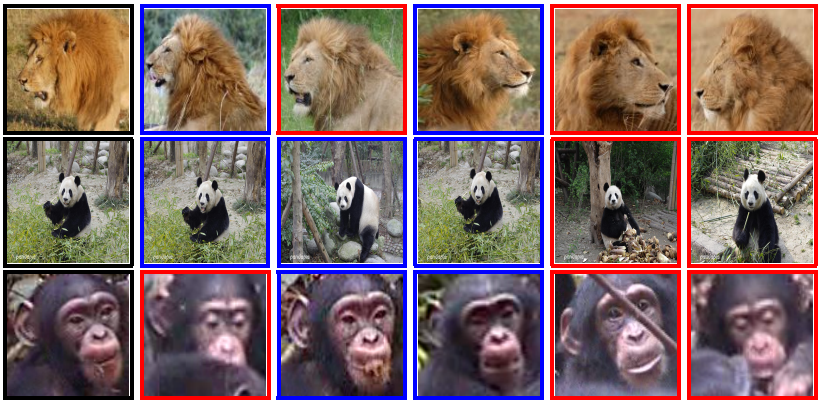


FIGURE 4.2: Faces: The first image is the test image, with five retrieved neighbours: the blue border is correct retrieval and red border incorrect retrieval [2]

4.5 Limitations

The results of the current work are limited to the models, datasets considered, and experimental design choices made. Changing the design of the experiments can change affect the results even when the same datasets are used. The current work relied on datasets that other researchers labeled. These were regarded as correct.

4.6 Conclusion

This chapter presented the results where we compared the effect of changing loss function on model performance. This comparison is meant to show which loss function is best suited for individual wild animal re-identification.

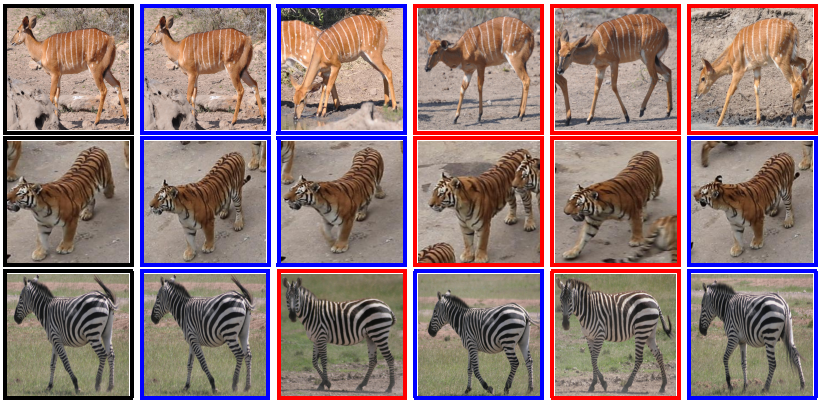
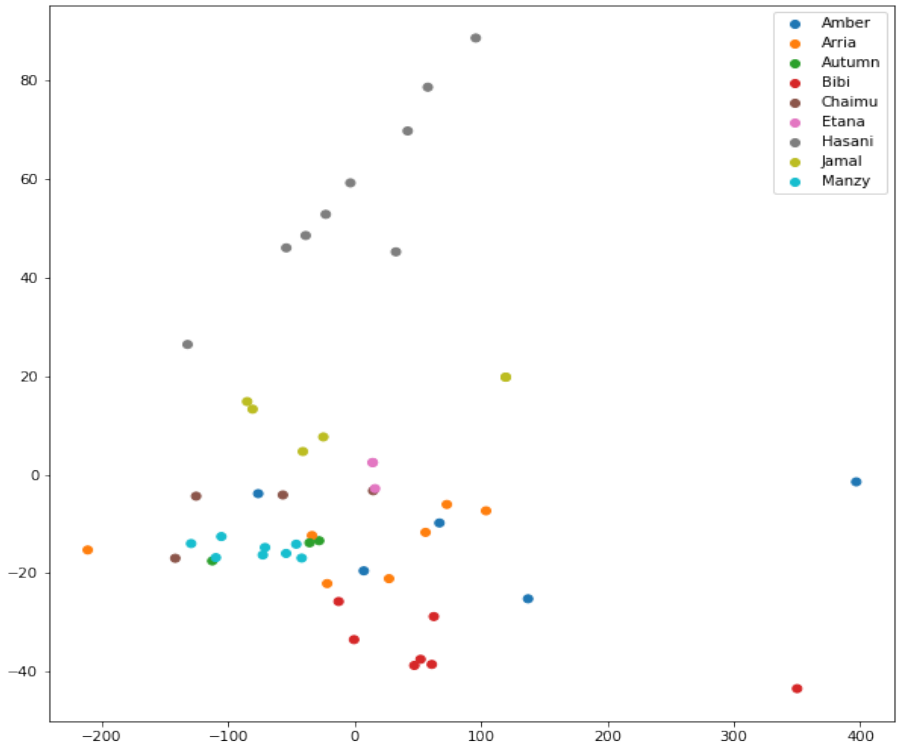


FIGURE 4.3: Flanks: The first image is the test image, with five retrieved neighbours: the blue border is correct retrieval and red border incorrect retrieval [2]

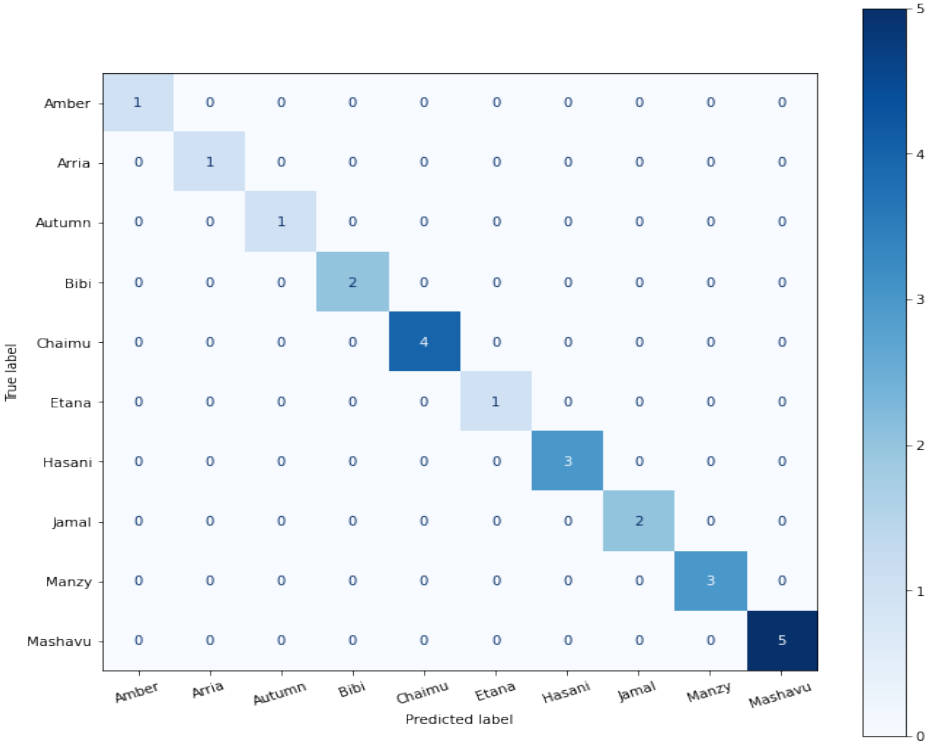
The last part of our experiments was to find out the best performing neural network architecture per dataset, and this is where we compared our work with the results of other researchers. The previous performance achieved by various researchers on the datasets is presented with a red bar in Figure 4.1.

