DEPARTMENT OF COMPUTER SCIENCE

Recognition of Individual Cattle from Depth Imagery Using Deep Metric Learning

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

Tuesday 2nd May, 2023

Abstract

The exploitation of biometrics for the individual identification of cattle is an active and important research area within the sphere of agricultural digitalisation. Previous work successfully uses deep metric learning to discriminate individual Holstein-Friesian cattle by their black and white coat patterns [7]. However, cattle in beef herds typically exhibit little-to no discriminating coat patterns with predominantly black or brown colourings [50]; we explore the potential for depth imagery to be used for discriminating such individuals. Previous unpublished work identifying cows in the RGBDCows2020 dataset quoted an accuracy of **60.62%** in this task. Through experiments with loss functions, data augmentation and network adaptation we find that **accuracies up to 75% can be reached.** We propose a network combining a ResNet backbone with a spatial context module, which projects examples into an embedded space facilitating K-nearest-neighbour classification, with full details given in figure 1. To accompany the model, we present the preparation of CowDepth2023, a new dataset fit for our research purpose which achieved accuracies within the range of 69.54% to 97.60%, dependent on temporal bias removal. We discuss where our findings and research stand within the sphere of animal biometric research, and hence identify paths for future research both in the veterinary and image recognition disciplines as well as conducting our own **novel research into what makes the spinal patterns of cattle individually unique**. Relevant source code, model weights, and a suite of utility functions are available publicly [1], and datasets used for training can be provided upon request.

My main five contributions achieved through this project are:

1. Curation of a **new dataset** fit for our research purpose
2. Purpose design of **network architecture and loss function**, including the evaluation of an additional spatial context module
3. Evaluation of **data augmentation, loss functions, and model backbones** and their effects on model performance
4. Execution of **ablation studies** to facilitate interpretation and discussion of quantitative results
5. Spatial localisation of individuality between classes using a Grad-CAM algorithm to facilitate discussion with **veterinary implications**.

[1] Accompanying source code [redacted]

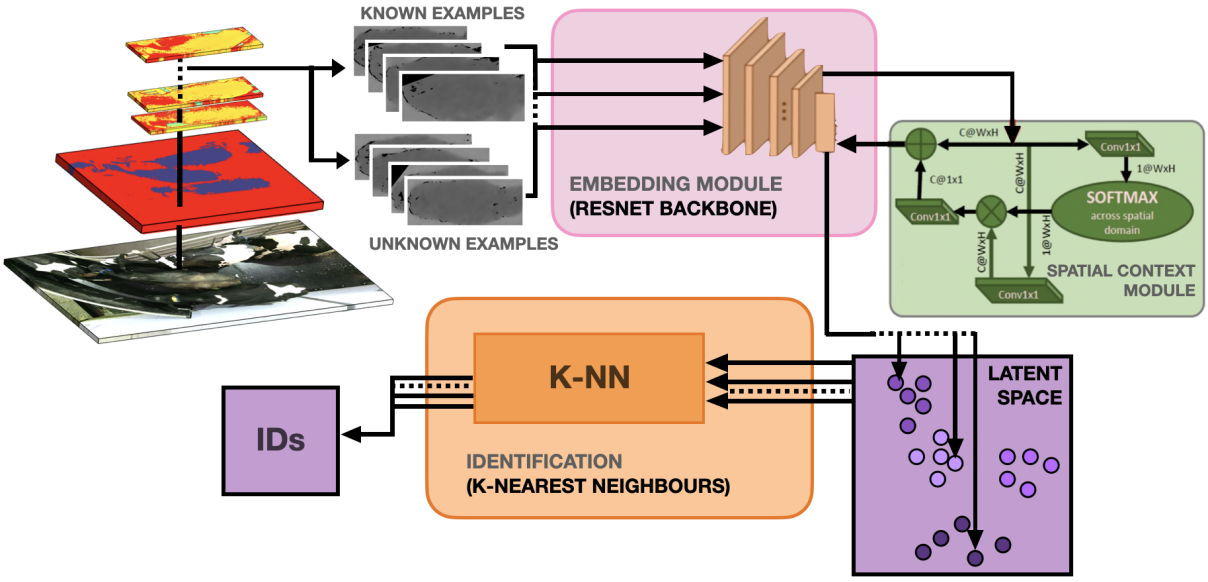


Figure 1: **Identification Network Pipeline Overview.** Top down-imagery is segmented and classified into individual cattle depth images. The network takes these depth images as its input, accompanied by a supplementary json file which designates the specific individuals to be used either as known or unknown entities. This classification corresponds to whether the individuals have been previously observed during the training phase or are entirely unseen. A ResNet-driven module for dimensionality reduction then projects each example onto a latent space embedding, where each clusters according to the features deemed of interest in terms of identification by the model. This process is aided by a Spatial Context Module (SCM), which is driven by a self-attention mechanism for emphasising the most important elements of a feature (SCM figure component taken from Yang et al. [68]). The obtained clusters may be classified utilizing a lightweight method such as K-Nearest Neighbours, resulting in the determination of unique identifiers for the cattle. As such, previously unseen data points may be mapped onto this identification space, such that they form distinct clusters that can be differentiated from previously observed individuals based on their distance from others in the space. Images used in the figure are from the CowDepth2023 dataset.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

[redacted] Tuesday 2nd May, 2023

Contents

List of Figures

List of Tables

Ethics Statement

This project did not require ethical review, as determined by my supervisor, Tilo Burghardt.

Supporting Technologies

This project would not have been possible without the use of third-party resources, of both hardware and software components. A non-exhaustive list is given below, highlighting the main components influencing the project:

Hardware

- An **Intel RealSense Depth Camera D435** was used to capture images in the RGB-DCows2020 dataset.
- A **Kinect sensor** was used to capture images in the CowDepth2023 dataset.
- GPU compute was used for the training and evaluation of models on the University of Bristol's supercomputer **BlueCrystal Phase 4**.

Software

All of the project implementation was carried out in Python (Version 3.10.9). A non-exhaustive list of the packages and libraries used are given below.

- Image processing packages such as **opencv** (4.7.0.68), **Pillow** (9.3.0), **scikit-image** (0.19.3), and **Albumentations** (1.3.0) were used throughout project execution.
- All deep learning involved in the project was implemented in **PyTorch** (1.13.1).
- Networks were built using **ResNet** backbones with weights pre-trained on **ImageNet**.
- Data handling and plotting functions were used from libraries such as, but not limited to **Numpy** (1.23.5), **Pandas** (1.5.3), **Matplotlib** (3.7.1), and **Seaborn** (0.12.2).
- Embedding visualisations were produced by adapting the pre-defined t-SNE function given by **scikit-learn** (1.2.1).

Notation and Acronyms

FAD	:	Foreign Animal Disease
PLF	:	Precision Livestock Farming
FBO	:	Food Business Organisation
FMD	:	Foot-and-Mouth Disease
BSE	:	Bovine Spongiform Encephalopathy
ASIFT	:	Affine Scale-invariant Feature Transform
RGB	:	Red-Green-Blue colour model
CNN	:	Convolutional Neural Network
KNN	:	K-Nearest Neighbours (clustering algorithm)
RFID	:	Radio Frequency Identification
t-SNE	:	t-distributed Stochastic Neighbor Embedding
SCM	:	Spatial Context Module
	:	
$d_{euc}(x_1, x_2)$	:	the Euclidean distance between x_1 and x_2
$d_{cos}(x_1, x_2)$	:	the cosine difference between x_1 and x_2
$d_{combined}(x_1, x_2)$	:	a combination of cosine difference and Euclidean distance between x_1 and x_2

Chapter 1

Introduction

1.1 Moo-tivation

The agricultural sector has been benefiting from the continuous development of technology, with aims such as cost reduction, increasing efficiency of practises, and improving welfare of livestock [10]; one such avenue of digitalisation is the automation of individual cattle identification [9].

1.1.1 Importance of automated identification

It is widely acknowledged that the individual identification of cattle is a challenge of great significance in cattle farming. The development of automated approaches affords great promise to solve both logistical and economic challenges, improving not only commercial viability but the welfare of the animals in question.

Tracing of disease. Sophisticated animal identification systems are essential to the tracking and ultimately the elimination of FADs (foreign animal diseases), through reducing the time taken to locate infected animals and hence limiting opportunities for exposure to other individuals [16]. The success of disease prevention programs therefore depends heavily on the efficiency and reliability of the identification system to facilitate rapid tracing. Economic impacts of such systems could be drastic, particularly evidenced during the catastrophic foot-and-mouth disease (FMD) and bovine spongiform encephalopathy (BSE) outbreaks in the UK [18]. The losses in the US beef industry following restrictions due to BSE further motivate this statement; Coffey et al. estimated losses in 2004 alone to be between $3.2 billion and $4.7 billion [13]. Clearly, an identification system to facilitate the intervention of such outbreaks could offer significant economic benefits.

Yield monitoring. On dairy farms, there would be significant utility in an automated identification system which would bring attention to cows of interest as they pass through the milking parlour. Such individuals might include those which are pregnant, on medications, or are currently being treated for diseases such as mastitis [15] or foot ulcers [19]. Mastitis in particular is a disease affecting the udder which is of high incidence and extensive costs [42]. Interventions such as discarding milk produced by infected individuals rely on the identification and tracking of such individuals, which could be facilitated by an automated identification system mounted above a milking parlour.

Animal welfare. For moral, ethical, and even economic reasons, it should be of high priority that animals are reared such that they are healthy and happy in their social environment [61]. Precision Livestock Farming (PLF) aims to develop technologies to both evaluate animal-based welfare indicators and to subsequently address the welfare issues identified [57]. Automated identification systems are often required for such systems to reduce the time taken to track the whereabouts of, for example, lame cows which require intervention.

1.1.2 Current regulations and standard methods

Cattle identification regulations. The Food Standards Agency mandates that Food Business Organisations (FBOs) must have systems and procedures in place to ensure the traceability of food-producing

animals, specifying specifically that all cattle born after 1st January 1998 must be tagged with an approved tag in each ear [21]. RFID is a current standard for tagging which enables tracking of individuals automatically and wirelessly; a basic RFID system consists of a label, antenna, and a processing system [59]. Implementation of this method can be expensive and time-consuming, with costs for tagging a herd of 200 reaching up to $1000 [3]. This can become even more costly due to loss of tags, with farmers reporting that up to 80% of cows will lose their tags at some stage in their lives [3]. The negative impact of economic loss comes coupled with the physical impact to cows' ears; research into welfare implications of identification of cattle by ear tags showed that only 2.9% of ears with metal tags were free from long-term damage [28].

Current method overview. Other permanent methods of identification are shown in figure 1.1, each of which has shortcomings and drawbacks. Classical methodologies (figure 1.1, left) have been employed extensively for prolonged periods of time, and have facilitated extensive research. However, these techniques are susceptible to challenges such as data loss, distortions, and breakage, in addition to concerns regarding the welfare of animals [9]. Clearly, this motivates the need for a modern hands-off system which can be used on a large scale to reliably track individuals. The right branch of figure 1.1 gives examples of emerging methods which exploit recent machine learning advances to identify cows using biometrics; in section 1.1.3 we explore the extent to which these methods are successful and identify where our approach lies within this category of techniques.

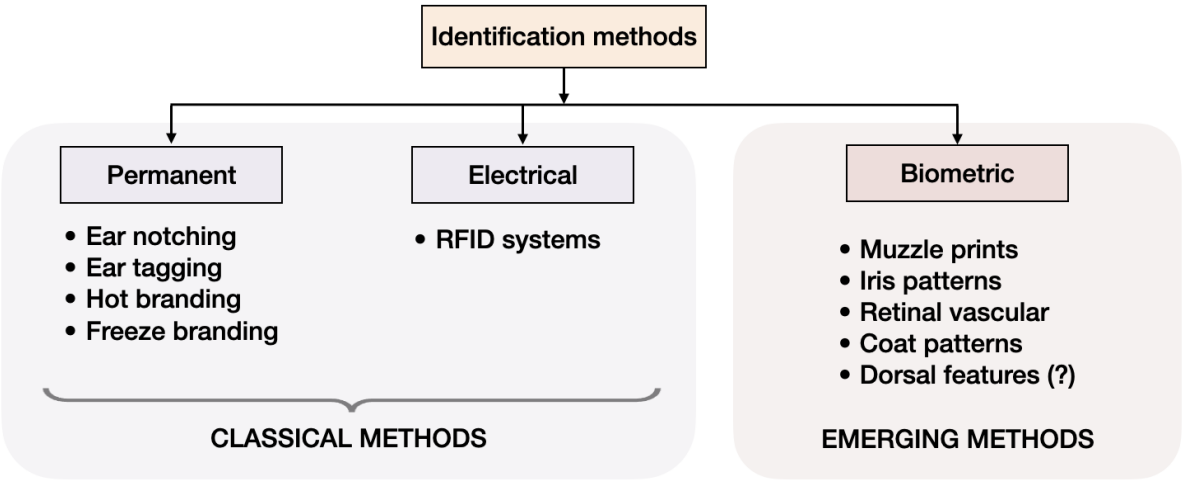
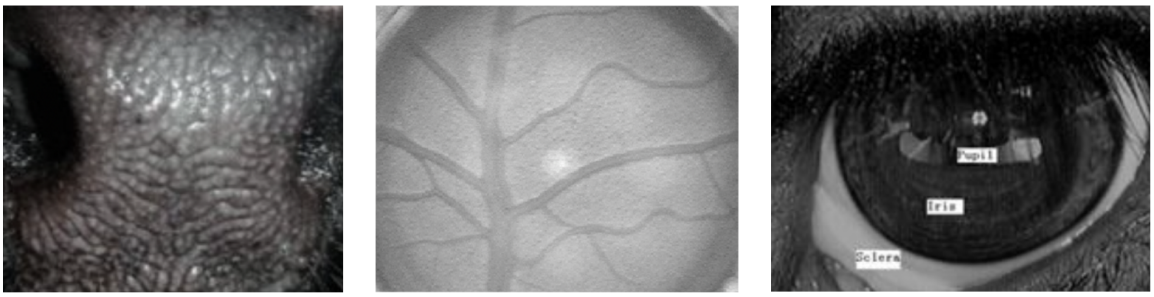


Figure 1.1: **Identification methods categorised as classical and emerging.** The primary traditional techniques for identifying cattle include notching the ears, tattooing/branding the ears, and attaching ear tags with RFID sensors, while deep neural network advances have facilitated recent research into methods using biometrics such as muzzle prints, iris patterns and coat patterns. Our work joins this emergent category using biometrics as an identification method.

1.1.3 Bovine biometrics

Approaches involving features comparable to human fingerprints offer non-invasive methods for identification. Figure 1.2 shows three examples of such biometrics, each of which have been exploited by deep learning systems with varying degrees of accuracy and scalability.

Muzzle imagery. A notable example which has been explored with great success in terms of accuracy metrics is through the use of muzzle imagery, reaching identification accuracies of 98.7% [33]. The nose area of a cow has small protuberances called beads that can be round, oval, or irregularly shaped, as well as elongated grooves and valleys arranged in a specific pattern referred to as ridges. Such patterns are shown in figure 1.2 (a). These features lie in patterns which are unique to each individual, hence providing an opportunity for identification. However, muzzle images are time consuming and difficult to acquire due to the proximity to the animals required for image capture, and many images must be filtered out due to low quality due to movement of animals [32].



(a) - Muzzle (b) - Iris (c) - Retinal vascular

Figure 1.2: **Example images from existing biometric methods of bovine identification.** Viability of each of these methods for identification is well-documented in literature, with each bringing its own strengths and weaknesses. **(a)** shows a muzzle pattern, where beads and ridges form uniquely identifiable patterns which can be learned through deep learning [33]. **(b)** is an image of an iris of a cow. Patterns such as arching ligaments, furrows, ridges, crypts, rings, etc. can be learned to distinguish individuals [35]. **(c)** is a retinal vascular image; the quantity, location, and diameter of iris branches offer information which can be used for identification [66].

Iris and retinal imagery. Two other notable methods of identification via biometrics are through iris [35] and retinal imagery [66], shown in figure 1.2 (b) and (c) respectively. Both methods require a challenging image capture process to perform model inference; the capture of high-quality iris images is difficult not only because of the small size of eyes but because of the high potential of obstructions due to eyelashes, reflections, and other occlusions. Retinal imagery requires precise alignment of the camera setup, which is difficult to achieve with a moving animal, resulting in a lower image quality. Despite these challenges, identification with both methods has been successful in the individual identification task in research settings where image capture conditions can be given careful attention. However, a solution more suited to the agriculture industry might involve image capture from overhead drone footage, motivating the advantages of identification through top-down imagery alone.

Top-down imagery recognition. Previous work (described in detail in section 2.1.1) has had great success in the task of classifying Holstein-Friesian cattle using top-down RGB imagery while struggling to tell apart those which lack distinctive coat patterns, such as those which are entirely black [7]. The majority of cattle bred for beef are not discriminable by coat pattern; with almost 330,000 registered animals, the Black Angus is the most prevalent breed of beef cattle in the US [53]. This is where the motivation for our work lies; we seek to answer the question of whether such cows can be identified not through their markings but through analysis of dorsal features alone, such as spinal patterns and body shapes.

1.2 Novelty of work

Our research uses local farm data to serve as a proof of concept for the automated identification of non-visually patterned herds using top-down depth imagery, which has not previously been employed for this purpose in published literature. Figure 1.3 visualises the success of our work in this task through an embedded space visualisation produced by our network. We also explore the localisation of individuality within images to spark novel discussions surrounding bovine spinal structures.

Dataset opportunities. Our experiments leverage the RGBDCows2020 dataset, which was compiled by William Andrew to facilitate research for an unpublished paper[1]. We further contribute through the creation of a new dataset, CowDepth2023, to address the limitations of RGBDCows2020 which became apparent during project execution. Prior work (full detail in section 2.2.1) carried out by internal University of Bristol connections required depth imagery for the segmentation of cattle from top-down RGB images, which resulted in a wealth of raw data comprising corresponding RGB and depth images thus enabling the creation of CowDepth2023. Images from both datasets were collected from the highly

[1] Unpublished paper is described in further detail in section 2.1.2

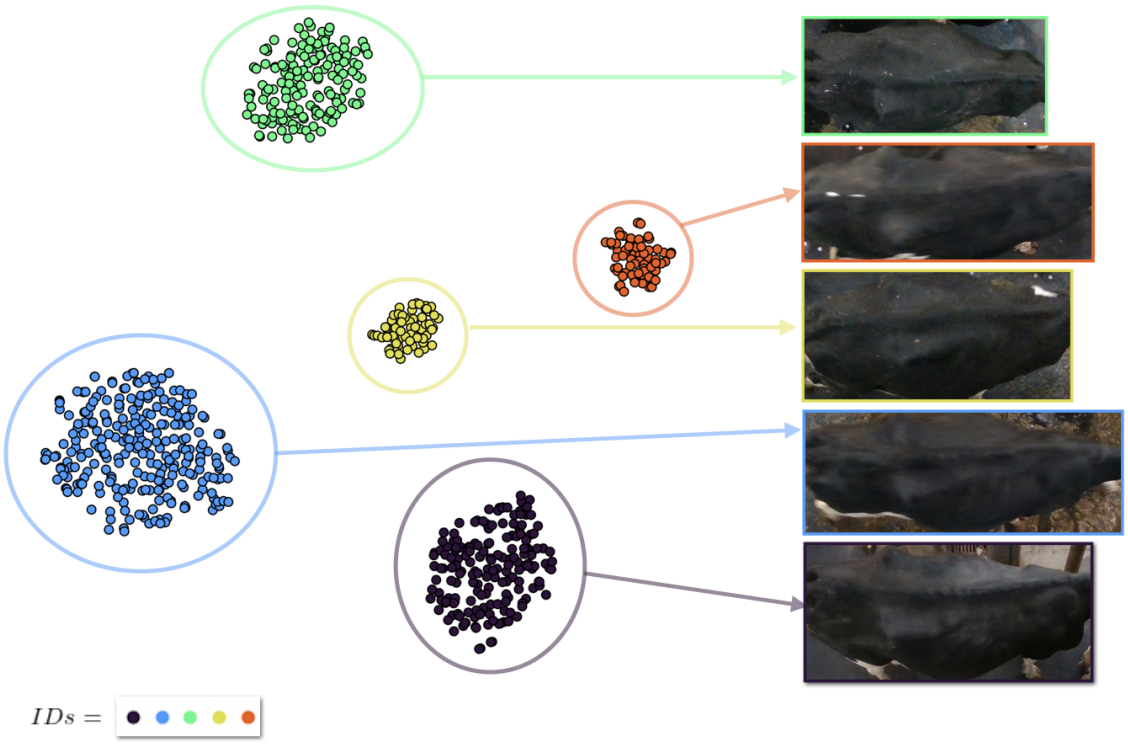


Figure 1.3: **Training embeddings of unpatterned cows.** A t-SNE algorithm was used to visualise the embeddings formed when our network was trained on black cattle, which previous work deemed "challenging" examples. Sample corresponding RGB images are given of each individual (right). Notice the distinct clustering of examples of each individual in the embedded space.

instrumented commercial farm at Wyndhurst; sections 3.1 and 3.2 give full details of the data acquisition processes of RGBDCows2020 and CowDepth2023 respectively.

