



## Unique Lion Identification Using Triplet Loss and Siamese Networks

Glen Bennet Hermon<sup>1\*</sup>, Durgansh Sharma<sup>2</sup>

<sup>1</sup> School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, India

<sup>2</sup> Department of Cybernetics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, India

Corresponding Author Email: [gbhermon@ddn.upes.ac.in](mailto:gbhermon@ddn.upes.ac.in)

<https://doi.org/10.18280/ria.340603>

**Received:** 28 October 2020

**Accepted:** 13 December 2020

### Keywords:

*pattern recognition, pattern matching, triplet loss, animal biometrics, animal identification, automated photo identification, computer-vision, non-invasive techniques*

### ABSTRACT

Former techniques for the identification of lion individuals (*Panthera leo*) relied on manual methods of recording data. Such processes have various shortcomings due to the manual nature of recording this data. This research work aims to automate the process of encoding the uniqueness within the whisker spot patterns for each lion individual by non-invasively using photographs. Towards this research work the main bottleneck was the availability of image data for individual lions. The proposed model embeds the uniqueness within the patterns for a specific individual as a unique cluster within its embedding space. This is achieved by using a triplet loss function which, due to its one-shot learning nature trains a deep inception network with less training data. Photographic images are known to have variations in lighting, pose variation, angle variation and other inconsistencies. Since the nature of these issues are nonlinear, it is preferred to create the target model using deep learning techniques. An inception network is trained to generate 128-dimensional vectors unique to each lion. This research paper elaborates on such deep machine learning techniques and other processes that are used to create this model.

## 1. INTRODUCTION

The task of animal identification can be undertaken by a wide variety of techniques that may be invasive or non-invasive in nature. The harmony in lifestyle of animal individuals identified by the invasive processes may suffer due to the intrusive nature of such techniques. There is thus incentive to make progress in the direction of non-invasive identification techniques. Using various image processing techniques to analyse unique natural markings captured non-invasively in photographs is the predominant technique that is able to satisfy such a necessity.

Image processing and pattern recognition have come a long way since its inception and we have seen many useful implementations of the same in various fields and sciences [1]. But there is always scope for more. Constructing a tighter confidence interval for similarity between each image does solve the problem in most cases. This work deals with such an optimization.

The classification of many animals in perspective of the wildlife heritage that we are blessed with is comparatively easier when compared to that of the Lion. The reason being that there is a lack of peculiar patterns like spots or stripes over the Lion's body.

Visual animal biometrics is non-invasive and cost effective for wildlife monitoring activities. There is a problem of detection of individual species in multiple captured images. We have a large collection of images which is hard to manage and to recognize using manual identification techniques. There is a lack of automation in the detection process while retaining robustness to image ambiguities (like blur, fading, occlusions and pose variation).

To identify individual Lions (*Panthera leo*) noninvasively there is a current requirement for developing an automated Lion detection and recognizing system for individuals based on whisker spot patterns that are present on each side of the Lion's face. No two whisker spot patterns are the same and they do not change over time [2, 3].

By identifying each lion, one can track individuals using advancements through the proposed technique and thus identify pride home ranges and population trends. This allows for effective conservation.

The identification process is done by using various deep machine learning techniques to generate a learned model that is able to embed the uniqueness of individuality within a 128-dimensional embedding space. This model is thus able to create unique identification vectors for each lion that acts as a key while trying to identify a lion, given a new photograph. During the identification process the system then just needs to compute the distance of the current vector with the set of stored vectors of known lions to check for a match with the least distance. To have a higher confidence, a tighter threshold of similarity can be used.

## 2. CURRENT TECHNIQUES AND ISSUES IN IDENTIFICATION

Currently, individual identification of Lions is done manually by recording the spot patterns from the photographs, which is tedious, erroneous and time consuming. Though the record keeping and the matching process is done with the help of computational devices, the whole process of the data entry is done manually including the entries that record each

location of the whisker spots in a grid like convention. The rows of the grid correspond to the rows of the lion's whiskers. Various other information along with the whisker patterns such as the presence of scars, notches on the ears and other metadata of the photograph taken are also manually recorded to aid the identification process.