TABLE 4.5: Classification experiments results: F1-score results of ten classes per dataset for VGG-11 Network architecture with softmax top layer.

F1-Score %	
Dataset	F1-Score
Lion	100
Chimp	100
Panda	99
Zebra	97
Tiger	100
Nyala	95

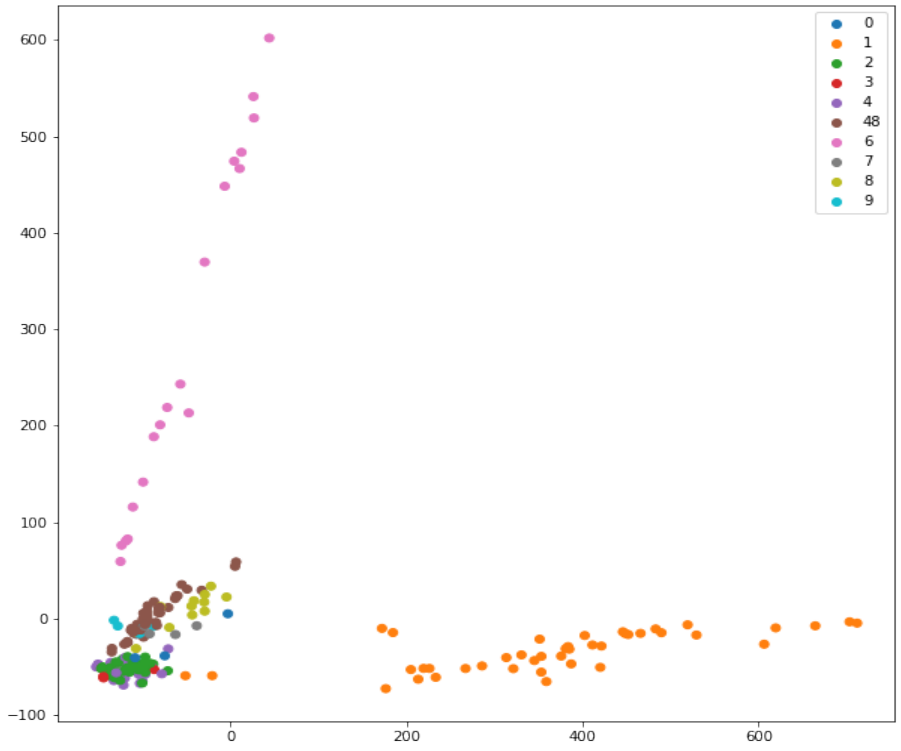


(A) Lion training set clusters

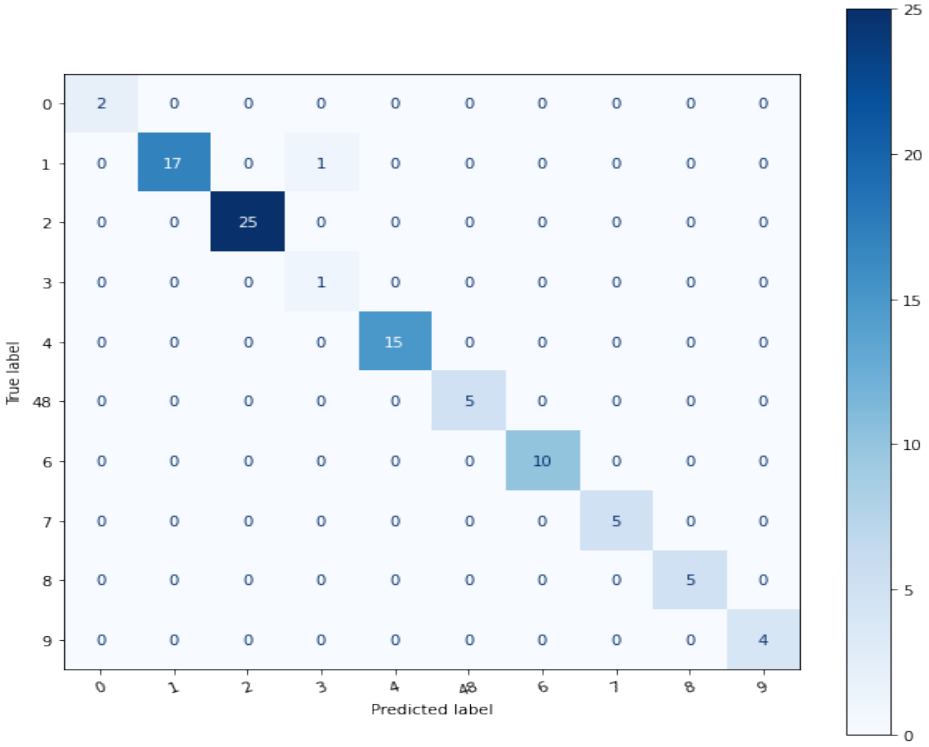


(B) Lion test set confusion matrix

FIGURE 4.4: Lion

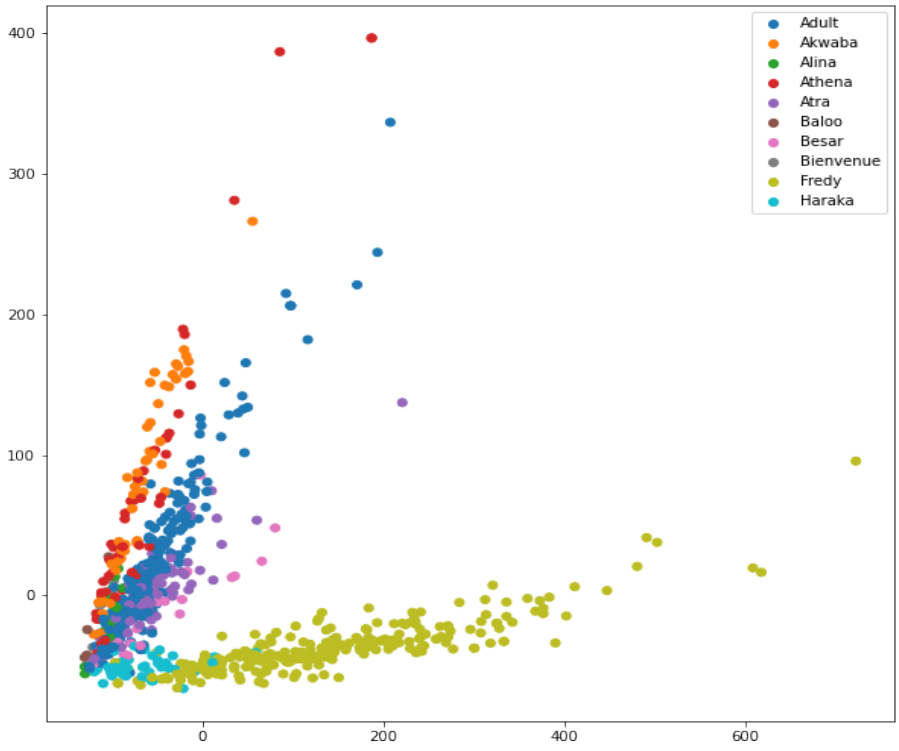


(A) Panda training set clusters

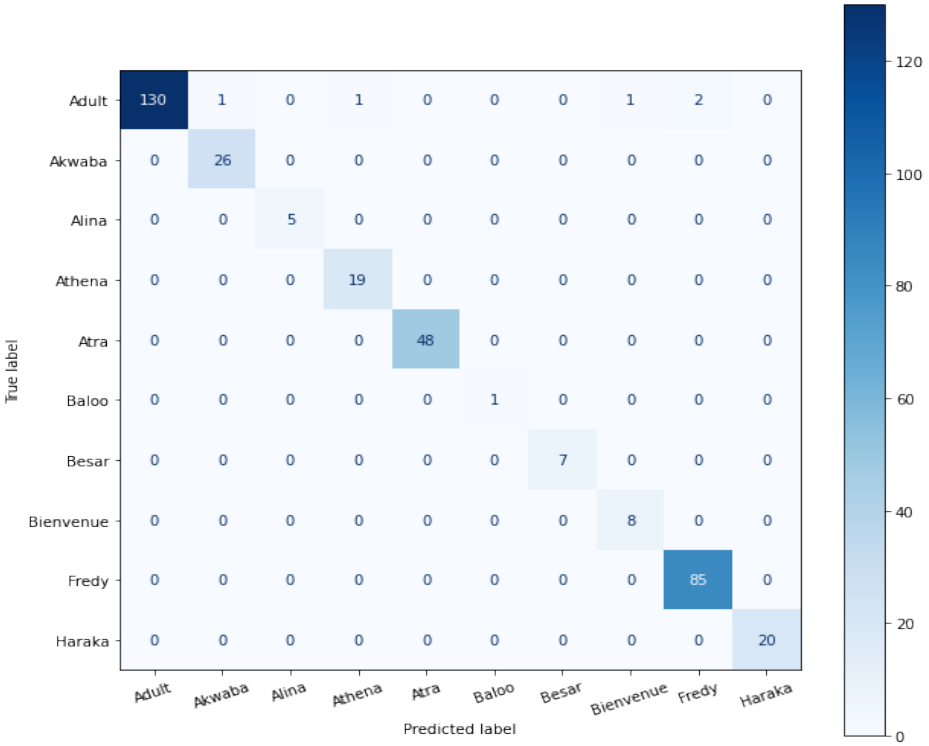