Unanswered questions of depth potential. The combination of data availability and hardware resources allows us to experiment with and evaluate cutting edge network advances from recent literature in order to explore whether there is enough information encoded in the depth imagery alone to discriminate individual cows. Depth data has been used previously for purposes such as segmenting body parts of cows [27], and the 3D measuring of cattle shape [46], but has not been used before in the task of top-down individual cattle identification. Our work aims to serve as a proof-of-concept that it is possible to leverage neural network advances for this purpose.

Opportunities for veterinary discussion. Since the use of top-down depth imagery for identification of cattle is yet to be explored in published work, the extent of biometric potential of the conformation of cows' backs remains unknown. We aim to forge new paths in this area through providing qualitative results in section 3.11.3 to facilitate discussions surrounding to what extent spinal patterns are individually unique to each cow. These findings hold the potential to inspire further research into the individuality of cow spines, an area which previously lacked established methodologies.

1.3 Challenges

The state-of-the-art nature of our approach presented several significant challenges, the most prevalent of which we describe here.

Dataset creation. Building a dataset from vast image sequences (in our case, roughly 50GB of data) is a challenging endeavour due to several factors. The decisions made in the capture of the original images such as camera resolution, orientation, and height of camera above the ground all affect the quality and completeness of the dataset and obviously cannot be tweaked after data collection. Variations in illumi-

nation, occlusion, and viewpoint are common difficulties faced in image processing applications [60], all of which can make it difficult to identify objects within a crowded scene. In our case, occlusions caused by cattle standing close together caused a significant challenge in the segmentation process. The manual annotation process was not only costly in terms of time but was also challenging since inconsistencies in labelling would cause the model to learn incorrect associations leading to a negative impact on its overall performance. Ensuring the resulting dataset was representative and accurate was critical to ensure experimental results were directly illustrative of the extent of the model's capabilities.

Comparability of results. We sought to conduct all experiments in a way which facilitated immediate comparability with the state-of-the-art. Therefore, we ensured that initially all network parameters and decisions were in-line with previous works. This not only allowed for meaningful ablation studies, but also made it abundantly clear as to where our work improved upon previous achievements. However, we ran into issues in making results gathered from our bespoke dataset directly comparable to other metrics as a result of skew due to temporal bias within training and testing examples. While it is undesirable that these results cannot be compared directly, the experiments carried out remain meaningful in their own context.

Inferring real-world application performance. While the validity of our quantitative results is supported through ablation studies, it is important to remember that our research is experimental and might not be representative of performance in an industrial setting. For example, we test our network primarily on imagery from a herd of Holstein-Friesian cattle from a local farm, whereas it would be of benefit to test performance on beef herds which lack distinctive coat patterns since these are the individuals which previous works struggled to differentiate. The lack of variation in our datasets might also create a facade which might not extend to real-application model performance. For example, variables which were controlled in our image capture process such as camera orientation and light-level might not be as easily controlled in an industrial setting where time-critical inference needs to be performed.

Open-endedness of the task. It was particularly challenging to make high-level decisions as to which paths were the most fruitful in terms of meaningful research. We aim for our paper to provide a breadth of research evaluating state-of-the-art neural network advances with novel research into the spinal structure of cattle, while also giving a depth of discussion into both qualitative and quantitative results. Trade-offs such as these are evaluated in chapter 4, which discusses options and decisions which were taken during the project and compares alternative pathways which could have been taken and leads into a discussion of possible future work.

1.4 Objectives

The high-level objective of this project is to build upon previous work for identifying individual cattle using top-down imagery, in order to have a resulting network fit for the purpose of identifying cattle through depth imagery alone, allowing the identification of herds which don't display individually identifying coat patterns.

More specifically, the concrete aims are to:

1. Outline the current knowledge in literature in order to identify novel paths for our own research.
2. Apply state-of-the-art network advances to investigate the extent to which individuals can be identified through top-down depth imagery alone.
3. Evaluate how well existing datasets perform for our research purpose, and proceed accordingly, manipulating or creating an entirely new dataset purpose-built for our research purpose if deemed necessary.
4. Perform ablation studies to facilitate interpretation and discussion of network abilities.
5. Conduct novel research into possible veterinary applications of our work.

Chapter 2

Background

Previous work demonstrates the efficacy of deep neural networks in the task of individual identification of cattle using biometrics. The inherent ability of such networks to learn the complex relationships between input images and output labels makes them able to identify subtle differences which are unique to each individual. Here we give an in-depth discussion of related literature as well as a comprehensive technical background of concepts discussed in our paper.

2.1 Top-down imagery identification

Whilst section 1.1.3 shows that there are many potential options for identification through the learning of biometrics, this work focuses on approaches which utilise top-down imagery of cows' backs. This section outlines the current state of research in this area at the time of writing, in turn motivating the avenues for our work.

2.1.1 On the shoulders of giants

RGB coat pattern identification. Our work builds almost directly upon the network which accompanies Andrew et al.'s paper 'Visual Identification of Individual Holstein-Friesian Cattle via Deep Metric Learning' (2020) [7] [1]. This research used the OpenSetCows2020 dataset [64] to provide a framework for learning the distinctive coat patterns of Holstein-Friesians.

(a) (b)

Figure 2.1: **Examples and embeddings from the OpenSetCows2020 dataset.** An example image for each of the 46 individuals is given in **(a)**, grouped by image collection method. A t-SNE plotting function was used to produce **(b)**, which is a two-dimensional visualisation of clustering of the testing set in the latent space. Each cluster has the ID of the cow it represents printed on top of the scatter plot. This figure is taken directly from Andrew et al. [7].

[1] Source code: https://github.com/CWOA/MetricLearningIdentification

Network architecture. The proposed network demonstrates high efficacy in discriminating between individual cows, attaining consistent accuracy measures approaching 95%. A ResNet-50 backbone performs dimensionality reduction to produce a latent space mapping such that examples of each individual naturally cluster together; the outputs of the network are fed into a fully connected layer with $n = 128$ outputs, which in turn defines the dimensionality of this latent ID space. This choice of dimensionality is consistent with previous works in classic classification tasks such as facial recognition [51]. Since distances in the latent space are representative of the similarity measure between inputs, the process can be described as a form of metric learning [69]. Once the latent space has been learned, individuals can be classified using a lightweight clustering algorithm such as KNN. Figure 2.1 shows example images of each of the 46 cows in the OpenSetCows2020 dataset (a), alongside a t-SNE visualisation showing an example embedded space with the training set projected into two dimensions (b). The network is able to generalise to unseen cattle by generating separable embeddings not only for the classes seen during training but also for those which are 'unseen'. Success in the separation of classes in the latent space was largely due to the choice of triplet loss function, which we explore in relation to our work in section 3.6.

Avenues for improvement. For Andrew et al.'s research to be used in an industrial setting, an investigation must be done to determine the scalability of the network's current capabilities with larger populations than those used in development. Our project's intention to learn a general representation from depth imagery alone could provide an avenue for this approach to generalise to new herds without the need for any pre-training, as learning the structural features of the cows could allow for greater generalisability. Figure 2.2 shows a random RGB image from CowDepth2023 compared with its corresponding colourised depth image to highlight the difference between the image formats. In depth imagery, each pixel value corresponds to the distance between the object and the camera.

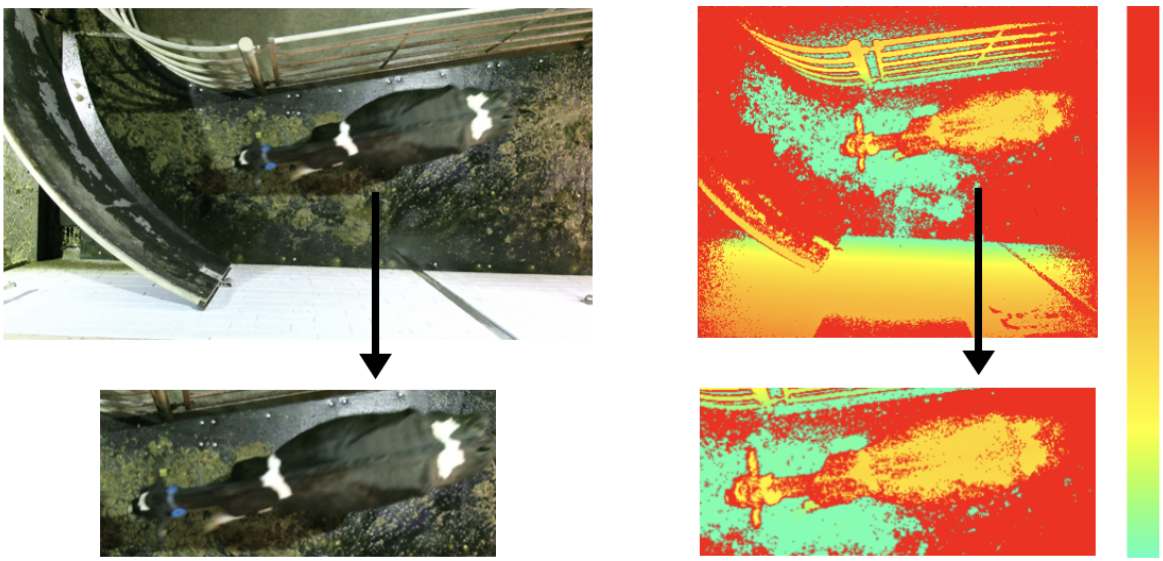


Figure 2.2: **RGB vs depth images.** **(a)** and **(b)** show RGB and depth images respectively, taken simultaneously of the same individual. **(b)** is colourised to allow visualisation, since raw depth image appears entirely black when in print. The colour bar (right) shows the colour spectrum applied to the pixel values; red corresponds to small pixel values (high distance from camera), yellow colours correspond to intermediate values, and green corresponds to large pixel values (smaller distance from camera). Images are taken from the CowDepth2023 dataset.

2.1.2 Preliminary depth experiments

Low accuracy yet promising results. Unpublished work carried out by William Andrew [6] subsequent to the work described above attempted to utilise depth data for identification but found limited success with validation accuracies struggling to surpass 60%. It was tentatively concluded that advancements would require additional data collection.

(a) (b) (c) (d)

Figure 2.3: **Difficult examples (black Holstein-Friesians). (a)** through **(d)** depict RGB examples of cows which the coat pattern model struggled to differentiate, due to their lack of markings.

This unfinished study also explored the possibility of combining RGB and depth imagery to encourage simultaneous learning of features, with results shown in table 2.1. The findings of this unfinished study suggest that although using depth imagery alone yields identification outcomes to some extent, its integration with RGB inputs does not appear to rectify misclassifications that occurred in the RGB space. However, this does not mean that depth imagery alone is or will always be insufficient for large-scale classification purposes. When compared to the random case (in this setup, there are 186 classes which corresponds to a random accuracy of 0.538%), the tentative 60% accuracy results achieved prove that there is enough discriminating information encoded in the dorsal features of cows for the identification purpose. Whether these features can be extracted via improved data collection methods or through network optimisations, the baseline results at least confirm the hypothesis that dorsal features alone exhibit the potential to be used for identification. The "difficult" examples in question are shown in figure 2.3. While it's easy to see how the RGB model would struggle to set these individuals apart, it is also apparent that they do indeed display visually distinctive spinal patterns despite their lack of coat pattern variation which further motivates the avenues for our work.

	(a) Difficult examples included in training	(b) Difficult examples excluded from training
Image Type	**Accuracy (%)**	**Accuracy (%)**
RGB (William Andrew, 2020)	98.00	99.46
Depth (William Andrew, 2020)	**60.62**	62.03
RGB & Depth (William Andrew, 2020)	97.63	99.15

Table 2.1: **Accuracy results from a variety of image types.** Results drawn from unpublished experiments carried out by William Andrew exploring the impact of different image types as input to the network, including passing RGB and Depth data simultaneously as separate streams. 'Difficult examples' were identified as those which the RGB model failed to differentiate due to lack of visually distinguishing coat pattern. The value in bold (60.62%) corresponds to the highest accuracy attained on the entire RGBDCows2020 dataset using depth imagery alone, which is the metric which our work improves upon.

2.2 Alternate identification methods - honourable mentions

It is worth noting that while we focus on deep metric learning for our approach, other network methods have had success in the task of identification with RGB imagery.

2.2.1 Pattern matching methods

As well as through deep learning approaches described in section 2.1.1, top-down imagery of cattle has been used for identification using a pattern matching approach. Andrew et al. found that by the process of extracting a subset of affine scale-invariant feature transform (ASIFT) [40] coat descriptors which distinguish cattle individually, predictions can be generated using a support vector machine with a radial basis function kernel [5]. The system proposed by this work yielded an identification accuracy of 97%. The authors did not include results from the system on difficult animals such as those in figure 2.3 or on species other than Holstein Friesians, but it is clear that the system would not generalise well to individuals without distinguishing coat patterns since the feature matching approach relies heavily on differences in dorsal coat pattern.

2.2.2 Spatial-temporal approaches

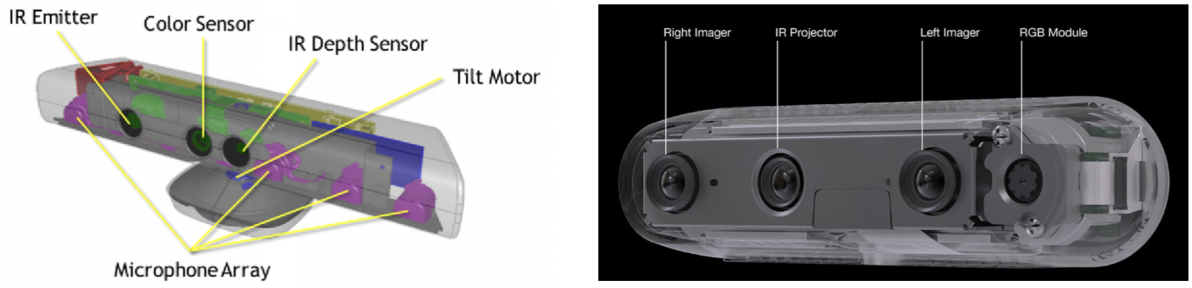
Andrew et al.'s approach in section 2.1.1 used top-down imagery without any notion of temporal correlation between examples. However, the dataset was gathered in such a way that the data was, in its raw form, sequential. Alternate approaches to identification have exploited this feature to varying success.

BiLSTM models. An alternate network approach to cattle identification using deep learning includes Bidirectional Long Short-Term Memory (biLSTM) model elements which utilise spatial-temporal information encoded in video segments for identification [45]. This approach used rear-view images of cattle, which were fed into a CNN to extract features before capturing the spatial-temporal information using a biLSTM layer ahead of the standard fully connected layer allowing individual classification. The authors quote accuracies of 91% using video sequences each containing 30 frames. It would be interesting to know whether this spatial-temporal approach would perform to a better standard when trained on top-down imagery instead of the rear-view data used in this research.

2.3 Depth data developments

Depth imagery has been exploited for many uses in precision livestock farming other than individual identification. Here we outline common practises in terms of data acquisition and transfer learning, as well as giving some example applications.

Common practises for depth image capture. It is common within the publications we discuss in this section for either a Kinect sensor or the Intel RealSense D435 to be used for image capture. Diagrams of each are shown in figure 2.4. Kadlec et al. (2022) [29] provide an evaluation of image capture methods using the D435 RGB-D camera for top-down depth imagery, concluding that image quality results of picture and video-based methods of image acquisition are comparable, with video of course being less preferable in terms of storage. Pulling our view back out of the cattle imagery realm into a broader context, we can look to works which evaluate depth sensors for general computer vision purposes for a more concrete understanding of the options available, such as Andersen et al.'s evaluation of the Kinect sensor [4]. The performance of the Kinect sensor is particularly well documented in literature, such as in the fields of robotics [17], human motion estimation [36], and autonomous vehicles [20]. In terms of deciding which sensor is best for a specific application, Mejia-Trujillo et al. provide a direct comparison between the Kinect and Intel RealSense D435 [39]. However, it is clear from the success of publications using both sensors that either is appropriate for a top-down imagery purpose and one might only prevail over the other due to intricacies in ability to deal with parameters such as ambient light.



(a) Kinect

(b) Intel RealSense D435

Figure 2.4: **RGB-D sensors. (a)** shows a labelled Kinect sensor, taken from Nag et al. [41]. The two separate sensors are shown, which produce RGB and depth imagery simultaneously. **(b)** is an image of the Intel D435, taken directly from the Intel RealSense website [26], with labels from left to right showing the left sensor, IR projector, right sensor, and RGB module.

RGB transfer learning with depth. Many examples can be found of literature where great success has been had in using transfer learning (explained in section 2.4.3) of RGB networks with depth imagery in multiple applications. Gopalapillai et al. [22] explored the effects of converting single-channel

depth images into three-channel images in the context of environment classification for autonomous mobile robots. In these complex settings, it's common to extract horizontal disparity, height, and angle of each pixel's local surface normal as the three streams such that they can be used directly with networks pre-trained on RGB data, and the work proposes low-computational cost solutions for this extraction process. This level of channel extraction is not viable with the RGBDCows2020 dataset due to format of image storage during the acquisition process (see section 3.1), but it could be important to consider these methods of data streaming for future depth imagery applications.

Other applications of cow depth data. Depth-camera based systems have been used with cattle for applications such as 3D model generation [47] [48] [37] and body measurement estimation [49] [34]. Measurements such as these can be used, for instance, to predict body weight for applications such as health monitoring. The manual measuring process which otherwise takes place requires timely labour, which is not only impractical for farmers but can cause stress for the animals. These motivations are often seen within the research realm of non-invasive measurements and identification of livestock and are reflected in our own motivations for individual identification.

2.4 Technical background

Assuming a foundational understanding of artificial neural networks, we proceed to discuss the specific architecture of convolutional neural networks and motivate their suitability for our task. We additionally introduce the concepts of deep metric learning and embedding networks and describe how we leverage such techniques in the construction of our model.

2.4.1 Why convolutions?

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used in tasks such as image classification, object detection, and other applications involving sequential data [44].

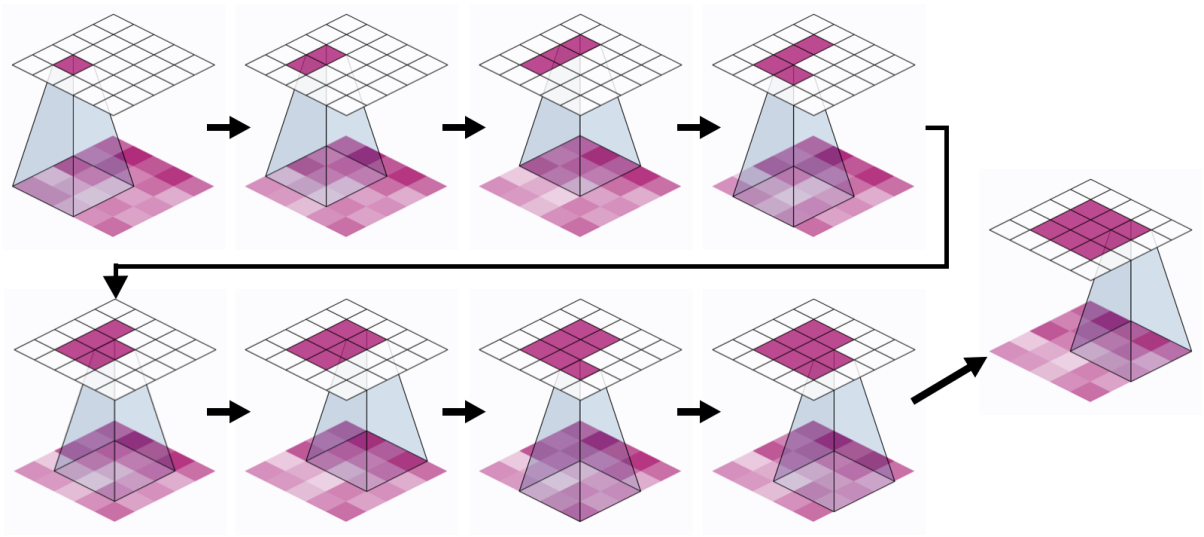


Figure 2.5: **Convolution filter being applied.** Figure adapted from Kaggle tutorial [a]. A 3×3 kernel is applied across the 5×5 input image (bottom of each subfigure) to form the resulting 3×3 output vector (excluding the bounding edges) shown on the right.
[a] https://www.kaggle.com/code/ryanholbrook/the-sliding-window

CNN crash-course. The intrinsic ability to learn discriminative features of images makes convolutional neural networks well suited to many computer vision applications. Analogous to a standard artificial neural network, CNNs are made up of neurons which learn through a process of self-optimisation. Each layer in a CNN applies a certain filter across the input vector (visualised through figure 2.5), thereby learning feature detectors in a hierarchical structure. During training, the early layers learn a representation of

smaller features such as edges and corners, while deeper layers learn increasingly abstract and complex patterns. In the context of coat pattern-based identification described in section 2.1.1, the lower layers might detect small features on the cow's backs, while the layers further downstream could detect complex sections of coat patterns. After the convolutional layers, a pooling layer performs down sampling as a form of dimensionality reduction before information is passed to a fully connected layer. In a standard CNN for classification, outputs of the fully connected layer are used to produce per-class scores.