An issue with this system is that it lacks granularity of the information being entered, leading to ambiguity in the entry of the spot locations within the grid system. This ambiguity is arbitrarily resolved only by the decision-making process by the person who is entering the data onto the system, causing potential disagreement when there is more than one person entering this data into the system. Hence this system has a degree of bias to the decision-making process of the person making the entries. Such an issue may be solved by taking multiple readings with the help of different people [4] making the entries into the system, and having ambiguities handled by having the most repeated entry chosen or to have the system take an average reading of all entries. But even this approach is costly both in terms of effort as well as time. The accuracy of the current manual system has a lot of scope for improvement.

Looking towards computational techniques that input the image as a whole and not just a tabular representation of the spot patterns, one must consider the nonlinear variations in pose that occur in different images of the same individual. Mathematically, the various transformations that may be applied to various degrees are that of rotation, shear, scale variation, distortion (linear angular variation), deformation (non-linear) and occlusions in the extreme case. Such manipulations can be translated back and forth by mapping out the various differences in the transformations and creating an affine map, a Euclidean map or an advanced 3-D model that acts as a map to the information represented in the images. Manually charting out such a mathematical representation is tedious and is open to error for new representations of the same data with the absence of any formal technique to prove for accuracy. Also, such a model is more rigid towards accommodating such new representations or variations [5, 6].

Apart from the issues in pose variation and the prospects of creating a mathematical model to represent all such variations,

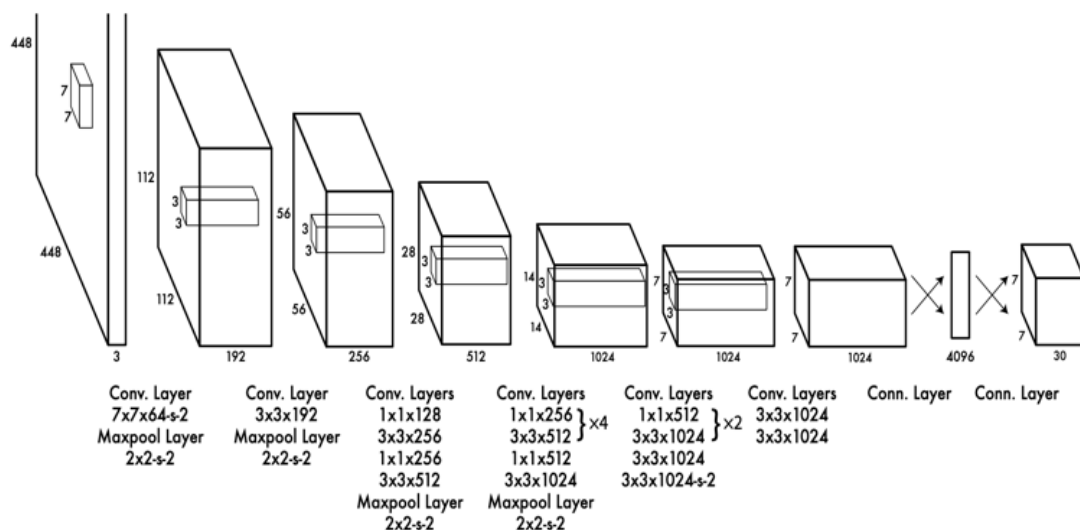
there exists other complications that come along with the computational analysis of images, such as background noise, lighting, shadows, contrast ratios and the complications of various environmental and climatic conditions that get recorded on these images [7]. Charting out these fluctuations mathematically by separately modeling each kind of noise is possible but costly and will still have room for error accounting for a new possibility of such combined image noises.

Along with these variations that may be estimated, it has also been observed that lions may have scars over their muzzle that results in the damage of their whisker spots. In this case the scar could be taken as an identification feature. A similar situational issue of the presence of flies on the lion's face pose a greater problem of mistaking a fly for a whisker spot. Such issues are difficult and near impossible to be mathematically modeled.

Thus, to automate the creation of a model that combines the mathematical representation of the variability in the lion's pose along with the removal of the various image related and situational noises, we have used deep machine learning techniques, to first isolate the region of interest (the lion's whisker spot region), and then subsequently generate a unique vector for the identification process.