(B) Panda test set confusion matrix

FIGURE 4.5: Panda

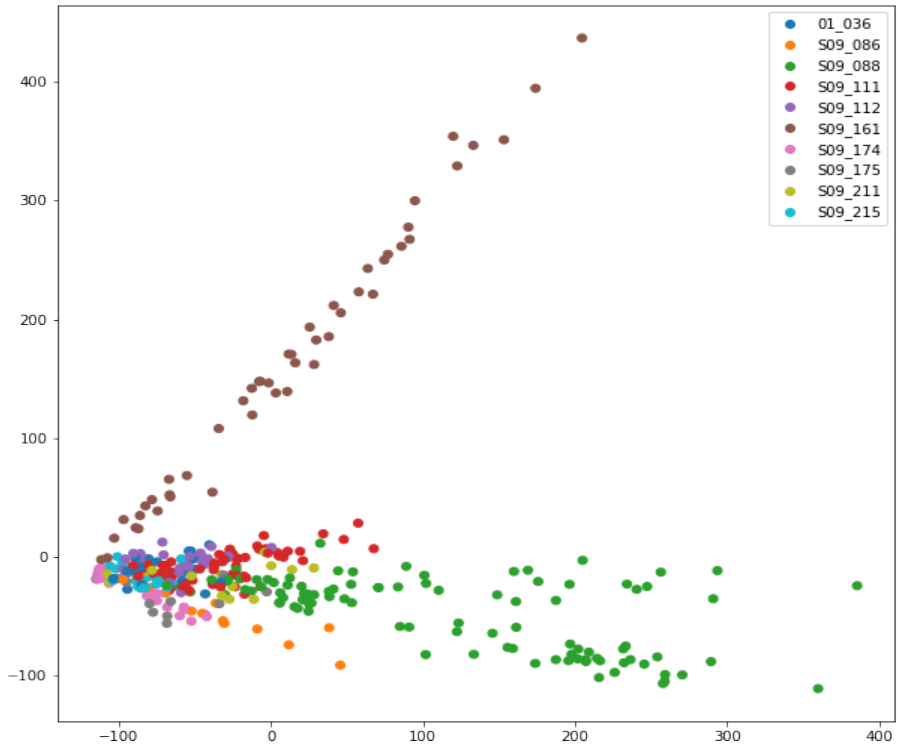


(A) Chimpanzee training set clusters

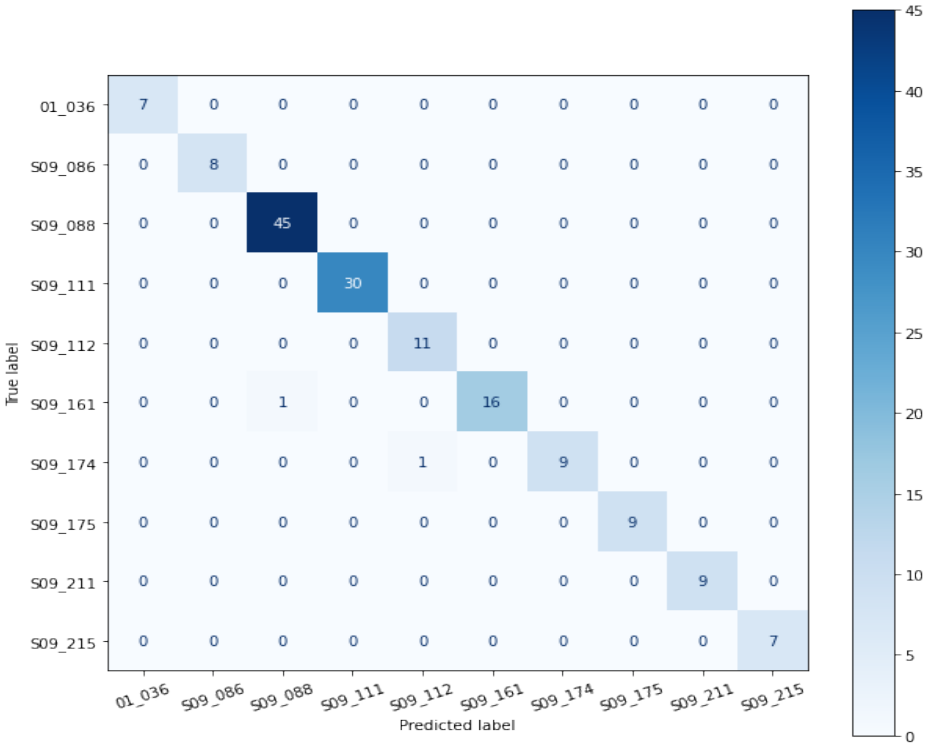


(B) Chimpanzee test set confusion matrix

FIGURE 4.6: Chimpanzee

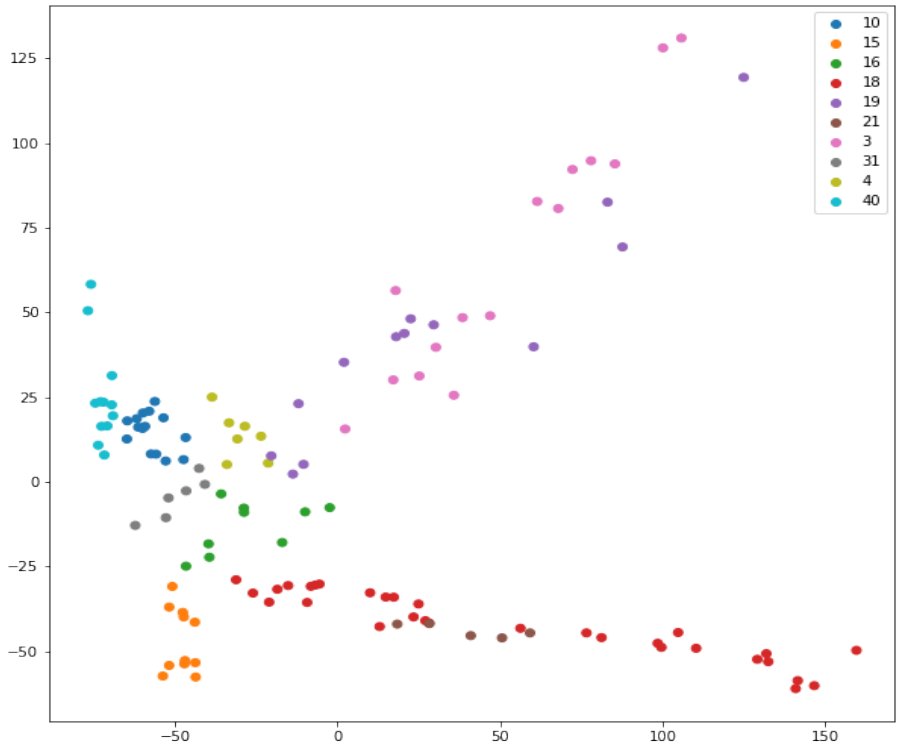


(A) Zebra training set clusters

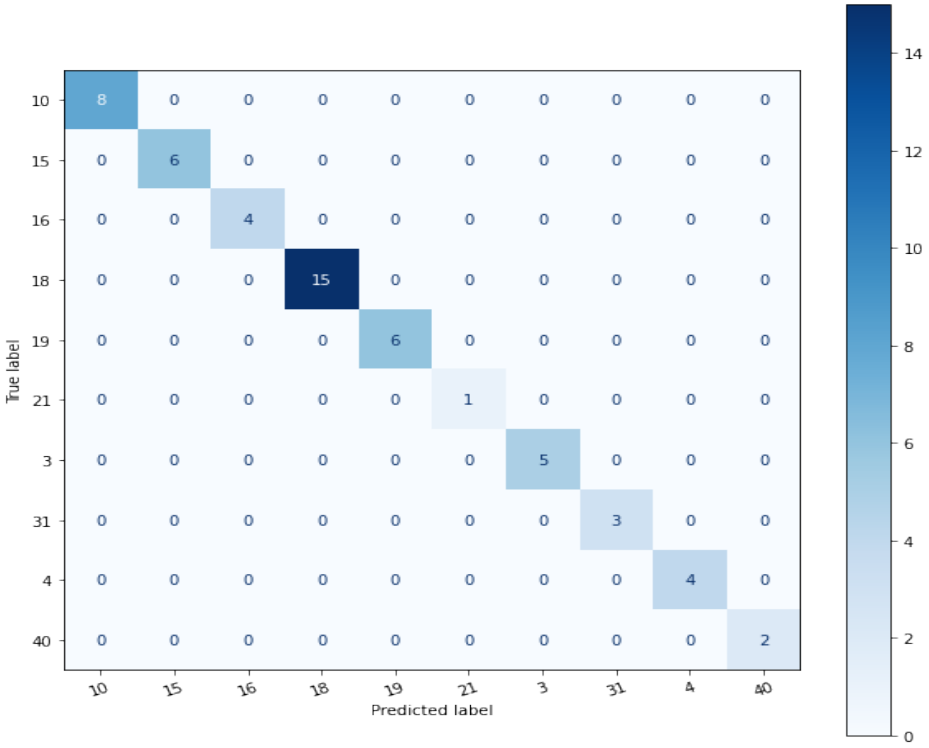


(B) Zebra test set confusion matrix

FIGURE 4.7: Zebra

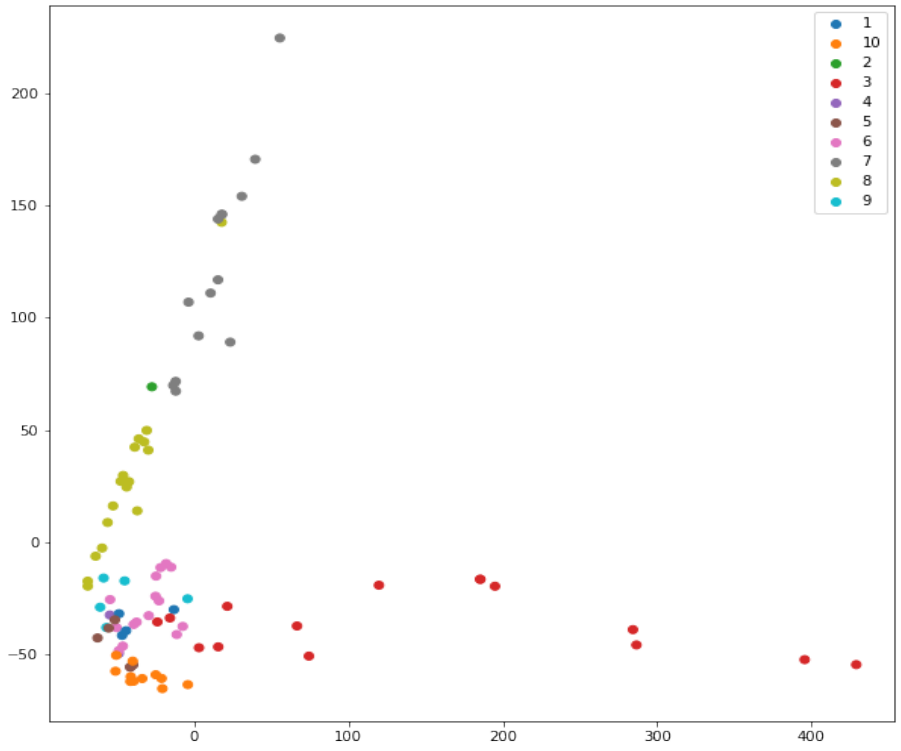


(A) Tiger training set clusters

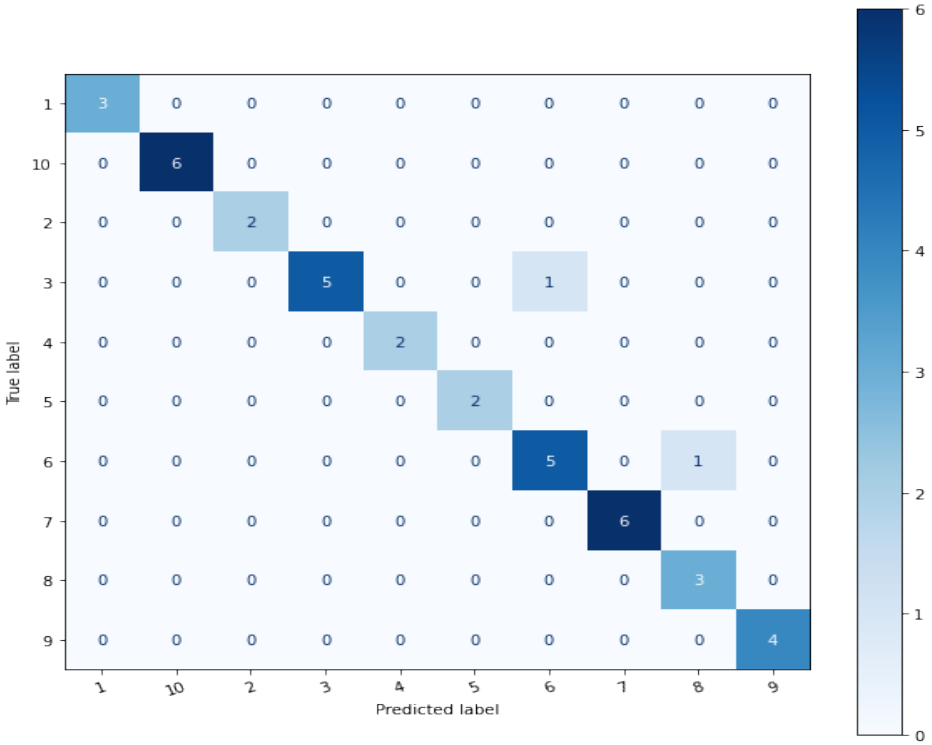


(B) Tiger test set confusion matrix

FIGURE 4.8: Tiger



(A) Nyala training set clusters



(B) Nyala test set confusion matrix

FIGURE 4.9: Nyala

Chapter 5

Findings and Discussions

5.1 Introduction

This chapter discusses the experimental results presented in Chapter 4. The first section gives an overview of which loss function is better for individual wild animal re-identification. The second section draws comparisons between the results we presented with the best results obtained by other researchers. The last section concludes the chapter.