2.4.2 Embedding networks

Contrary to the standard classification approach described above, our application exploits metric learning to learn a general latent representation of classes rather than producing per-class logits.

Deep metric learning. Metric learning has the objective of quantifying similarity between examples through use of a distance metric [31]. In a similar vein, deep metric learning is a technique which uses an embedding function to project examples onto a latent space where the distance between points reflects the similarity of examples in their original space. Therefore, the objective of most deep metric learning applications is to learn an embedding function which clusters examples of each class in the latent space in the optimal way, i.e., to facilitate the greatest inter-class separation [30]. Challenges involved in building such a network involve choosing appropriate distance metrics, network structures and loss functions [30], all of which we discuss where appropriate in chapter 3.

Triplet Loss. Since its introduction by Schroff et al. in 2015 [52], it has been common for embedding networks to use some variation of triplet loss to encourage the simultaneous learning of both similarities and dissimilarities between classes. With (X_a, X_p, X_n) denoting an anchor, positive example, and negative example in the latent space respectively, the function encourages there to be a minimal distance between x_a and x_p while maximising the distance between x_a and x_n. We can therefore represent standard triplet loss by the formula 2.1, where $d(x_1, x_2)$ denotes the Euclidean distance between vectors x_1 and x_2 in the high-dimensional space. We discuss the efficacy of other distance measures in section 3.7.

$$\mathbb{L}_{TL} = \max\left(0, d(x_a, x_p) - d(x_a, x_n) + \alpha\right) \tag{2.1}$$

A visualisation of the triplet loss learning objective is given in figure 2.6, the idea being that the triplet on the left shows each example in the latent space before learning has taken place and the right side of the image depicts the same triplet after the loss function has successfully been applied.

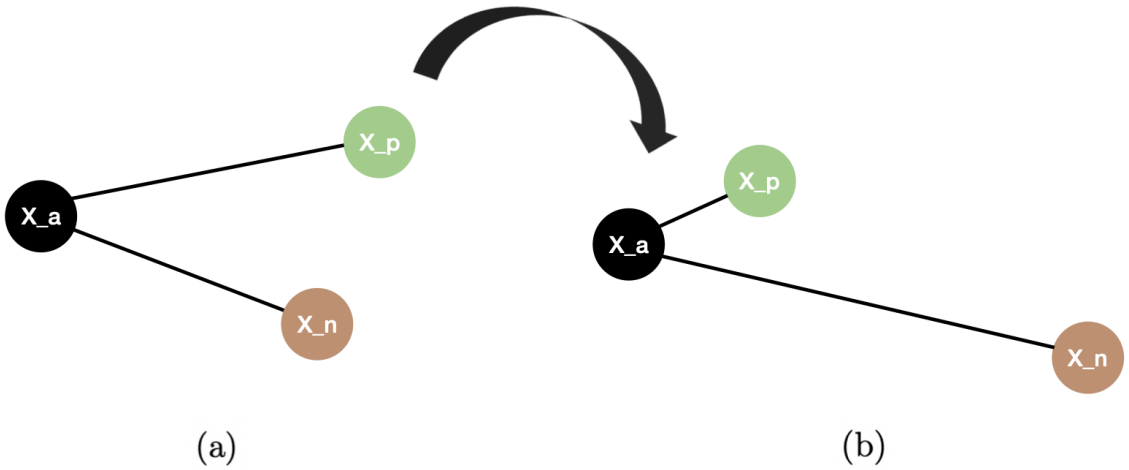


Figure 2.6: **Triplet loss visualisation. (a)** represents a triplet before learning has taken place, while **(b)** shows how a triplet might look after an iteration of the triplet loss function has been completed. As before, X_a, X_p, and X_n represent an anchor, positive example, and negative example in the latent space.

2.4.3 Backbones for detecting bony backs

The model architecture proposed in this paper relies on a ResNet backbone pre-trained on ImageNet, as seen in the network flow diagram of figure 1. We motivate the use of this particular backbone and experiment with alternatives in section 3.8.

Transfer learning crash-course. In current literature, it is common for image detection systems to use some form of transfer learning within the network [24] [55]. The general idea of transfer learning is to use the parameters of a pre-trained model in a new task with a smaller dataset to leverage the learned knowledge of the large pre-trained model. This has many advantages, namely the effect of reducing the amount of labelled training images needed for achieving good performance which massively reduces expenses in terms of data collection and computational resources involved in model training and development. ResNet [25] is a network backbone which is widely acknowledged in literature to have great success in many different image recognition applications. We provide a diagram of the standard ResNet-50 architecture in figure 2.7. Specifically, ResNet (Residual Network) addresses the vanishing gradient issue using a large number of skip connections. Since the identification of individual cattle requires the learning of both subtle and complex features in depth maps, ResNet improves model accuracy by allowing better propagation of gradients through the network so that vanishing gradients do not render early layers useless.

Forming our network. Following suit of Andrew et al.'s work described in section 2.1.1, our work uses a ResNet [25] backbone pre-trained on ImageNet, with the impact of differing implementations tested and evaluated in section 3.8. To form the network proposed in figure 1, the input triplets are passed into the ResNet module which is adapted to produce embeddings of dimensionality 128 to be projected onto the latent space embedding. Hence, the ResNet backbone is acting in this case as a dimensionality reduction module to produce an embedding for each example.

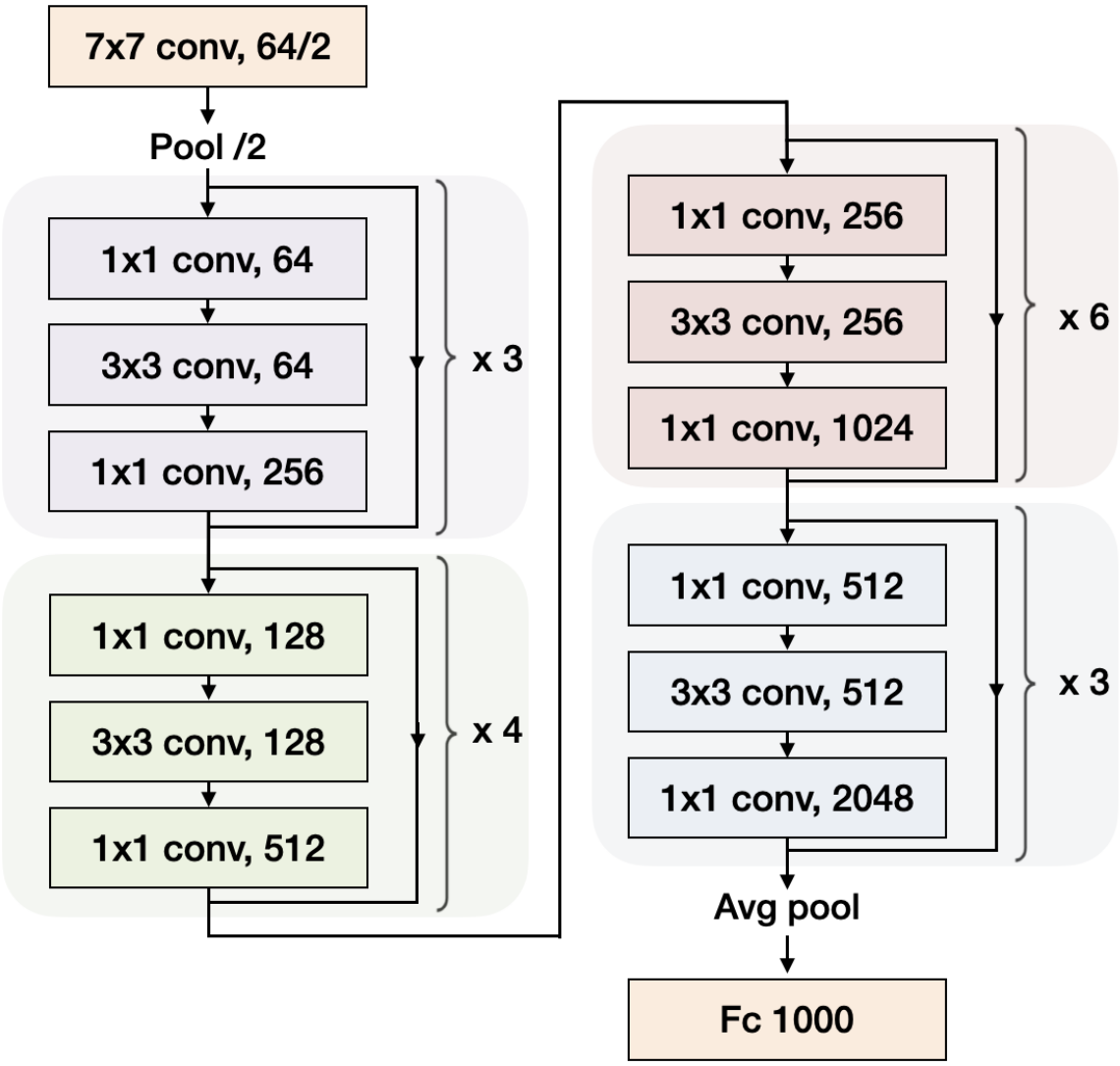


Figure 2.7: **ResNet architecture diagram.** Here we give a network flow diagram of a standard ResNet-50 architecture. The diagram was drawn using column 5 of table 1 from the original ResNet paper [25] as a reference for network structure. In block labels, '$x \times x$ conv' corresponds to size of each convolutional layer, where the value following each comma corresponds to the output size of each block.

Chapter 3

Project Execution

This section describes the data acquisition and preparation processes for both the existing RGBD-Cows2020 dataset and the new CowDepth2023 dataset. We then give details of our experimental setup before beginning to discuss initial network results. We describe experiments carried out in order to evaluate datasets, refine network architecture, and discuss state-of-the-art advances in deep metric learning approaches.

3.1 RGBDCows2020

RGBDCows2020 was created in previous unpublished work by William Andrew from a subset of the publically available OpenCows2020 dataset [1], containing corresponding RGB and depth top-down imagery of cattle.

Dataset preparation. Images in the RGBDCows2020 dataset were captured using an Intel RealSense™ Depth Camera D435 [26], which was mounted around 4m above the ground over the walkway between the milking parlour and holding pens at Wyndhurst farm. The D435 yields depth images of resolution 1280×720 using Stereoscopic depth technology and captures RGB images to correspond with each depth image. A threshold in optical flow was used to automatically trigger capture, with capture halting when there was no additional movement detected in a 5 second period. The dataset was produced over a period of one month, recording the entire herd as they passed through the capture area for milking once a day. The capture area was large enough for the cows to be facing any direction during capture, however they typically were imaged when facing direction of travel, so their heads appear on the right side of each image as seen in figure 3.1 which gives an example of an RGB and depth image pair.

(a) (b)

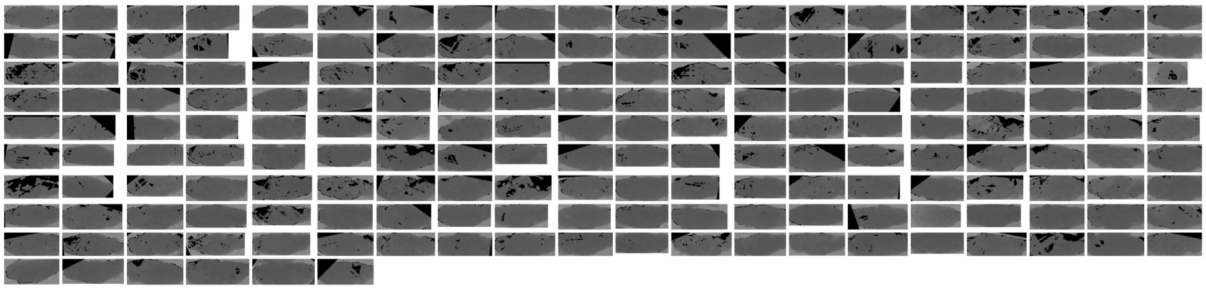
Figure 3.1: **Example images of cow #000 from RGBDCows2020. (a)** shows a depth image where each pixel value corresponds to a distance from the camera, while **(b)** is the corresponding RGB image.

Captured cows. The resulting dataset is made up of 186 individuals, each of which were manually labelled. For each individual, there exist corresponding folders of RGB and depth images, initially labelled 000 through 185. A random example of each cow is shown in figure 3.2. Visually inspecting

[1] https://research-information.bris.ac.uk/en/datasets/opencows2020

these samples highlight the apparent similarity between individuals, and it is a very challenging task to manually classify each cow; this emphasizes the remarkable aptitude of deep neural networks for performing classification tasks of this nature. It also gives rise to a discussion about which parts of the image the networks pay most attention to, whether that be features such as body size or spine structure; we explore possible answers to this question in section 3.11.3.

(a) RGB image examples



(b) Depth image examples

Figure 3.2: **Visualisation of associated RGB and depth images from RGBDCows2020.** One image of each cow has been randomly selected to be visualised, with an RGB (top) example and depth (bottom) example for each, resulting in 186 RGB images and 186 corresponding depth images. The dataset has a total of 14412 examples, so this figure visualises 1.3% of the total images. This figure is taken directly from unpublished work carried out by William Andrew.

Preparing for training. Five random images for each individual were removed to serve as test data, discarding any cows in the process which had fewer than 10 images to allow for an absolute minimum of 6 training examples per cow. This took the total number individuals down to 182, with an average of 74 training instances per class. There exists a significant variance in the quantity of training examples available for each cow, with some having a very limited number of samples while others possess nearly 200. This is largely a result of the cows having the liberty to spend as much or as little time passing through the capture area as they wished, causing some to pass through too fast for the camera to capture a full image, capturing them only in the boundaries of the capture area.

Data quality evaluation. As with almost any large-scale dataset, there are other forms of imperfections to be found within the samples. Most notably, there are regions of each depth image which are partially or entirely black, which in depth imagery terms corresponds to an area of "infinite" distance from the camera lens. A selection of depth images from cow #000 are shown in figure 3.3 to illustrate this phenomenon. This effect can be put down to imperfections in the camera setup, as the images have been taken from around $3-4\,\mathrm{m}$ from the camera lens while the recommended distance is $0.3-3.0\,\mathrm{m}$ [26]. This is likely to have adverse effects on model performance since visual artifacts such as these can result in occlusions in images. As a result of the dataset's handling prior to the start of this project, we are only able to access each image as an 8-bit data stream, despite the original resolution being 16-bit. This inevitably results in a significant loss of information which can negatively affect the ability to learn distinctive identifying features, specifically due to the downscaling of bit-depth resulting in similarities among adjacent pixels.

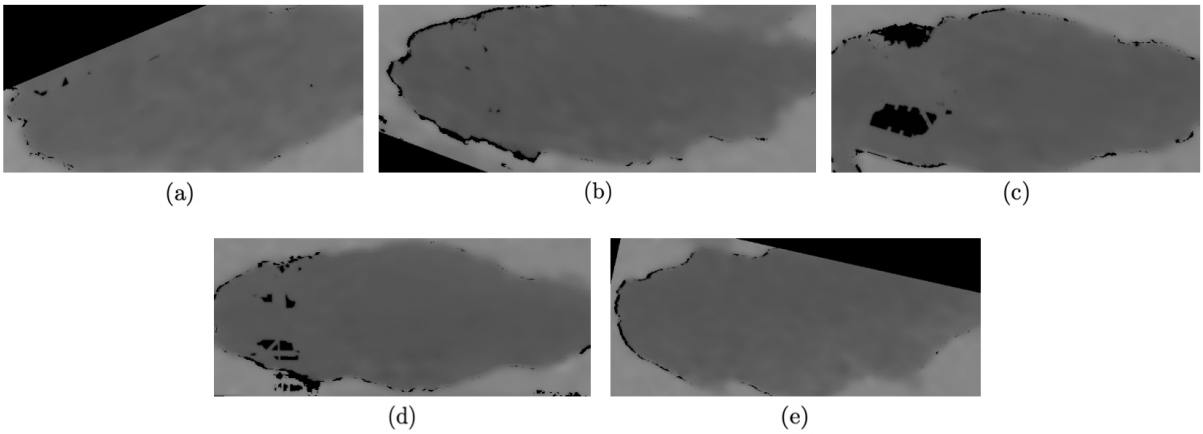


Figure 3.3: **Example depth images of cow #000 from RGBDCows2020.** These images highlight the issue of imperfections in the image capture process resulting in black patches in images. Some such artefacts are a result of the auto-alignment used in the data preparation process, seen for example at the top-left of image **(a)** and the top-right of image **(e)**.

3.2 CowDepth2023

We present a new dataset, CowDepth2023, which was developed to address the limitations of RGBD-Cows2020. Images in CowDepth2023 have 16-bit resolution, which we found to be crucial in order for accurate information to be supplied to models. In this section, we describe the processes of collecting and annotating the data, as well as measures which were taken to ensure the quality and reliability of images in the new dataset.

Image capture. A Kinect sensor was used for depth image capturing in this case, in contrast to the Intel D435 which captured the RGBDCows2020 dataset. The Kinect sensor captures images of size 640×480 pixels at a rate of 30Hz, and metadata files were recorded so that each depth image could be temporally paired with a corresponding RGB image. Depth images are created by the sensor by capturing infrared light reflecting off objects; light which has reflected from a closer object has a shorter time of flight than those which are further away, allowing time of flight measurements to be used alongside pattern deformation in the creation of the resulting depth map [23].

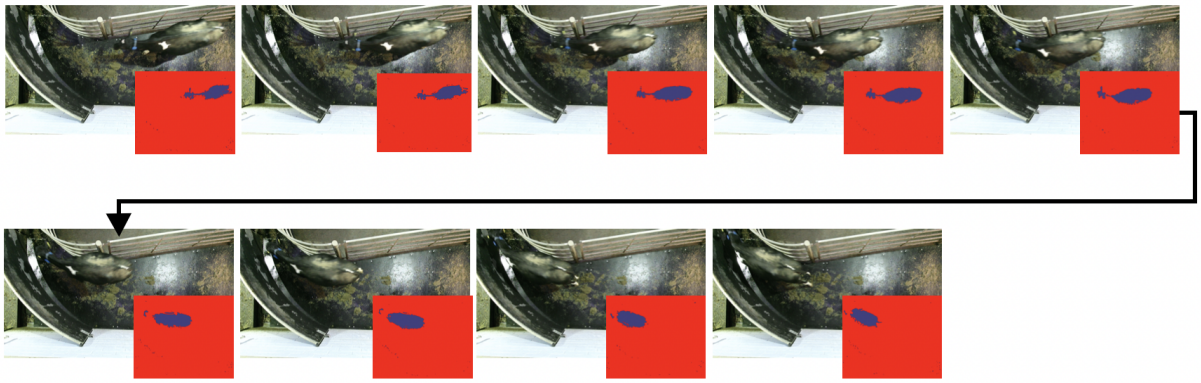


Figure 3.4: **A cow walking through the capture area.** A selection of sequential RGB (large images) and their corresponding segmented depth images (red and blue) are shown, depicting the progression of a single cow walking towards the milking parlour. This shows how the path taken by each cow is generally right-to-left with a clockwise rotation at the end as the cow turns to its right to walk through the gate.

The data collection process yielded 14 sequences of image capture, which in total corresponded to around 50GB of data. The RGB camera and Kinect sensor were set up directly above a path outside the milking parlour of Wyndhurst farm, which explains the leftwards orientation of each cow. The path walked by each cow follows a very similar pattern, resulting in a stream of sequential data where the cow can be

seen to walk towards the top-left of the frame. This is shown through figure 3.4, which shows colour visualisations of depth images of cow #000 walking through the frame paired with their segmented counterparts (red and blue images).