### 3. DEEP MACHINE LEARNING APPROACHES

Convolutional neural networks (CNNs) are able to train multiple filters of various configurations stacked in layers, such that each layer becomes capable of recognizing patterns of different complexity. Deeper convolutional layers are capable of having their filters trained such that based on the inputs from the previous layers they are able to identify more complex patterns. [8, 9] This capability of convolutional nets is a huge advantage considering the problem at hand. Given sufficient training data, the trained filters of our model are able to isolate the patterns of interest and separate them from the unwanted noise. Thus, being able to extract desired features defined by the loss function as well as our training data with high efficiency.



**Figure 1.** Representation of the convolutional layers within the YOLO architecture

The convolutional approach is used to tackle the requirement of object localization and detection as well. Object localization is done by including the bounding box information in the output vector for each element in the training set that contains lion faces. Subsequently, this process is replicated of the whisker regions as well (as shown in Figure 3 and Figure 4, below in the methodology section). This enables the prediction of the CNN to use the loss generated from the provided ground truth to increase the accuracy of the predicted bounding box information. The Object detection is done by scanning portions of the image by a sliding window mechanism and then localizing for the presence of the lion face and the whisker region. This sliding window mechanism is optimised by handling these separate windows into one step by passing the whole image into the convolution architecture defined for training. Thus, just the lion face or the whisker region size incorporated is created as a viewport. Hence the size of the window is the size with which the lion face or whisker spot location is trained on. Given any input image larger than this size, the convolutional output resembles a sliding window output within a single step. This increases the output volume but in turn gives us the results for the location of the lion face or the whisker region in one step as described by the YOLO (You Only Look Once) architecture [10].

The network architecture of the YOLO detector has 24 convolution layers along with 2 fully connected layers. The convolutional layers learn the unique patterns of the classification problem presented to the network and the respective maxpool layers reduces the parameter count passed down to a deeper layer. The fully connected layers are used to flatten the convolutional outputs and then reduce it to the format of the truth representation that has the bounding box information. Given below is a brief representation of the network architecture (Figure 1).

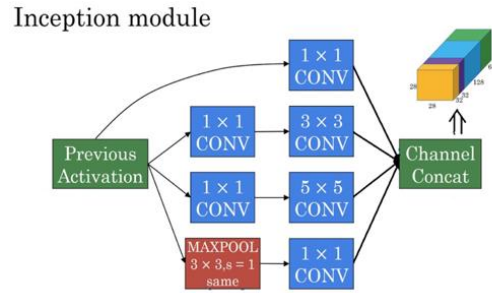
The YOLO architecture also handles multiple predictions by thresholding the Intersection over Union (IoU) calculated for each repeated prediction by using non-max suppression over the confidence of each prediction. The YOLO architecture is thus used for our Lion face detector as well as the whisker region detector as the most optimized approach. Described below is the loss function described by the authors of the YOLO architecture (Eq. (1)).

$$\begin{aligned}
\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
+ \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \quad (1)$$

In the YOLO loss function Eq. (1),  $x$  and  $y$  are the coordinates to the centroid of the anchor box,  $\hat{x}$  and  $\hat{y}$  being

the predicted values. Similarly,  $w$  and  $h$  are the width and height of the anchor boxes along with their predicted values. They are under square root to make predictions for smaller objects more precise.  $C$  denotes a score of the presence of an object or not. The value of  $p(c)$  is the classification loss summed over all the classes. To train the network in a reinforced fashion, respective masks for the presence and absence of objects are embedded,  $\mathbb{1}_{ij}^{obj} = 1$  when the object exists and  $\mathbb{1}_{ij}^{obj} = 0$  when there is no object inversely,  $\mathbb{1}_{ij}^{noobj}$  is 1 when there is no object and 0 when there is an object.  $S^2$  represents the total number of grid cells the input gets divided into. The  $\lambda$  values represent constants with higher values for coordinates ( $\lambda_{coord}$ ), to have more focus on the recognition problem. Finally,  $B$  is the number of anchor boxes defined. The loss is computed for each of the cells within the grid  $S^2$  specified.