5.2 Loss Function Comparisons

The loss functions we studied in the current work are Proxy-NCA and triplet loss. Our results depict that the triplet loss function performs better than Proxy-NCA. However, we observed that the performance from triplet loss models is not significantly different from the performance of the same model trained on the Proxy-NCA loss function. In the chimpanzees dataset, the VGG-11 model trained on triplet loss obtained Recall@1 of 79.0% while the same network trained on Proxy-NCA obtained 78.9% Recall@1. A study was done by Musgrave *et al.* [109] reveals that pair-based loss functions, like the contrastive loss, performed better than Proxy-NCA loss in some datasets. The idea of class-aware loss functions, specifically the Proxy-NCA, was aimed at substituting the pair-based loss functions because of superior performance [91].

Our results demonstrate that superior model performance gains may be coming from other tuned model parameters because the loss functions in isolation do not bring significant improvements in model performance. We studied seven datasets in five different neural network architectures, and the trend we observed is consistent; there is no significant performance gain when we compare triplet loss and Proxy-NCA.

5.3 Current Work vs Benchmark Results per Dataset: Recall @1

Figure 4.1 highlights the results we got in our experiments and the best results obtained by other researchers, the previous research results are plotted in the red bar.

5.3.1 Panda Dataset

For the panda dataset, we were able to replicate the results obtained by Chen *et al.* 2020 [53], our VGG-11 triplet loss model achieved 91.2% Recall@1 while Chen *et al.* [53] achieved 92.1%. We achieved similar results on the panda dataset without the need to do extensive image pre-processing as done by Chen *et al.* [53]. The significance of our result is that we remove the computation overhead of extensive pre-processing of the panda face images before we can train our re-identification models, and this does not suffer a significant loss in model performance. Also, removing extensive pre-processing in wild animal re-identification is important to preserve the uncontrolled environment that wild animals live in, and still build models that can reliably re-identify individuals in the animal population.

5.3.2 Tiger Dataset

In the tiger dataset, we obtained better results compared to the best result we found in the literature. Our VGG-11 triplet loss obtained 88.9%, VGG-11 Proxy-NCA (87.0%) and ResNet-18 triplet loss obtained 87.1%. Schneider *et al.* [126] got 86.3% from a ResNet-50 model pre-trained on ImageNet data. Our results show that some neural network backbones can achieve better results on a particular dataset; therefore, investigating more than one neural network in an experiment has the advantage of giving a wider scope in the search for better models.

5.3.3 Zebra Dataset

Similar to the tiger dataset, we found best performing models in obtained best results in the zebra dataset than results reported by Van Zyl *et al.* [10]. All our models performed better. However, the VGG-11 and ResNet-18 models achieved the best results: 94.6% and 94.8%, respectively. Our work has set a new benchmark result in the zebra dataset. There is, however, more

work that can be done to further improve the re-identification accuracy, and we only experimented with VGG, ResNet, DenseNet, and a combination of two-loss functions, namely: triplet loss and Proxy-NCA. It is possible that a different neural network backbone can do better. Searching for all possible neural network backbones was outside the scope of the current research, as more time would be required.

5.3.4 Nyala Dataset

The nyala dataset was found to be the most difficult dataset [10]. Our results confirm this assertion. We could not surpass the performance achieved by Van Zyl *et al.* [10]. The best model for the nyala dataset is VGG-19 that obtained 72.3% Recall@1. Reasons that explain why the nyala dataset is difficult would need to be concluded by further research. Further research may include experimenting with other neural network backbones, loss functions, and data pre-processing that were not considered in the current work.

5.3.5 Chimpanzees Dataset

Even with the chimpanzees' dataset, we only sightly outperformed the best results from prior research. An observation we made on the chimpanzee experiments is that all the models trained on triplet loss got almost the same results, ranging between 77.9% and 79.0%. Just like the nyala dataset, the chimpanzee data needs further research to find better-performing models and experimental designs.

5.3.6 Lion Dataset

This is the smallest dataset compared to all the other datasets we studied. VGG-19 gave us the best performance, Recall@1 of 71.3%. Similar to the chimpanzees' dataset, the range in performance of our models is between 67% and 71%.

There is a trend that we observe in our results. Our models performed better in the tiger, panda, and zebra datasets. The discriminating features in this dataset are bold stripes for zebra and tigers, while for the panda datasets are spots on the panda's faces. However, for lions, chimpanzees, and nyala, there are no bold discriminating features in the faces or body flanks. This can partly explain the performances we obtained from these datasets. Collecting more data for these datasets may improve model performance.

5.4 Current Work: Mean Average Precision at R

In the current work, we have introduced the mean average precision at R (MAP@R) model performance measurements. There is no existing literature where researchers have reported on the MAP@R metric for the datasets we used in our experiments. Musgrave *et al.* [109] measured this metric for the Cars196 [132] dataset, the CUB200 [133] dataset and the Stanford Online Products [134]. The MAP@R for all these datasets was below 48%. The reported MAP@R Stanford Online Products is 47.4%, while for Cars196 the reported MAP@R is 29.9% and for CUB200 dataset the reported MAP@R is 37.5%. We cannot compare our MAP@R results with these because of the different datasets we used. However, we can demonstrate that more work still needs to be done to improve models MAP@R. For all our datasets, the MAP@R we obtained ranged between 9.8% and 32.0%.

The mean average precision at R and the model performance metrics like Recall@1 answers two different questions in information retrieval. Considering ten retrieved images, the Recall@1 checks, if the first image retrieved is a true positive (is the most relevant retrieved image ranked at the top?), then Recall@1 is 100%. However, MAP@R checks what proportion of the ten retrieved images are true positives. It is for these differences that we cannot discard the Recall@1 results because the MAP@R results are lower.

5.5 Additional Classification Experiments: VGG-11 with a 10 Class Softmax Activation Function

The principal component analysis plots show that the training set in our datasets depicted that the individual classes are forming distinct clusters. The chimpanzees' cluster plots in Figure 4.6a show clusters with minimal overlap. Since the test set was drawn from the distribution depicted by the clusters, it is not surprising that the confusion matrix Figure 4.6b has very few type-1 and type-2 errors. Similar to the chimpanzees, the panda clusters in Figure 4.5a and the confusion matrix in Figure 4.5b demonstrate a plausible class separability with few type-1 and type-2 errors. The same pattern was observed with the zebra dataset clusters in Figure 4.7a and the confusion matrix in Figure 4.7b.

With the lion clusters in Figure 4.4a and tiger clusters in Figure 4.8a there is class separability. However, the majority of the classes are spread out, and this creates a significant overlap amongst the classes. This overlap can be

attributed to the fact that these datasets contain fewer samples per individual. Classifying the test data points produced good results, as shown in the confusion matrix for these datasets. Figure 4.4b and Figure 4.8b show the confusion matrix for lion test set and tiger test respectively with minimum type-1 and type-2 errors.