Segmentation. Previous work on coat pattern identification [5] used depth imagery in a similar manner to perform D-segmentation on cattle imagery. Taking inspiration from the scripts used in this process, we performed our data preparation with a similar approach; our resulting suite of data preparation functions are available publicly [2], should they be useful for any future work of a similar nature. The first task was to segment each depth image to produce bounding boxes around sections of each image where cows appear. We describe the process here, relating each step to the flowchart given in figure 3.5. Firstly, each depth image (a) was thresholded (b) such that all values outside of the range 2000-3400 were removed since we found by a trial-and-error process that all pixels containing cows lay within this range only. Next, a mask was applied to remove as best as possible the gates and such structures which were present in each image (c). This mask was created using an image found at the beginning of one of the data streams which contained no cows, so only the structures we sought to contain in the mask were present. A median filter was then applied to smooth each image (d), before using SKLearn's label function to extract the largest regions in the images which corresponded to individual cows, which then had bounding boxes drawn around them (e). This resulted in one folder for each depth image containing a segmented image for each cow found in the frame, ready for the labelling process.

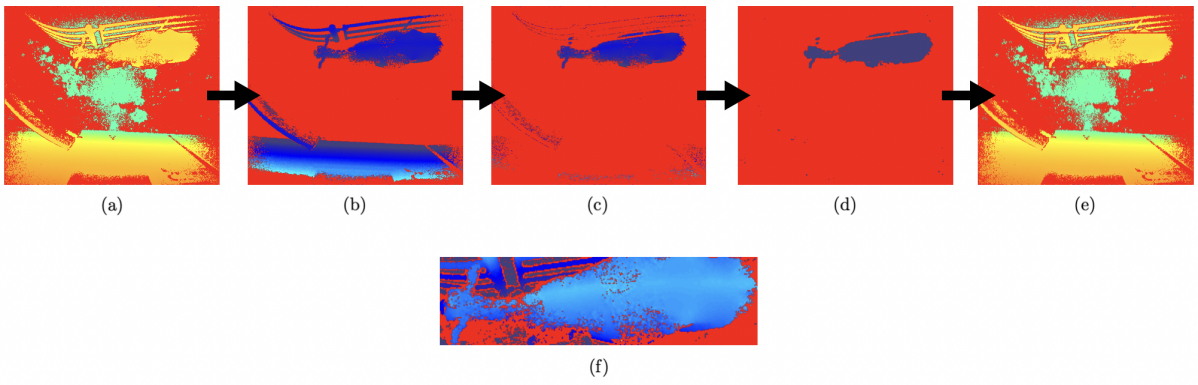


Figure 3.5: **Segmentation process flowchart.** RGB colourisation is used to show the process of segmenting cows from a depth image. This example in particular shows the segmentation process of a frame containing only a single cow. Each subfigure depicts a stage in the segmentation process as follows: **(a)** Original depth image **(b)** Depth image after thresholding function is applied **(c)** Depth image after mask to remove gate structures is applied **(d)** Depth image after "blob" segmentation process is applied **(e)** Original depth image with bounding box produced in step d applied around the cow **(f)** Resulting cropped depth image depicting the final section ready to be labelled.

Labelling. The labelling process was straightforward for image sequences which contained one cow only; each segmented image from the sequence was taken to be an individual. This process was more difficult for frames containing more than one cow and resulted often in loss of examples, for instance if cows were standing directly next to each other the algorithm would put a bounding box surrounding them both together, resulting in the segmented image being thrown away. Figure 3.6 shows an example of this phenomenon, where three cows appear in the frame but only one was correctly segmented by the script. For the instances where multiple cows were successfully segmented from a single frame, corresponding RGB images were used to manually label each cow. This process took a few days of manual work. The depth images themselves are not easy to tell apart directly without their RGB counterparts, and as such there was possibility for mistakes to be made which would result in some incorrect depth images being sorted into the folder for a different cow. To reduce the chance of these sorts of error, periodic selections of images labelled as each individual were visualised with RGB colour to ensure no errors had been made.

[2] [redacted]

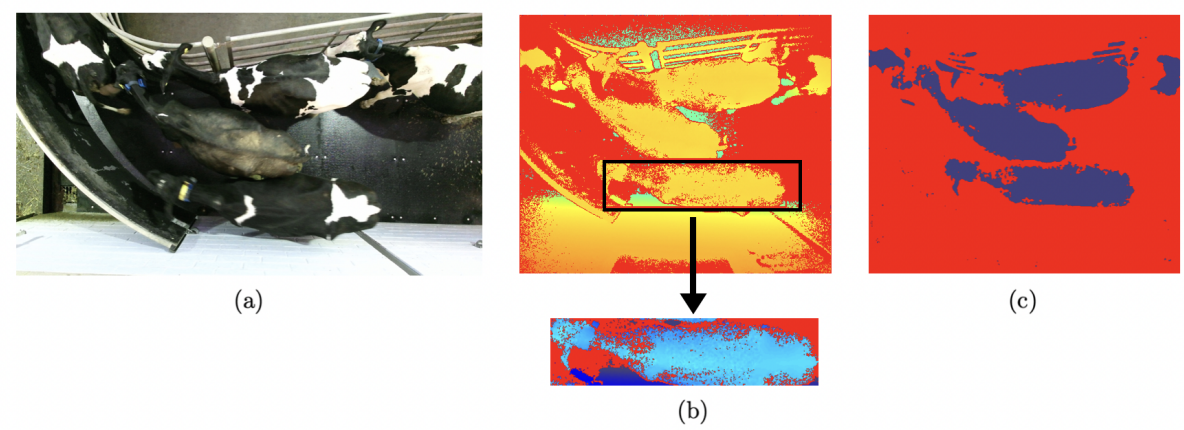
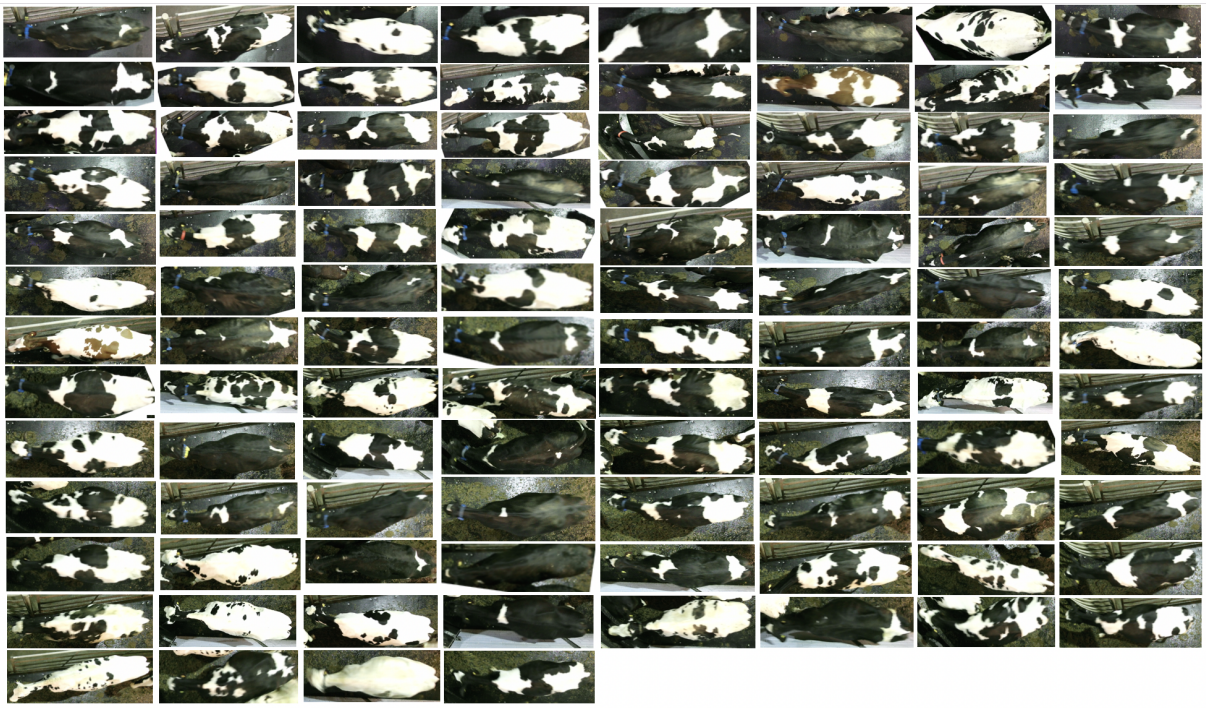


Figure 3.6: **Challenging image containing multiple cows close together.** The segmentation process fails to detect the three separate cows in the frame, since the cows at the top of the image are physically touching each other and so are not seen as separate entities by the script. **(a)** Corresponding RGB image with three full cows in frame. **(b)** Colourised depth image after bounding box is applied, along with a single segmented cow found by the script. **(c)** Segmented image showing how the top two cows in the frame are physically touching.

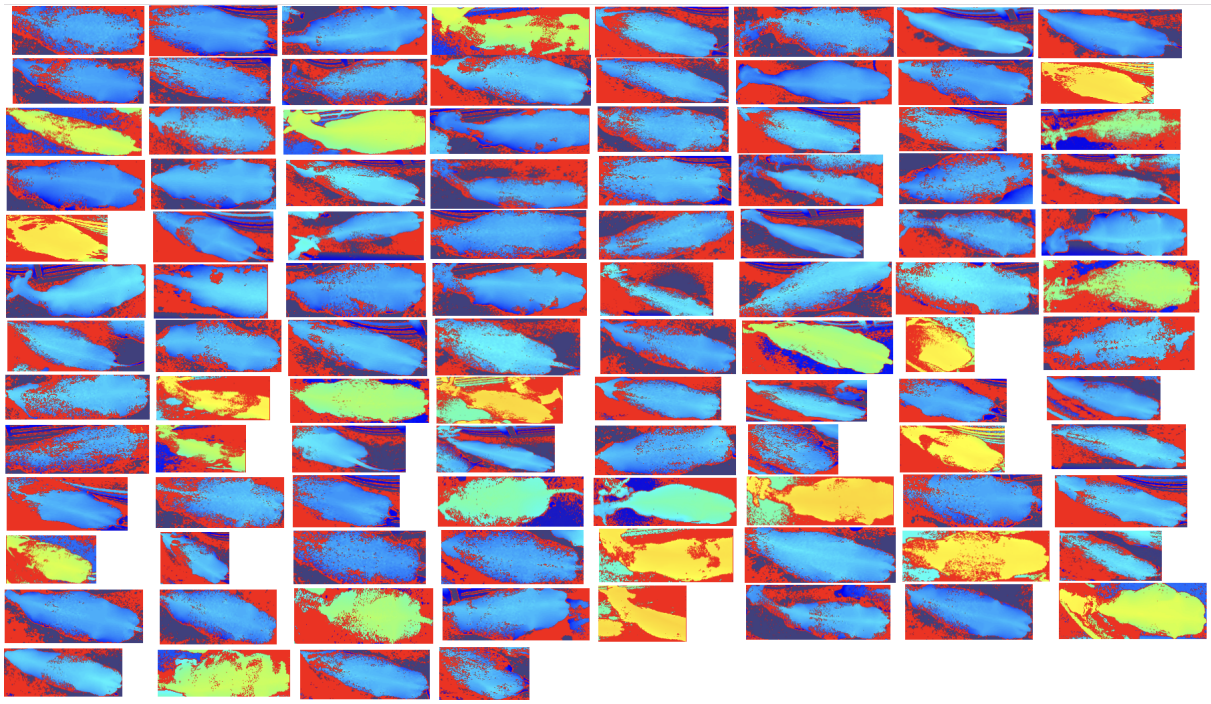
Dataset preparation. At the end of the segmentation and labelling process, 102 folders of images were produced each corresponding to an individual cow. However, after scrutiny of RGB images it transpired that two cows had managed to walk through the frame twice at different times, meaning that these two individuals were counted twice in the labelling process. After this was amended we were left with 100 individual cows, amounting to a total of 43195 images in the dataset. This meant that in contrast to RGBDCows2020's 74 training instances per class, the cows in CowDepth2023 have a mean of 426 examples. Again, due to the differing length of time each cow spent walking through the field of view there is high variance in number of examples per cow; the smallest number being 39, and the largest being 4062. A similar process to that described in section 3.1 was applied to produce test/train sets for the data, consisting of 5 test examples selected from each folder without replacement.

Pre-processing (or lack thereof). The defining difference between CowDepth2023 and RGBD-Cows2020 is the disparity in resolution of images, however there are a few other variables to note which may influence differences in model performance. The pre-processing involved in the creation of CowDepth2023 was kept to a minimum, to ensure that as much information as possible was kept for the models to learn from. For example, we did not perform any hole-filling procedures which are commonly performed on depth imagery [8]. The idea of this was to allow the model to decide for itself which parts of the image constitute a differentiating feature. Another decision was made not to orientate bounding boxes to keep all cow backs at the exact same orientation in the frame. As seen in figure 3.4, most cows are oriented very similarly anyway due to the path they walk through the gate, although there is a degree of clockwise rotation as the cows move up through the gate. This decision was made on the grounds that by their nature, convolutional networks are invariant to rotation, so supplying the model with images of different rotations made sense to make the model more robust.

Resulting dataset. Figure 3.7 shows a random RGB frame of each of the 100 cows used in the final dataset, as well as 100 corresponding colour visualisations of depth images of each cow.



(a) **RGB images of each individual.** A random RGB frame has been selected of each individual via a manual image selection process.



(b) **Depth images of each individual.** Each depth image has been colourised for visualisation using skimage's exposure function in Python. Note that depth images correspond to RGB images in **(a)** in terms of individual cow but do not match in terms of frame captured.

Figure 3.7: **Example images from CowDepth2023 dataset.** One random RGB image is shown in **(a)** for each cow existing in the new dataset, totalling 100 individual cows, while **(b)** shows a random depth image for each cow. All cows are facing the left direction due to orientation of the camera above the route to the milking parlour.

3.3 Experimental setup

Given the large amounts of data involved in training paired with model complexity, it was imperative to perform experiments on a GPU cluster. We set up source code such that CUDA is used whenever it is found to be available.

GPU node setup. Experiments were run on Nvidia Pascal P100 GPUs on the University of Bristol's supercomputer, Blue Crystal (Phase 4) [1]. All models were trained for 10 hours which typically transpired to the completion of 50 epochs, given a typical 12 minutes to complete one epoch with a batch size of 16. Models tended to converge after around 10 epochs, so the time limit set for jobs of 10 hours was very much an upper bound to be certain convergence was reached in all experiments.

Parameters and checkpoints. To be consistent with previous work, stochastic gradient descent was used with an initial learning rate of 1×10^{-3}, momentum of 0.9, weight decay 1×10^{-4}, and a margin for triplet loss of 0.5. The model was set up to evaluate on the validation set every other epoch, saving model weights in .pkl format if the previous best results get surpassed. Along with 'current' and 'best' weights, training and testing embeddings are also saved upon each checkpoint to allow for visualisations.

Hyperparameter	Value used in training
Initial learning rate	1×10^{-3}
Momentum	0.9
Weight decay	1×10^{-4}
Triplet loss margin	0.5
Batch size	16

Table 3.1: **Hyperparameter settings.** All results quoted from experiments in this paper were run with these settings unless otherwise stated.

3.4 Baseline results

An initial step of project execution involved the training of a baseline neural network on the depth dataset; this was accomplished by utilizing the model described in section 2.1.1, with hyperparameters described in 3.1 and the default loss function 'ReciprocalSoftmaxLoss' (see section 3.6 for loss function details).

Adapting network for depth data. Since it was built for use with RGB images, the original network requires three streams of data to be provided as input for each image in the dataset. We found that feeding depth data into the original network as three streams according to the PILLOW [2] package's "new Image" function with the "RGB" option specified allowed the model to learn with respect to depth values.

Initial results and training curves. We give a training loss curve and learning curves produced from an initial attempt at training the model in figures 3.8 (a) and 3.8 (b) respectively. This approach reached a maximum validation accuracy of 52.8%, shown by the purple line in figure 3.8 (b). Clearly, there is a large degree of overfitting exhibited through the disparity between training and validation accuracies, along with the issue of training accuracy hitting the ceiling after less than 30 epochs which does not allow for any subsequent learning.

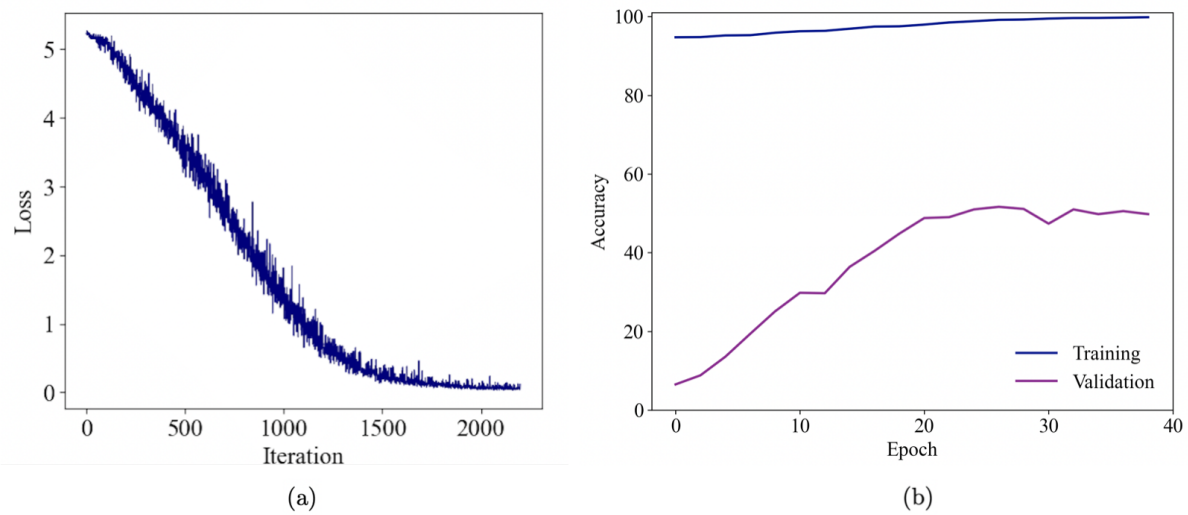


Figure 3.8: **Training curves for the baseline model.** Graphs are drawn from 40 epochs of training on the entire RGBDCows2020 dataset, using reciprocal softmax loss and default hyperparameters described in section 3.3. **(a)** represents the training loss logged at every iteration, while **(b)** shows the training and validation accuracies logged upon every evaluation on the validation set.

Embedding visualisation. The t-SNE [65] technique affords the opportunity to visually represent high-dimensional embedded spaces, such that clusters of points can be seen. This can be used not only as a quick metric for model evaluation, but as an avenue for pinpointing areas where the model can be improved. Figure 3.9 shows this technique having been applied to the training and testing embeddings of the trained baseline model in its best state. It is clear to see that while the test set embeddings have begun to cluster, there are also many misclassifications, as anticipated given the validation accuracy metric of 52.8%. This indicates that a different loss function might be more appropriate to separate classes more effectively, which we explore in section 3.6.