To perform individual identification, the capability of learning various patterns and variations at multiple scales is the top requirement. An Inception network is of the best interest as it encompasses multiple convolution filter configurations that are channel concatenated to a single layer (as shown in Figure 2). This structure is then repeated over 22 layers [9]. A single inception module is described in the figure below (Figure 2).



**Figure 2.** The inception model: the constituent convolutional filters within a single layer of the inception network

The inception network is capable of producing embeddings of reduced dimensions for each individual by training with loss functions over this low dimensional (finite) embedding space.

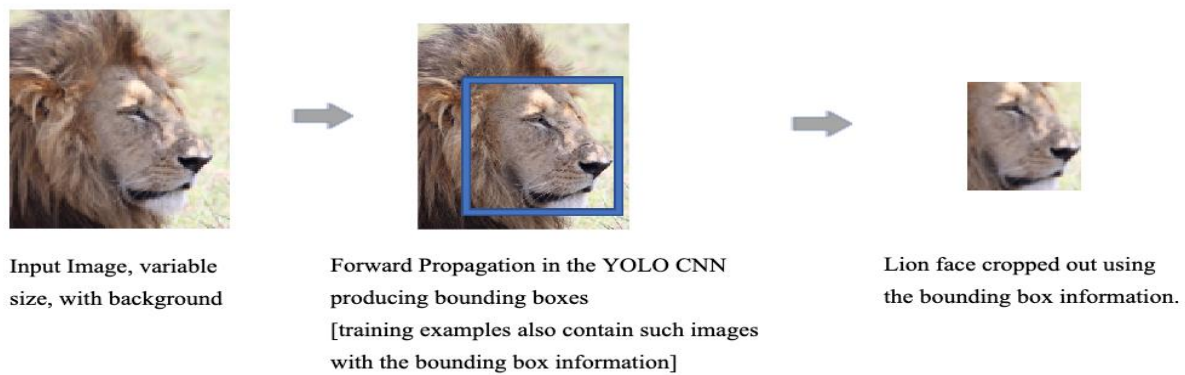
#### 4. METHODOLOGY

The identity estimation process is achieved by identifying the uniqueness within the whisker spot patterns over the mystacial region of the lion-face's muzzle area [2, 11, 12]. This system thus extracts this uniqueness and is able to cluster similar patterns within the embedding space of the learned model.

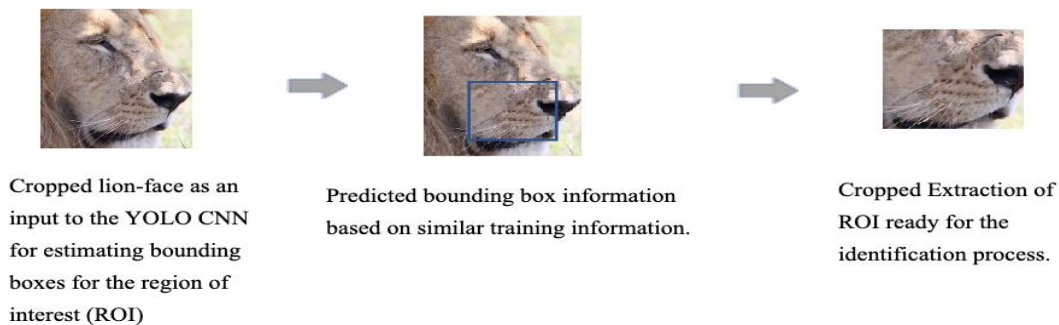
To create such a system, supervised datasets of labeled examples had to be created. A dataset of labelled lion-faces over whole images for training an initial Convolutional Neural Network that implements the YOLO algorithm was created. A dataset for labelled whisker locations over cropped lion-faces to train a second YOLO network was made. Datasets comprising the images of cropped whisker regions separately for each individual lion were also created. These datasets for each individual's whisker images provide for training the inception convolutional network by the triplet loss function over precomputed triplets. To create all the datasets the images

for each lion was manually downloaded from the website that hosts the photographs taken in the Mara Predator Project that has lion photographs documented by individuals over the time period of 2008 to 2013 at the Maasai Mara National Reserve located in Kenya [13].