These results demonstrate that the classification approach can give good performance in the re-identification of individuals in the wild animal population. The f-1 scores in Table 4.5 for all the individual re-identification are greater than 95% if the number of different individuals is known. The challenge with the classification approach is that, in the wild, knowing the total number of individuals in a wild animal population is difficult to attain. Wild animals roam around spaces, and observing new individuals is inevitable.

5.6 Conclusion

Our results on the effect of changing a loss function from a pair-based loss function to a class-aware loss function in wild animal re-identification did not show improvements. We have kept all parameters constant and changed the loss function both triplet loss and Proxy-NCA achieved similar results. Even though the majority of the better performance was achieved by the models that were trained on triplet loss function with semi-hard negative pair mining, the difference in performance with the Proxy-NCA models was not significant.

The second part of our research was to determine the best architecture per dataset. This section compares our results with the best performance found in literature in a particular dataset. For the chimpanzees, the nyala, and zebra datasets, we were able to outperform the benchmark results.

The last part of our research was to do classification experiments where in each population, there is a known number of different individuals. This approach has a limitation in that the number of individuals should be known. In wildlife, this can be hard to attain. The nature of wild animals is that they roam wide areas, and observing new individuals in a specific colony is inevitable. The other limitation of classification is that the trained model cannot be used when a new class joins the population. The classification model will need to be trained again. This limitation is addressed by similarity learning, and this is the reason why the current work has focused mainly on similarity learning as opposed to classification.

Chapter 6

Future Direction and Conclusion

6.1 Summary of Research

Successful and robust individual wild animal re-identification using non-invasive methods like image analysis can promote accurate reporting on wild animal population [135]. The automated re-identification can potentially reduce human errors in counting and re-identifying missing individuals in the endangered wildlife population. The non-invasive methods of re-identification also ensure the animal's health and the animal's natural habitat is preserved. This is in contrast to the invasive methods like the implantation of a tracking device under the animal skin which can alter the host animal's physiology and affect the host animals' reproductive system [136]. Our study not only shows that re-identification of lions can be done using lion face images but also that improved model performance can be achieved in animal re-identification through the search for the best-suited neural network backbone and loss functions.

6.2 Summary of Results

In contrast to our expectations that class-aware loss functions, like the Proxy-NCA, achieve better results compared to the pair-based loss functions like triplet loss. The difference in performance between triplet loss and Proxy-NCA models is not significantly large. However, using the Proxy-NCA loss function simplifies the experiment design because training pairs are randomly selected. The triplet loss experiments need to be coupled with effective training pairs sampling techniques like semi-hard negative pair mining. This extra step before training the models adds some computation overhead.

6.3 Future Work

There is work that needs to be done to improve the performance in the re-identification of individual lions, nyalas, and chimpanzees. We did not do parameter tuning for all the models that we trained, and possible performance improvements can come from finding the optimal learning rates during training. We used one loss optimizer function, Adam [137], for all our experiments because our experimental variable was the effects of changing the loss function on model performance. We kept all other parameters constant. Wang *et al.* [138] experimented with different optimizer functions on datasets and found out that certain optimizer functions are best suited for some datasets.

6.4 Contributions

We were able to set new best results in the re-identification of tiger [11], zebra [49] and chimpanzees [51]. The top performances were achieved by VGG-11, ResNet-18, and DenseNet-201. We find that VGG-11 is generally the best suitable model for the re-identification task in the datasets we studied. The VGG-11 models achieved the best or second-best result for the majority of the datasets we studied.

Bibliography

[1] N. Dlamini and T. L. van Zyl, "Automated identification of individuals in wildlife population using siamese neural networks", in *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, IEEE, 2020, pp. 224–228.

[2] ——, "Comparing class-aware and pairwise loss functions for deep metric learning in wildlife re-identification", *Sensors*, vol. 21, no. 18, p. 6109, 2021.

[3] F. Mutua, A. Kihara, J. Rogena, N. Ngwili, G. Aboge, J. Wabacha, and B. Bett, "Piloting a livestock identification and traceability system in the northern tanzania–narok–nairobi trade route", *Tropical animal health and production*, vol. 50, no. 2, pp. 299–308, 2018.

[4] A. K. Jain, A. Ross, S. Prabhakar, *et al.*, "An introduction to biometric recognition", *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, 2004.

[5] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces", in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1991, pp. 586–591.

[6] B. G. Weinstein, "A computer vision for animal ecology", *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545, 2018.

[7] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling", in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 32–39.

[8] G. K. Verma and P. Gupta, "Wild animal detection using deep convolutional neural network", in *Proceedings of 2nd international conference on computer vision & image processing*, Springer, 2018, pp. 327–338.

[9] S. Schneider, G. W. Taylor, and S. Kremer, "Deep learning object detection methods for ecological camera trap data", in *2018 15th Conference on Computer and Robot Vision (CRV)*, IEEE, 2018, pp. 321–328.

[10] T. Van Zyl, M Woolway, and B Engelbrecht, "Unique animal identification using deep transfer learning for data fusion in siamese networks", in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, IEEE, 2020, pp. 1–6.

[11] S. Li, J. Li, H. Tang, R. Qian, and W. Lin, "ATRW: A Benchmark for Amur Tiger Re-identification in the Wild", *arXiv e-prints*, arXiv:1906.05586, arXiv:1906.05586, Jun. 2019. arXiv: 1906.05586 [cs.CV].

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[14] D. D. Zhang, *Automated biometrics: Technologies and systems*. Springer Science & Business Media, 2013, vol. 7.

[15] R. Clarke, "Human identification in information systems: Management challenges and public policy issues", *Information Technology & People*, vol. 7, pp. 6–37, 1994.

[16] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to biometrics*. Springer Science & Business Media, 2013.

[17] I. Buciu and A. Gacsadi, "Biometrics systems and technologies: A survey", *International Journal of Computers Communications & Control*, vol. 11, no. 3, pp. 315–330, 2016.

[18] W. Kabir, M. O. Ahmad, and M. Swamy, "A multi-biometric system based on feature and score level fusions", *IEEE Access*, vol. 7, pp. 59 437–59 450, 2019.

[19] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems", *Journal of medical systems*, vol. 37, no. 5, p. 9972, 2013.

[20] J. J. Hathaliya, S. Tanwar, S. Tyagi, and N. Kumar, "Securing electronics healthcare records in healthcare 4.0: A biometric-based approach", *Computers & Electrical Engineering*, vol. 76, pp. 398–410, 2019.

[21] M. Tistarelli and C. Champod, *Handbook of biometrics for forensic science*. Springer, 2017.

[22] C Shanahan, B Kernan, G Ayalew, K McDonnell, F Butler, and S Ward, "A framework for beef traceability from farm to slaughter using global standards: An irish perspective", *Computers and electronics in agriculture*, vol. 66, no. 1, pp. 62–69, 2009.

[23] F. Dabbene and P. Gay, "Food traceability systems: Performance evaluation and optimization", *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 139–146, 2011.

[24] K. G. Buk, V. C. van der Merwe, K. Marnewick, and P. J. Funston, "Conservation of severely fragmented populations: Lessons from the transformation of uncoordinated reintroductions of cheetahs (acinonyx jubatus) into a managed metapopulation with self-sustained growth", *Biodiversity and conservation*, vol. 27, no. 13, pp. 3393–3423, 2018.

[25] X. Dou and J. Day, "Human-wildlife interactions for tourism: A systematic review", *Journal of Hospitality and Tourism Insights*, vol. 3, no. 5, pp. 529–547, 2020.