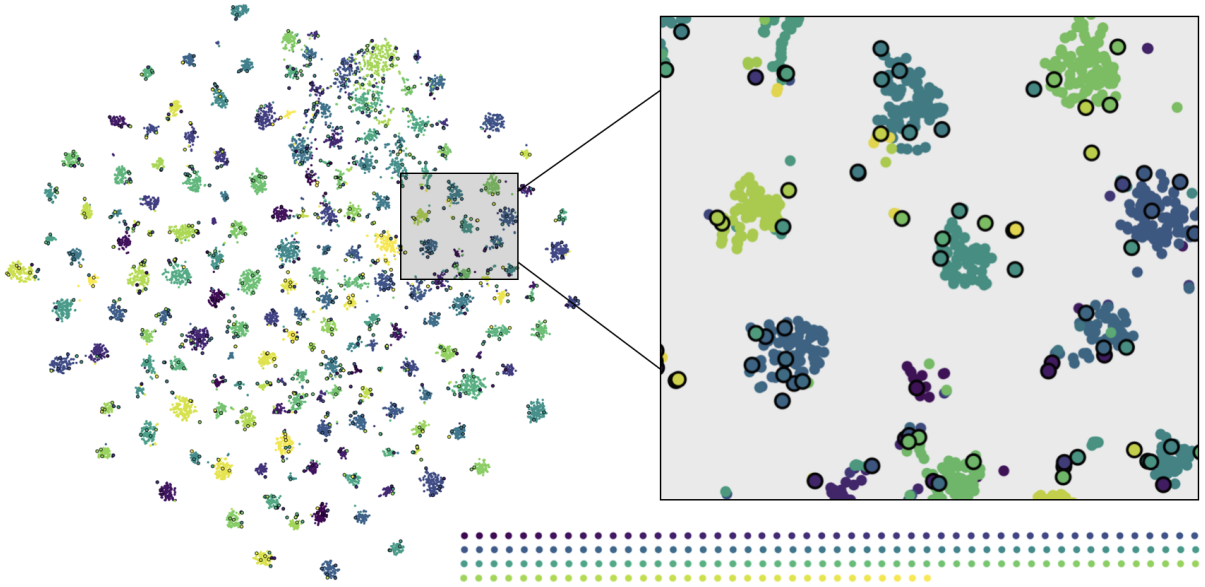


Figure 3.9: **Embedded space visualisation for baseline model.** t-SNE was used to reduce dimensionality of points from 128 to 2. Each colour corresponds to an individual cow (182 discrete colours shown in bottom right correspond to the 182 individuals RGBDCows2020), where regular points represent training examples and points with a black border represent test examples. A section of the embedded space is magnified (right) to highlight how examples have separated into classes, but with many imperfections. In particular, misclassifications can be seen in the black-bordered test example points.

3.5 Data augmentation

Data augmentation is a widely adopted technique in image recognition systems [56]. This process tackles the challenge of limited training data by leveraging image transformations such as rotations and zooms to create new examples to be used in training alongside those which already exist. By increasing the diversity and variability of training data in such a way, the model's capacity to learn features becomes more robust and capable of generalising the new unseen testing data.

Motivation. Figure 3.8 makes it immediately evident that the model reaches almost perfect accuracy on the training set very quickly, indicating limited potential for learning a general solution. Ideally we would like to see a more gradual curve, starting much further down the graph than it does in figure 3.8; this suggests that we should make the dataset more challenging for the model to learn, in order to bring down the initial training accuracy to encourage more general learning. Data augmentation presents itself as a clear first option, with consistently proven success in image recognition networks [56]. Obviously, care must be taken with which augmentations are performed on the images, as it would be counterproductive to create images which distort the distinguishing features between individuals. For example, it was important in our case to preserve the aspect ratio of images. It is also common to distort colour spaces in training images, however distorting colours in our case would translate to a change in distance of each pixel from the camera due to how depth imagery is represented, which would be counterproductive.

Augmentation process. A minimal solution used the Augmentor [38] package to produce 60 new images for each cow from random rotations, translations, and croppings of the original images. Figure 3.10 shows a random selection of augmented images produced from one training example. In the case of rotation, an operation where $n\%90 \neq 0$, where n is the number of degrees used in the rotation would result in the image being padded in each corner. However, Augmentor's default behaviour is to alleviate this issue by cropping the image to retain the largest possible crop while maintaining the original aspect ratio of the image. This behaviour made the package a suitable choice since the creation of new artefacts in augmented images would be undesirable.

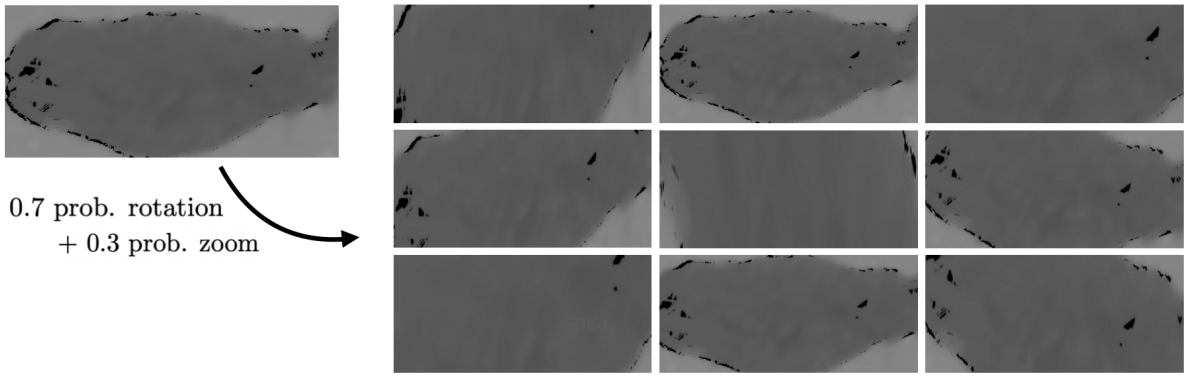


Figure 3.10: **Image augmentation visualisation.** A randomly selected example original image (left) with a random sample of 9 augmented images produced from this one original (right). The augmented images are produced with a 0.7 probability of rotation of maximum 18 degrees, combined with a 0.3 probability of a zoom operation. The original image was taken from the RGBDCows2020 dataset.

We also created a dataset which built upon the dataset with 60 additional rotated/zoomed images to contain an additional 30 images for each individual exhibiting varying degrees of Gaussian noise. The function 'GaussNoise' from the Albumentations package [11] was used to produce these additional images, using parameters mean = 0 and variances σ^2 ranging from 10 to 80. Example images are shown in figure 3.11.

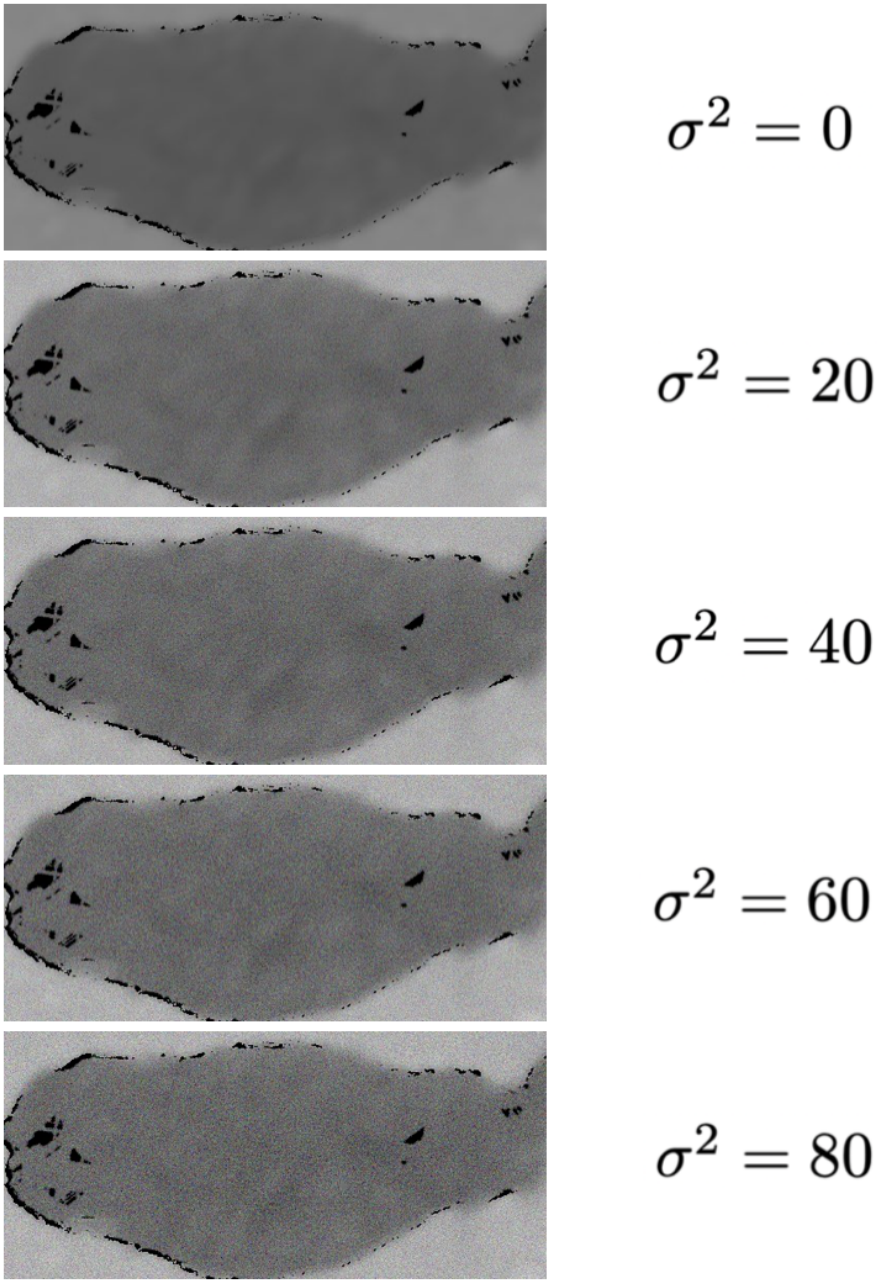


Figure 3.11: **Gaussian noise visualisation.** Top image shows an example original depth image where variance $\sigma^2 = 0$, and subsequent images depict this image where Gaussian noise has been added with incremental variances. Contrast of each image has been increased by 5% for visualisation purposes. The original image was taken from the RGBDCows2020 dataset.

3.5.1 Results

The addition of rotated and zoomed images to the dataset translated to an increase in accuracy of around 2%, reaching a high of 54.7%. Table 3.2 shows the results of each experiment. First a baseline was run, before incrementally increasing numbers of augmented images for each individual. Our experiments saw accuracies increase by roughly 1% for each 30 additional images added, however we limited experiments in the interest of time to 60 additional "rotation and zoom" images. It would have been beneficial to explore the impact of a larger number of extra images on the accuracy, in order to determine the point at which further increases no longer yield improvement. Adding Gaussian noise to produce extra images resulted in a decrease in accuracy, however. Note that these experiments were run with the baseline model using reciprocalSoftmaxLoss, which explains why accuracies are lower than those quoted further on. Despite this, the relative increases in accuracies shown in these results remain meaningful.

Dataset	No. extra images	Validation accuracy (%)
Original	0	52.78
Rotation + zoom	30	53.57
Rotation + zoom	60	**54.67**
Rotation + zoom + Gaussian noise	90	53.79

Table 3.2: **Effect of augmenting data on model performance.** Experiments were performed to explore how adding augmented images to the input dataset affects model performance. The baseline set included only the original images, then subsequent tests were performed with increasing numbers of augmented images. All tests were run using the baseline model in section 3.4, using the reciprocal softmax loss function.

3.6 Loss functions

In the realm of deep learning, the selection of a loss function plays a pivotal role in determining model performance, and such is the case for our application of metric learning.

Scope for investigation. On RGB data, Andrew et al. [7] investigated the use of various loss functions for constructing metric latent spaces. The embedding functionality of the network sees a reduction in dimensionality through the network from a tensor of shape $width \times height \times channels$ onto a latent representation of size $\mathbb{R}^n$ where n corresponds to the chosen embedded space dimensionality. For all experiments presented in this paper, the value of n was chosen to be 128 in order to keep results directly comparable with previous work. Finding a loss function which yields an identity-clustered embedded space is paramount to model success; we ran experiments using various loss functions to verify the efficacy of each on the RGDBCows2020 depth dataset.

3.6.1 Loss function descriptions

Triplet Loss (TL). We give a comprehensive explanation of the standard triplet loss function in section 2.4, and re-state its formula here (3.1) for completeness.

$$\mathbb{L}_{TL} = \max\left(0, d(x_a, x_p) - d(x_a, x_n) + \alpha\right) \tag{3.1}$$

Reciprocal Triplet Loss (RTL). RTL in this context is a method to alleviate the issue of triplet margins being satisfied at any distance from the anchor. It would be undesirable for the margin to be satisfied at a large distance from the anchor, since we wish to keep the anchor close to the positive example in the embedding. Reciprocal triplet loss is defined in formula 3.2.

$$\mathbb{L}_{RTL} = d(x_a, x_p) + \frac{1}{d(x_a, x_n)} \tag{3.2}$$

Inclusion of softmax (RSL) and (TSL). We ran experiments on RTL and TL with and without the inclusion of a softmax function, resulting in the functions 'ReciprocalSoftmaxLoss' and 'TripletSoftmaxLoss'. This ablation study is consistent with the loss function experiments carried out on RGB data in Andrew et. al's work. A standard softmax function is defined in formula 3.3 where N = total number of classes, which is in turn used to derive RSL and TSL in formulae 3.4 and 3.5 respectively.

$$\mathbb{L}_{Softmax}(x_i) = -\log\left(\frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}\right) \tag{3.3}$$

$$\mathbb{L}_{RSL} = \mathbb{L}_{Softmax} + \lambda \cdot \mathbb{L}_{RTL} \tag{3.4}$$

$$\mathbb{L}_{TSL} = \mathbb{L}_{Softmax} + \lambda \cdot \mathbb{L}_{TL} \tag{3.5}$$

In both $\mathbb{L}_{RSL}$ and $\mathbb{L}_{TSL}$, λ is a weighting constant selected as $\lambda = 0.01$, again consistent with Andrew et. al's work.

3.6.2 Results

Experimental results are shown in table 3.3. We compare validation accuracies from the four loss functions ReciprocalSoftmaxLoss 3.4, ReciprocalTripletLoss 3.2, TripletLoss 2.1, and TripletSoftmaxLoss 3.5.

Row	Loss function	Validation accuracy (%)
1	ReciprocalSoftmaxLoss (RSL)	54.01
2	ReciprocalTripletLoss (RTL)	70.03
3	TripletLoss (TL)	60.04
4	TripletSoftmaxLoss (TSL)	**73.77**

Table 3.3: **Loss function comparison.** For implementations of each function, see "utilities/loss.py" in the repository accompanying this paper. All experiments were run on the RGBDCows2020 dataset with standard hyperparameters.

In contrast to Andrew et al.'s findings with RGB data, reciprocal loss is not seen to improve accuracy when depth data is used as input to the network, as seen in rows 1 and 2 of table 3.3. The TripletSoftmaxLoss function saw accuracies increase as far as 75.41% (row 4), deeming it the most suitable loss function to use for the task of identification with depth data alone. For completeness, we give loss curves and embedded space visualisations for all four loss functions in figures 3.12 and 3.13 respectively. In particular, figure 3.12 (b) and (c) highlight the instability of loss metrics when the selected loss function does not incorporate a softmax function. In contrast, figure 3.12 (d) shows that the combination of triplet loss with the softmax function results in a smooth curve with the fastest convergence.

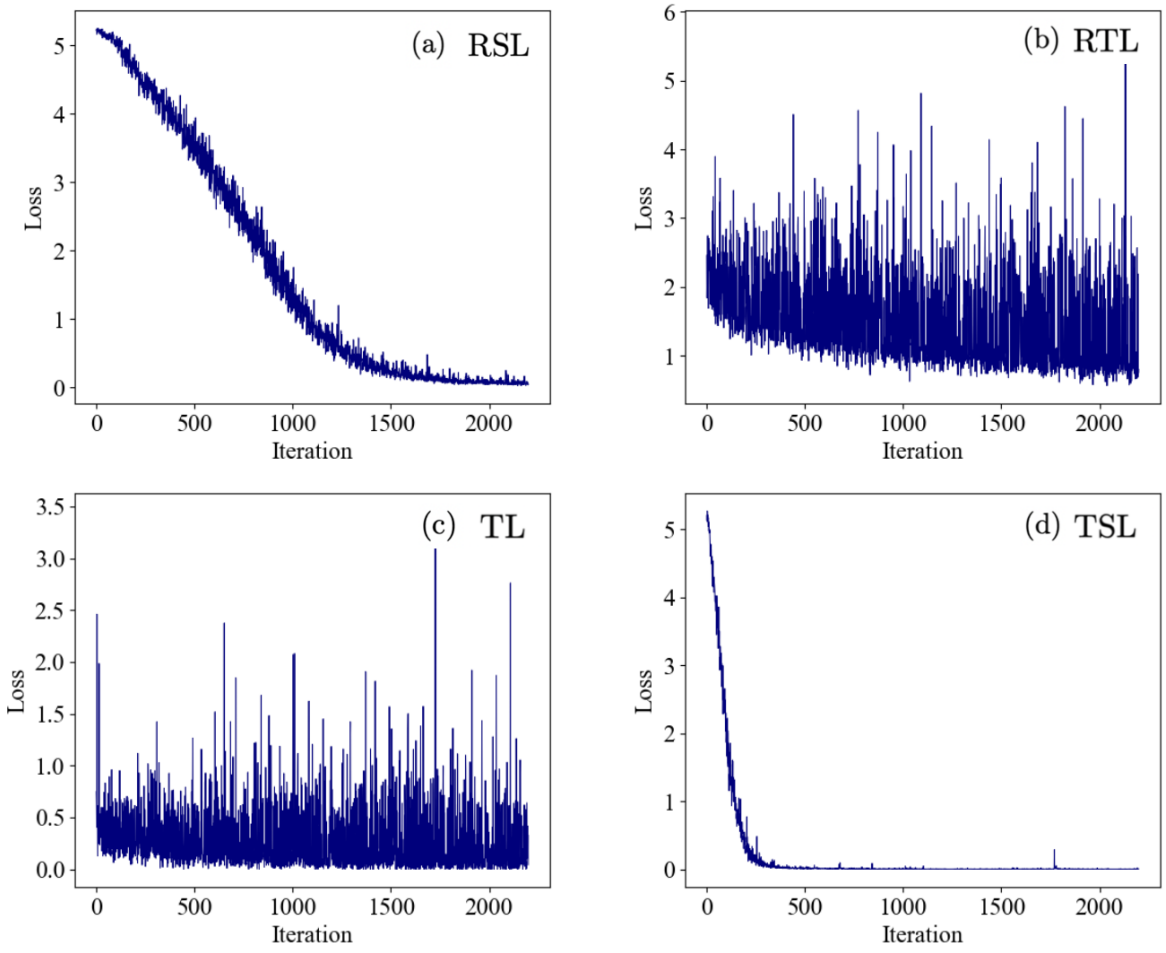


Figure 3.12: **Loss curves of model with different loss functions.** Figures highlight the instability of loss values when the softmax function is not applied within the loss function, in contrast to the smoother curves of RSL **(a)** and TSL **(d)** which both incorporate softmax.

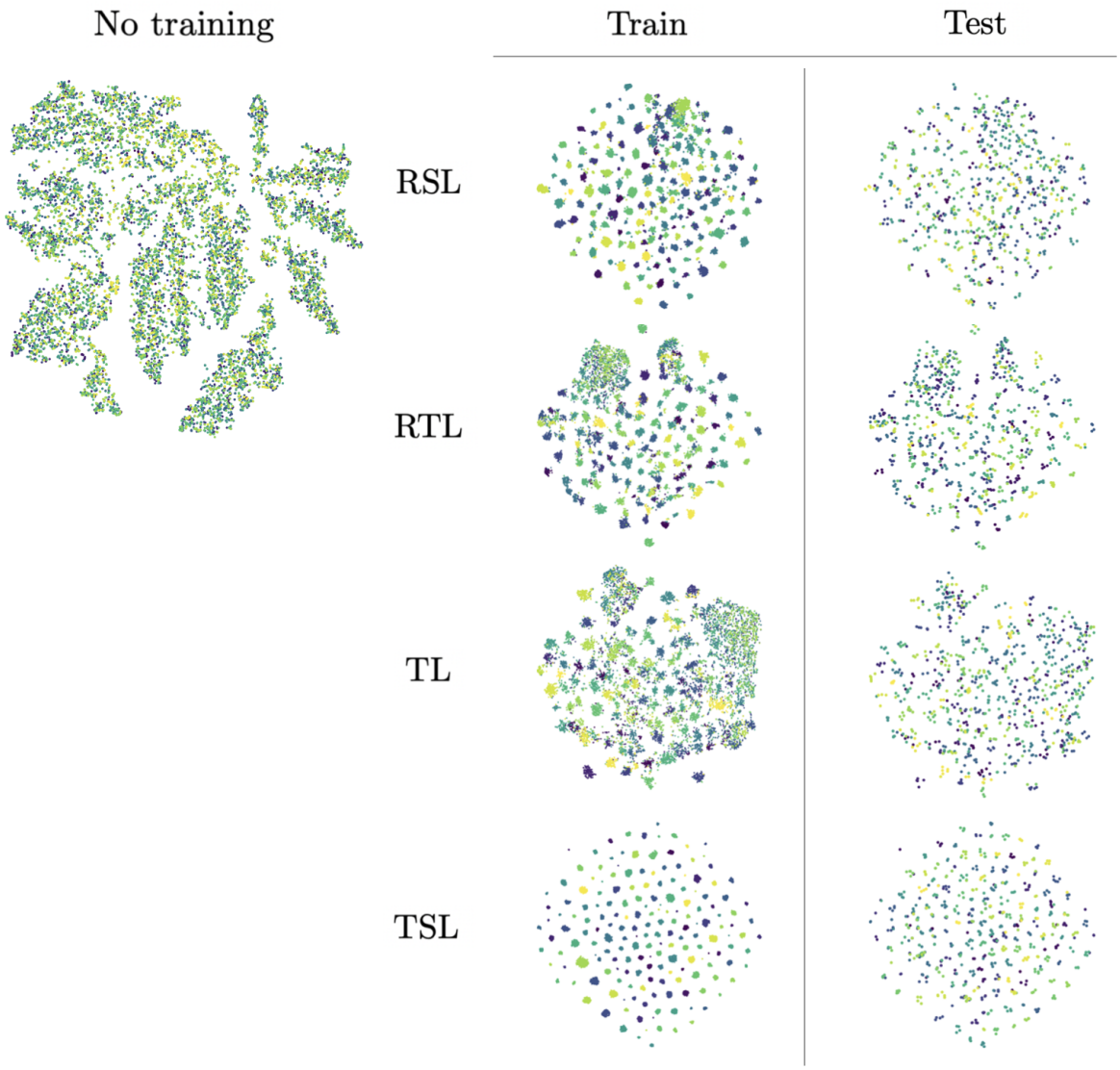


Figure 3.13: **Embedding visualisations for different loss functions.** Visualisations were produced using t-SNE. The top-left shows the embeddings of a model which had not yet performed an iteration with any loss function. The figures in the top-right and bottom-right portion of the figure show training and testing embeddings respectively, produced from the best model state after 50 epochs of training with each loss function.