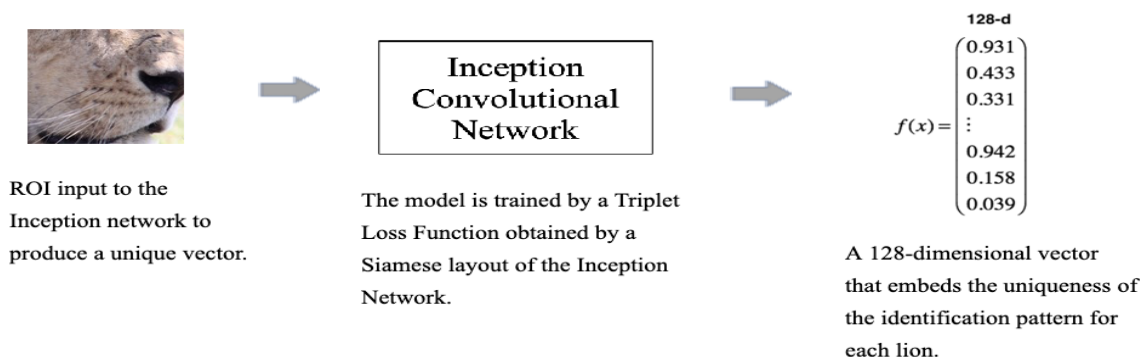
The first step towards the identification process is to locate a Lion face within the Image provided to the system. This is achieved by feeding the input image to a Convolutional Neural Network that implements the YOLO algorithm to get the bounding boxes for the location of the lion face [10]. The training set for this neural network comprises of a set of labelled Images with bounding boxes for three different classes of lion faces: the front face, the right face and the left face. This step also provides for the information of the side of the lion’s muzzle from which the whisker patterns are extracted. Forward propagating through this trained network, we get the coordinates to the bounding boxes for the located lion-face in the image. As a final process of this step, we use the bounding box coordinates to crop out only the lion face from the given image (Figure 3). This ensures the reduction in background noise for our next step.