[26] F. Mancini, "Managing the wildlife tourism commons", PhD thesis, University of Aberdeen, 2019.

[27] T. Wyatt, K. Johnson, L. Hunter, R. George, and R. Gunter, "Corruption and wildlife trafficking: Three case studies involving asia", *Asian Journal of Criminology*, vol. 13, no. 1, pp. 35–55, 2018.

[28] S. E. McMillan, C. Dingle, J. A. Allcock, and T. C. Bonebrake, "Exotic animal cafes are increasingly home to threatened biodiversity", *Conservation Letters*, vol. 14, e12760, 2020.

[29] R. A. Sollund, *The crimes of wildlife trafficking: Issues of justice, legality and morality*. Routledge, 2019.

[30] M. Neary and A. Yager, "Methods of livestock identification", 2002.

[31] M. Ariff and I Ismail, "Livestock information system using android smartphone", in *2013 IEEE Conference on Systems, Process & Control (ICSPC)*, IEEE, 2013, pp. 154–158.

[32] A. I. Awad, "From classical methods to animal biometrics: A review on cattle identification and tracking", *Computers and Electronics in Agriculture*, vol. 123, pp. 423–435, 2016.

[33] M. Cabanac and S. Aizawa, "Fever and tachycardia in a bird (gallus domesticus) after simple handling", *Physiology & behavior*, vol. 69, no. 4-5, pp. 541–545, 2000.

[34] L. Hou, M. Verdirame, and K. C. Welch Jr, "Automated tracking of wild hummingbird mass and energetics over multiple time scales using radio frequency identification (rfid) technology", *Journal of Avian Biology*, vol. 46, no. 1, pp. 1–8, 2015.

[35] M. L. Gillenson, X. Zhang, A. Muthitacharoen, and P. Prasarnphanich, "I've got you under my skin: The past, present, and future use of rfid technology in people and animals.", *J. Inf. Technol. Manag.*, vol. 30, no. 2, pp. 19–29, 2019.

[36] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John wiley & sons, 2010.

[37] K. Albrecht, "Microchip-induced tumors in laboratory rodents and dogs: A review of the literature 1990–2006", in *2010 IEEE International Symposium on Technology and Society*, IEEE, 2010, pp. 337–349.

[38] A. Jain, L. Hong, and S. Pankanti, "Biometric identification", *Communications of the ACM*, vol. 43, no. 2, pp. 90–98, 2000.

[39] D. L. Borchers, W. Zucchini, and R. M. Fewster, "Mark-recapture models for line transect surveys", *Biometrics*, vol. 54, no. 4, pp. 1207–1220, 1998.

[40] H. J. Skaug and T. Schweder, "Hazard models for line transect surveys with independent observers", *Biometrics*, vol. 55, no. 1, pp. 29–36, 1999.

[41] C. J. Schwarz and W. T. Stobo, "Estimating temporary migration using the robust design", *Biometrics*, vol. 53, no. 1, pp. 178–194, 1997.

[42] T. Burghardt and N. Campbell, "Individual animal identification using visual biometrics on deformable coat patterns", in *5th International Conference on Computer Vision Systems (ICVS)*, 2007.

[43] U. G. Barron, G Corkery, B Barry, F Butler, K McDonnell, and S Ward, "Assessment of retinal recognition technology as a biometric method for sheep identification", *Computers and electronics in agriculture*, vol. 60, no. 2, pp. 156–166, 2008.

[44] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna", *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[45] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8769–8778.

[46] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 456–473.

[47] S. Beery, E. Cole, and A. Gjoka, "The iwildcam 2020 competition dataset.", *CoRR*, vol. abs/2004.10340, 2020. [Online]. Available: https://arxiv.org/abs/2004.10340.

[48] D. Deb, S. Wiper, S. Gong, Y. Shi, C. Tymoszek, A. Fletcher, and A. K. Jain, "Face recognition: Primates in the wild", in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, 2018, pp. 1–10.

[49] M. Lahiri, C. Tantipathananandh, R. Warungu, D. I. Rubenstein, and T. Y. Berger-Wolf, "Biometric animal databases from field photographs: Identification of individual zebra in the wild", in *Proceedings of the 1st ACM international conference on multimedia retrieval*, 2011, pp. 1–8.

[50] A. Polzounov, I. Terpugova, D. Skiparis, and A. Mihai, "Right whale recognition using convolutional neural networks", *arXiv e-prints*, arXiv:1604.05605, arXiv:1604.05605, Apr. 2016. arXiv: 1604.05605 [cs.CV].

[51] A. Freytag, E. Rodner, M. Simon, A. Loos, H. S. Kühl, and J. Denzler, "Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates", in *German Conference on Pattern Recognition*, Springer, 2016, pp. 51–63.

[52] M. Körschens and J. Denzler, "Elpephants: A fine-grained dataset for elephant re-identification", in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, 2019, pp. 263–270.

[53] P. Chen, P. Swarup, W. M. Matkowski, A. W. K. Kong, S. Han, Z. Zhang, and H. Rong, "A study on giant panda recognition based on images of a large proportion of captive pandas", *Ecology and evolution*, vol. 10, no. 7, pp. 3561–3573, 2020.

[54] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance", *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[55] S. Kumar and S. K. Singh, "Visual animal biometrics: Survey", *IET Biometrics*, vol. 6, no. 3, pp. 139–156, 2016.

[56] D. G. Lowe, "Object recognition from local scale-invariant features", in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[57] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns", *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[58] S. Routray, A. K. Ray, and C. Mishra, "Analysis of various image feature extraction methods against noisy image: Sift, surf and hog", in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, 2017, pp. 1–5.

[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[60] C. Finn, J. Duyck, A. Hutcheon, P. Vera, J. Salas, and S. Ravela, "Relevance feedback in biometric retrieval of animal photographs", in *Mexican Conference on Pattern Recognition*, Springer, 2014, pp. 281–290.

[61] Z. Zhang and E. Sejdić, "Radiological images and machine learning: Trends, perspectives, and prospects", *Computers in biology and medicine*, vol. 108, pp. 354–370, 2019.

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

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[64] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size", *arXiv e-prints*, arXiv:1602.07360, arXiv:1602.07360, Feb. 2016. arXiv: 1602.07360 [cs.CV].

[65] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

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

[67] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning", in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.

[68] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[69] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer, "Imagenet training in minutes", in *Proceedings of the 47th International Conference on Parallel Processing*, 2018, pp. 1–10.

[70] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition", in *ICML deep learning workshop*, Lille, vol. 2, 2015.

[71] H. Manack and T. L. Van Zyl, "Deep similarity learning for soccer team ranking", in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, IEEE, 2020, pp. 1–7.

[72] J. Burns and T. L. van Zyl, "Automated music recommendations using similarity learning", in *SACAIR 2020*, 2020, p. 288.

[73] M. Z. Variawa, T. L. Van Zyl, and M. Woolway, "Transfer learning and deep metric learning for automated galaxy morphology representation", *IEEE Access*, 2022.

[74] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843.

[75] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1763–1771.

[76] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking", in *European conference on computer vision*, Springer, 2016, pp. 850–865.

[77] E. Hoffer and N. Ailon, "Deep metric learning using triplet network", in *International Workshop on Similarity-Based Pattern Recognition*, Springer, 2015, pp. 84–92.

[78] T. P. Nguyen, C. C. Pham, S. V.-U. Ha, and J. W. Jeon, "Change detection by training a triplet network for motion feature extraction", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 433–446, 2018.

[79] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 403–412.

[80] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey", *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[81] H. Xuan, A. Stylianou, and R. Pless, "Improved embeddings with easy positive triplet mining", in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2474–2482.