3.7 Exploring distance measures

As previously explained and shown in formula 2.1, triplet loss requires the use of a distance measure in order to calculate the distance between points in the high-dimensional embedded space to allow for classes to be separated. In recent literature, cosine similarity has been widely adopted in place of the standard Euclidean distance measure, with many researchers in fields such as speech and facial recognition citing an improvement in cluster ID separation [43] [70] [67]. All experiments in this section adapt the standard TripletSoftmaxLoss function described in formula 3.5 and row 4 of results table 3.3.

3.7.1 Motivating cosine difference

All previously described loss functions make use of a Euclidean distance measure, described in equation 3.6. Cosine similarity (to which we refer interchangeably as cosine difference) takes into account not only the distance between each example used in triplet loss but also the angle at which they reside in respect to each other. We define the cosine difference measure in formula 3.7.

$$d_{euc}\left(x_1, x_2\right) = \sqrt{\sum_{i=1}^{n}\left(x_{1i} - x_{2i}\right)^2} \tag{3.6}$$

$$d_{cos}\left(x_1, x_2\right) = 1 - \frac{\sum_{i=1}^{n}\left(x_{1i}x_{2i}\right)}{\sqrt{\sum_{i=1}^{n} x_{1i}^2}\sqrt{\sum_{i=1}^{n} x_{2i}^2}} \tag{3.7}$$

The aim of using the cosine difference measure is to encourage the gradual orthogonality of points which belong to different classes. We first experimented with adapting the TripletSoftmaxLoss function to use cosine difference alone, performing training from scratch. After gaining an accuracy metric of 69.48% which proved the distance measure was capable of training, we conducted experiments using weights from pre-trained networks. A first approach was to load the model state from our best model which used the standard Euclidean distance [3] (row 4 of table 3.3), and then perform additional training using the adapted cosine difference measure. Another experiment was run with the concept of creating a space where all classes are orthogonal through training with cosine difference, before subsequently separating the distances between each class using Euclidean distance. We also experimented with combinations of both distance measure within the same loss function, described in terms of a weighting parameter λ in formula 3.8.

$$d_{combined}\left(x_1, x_2\right) = d_{euc} + \lambda \cdot d_{cos} \tag{3.8}$$

3.7.2 Results

Results drawn from these experiments are given below, where the maximum validation accuracy reached by each experiment is given.

Row	Distance measure		Pre-trained?	Validation accuracy (%)
1	Euclidean		✗	73.77
2			✓ - on run 3	73.55
3	Cosine		✗	69.48
4			✓ - on run 1	**73.98**
5	Combination	$\lambda = 1$	✗	**73.98**
6		$\lambda = 0.1$	✗	73.00

Table 3.4: **Distance measure experiments on network accuracy.** All experiments run with adapted TripletSoftmaxLoss functions, with a ResNet-50 backbone on the 'baseline' network without the SCM component.

From these results we can infer that, contrary to the findings of other works using cosine difference with triplet loss, there was not any substantial improvement in accuracy of the network when trained using cosine difference instead of Euclidean distance. Rows 4 and 5 of table 3.4 show the highest results gained were 73.98%, which are not an increase distinguishable from random noise from the baseline of 73.77%. Surprisingly, the network performance was not seen to improve by any amount distinguishable from random variation even when the network was pre-trained using Euclidean distance as seen in row 4 of table 3.4. We also did not see a substantial increase in accuracy above that gained from using Euclidean distance when training with a combination of Euclidean and cosine difference, given in rows 5 and 6.

3.8 Network backbones

The choice of network backbone affects model performance due to varying levels of complexity and capacity which will directly affect feature extraction capabilities.

[3] model weights hosted at URL

3.8.1 Experimental results

The baseline model proposed in section 3.4 utilised a ResNet-50 backbone with weights pre-trained on ImageNet. We made this decision in order for results to be directly comparable with previous works, however it is of interest to justify the decision to use a backbone of this size through our own experiments. Results using TripletSoftmaxLoss are given in table 3.5 (a full ablation study with all loss functions is given in table 4.1).

Backbone	Validation accuracy (%)
ResNet-50	**73.77**
ResNet-101	72.12
ResNet-152	71.90

Table 3.5: **Effect of backbone on model performance** Experiments were performed to explore how larger network backbones affect learning. The TripletSoftmaxLoss function was used for all experiments.

3.8.2 Discussion

For our combination of network and dataset, increasing the size of backbone does not transpire to better model performance. To reiterate, the use of ResNet with weights pre-trained on ImageNet was a decision passed down from previous work, but of course the success of ResNet is not isolated in the realm of transfer learning; since the publication of ResNet in 2014 [25], there have been many variations and alternatives which have the possibility of improving network performance. Future work could evaluate alternative pre-trained models such as VGG-16 [58], Inception [62], and EfficientNet [63]. Despite the non-exhaustive nature of this study, we can conclude that a ResNet-50 backbone is indeed sufficient for our network in terms of complexity.

3.9 Attention-based Spatial Modules

Motivation. A potential pitfall of the original network is its inability to focus on features which are most influential in discriminating individuals. Spatial context modules allow networks to focus on the most relevant regions of an image and to weight the contribution of each region in the identification process. By building a spatial context module into the network before the fully connected layers (shown in the green section of figure 1 and with more detail in figure 3.14), the intuition is that the accuracy could be boosted by homing in on the most important aspects of each image, namely the spine structure of individuals. It is possible that the module will even pick out feature elements which we would not have considered to be of importance. Previous works have had great success with this approach, specifically in the detection of great apes from camera trap footage [68]. The SCM in this case helped with the difficulties faced due to occlusion in camera footage; it could be the case that this extends to cow depth imagery since there are some obvious anomalies in the images due to imperfections in capture of the camera, as section 3.1 explains. The spatial context module has the possibility of ignoring these imperfections in the input images to focus only on the important features. A further discussion is given in section 3.9.3.

3.9.1 Module architecture

The conceptual idea behind spatial context modules is to allow networks to learn contexts in which each feature appears, enabling them to take relationships between features into account as well as just the presence or absence of the features themselves. A common approach to this is implementing a technique called self-attention, which allows the weighting of different parts of images according to their perceived importance. This is achieved through the computation of a set of weights for each feature based on the relationship between this feature and others in the same image. Yang et al. [68] describe this as a spatial "blending" process, visualised in figure 3.14.

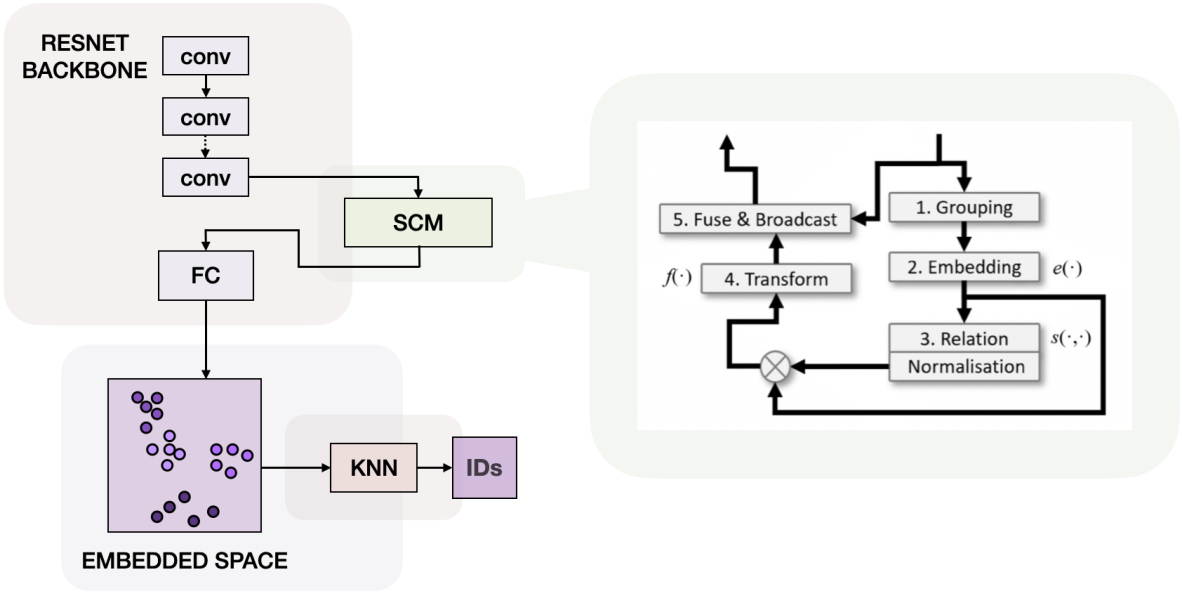


Figure 3.14: **Simplified network diagram including SCM component.** SCM component diagram (right) is taken from [68]. 1. Features are grouped on spatial location. 2. Each part of the feature map is embedded. 3. A correlation map is produced, then applied to the embeddings via matrix multiplication denoted as $\otimes$. 4. The result is transformed and fused 5. with the original input to produce the blended output.

3.9.2 Results

Table 3.6 shows the improvements in validation accuracy gained when training the model on RGBDCows2020 with the SCM component added to the network as described above. An improvement of around 1% was shown consistently through experiments. The right-most column of table 3.6 shows improvements on model accuracy when training with the dataset containing augmented images described in section 3.5; a validation accuracy of 75.63% was attained with this experimental setup.

	RGBDCows2020 (orig.)	RGBDCows2020 (w/ aug.)
Baseline	73.77%	74.53%
SCM	**74.20%**	**75.63%**

Table 3.6: **Effect of SCM on model performance.** Comparison of model performance with and without an SCM component before the fully connected layer. Experiments were run on both the original dataset (column 1) and the extended dataset with augmentations (column 2) explained in section 3.5.

3.9.3 Discussion

In terms of percentage increase in validation accuracy, improvements gained from data augmentation and the addition of an SCM are comparable, which points towards a possible conclusion that both methods are solving the same issue. One explanation could be that the SCM can eliminate errors due to black spots in the input images, which could also feasibly be the same problem that the data augmentation is solving. An alternative hypothesis, noting of course that the true explanation may be a combination of these reasons, is that the increase in accuracy seen through the addition of the SCM is explained by the added capability of the network to detect rotations. By their nature, CNNs do not inherently possess the ability to recognise rotations or orientations in images; convolutional layers rely on only spatial arrangement of features for extracting information, which results in an inability to detect rotation. SCMs provide an avenue for combatting this limitation, since their learnable parameters can be used to manipulate the spatial arrangement of features within the network. This means it has the capability to augment the data, possibly through rotations and scaling, during training.

3.10 Hi-Res Data Experiments

This section outlines the methodologies and results of training models on the newly created CowDepth2023 dataset described in section 3.2. We experiment with network performance and evaluate the comparability of results due to temporal bias between examples.

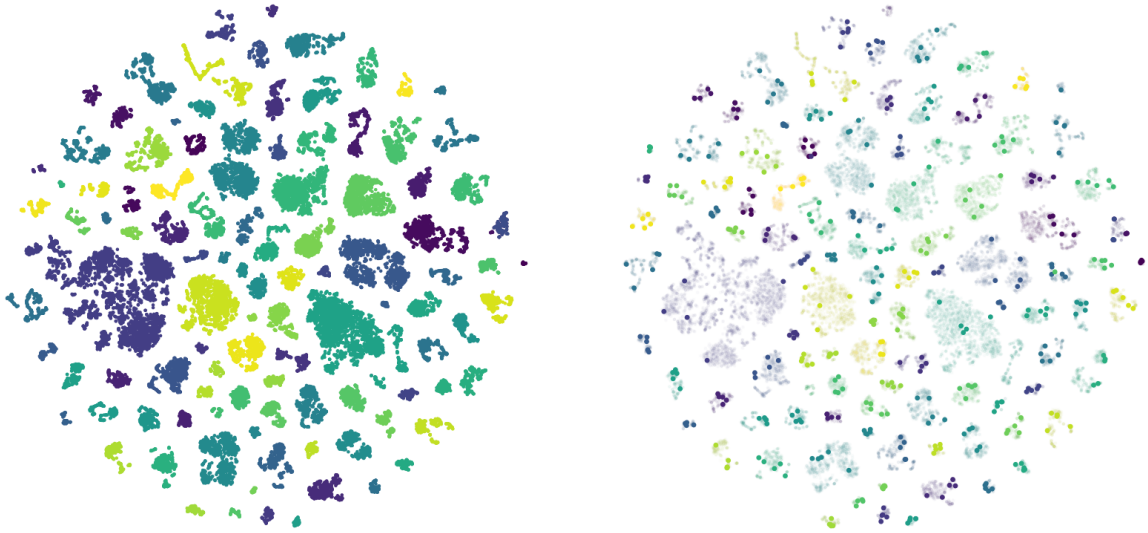
3.10.1 Preliminary Results

Table 3.7 compares validation accuracies resulting from training models on RGBDCows2020 versus CowDepth2023. For completeness, results from the addition of an SCM component in the network are shown in row 2 as well as a baseline result without this component. The results quoted for CowDepth2023 are the result of removing two images from the training set either side of each test example (see section 3.10.2 for explanation). We visualise these results before conducting an experiment into their reliability in section 3.10.2.

	RGBDCows2020	CowDepth2023
Baseline	73.77%	96.01%
SCM	**74.20%**	**97.21%**

Table 3.7: **Dataset performance.** Comparison of validation accuracies when the same network is trained on RGBDCows2020 and CowDepth2023.

Inferring embeddings. Figures 3.15a and 3.15b respectively depict the training and testing embeddings projected onto a two-dimensional space for the preliminary training of this 100-cow dataset. It's clear to see that the model has succeeded in projecting each individual onto a different space in the embedding.

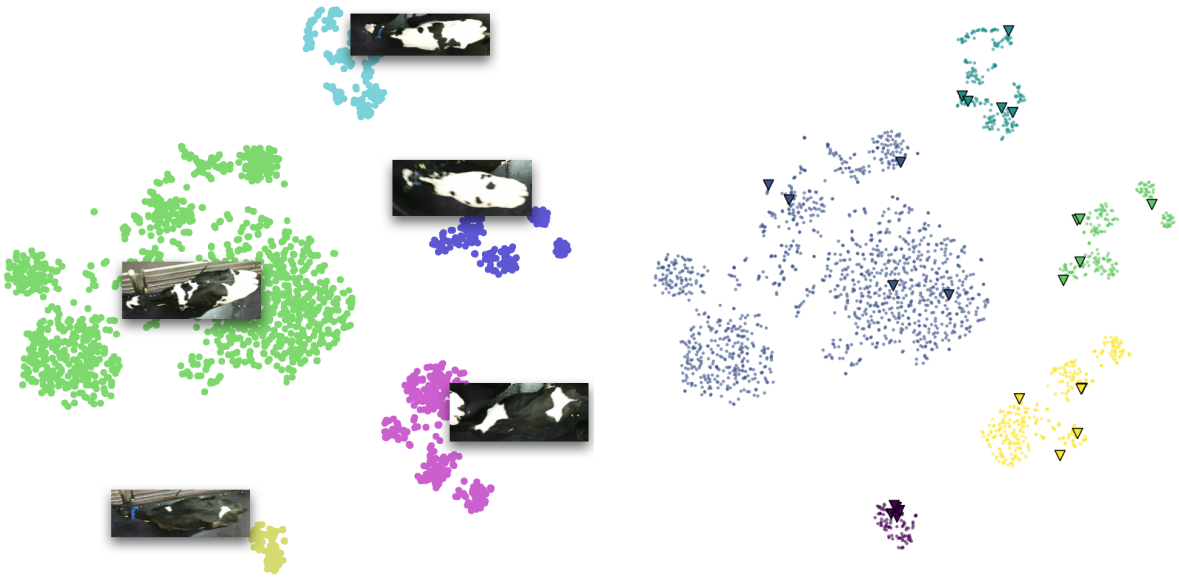


(a) Train embeddings

(b) Test embeddings

Figure 3.15: **t-SNE embedding visualisation plots for CowDepth2023 dataset.** Points have been projected down to a 2-D space using t-SNE dimensionality reduction. **(a)** shows how training set embeddings have clustered into ID groups, and **(b)** shows training embeddings plotted with a low alpha value so that we can see test examples plotted on top. Colours correspond to individual IDs; notice how test examples of each class are projected onto clusters of corresponding colours due to classification success.

In order to more clearly visualise how each individual is projected into the embedded space, figure 3.16 gives a t-SNE plot produced by training a model on only five cows from the dataset (individuals #000 through #004).



(a) **Train embeddings.**

(b) **Test embeddings.**

Figure 3.16: **t-SNE plots of five individuals. (a)** shows the training embeddings of the five cows, where RGB images have been placed on top of the projections of each corresponding cow in the small five-cow dataset used for training. The colour of each plot corresponds to each individual. **(b)** shows the training embeddings (circle markers) with test set embeddings plotted on top (triangle markers), with five clusters corresponding to the five cows tested on. Each cluster has five plots since five test examples were used for each individual.

3.10.2 Temporal bias removal

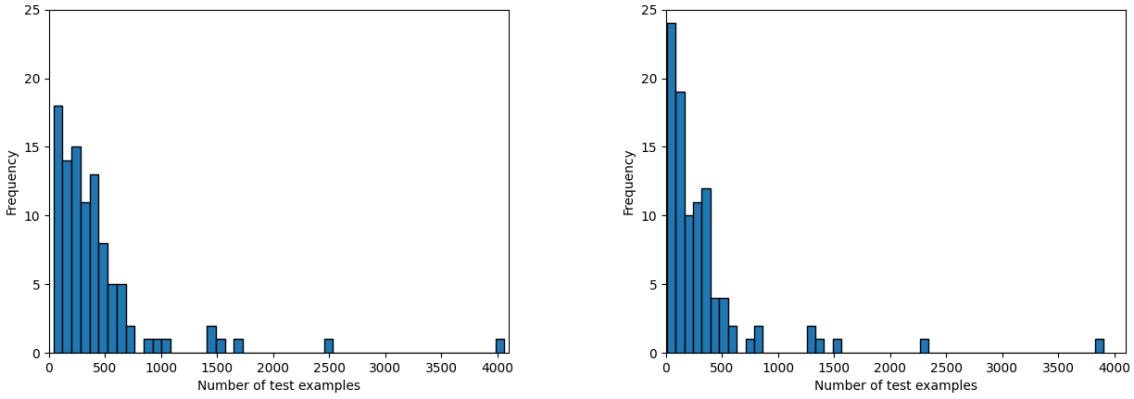
Methodology. When the model was initially trained on the dataset, a validation accuracy of 97.60% was attained; due to the sequential nature of the dataset it was important to explore to what extent temporally neighbouring examples differ. So-called "leave-sequence-out" training was performed on the model, such that two temporally neighbouring images either side of each test image were removed from the training set if they were found to exist. This method aimed to prevent the likelihood of a test example having a temporally adjacent image in the training set that was similar enough to the test example to suggest that the model had already learned from that image. We henceforth refer to this phenomenon as "temporal bias". Results from these experiments are shown in table 3.8, where n is the number of samples left out of the training set surrounding each test example.

n	Min. timestep (ms)	Avg. training examples	Accuracy
0	34	426	97.60%
2	102	408	95.41%
4	170	392	93.81%
8	306	362	81.04%
16	578	330	69.54%

Table 3.8: **Leave-sequence-out experiment results.** Comparison of validation accuracies where n examples either side of each test image temporally are removed from the training set. The minimum timestep column refers to the respective shortest lengths of time there can be between each test example and its temporally neighbouring training examples in each trial, given that the Kinect sensor captures at a rate of 30Hz (roughly one image per 34ms). This is a minimum rather than an absolute value since datasets are not made up of entirely sequential images, so often timesteps are larger than the minimum.