**Figure 3.** Using convolutional neural networks (YOLO) to crop the lion-face from the input image

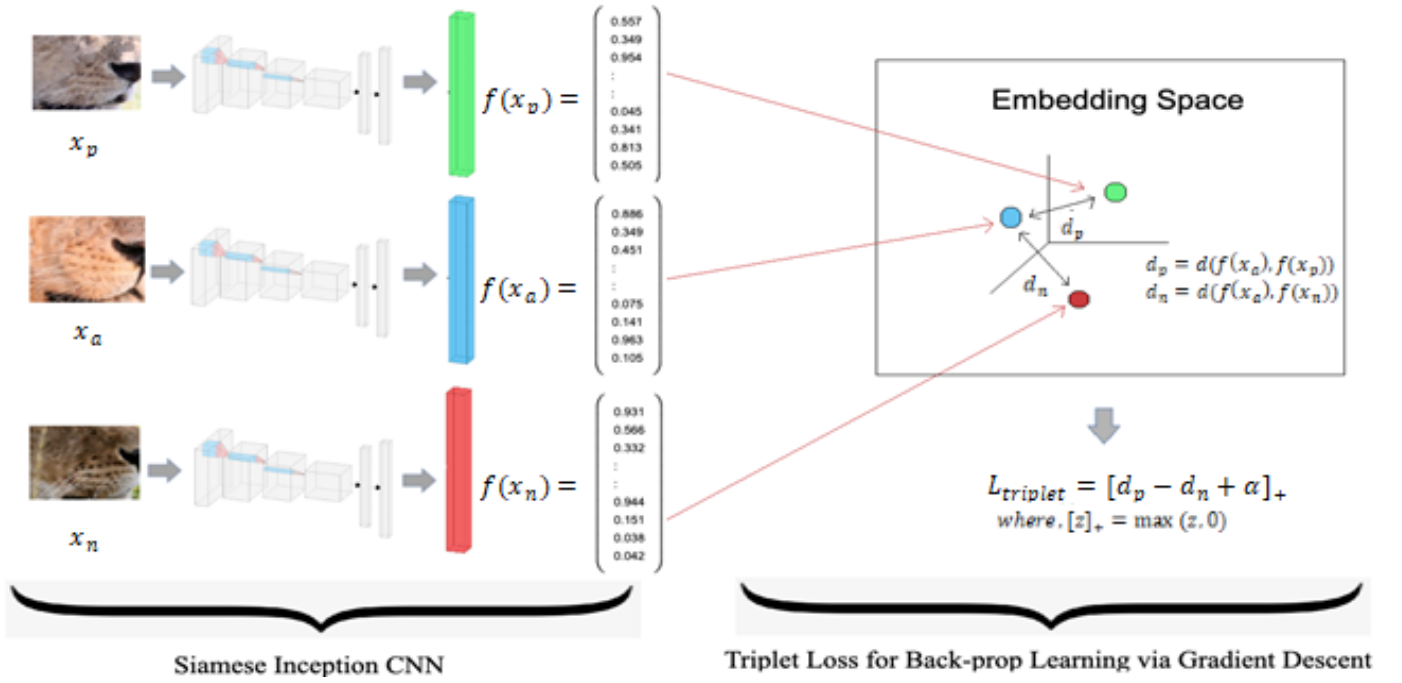


**Figure 4.** Using convolutional neural networks (YOLO) to crop the whisker region (ROI) from the lion-face input



**Figure 5.** Generating 128-Dimentional vectors for each lion with an inception network trained using a triplet loss function





**Figure 6.** An intuitive representation of the Siamese Network for the Triplet loss Function along with the Embedding Space

The triplets are created by grouping the vector data of three different images. Out of these three images, two are different images of the same lion and the third is an image of a totally different lion. The triplet loss function is formulated by these images namely, the anchor image, the positive match and the negative match. To form these triplets for each anchor image, the vectors of the positive match as well as the negative match are precomputed each by a forward pass of the respective images through the inception net, so that the loss function can be computed over the current pass of the anchor image. Such a network of using the outputs of multiple forward passes through a same neural network, is known as a Siamese network. Minimizing the difference between the positive pair of the anchor and the positive match and maximizing the difference between the negative pair of the anchor and the negative match is the goal of the triplet loss function [14, 15]. An intuitive representation of these processes is shown in Figure 6. Backpropagating this loss function through the inception network, creates a model for the various unique patterns and their nature of similarity and dissimilarity. Described below is the triplet loss function (Eq. (2)):

$$J = \sum \left( \|f(A_{(i)}) - f(P_{(i)})\|_2^2 - \|f(A_{(i)}) - f(N_{(i)})\|_2^2 + \alpha \right) \quad (2)$$

In the triplet loss function, A is the anchor image, P is the image with the positive match and N is the image with the negative match. The function  $f$ , denotes a neural network forward pass that results in a 128-dimensional vector. The L2 norm (Euclidean distance) is taken for each pair with the anchor by computing the sum of squared differences. Finally,  $\alpha$  denotes the margin that increases the distance between the negative match pairs and decreases the distance between the positive match pairs. This loss function operates on the triplets for all anchors used for training, hence the summation over the whole function.

Hence while forward propagating for a test image of the cropped region of interest, the network thus outputs a 128-dimensional vector which is used to search for a match within the stored set of vectors of known lions. The matching process is done by finding the Euclidean distances for each pairwise comparison with the vectors of every lion within the database. The lesser the distance the more the similarity, therefore we use a threshold value to find a match.

---

**Algorithm 1:** Lion Mystacial Features by Triads (LMFT): Extraction training and Clustering

---

1. **Input:** Datasets (LFlab, CLWlab, CLwin), Neural Nets (Y1, Y2, Incep), Triplets Template
  2. **while** yolo-bounding-box-prediction-loss decreases **do**
  3.   update Y1 with Y1\_BackProp(LFlab)
  4. **end while**
  5. **while** yolo-bounding-box-prediction-loss decreases **do**
  6.   update Y2 with Y2\_BackProp(CLWlab)
  7. **end while**
  8. Precompute initial vectors by Incep\_ForwardProp(CLwin)  $\Rightarrow T$
  9. Triplet Template (T)  $\Rightarrow$  Triads
  10. **while** triplet-loss decreases **do**
  11.   Incep\_BackProp(Triads)
  12. **end while**
  13. **Return** Incep(Triads)
- 