[82] M. Sikaroudi, B. Ghojogh, F. Karray, M. Crowley, and H. R. Tizhoosh, "Batch-incremental triplet sampling for training triplet networks using bayesian updating theorem", in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 7080–7086.

[83] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling", in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6450–6458.

[84] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions", *City*, vol. 1, no. 2, p. 1, 2007.

[85] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance", *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[86] J. Ye, "Cosine similarity measures for intuitionistic fuzzy sets and their applications", *Mathematical and computer modelling*, vol. 53, no. 1-2, pp. 91–97, 2011.

[87] T. Kim, I. R. Chen, Y. Lin, A. Y.-Y. Wang, J. Y. H. Yang, and P. Yang, "Impact of similarity metrics on single-cell rna-seq data clustering", *Briefings in bioinformatics*, vol. 20, no. 6, pp. 2316–2326, 2019.

[88] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere", in *International Conference on Machine Learning*, PMLR, 2020, pp. 9929–9939.

[89] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric Learning with Adaptive Density Discrimination", *arXiv e-prints*, arXiv:1511.05939, arXiv:1511.05939, Nov. 2015. arXiv: 1511.05939 [stat.ML].

[90] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5022–5030.

[91] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368.

[92] J. Goldberger, G. E. Hinton, S. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis", *Advances in neural information processing systems*, vol. 17, pp. 513–520, 2004.

[93] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization", in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[94] E. W. Teh, T. DeVries, and G. W. Taylor, "Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis", in *European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 448–464.

[95] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1861–1870.

[96] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition", in *International conference on machine learning*, 2014, pp. 647–655.

[97] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[98] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training", in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 4918–4927.

[99] S. J. Pan and Q. Yang, "A survey on transfer learning", *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[100] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng, "Zero-shot learning through cross-modal transfer", in *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, 2013, pp. 935–943.

[101] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–37, 2019.

[102] M. Rezaei and M. Shahidi, "Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review", *Intelligence-based medicine*, p. 100 005, 2020.

[103] J. Lu, P. Gong, J. Ye, and C. Zhang, "Learning from Very Few Samples: A Survey", *arXiv e-prints*, arXiv:2009.02653, arXiv:2009.02653, Sep. 2020. arXiv: 2009.02653 [cs.LG].

[104] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal, "Improved few-shot visual classification", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 493–14 502.

[105] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning", *Advances in Neural Information Processing Systems*, vol. 30, pp. 4077–4087, 2017.

[106] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning", in *International Conference on Machine Learning*, PMLR, 2020, pp. 8242–8252.

[107] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces", in *European Conference on Computer Vision*, Springer, 2020, pp. 741–757.

[108] J.-H. Kim, B.-G. Kim, P. P. Roy, and D.-M. Jeong, "Efficient facial expression recognition algorithm based on hierarchical deep neural network structure", *IEEE access*, vol. 7, pp. 41 273–41 285, 2019.

[109] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check", in *European Conference on Computer Vision*, Springer, 2020, pp. 681–699.

[110] D. T. Bolger, T. A. Morrison, B. Vance, D. Lee, and H. Farid, "A computer-assisted system for photographic mark–recapture analysis", *Methods in Ecology and Evolution*, vol. 3, no. 5, pp. 813–822, 2012.

[111] M. J. Kelly, "Computer-aided photograph matching in studies using individual identification: An example from serengeti cheetahs", *Journal of Mammalogy*, vol. 82, no. 2, pp. 440–449, 2001.

[112] D. Schofield, A. Nagrani, A. Zisserman, M. Hayashi, T. Matsuzawa, D. Biro, and S. Carvalho, "Chimpanzee face recognition from videos in the wild using deep learning", *Science Advances*, vol. 5, no. 9, eaaw0736, 2019.

[113] J. Guo, H. He, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha, *et al.*, "Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing.", *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.

[114] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[115] S. Li and W. Deng, "Deep facial expression recognition: A survey", *IEEE Transactions on Affective Computing*, pp. 1–1, 2020. DOI: 10.1109/TAFFC.2020.2981446.

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

[117] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv e-prints*, arXiv–2010, 2020.

[118] N. Park and S. Kim, *How do vision transformers work?*, 2022. DOI: 10.48550/ARXIV.2202.06709. [Online]. Available: https://arxiv.org/abs/2202.06709.

[119] M. Sun, H. Ma, G. Kang, Y. Jiang, T. Chen, X. Ma, Z. Wang, and Y. Wang, "VAQF: fully automatic software-hardware co-design framework for low-bit vision transformer", *CoRR*, vol. abs/2201.06618, 2022. arXiv: 2201.06618. [Online]. Available: https://arxiv.org/abs/2201.06618.

[120] S. Shahinfar, P. Meek, and G. Falzon, "How many images do i need? understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring", *Ecological Informatics*, vol. 57, p. 101 085, 2020.

[121] T. Burghardt, J. Calic, and B. T. Thomas, "Tracking animals in wildlife videos using face detection.", in *EWIMT*, 2004.

[122] R. C. Buckley, J. G. Castley, F. de Vasconcellos Pegas, A. C. Mossaz, and R. Steven, "A population accounting approach to assess tourism contributions to conservation of iucn-redlisted mammal species", *PloS one*, vol. 7, no. 9, pp. 1–8, 2012.

[123] E.-J. Wagenmakers, R. Wetzels, D. Borsboom, H. L. van der Maas, and R. A. Kievit, "An agenda for purely confirmatory research", *Perspectives on Psychological Science*, vol. 7, no. 6, pp. 632–638, 2012.

[124] A. Genç and H. K. Ekenel, "Cross-dataset person re-identification using deep convolutional neural networks: Effects of context and domain adaptation", *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5843–5861, 2019.

[125] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection", in *Ijcai*, Montreal, Canada, vol. 14, 1995, pp. 1137–1145.

[126] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 44–52.

[127] D. Deb, S. Wiper, S. Gong, Y. Shi, C. Tymoszek, A. Fletcher, and A. K. Jain, "Face recognition: Primates in the wild", in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, 2019, pp. 1–10.

[128] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library", *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[129] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. Reboucas Filho, "Performance analysis of google colaboratory as a tool for accelerating deep learning applications", *IEEE Access*, vol. 6, pp. 61 677–61 685, 2018.

[130] L. Frank, "Living with lions: Lessons from laikipia", *Smithsonian Contributions to Zoology*, pp. 73–83, 2011.

[131] N. Craswell, "R-precision", in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 2453–2453, ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_486. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_486.

[132] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization", in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[133] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset", 2011.

[134] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4004–4012.

[135] R. Steenweg, M. Hebblewhite, R. Kays, J. Ahumada, J. T. Fisher, C. Burton, S. E. Townsend, C. Carbone, J. M. Rowcliffe, J. Whittington, *et al.*, "Scaling-up camera traps: Monitoring the planet's biodiversity with networks of remote sensors", *Frontiers in Ecology and the Environment*, vol. 15, no. 1, pp. 26–34, 2017.

[136] P. Rotter, B. Daskala, and R. Compano, "Rfid implants: Opportunities and and challenges for identifying people", *IEEE Technology and Society Magazine*, vol. 27, no. 2, pp. 24–32, 2008.

[137] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *arXiv e-prints*, arXiv:1412.6980, arXiv:1412.6980, Dec. 2014. arXiv: 1412.6980 [cs.LG].

[138] Y. Wang, J. Liu, J. Mišić, V. B. Mišić, S. Lv, and X. Chang, "Assessing optimizer impact on dnn model sensitivity to adversarial examples", *IEEE Access*, vol. 7, pp. 152 766–152 776, 2019.