Result discussion. Clearly, there was truth to the intuition that the sequential nature of the data was providing bias in the form of the model being trained on images which were very close temporally to the test examples. We should note here that due to the high variance in number of examples per class, results for $n = 16$ were taken from only 95 of the total 100 cows since individuals had to be removed in the cases where the training example removal process resulted in too few examples to be able to perform

triplet loss. The process also resulted in some cows having very few training examples, which could be a contributing factor to the reduction in accuracy as n decreases. We provide a histogram of number of examples per class for the original dataset ($n = 0$) compared with $n = 16$ in figure 3.17. To get a more concrete idea of how reducing the number of examples per class affects training accuracy we conducted a test where $n = 8$ and the average number of training examples were consistent with row 4 of table 3.8, but instead had random examples removed from the training set as opposed to specific temporally relevant examples as before. This resulted in a validation accuracy of 96.8%, which is a decrease from the baseline $n = 0$ of only 0.8%. This result shows that some of the decrease seen as we traverse the rows of table 3.8 is simply due to fewer training examples per class, however the decrease solely caused by removing temporal bias was far more substantial.



(a) $n = 0$ histogram.

(b) $n = 16$ histogram.

Figure 3.17: **Histograms showing number of training examples per cow in different experimental datasets.** The y axis refers to number of individuals which have a number of training examples within each bin.

Inferring comparability of results. Due to the conclusion that a large degree of temporal bias exists in the CowDepth2023 dataset, we cannot meaningfully compare results of experiments with those of RGBDCows2020; this means that the impact of a higher image resolution remains unknown. We therefore conclude that our network is capable of attaining accuracies in the range of 68.70% to 97.21% on the CowDepth2023 dataset and emphasise that this range is dependent on removal of temporal bias between examples.

3.10.3 Ablation study

Despite rows of the table being independent due to different numbers of training examples, we provide a final ablation study comparing network performance on CowDepth2023 with and without the SCM component. The addition of an SCM saw accuracies increase across almost all values of n; results of experiments are given in table 3.9.

n	**Avg. training examples**	**Accuracy without SCM (%)**	**Accuracy with SCM (%)**
0	426	97.60	97.21
2	408	95.41	96.81
4	392	93.81	94.41
8	362	81.04	82.04
16	330	69.54	68.70

Table 3.9: **Ablation study for SCM performance.** n refers to the number of training examples removed from either side of each image used in the test set. Accuracy metrics are taken as highest validation accuracy achieved when run with standard hyperparameters described in section 3.3, using tripletSoftmaxLoss.

3.11 Localising sources of individuality

Here we present a method for leveraging our network to extract information with the prospect of sparking veterinary discussion. We emphasise the novelty of our research path and discuss how our datasets and network facilitate evaluation studies relating specifically to cattle conformation.

3.11.1 Concept

We implemented an algorithm[4] inspired by grad-CAM [54] to identify and visualise where class-specific information lies spatially in the input images. The hope here is to gain insight into which physical features of each individual are being learned by our model as individually discriminating. To the best of our knowledge, no existing research has been conducted on which features of the conformation of a cow's back facilitate individual identification.

3.11.2 Implementation

Firstly, the model state was loaded in from our best model described in 4.2 trained using TripletSoftmaxLoss. We modified the model such that the network gradients get saved during execution to allow them to be used in visualisation. The network was also modified to return only the softmax function such that we had access to the logits of each prediction over every class; existing work has been done into using embeddings to calculate the probability of the example belonging to each class [12], however for our purpose we found that taking the softmax loss was enough. Then using a randomly chosen depth image from the RGBDCows2020 dataset, we performed a forward pass through the network. Since triplet loss requires three images, we simply passed the same image duplicated three times into the network. A single backwards pass was completed in order to 'hook' out the gradients of the network. Formula 3.9 shows that to obtain the importance of each neuron w^c, the gradient of the score for class c, y^c with respect to the activations of the final convolutional layer A were average pooled. [5]

$$w^c = AvgPool(\frac{\delta y^c}{\delta A}) \tag{3.9}$$

Formula 3.10 describes how the heatmap H was then created using a linear combination of the pooled gradients w^c with the activations A, which were then passed through a ReLU function.

$$H = ReLU(\sum w^c A) \tag{3.10}$$

The resulting heatmap H was superimposed onto the original depth image to produce the final heatmap images seen subsequently.

3.11.3 Resulting visualisations

Spinal attention. Example results of this method are shown in 3.18, which uses an image from the RGBDCows2020 dataset. The heatmap superimposed onto the original corresponding RGB image given in 3.18 (d) shows that the model was most 'focused' on the horizontal center of the depth image, which is evidently where the spinal cord of the animal is most prominent. Interestingly, the model does not seem to give much attention to the bodily shape of the cow, shown by the blue edges of the heatmap. Instead, the focus appears to be on the spinal structure above the cow's front two legs, shown by the dark red patch. Figure 3.19 shows four depth images (left) of cow #000 from RGBDCows2020 with their respective heatmaps superimposed. Each example used for this figure was identified correctly by the model we use for inference. The image on the right shows all four images superimposed onto each other, to highlight the visual pattern which is stable across examples. We provide a further evaluation of this study in section 4.3.

[4] Source code for Grad-CAM implementation hosted at [redacted]

[5] Using notation from the original Grad-CAM paper [54]

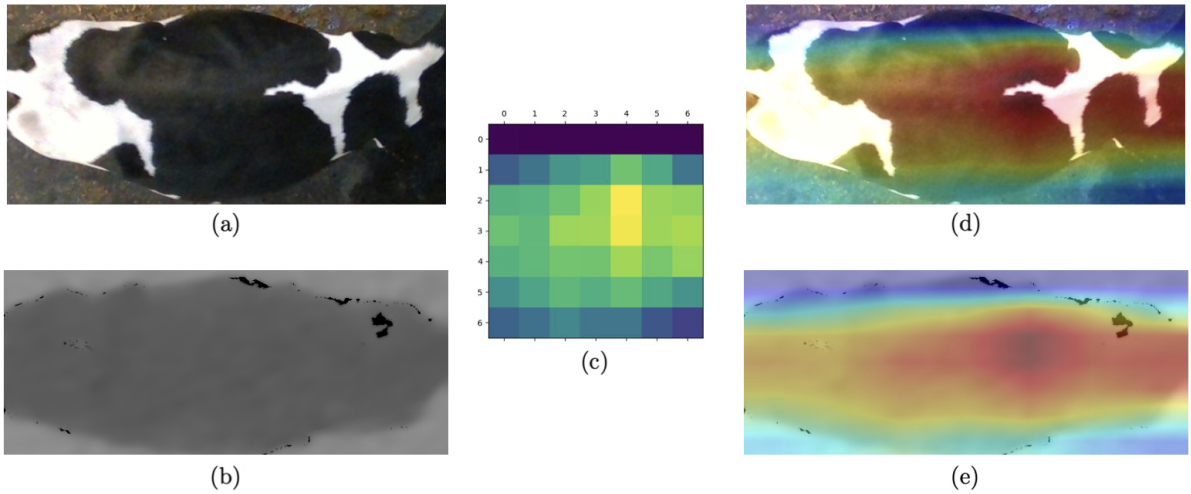


Figure 3.18: **Grad-CAM gradient visualisation.** **(a)** and **(b)** respectively show corresponding RGB and depth images of a cow, chosen at random from the RGBDCows2020 dataset. Depth image **(b)** was fed through a trained network, with weights recorded to produce the heatmap shown in **(c)**. **(d)** and **(e)** respectively then correspond to images **(a)** and **(b)** with the heatmap superimposed onto them.

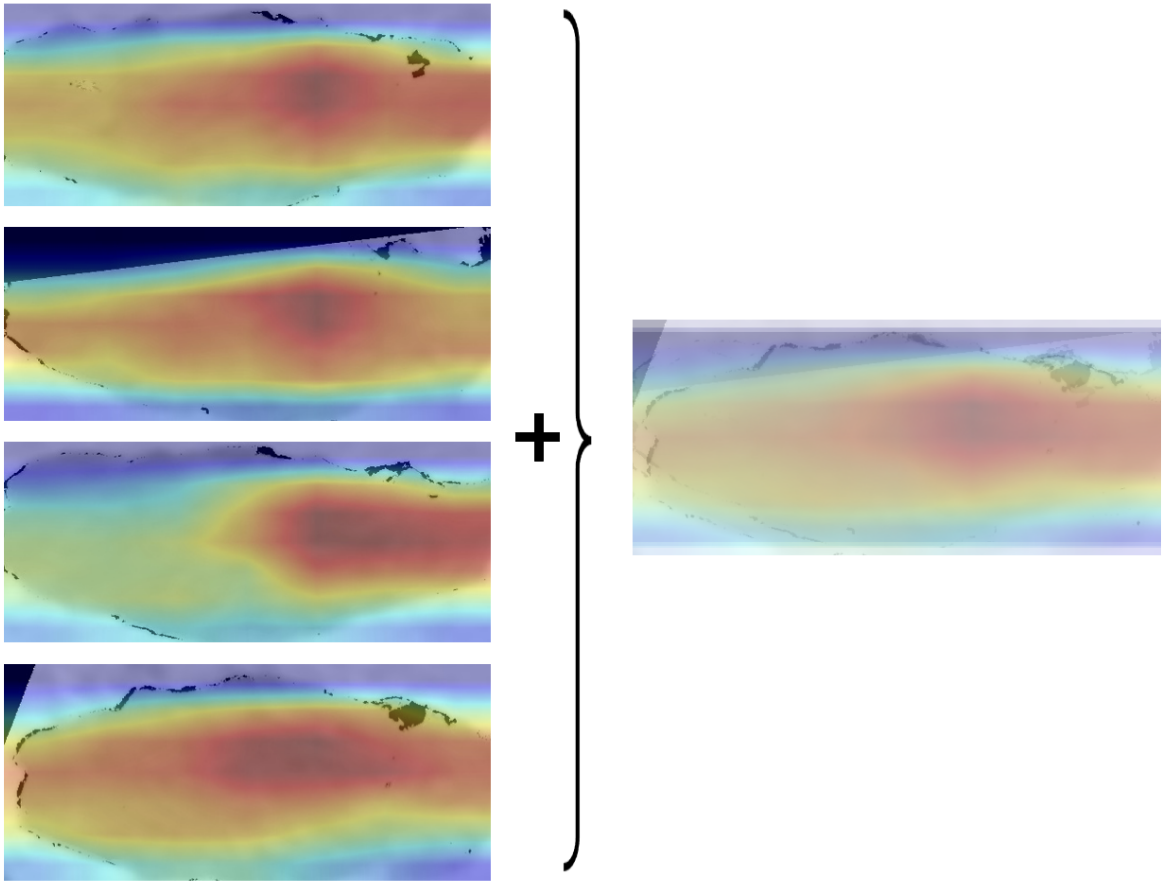


Figure 3.19: **Grad-CAM visualisations of cow #000 from RBGDCows2020.** We give four depth images (left) randomly selected from one individual which were passed through the Grad-CAM algorithm to produce gradient images. Each of these four visualisations are superimposed onto one general representation shown on the right.

Chapter 4

Critical Evaluation

This project's open-ended research aims allowed for an evolving progression of goals. This section presents an evaluation of the results obtained, the research process that facilitated these findings, and a comparison of the decisions made throughout the project's execution.

4.1 Collating quantitative network experiment results

Here we give summary tables collating our quantitative results. Some columns are incomplete due to restrictions in terms of supercomputer scheduling. Unless otherwise stated in table headings, experiments were run using default hyperparameters described in the experimental setup section.

		Accuracy (%)			
		RGBDCows2020			**CowDepth2023**
		resnet-50	resnet-101	resnet-152	resnet-50
Loss fn	ReciprocalSoftmaxLoss	51.59	50.49	50.01	97.21
	ReciprocalTripletLoss	70.03	61.69	67.29	95.61
	TripletLoss	60.04	56.31	56.86	94.21
	TripletSoftmaxLoss	**73.77**	72.12	71.90	**97.80**
TL dist. measure	Euclidean	73.77			
	Cosine	69.48			
	Combination ($\lambda = 1$)	**73.98**			
	Combination ($\lambda = 0.1$)	73.00			
Augmentation	Original	51.59			
	Rotation + zoom (30)	53.57			
	Rotation + zoom (60)	**54.67**			
	Rotation + zoom + GaussNoise	53.79			

Table 4.1: **Loss functions and augmentations.** All results were run on 'baseline' network i.e., not including the SCM component, in order that they are directly comparable with previous works. 'TL dist. measure' relates to section 3.7, each of which adapt the TripletSoftmaxLoss function. Augmentation variations were run using the ReciprocalSoftmaxLoss function with the Euclidean distance measure.

			Accuracy (%)	
		Avg. training examples	**Baseline** network	**Proposed** network
Minimum timestep n	$n = 0$	426	97.60	97.21
	$n = 2$	408	95.41	96.81
	$n = 4$	392	93.81	94.41
	$n = 8$	362	81.04	82.04
	$n = 16$	330	69.54	68.70

Table 4.2: **SCM ablation.** CowDepth2023 was used to show the effects of removing the SCM component from the network. 'Baseline network' refers to the network without the SCM, and 'proposed network' refers to the network with the SCM. All experiments use TripletSoftmaxLoss.

4.2 Evaluating our best model

Direct improvement. From our evaluation of quantitative results, it is clear that the TripletSoftmaxLoss loss function with a standard Euclidean distance measure and resnet-50 backbone is the optimum combination of parameters. Training a model on the RGBDCows2020 dataset with this setup resulted in a validation accuracy of 75.63%, which is a directly comparable improvement of 15.01% from the highest accuracy quoted by previous work.

Visualising embeddings. To demonstrate our best model's efficacy in distinguishing classes, we can visualise the embeddings produced during training, saved at the point of optimum validation set performance. Figure 4.1 shows such training and testing embeddings using t-SNE dimensionality reduction, in order to highlight the extent to which individuals have clustered in the embedded space. It is possible to see the general trend of test examples being projected onto their respective clusters, however it is also possible to see examples of test points projecting onto the incorrect cluster. An example of incorrect classification can be seen to the bottom of the yellow cluster in the second magnification (bottom right), where a green point and a blue point reside.

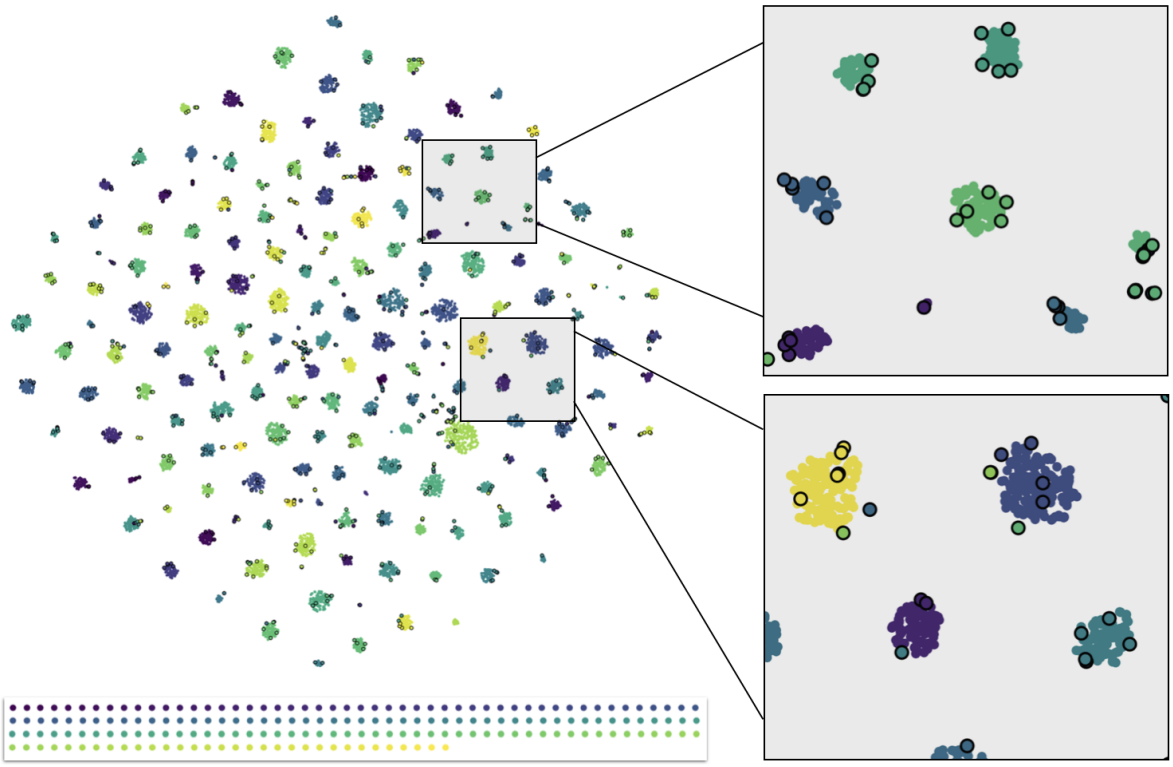


Figure 4.1: **Embedding visualisation of best performing model.** t-SNE dimensionality reduction has been performed on both the training and testing embeddings. Training examples are shown by filled circles while test examples correspond to filled circles with black borders. Colours of points correspond to the 182 individuals in the RGBDCows2020 dataset, with each discrete colour shown in the bottom left. We magnify two parts of the embedded space to highlight the extent to which classes have clustered together, with corresponding test examples being projected onto their respective identity clusters.

Suitability in application. While our validation accuracy result is an improvement on previous work, it's important to consider the suitability of the model in the context of real-world application. A model with a 75% accuracy rate is unlikely to be sufficient for deployment in a practical farm setting, as the consequences of misidentification could be significant in terms of animal welfare and economic loss. In section 5.3 we discuss opportunities for future work to improve on our network and suggest possible avenues for further experimentation.

4.3 Qualitative study evaluation

From our research into how each individual is identifiable through depth imagery, visualisations showed high inter-class diversity in the spinal patterns of the cattle, specifically on the part of the back which is above the two front legs. Before conducting this research, it was unknown whether our model would 'focus' on the bodily shape of individuals rather than their spinal structure, but it appears that examples are distinguishable from spinal structure alone. In particular, our model seemed to take interest in the regions containing the thoracic vertebrae. Figure 4.2 shows the spinal structure of a cow, highlighting the thoracic region in label 5.

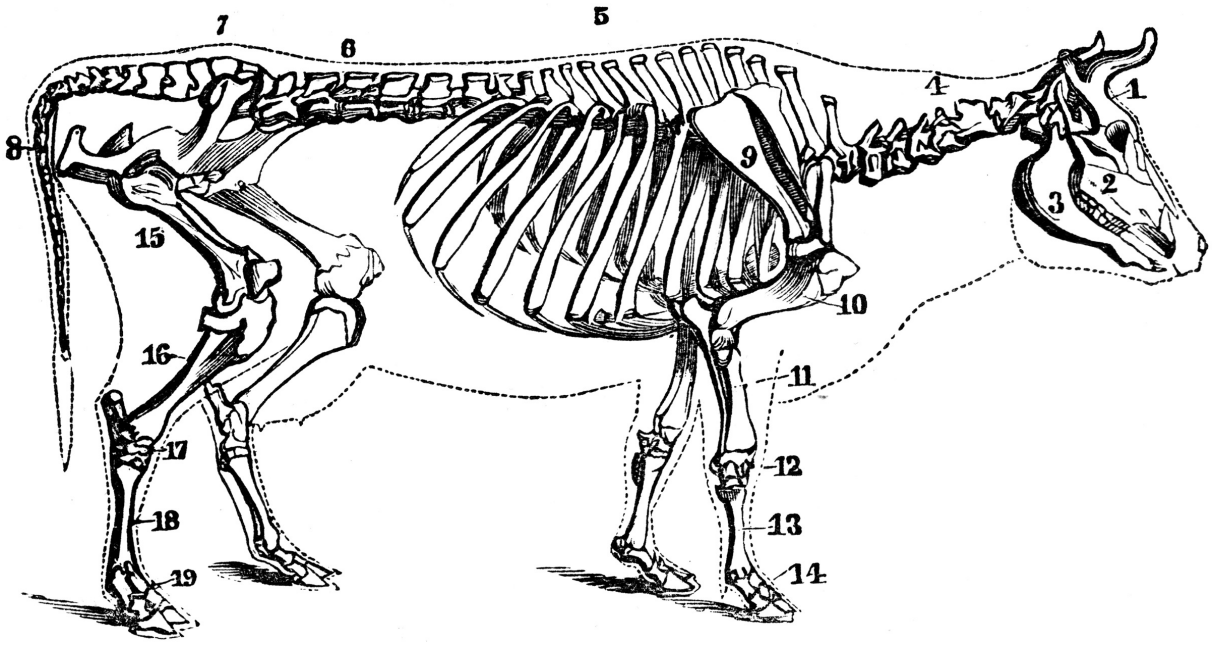


Figure 4.2: **Skeletal structure of a cow.** Diagram from Calvin Cutter's first book on analytic anatomy (1872) [14]. Label 5 (top) shows the location of the thoracic vertebrae of the cow, which our model highlights to be a possible biometric identifier between individuals.