Algorithm 1 describes the process of training the separate networks with respective datasets. The dataset that contains samples of labelled lion faces in whole images is denoted by the name 'LFlab'. 'CLWlab' is the dataset that contains samples of cropped lion-faces with the whisker region labelled. 'CLwin' is the dataset that contains samples of cropped whisker regions, this dataset is subdivided into sets for every individual separately. 'Y1' represents the YOLO neural net that estimates lion face label boxes. 'Y2' represents the YOLO neural net that estimates the whisker region label boxes. Finally, the 'Incep' neural net is the inception network that trains with the triplet loss function which is denoted as 'triplet-

loss'. The function with the '\_BackProp' postfix denotes the back-propagation training pass of the respective neural network. The function with the '\_ForwardProp' postfix denotes the forward-propagation pass of the respective neural network. The 'yolo-bounding-box-prediction-loss' is the loss function that governs the training of the respective YOLO networks to estimate the bounding boxes for the region it is trained for. 'Triplet Template' is a function that creates the triplets based on the lion identities. The trained inception network is the final outcome.

**Algorithm 2:** Lion Mystacial Identity (LMId): Forward pass prediction

- Input:** Image containing Lion Face (I), Datasets of known lion vectors(LiDb),
- Neural Nets (Y1,Y2,Incep)
- Bounding box 1 = Y1(I)
- LF = crop\_img(bounding box 1, I)
- ROI\_box=Y2(LF)
- ROI=crop\_img(ROI\_box,LF)
- Identity vector = Incep(ROI)
- Identity = Square\_diff ((**For** all Xi **in** LiDb), Identity vector) < Threshold\_diff
- Return** Identity

Algorithm 2 describes the various steps taken for the process of lion identification, when the system is provided with a new input image. The input image is denoted by 'I'. The various lion identities are stored as vectors in a database represented as 'LiDb'. The neural net 'Y1' represents the first YOLO net that estimates lion face locations. 'Y2' is the second YOLO net that estimates the whisker locations. Lastly, 'Incep' is the inception network that outputs a 128-dimensional vector that serves as a representation of the lion's identity. The 'crop\_img' function crops out the predicted region from the original image by taking both the output of the neural network as well as the original image as it's inputs. 'Square\_diff' is a function that computes the sum of the squared differences of all values between two vectors of the same dimension. This function computes the difference between the current vector along with all the vectors stored within the database. The 'Threshold\_diff' value is the threshold value that denotes that an identity is a match if the difference value from the previous step is below this threshold value. Hence, the final value returned is the identity of the identified lion.