Result validity. To corroborate the validity of heat-map visualisations, it would be of value to use a dataset which includes images of cattle at different orientations. This way, it would be possible to confirm that the localisation of the sources of individuality which we identified are not just artefacts of the network paying more attention to the center of the image, which we acknowledge as a possibility.

4.4 Suitability of datasets

Our research experiments were facilitated by both the RGBDCows2020 and CowDepth2023 datasets, each presenting their own limitations impacting the success of training, as we detail in the following summary.

RGDBCows2020. As discussed in section 3.1, the RGBDCows2020 dataset had multiple drawbacks which contributed to limitations in model training. Firstly, artefacts which were not present in the real frame appeared frequently in images due to imperfections in the image capture process as previously shown in figure 3.3. Secondly, and most critically, images in the RGBDCows2020 dataset are of 8-bit resolution due to having been previously saved in .jpeg format. This meant that there was far less variation of pixel values in each image than that of the raw captured data, causing the model to not be provided with important distinguishing features of each individual. This was the key aspect which we sought to combat with the creation of the CowDepth2023 dataset, which consists of 16-bit images.

CowDepth2023. The creation of CowDepth2023 was an interesting path for the project to take, not only for justification of how results translate between different individual cattle but to explore the benefits of higher resolution data. While it was undesirable for accuracies to decrease for higher values of

n during experimentation with temporal bias removal making comparisons between datasets difficult, more avenues appear for research into whether sequential data streams might be the way forward for identification tasks rather than still frames alone.

Future data. It would be of great benefit for future work to combine the successes of RGBDCows2020 and CowDepth2023 in the creation of a dataset which is of high resolution and also addresses the issue of temporal bias. Since both datasets used in this paper comprise of individuals from the Wyndhurst farm, it would be of interest to create a dataset of an entirely different herd to explore whether accuracies withstand this difference in classes. Ultimately, it would be of great benefit to test the depth data accuracies on a breed of cattle such as Black Angus, which have little-to-no discriminatory markings.

4.5 Discussion of project decisions

Shift in project focus. Every experiment conducted during the execution of the project offered interesting results and raised potential avenues for further investigations. In hindsight, it became clear that the choice of loss function had the greatest impact on the accuracy of our model and gave rise to perhaps the most impactful avenues of research, which suggests that a more in-depth exploration of loss function choice could have been fruitful.

Methodology drawbacks. With the benefit of hindsight, there are also some changes to decisions made which are beneficial to note should further research be done on related topics. For example, our results in sections 2.2.2 and 3.5 which involved experimenting with different data (respectively the removal of temporally related images, and the addition of augmented images) suffer from a relatively small number of experiments conducted. This was due to limitations in terms of time and storage space to carry out each experiment, since we chose to create the required dataset for each experimental run ahead of time, rather than performing the image selection/augmentation processes during training. The latter approach could have increased the scope for experimentation and hence enhanced results gathered.

Chapter 5

Conclusion

5.1 Contribution summary

Our primary objective throughout project execution has been to develop existing work on the identification of individual cattle using RGB imagery to produce a network capable of identifying cattle through depth imagery. The key motivation behind this effort was to enable individual identification of unmarked herds. In this paper, we have provided a comprehensive evaluation of the efficacy of depth imagery in the visual identification of individual cattle.

A network was developed and proposed, primarily using RGBDCows2020 as a means of experimentation. It was found that the addition of a spatial context module (section 3.8) before the fully connected layers results in increased model accuracy, as does spatial augmentation of the input images including rotations and zooms. In-depth discussions surrounding advantages and disadvantages of all experiments were given under each section, which pave the way for further research in each area. We presented a new dataset, CowDepth2023, specifically aimed at addressing the drawbacks of RGBDCows2020, with dataset collection methods described in section 3.2. This led to experiments with temporal bias removal, allowing discussions about the extent to which the timestep between training and test examples creates bias in model performance. Before this work was carried out, the question of whether depth imagery alone could be used for identification was open; our results have shown that there is conclusively enough information encoded in individual cow's spinal patterns and body shapes for them to be discriminated via these metrics alone.

We then looked further into what makes the spinal patterns of cattle an identifying feature, by using gradient visualisation algorithms in section 3.11. However, there is much more to be done in terms of improving our methodologies to make them applicable to real-world use-cases, namely improving data collection methods so that models have a suitable quality of data to train on. As with any scientific research, our work has opened more questions than it has answered; there are many avenues for research, experimentation, and verification to be completed, the most prevalent of which we outline below.

5.2 Quantitative result summary

Previous unfinished work quoted an accuracy of 60.62% when trained on the RGBDCows2020 depth data alone, which we treated as a baseline metric for our research to improve upon. Experiments with loss functions, data augmentation, and network advances resulted in an immediately comparable accuracy improvement to 75.63% when training our best model on the RGBDCows2020 dataset. The CowDepth2023 dataset, dependent on temporal bias, reached accuracies within the range of 69.54% to 97.60%.

5.3 Future work

The results from this research have demonstrated that there is certainly scope for deep learning applications to use depth imagery alone to classify individual cattle in a herd. The accuracy achieved by our proposed network suggests that it could be a viable solution for improving efficiency in the cattle farming industry, however further research is needed to fully evaluate the practicalities and extendibility of the

approach to real-world applications.

Variation of input imagery. Due to static data collection methods, our current approach relies on a fixed camera position. In order for this to be scalable in large farming operations, it would be important to develop the technique to handle 360-degree drone footage to enable identification from any angle. It also isn't clear from our experiments how well the system would hold up to variation in input depth imagery over time. For example, an interesting test would be to train a model using data containing images of cows in different situations, orientations, and locations rather than simply data from a single sequential image stream such as the one in our data collection method. If it transpired that the model performed poorly at inference time then further work would need to be completed in order to ensure the model is more robust to external factors such as change in environment.

Combining data sources. It could also be beneficial to explore whether the addition of other forms of input data would benefit the system. For example, developing a network where predictions are computed through the combination of RGB and depth inputs could result in a more robust system which would extend well to other herds. In a highly instrumented farming setting, it could even be useful to explore whether the use of additional sensors such as temperature could provide more contextual information about the cows and their environment, allowing further insights about the well-being of each cow. Extending the identification system in ways such as this would represent a significant advancement towards automating the monitoring of livestock welfare.

Additional data for study confidence. It would be of great benefit to test our proposed network on a dataset comprised of an entirely black herd. We have carried out our research on the intuitive assumption that depth imagery is intrinsically independent of changes in coat pattern. However, it would be of value to sanity-check this assumption through testing on a totally unpatterned herd to have total confidence that factors such as specular reflection do not influence the data captured by the depth sensors, which would lead to bias due to our model learning coat patterns instead of the intended dorsal features alone.

Economic implications. Evaluation of the economic viability of implementing our approach in real-world operations is also an avenue which should be explored, requiring a cost-benefit analysis which would take into account the cost of installing camera systems, the time required for model training, and the potential labour savings achieved by implementing the system.

Veterinary applications. Aside from applications in the agricultural industry, there are interesting veterinary science applications which could be explored. Our work paves the way for research into how bone structures of bovines differ not only from cow-to-cow but also between herds. Many interesting avenues could be explored using this depth imagery technique, for example exploring whether there are similarities in the bone structures of genetically related cows. This not only would allow for automated tracking of family groups but could be used in veterinary research relating to bovine bone structures.

Bibliography

[1] Advanced computing research centre. URL: https://www.acrc.bris.ac.uk/acrc/phase4.htm.

[2] Python package index - pillow. URL: https://pypi.org/project/Pillow/.

[3] Best alternative to cow rfid: Plate recognizer alpr, Mar 2023. URL: https://platerecognizer.com/best-alternative-to-cow-rfid/.

[4] Michael Riis Andersen, Thomas Jensen, Pavel Lisouski, Anders Krogh Mortensen, Mikkel Kragh Hansen, Torben Gregersen, and PJAU Ahrendt. Kinect depth sensor evaluation for computer vision applications. *Aarhus University*, pages 1–37, 2012.

[5] Will Andrew, Sion L Hannuna, Neill W Campbell, and Tilo Burghardt. Automatic individual holstein friesian cattle identification via selective local coat pattern matching in rgb-d imagery. In *2016 IEEE International Conference on Image Process (ICIP 2016)*, Proceedings of the IEEE International Conference on Image Processing (ICIP), pages 484–488, United States, March 2017. Institute of Electrical and Electronics Engineers (IEEE). 2016 23rd IEEE International Conference on Image Processing, ICIP 2016 ; Conference date: 25-09-2016 Through 28-09-2016. doi:10.1109/ICIP.2016.7532404.

[6] William Andrew. Rgbdcows2020. 2020.

[7] William Andrew, Jing Gao, Siobhan Mullan, Neill Campbell, Andrew W. Dowsey, and Tilo Burghardt. Visual identification of individual holstein-friesian cattle via deep metric learning. *Computers and Electronics in Agriculture*, 185:106133, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0168169921001514, doi:https://doi.org/10.1016/j.compag.2021.106133.

[8] Amir Atapour-Abarghouei and Toby P Breckon. A comparative review of plausible hole filling strategies in the context of scene depth image completion. *Computers & Graphics*, 72:39–58, 2018.

[9] Ali Ismail Awad. From classical methods to animal biometrics: A review on cattle identification and tracking. *Computers and Electronics in Agriculture*, 123:423–435, 2016.

[10] Manlio Bacco, Paolo Barsocchi, Erina Ferro, Alberto Gotta, and Massimiliano Ruggeri. The digitisation of agriculture: a survey of research activities on smart farming. *Array*, 3:100009, 2019.

[11] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. URL: https://www.mdpi.com/2078-2489/11/2/125, doi:10.3390/info11020125.

[12] Lei Chen, Jianhui Chen, Hossein Hajimirsadeghi, and Greg Mori. Adapting grad-cam for embedding networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2794–2803, 2020.

[13] Brian Coffey, James Mintert, John A Fox, Ted C Schroeder, and Luc Valentin. The economic impact of bse on the us beef industry: product value losses, regulatory costs, and consumer reactions. 2005.

[14] Calvin Cutter. *First Book on Analytic Anatomy, Physiology and Hygiene*. BoD–Books on Demand, 1872.

[15] Fred J DeGraves and John Fetrow. Economics of mastitis and mastitis control. *The Veterinary Clinics of North America. Food Animal Practice*, 9(3):421–434, 1993.

[16] WT Disney, JW Green, KW Forsythe, JF Wiemers, and SJRT Weber. Benefit-cost analysis of animal identification for disease prevention and control. *Revue Scientifique et Technique-Office International des Epizooties*, 20(2):385–405, 2001.

[17] Riyad A El-laithy, Jidong Huang, and Michael Yeh. Study on the use of microsoft kinect for robotics applications. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 1280–1288. IEEE, 2012.

[18] Levan Elbakidze. Economic benefits of animal tracing in the cattle production sector. *Journal of Agricultural and Resource Economics*, pages 169–180, 2007.

[19] C Enevoldsen, YT Gröhn, and I Thysen. Sole ulcers in dairy cattle: associations with season, cow characteristics, disease, and production. *Journal of dairy science*, 74(4):1284–1298, 1991.

[20] Niyonsaba Eric and Jong-Wook Jang. Kinect depth sensor for computer vision applications in autonomous vehicles. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 531–535. IEEE, 2017.

[21] Food Standards Agency. https://www.food.gov.uk/business-guidance/manual-for-official-controls, 2023. Online; accessed 21st February 2023.

[22] Radhakrishnan Gopalapillai, Deepa Gupta, Mohammed Zakariah, and Yousef Ajami Alotaibi. Convolution-based encoding of depth images for transfer learning in rgb-d scene classification. *Sensors*, 21(23), 2021. URL: https://www.mdpi.com/1424-8220/21/23/7950.

[23] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[24] XiaoZhen Han and Ran Jin. A small sample image recognition method based on resnet and transfer learning. In *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, pages 76–81. IEEE, 2020.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] Intel Corporation. https://www.intelrealsense.com/depth-camera-d435/, 2023. Online; accessed 13th February 2023.

[27] Nan Jia, Gert Kootstra, Peter Groot Koerkamp, Zhengxiang Shi, and Songhuai Du. Segmentation of body parts of cows in rgb-depth images based on template matching. *Computers and Electronics in Agriculture*, 180:105897, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0168169920331021, doi:https://doi.org/10.1016/j.compag.2020.105897.

[28] A M Johnston. Welfare implications of identification of cattle by ear tags. *The Veterinary record*, 138(25):612–614, 1996.

[29] Robert Kadlec, Sam Indest, Kayla Castro, Shayan Waqar, Leticia M Campos, Sabrina T Amorim, Ye Bi, Mark D Hanigan, and Gota Morota. Automated acquisition of top-view dairy cow depth image data using an rgb-d sensor camera. *Translational Animal Science*, 6(4):txac163, 2022.

[30] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[31] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[32] Santosh Kumar, Amit Pandey, K. Sai Ram Satwik, Sunil Kumar, Sanjay Kumar Singh, Amit Kumar Singh, and Anand Mohan. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement*, 116:1–17, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0263224117306991, doi:https://doi.org/10.1016/j.measurement.2017.10.064.

[33] Guoming Li, Galen E. Erickson, and Yijie Xiong. Individual beef cattle identification using muzzle images and deep learning techniques. *Animals*, 12(11), 2022. URL: https://www.mdpi.com/2076-2615/12/11/1453.

[34] Keqiang Li and Guifa Teng. Study on body size measurement method of goat and cattle under different background based on deep learning. *Electronics*, 11(7):993, 2022.

[35] Yue Lu, Xiaofu He, Ying Wen, and Patrick Wang. A new cow identification system based on iris analysis and recognition. *International Journal of Biometrics*, 6:18–32, 03 2014. doi:10.1504/IJBM.2014.059639.

[36] Roanna Lun and Wenbing Zhao. A survey of applications and human motion recognition with microsoft kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(05):1555008, 2015.

[37] Naoto Maki, Shohei Nakamura, Shigeru Takano, and Yoshihiro Okada. 3d model generation of cattle using multiple depth-maps for ict agriculture. In *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 11th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2017)*, pages 768–777. Springer, 2018.

[38] Marcus D. Bloice. https://pypi.org/project/Augmentor/, 2022. Online; accessed 13th February 2023.

[39] Jeison David Mejia-Trujillo, Yor Jaggy Castano-Pino, Andrés Navarro, Juan David Arango-Paredes, Domiciano Rincón, Jaime Valderrama, Beatriz Munoz, and Jorge Luis Orozco. Kinect™ and intel realsense™ d435 comparison: A preliminary study for motion analysis. In *2019 IEEE International Conference on E-health Networking, Application & Services (HealthCom)*, pages 1–4. IEEE, 2019.

[40] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[41] Arunava Nag and Sanket Deshmukh. Real time tracking system using 3d vision. 12 2015. doi:10.13140/RG.2.1.3513.4489.

[42] Christel Nielsen. *Economic impact of mastitis in dairy cows*, volume 2009. 2009.

[43] Sergey Novoselov, Vadim Shchemelinin, Andrey Shulipa, Alexander Kozlov, and Ivan Kremnev. Triplet loss based cosine similarity metric learning for text-independent speaker recognition. In *Interspeech*, pages 2242–2246, 2018.

[44] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[45] Yongliang Qiao, Daobilige Su, He Kong, Salah Sukkarieh, Sabrina Lomax, and Cameron Clark. Bilstm-based individual cattle identification for automated precision livestock farming. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 967–972, 2020. doi:10.1109/CASE48305.2020.9217026.

[46] A N Ruchay, K A Dorofeev, V V Kalschikov, V I Kolpakov, and K M Dzhulamanov. Accurate 3d shape recovery of live cattle with three depth cameras. *IOP Conference Series: Earth and Environmental Science*, 341(1):012147, oct 2019. URL: https://dx.doi.org/10.1088/1755-1315/341/1/012147, doi:10.1088/1755-1315/341/1/012147.

[47] Alexey Ruchay, Vitaly Kober, Konstantin Dorofeev, Vladimir Kolpakov, and Sergei Miroshnikov. Accurate body measurement of live cattle using three depth cameras and non-rigid 3-d shape recovery. *Computers and Electronics in Agriculture*, 179:105821, 2020.

[48] AN Ruchay, KA Dorofeev, VV Kalschikov, VI Kolpakov, and KM Dzhulamanov. Accurate 3d shape recovery of live cattle with three depth cameras. In *IOP Conference Series: Earth and Environmental Science*, volume 341, page 012147. IOP Publishing, 2019.

[49] AN Ruchay, KA Dorofeev, VV Kalschikov, VI Kolpakov, and KM Dzhulamanov. A depth camera-based system for automatic measurement of live cattle body parameters. In *IOP Conference Series: Earth and Environmental Science*, volume 341, page 012148. IOP Publishing, 2019.

[50] Burt Rutherford. U.s. beef herd is mostly black but changing slightly. *BEEF*. URL: https://www.beefmagazine.com/cattle-genetics/us-beef-herd-mostly-black-changing-slightly.

[51] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. doi:10.1109/CVPR.2015.7298682.

[52] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[53] Volia Schubiger. How many cows are in the world? URL: https://a-z-animals.com/blog/how-many-cows-are-in-the-world/.

[54] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[55] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[56] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[57] Severiano R Silva, José P Araujo, Cristina Guedes, Flávio Silva, Mariana Almeida, and Joaquim L Cerqueira. Precision technologies to address dairy cattle welfare: Focus on lameness, mastitis and body condition. *Animals*, 11(8):2253, 2021.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] Stevan Stankovski, Gordana Ostojic, Ivana Senk, Marija Rakic-Skokovic, Snezana Trivunovic, and Denis Kucevic. Dairy cow monitoring by rfid. *Scientia Agricola*, 69:75–80, 2012.

[60] Carsten Steger. Occlusion, clutter, and illumination invariant object recognition. *International Archives of Photogrammetry and Remote Sensing*, 34, 09 2003.

[61] Christine L Sumner, Marina AG von Keyserlingk, and Daniel M Weary. Perspectives of farmers and veterinarians concerning dairy cattle welfare. *Animal Frontiers*, 8(1):8–13, 2018.

[62] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[63] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[64] University of Bristol. Opencows2020. https://research-information.bris.ac.uk/en/datasets/opencows2020, 2020. Accessed: 2023-04-29.

[65] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[66] Jack Whittier, Jim Doubet, Dave Henrickson, Justin Cobb, and John Shadduck. Biological considerations pertaining to use of the retinal vascular pattern for permanent identification of livestock. 01 2003.

[67] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 748–756. IEEE, 2018.

[68] Xinyu Yang, Majid Mirmehdi, and Tilo Burghardt. Great ape detection in challenging jungle camera trap footage via attention-based spatial and temporal feature blending. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 255–262, 2019. doi:10.1109/ICCVW.2019.00034.

[69] Han-Jia Ye, De-Chuan Zhan, Xue-Min Si, Yuan Jiang, and Zhi-Hua Zhou. What makes objects similar: A unified multi-metric learning approach. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/8fecb20817b3847419bb3de39a609afe-Paper.pdf.

[70] Weiyu Zeng, Tianlei Wang, Jiuwen Cao, Jianzhong Wang, and Huanqiang Zeng. Clustering-guided pairwise metric triplet loss for person reidentification. *IEEE Internet of Things Journal*, 9(16):15150–15160, 2022.