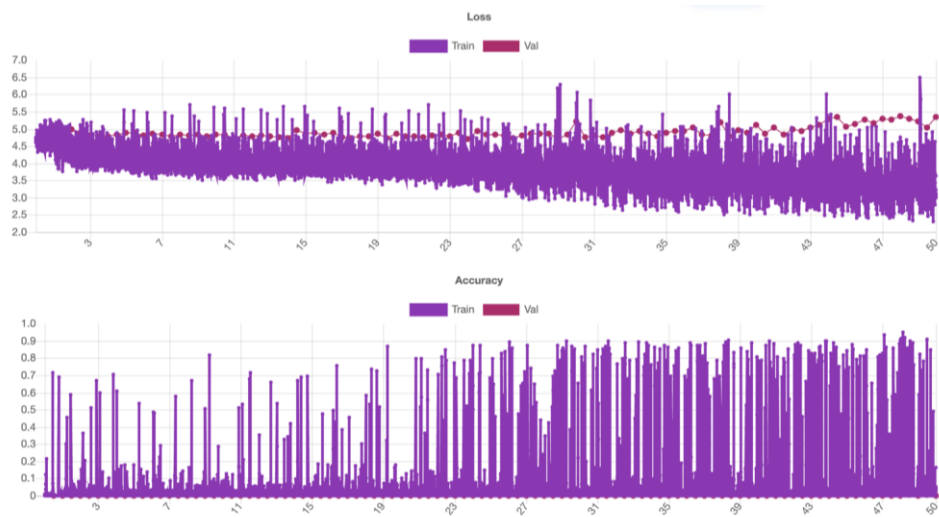
## 5. RESULTS AND DISCUSSION

To reduce the complexity of the system and to avoid the possibility of correlation between the whisker patterns of both the sides of the lion's muzzle, this research is done by training the model over the data of the whisker patterns of only the right side of every lion providing for a singular identification signature. Subsequently the approach of extracting the ROI for training was adopted, thus increasing the ease and accuracy of the model to 80 %. The training graph for the increase of model accuracy for the initial case is shown below (Figure 7). The initial training was experimented over training for the whole lion face that had a lot of noise. Subsequently the approach of extracting the ROI for training was adopted, thus increasing the ease and accuracy of the model to 80%. The training graph for the increase of model accuracy for the final case is shown below (Figure 8). The issues of scarcity of data (120 images, 40 lions with 3 images each) available for training has been solved by the creation of triplets for every training sample. Such a method of creating triplets has thus increased the number of training samples i.e. 28080 which is more than 200 times the original number of image samples (Table 1).

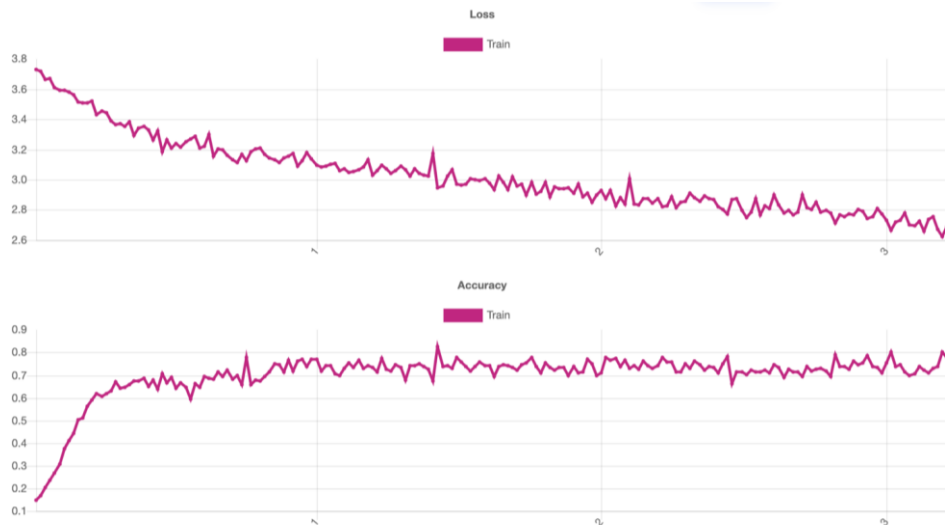
**Table 1.** Quantities within the dataset created

Data Description	Quantity
Unique lion individuals within the database	40
Minimum training samples per individual lion	3
Total images of all lions	120
Triplets for each lion	702
Total number of triplets for training	28080

This procedure of using triplets has thus made it viable to be able to train such an identification system. Due to the nature of the triplet loss function that trains the inception network, the uniqueness of the patterns learned for each lion within the training set develops a separate cluster within the embedding space of the model learned. This is predominantly due to the hyperparameter  $\alpha$  (margin parameter) within the triplet loss function, as well as the differences between the patterns that determine the location of these clusters within the embedding space.



**Figure 7.** Training the model without removing the background using YOLO



**Figure 8.** Training the model by extracting only the whisker regions

## 6. CONCLUSION AND FUTURE WORK

The unique method of training by triplets enables the system towards individual lion's identity and to have its own cluster within the embedding space of the learned model. Furthermore, reducing the overhead of creating separate mathematical models that handle each and every problem that is native to identification within photographs, such as lighting, contrast, shadows, etc., along with other occlusions due to the various features of the lion face's anatomy. The triplet loss function reduces the problem of identification to a problem of vector embedding within the embedding space of the model being trained. This model also has the advantage of getting even better accuracy on the exposure to more lions as this data may be used to create further triplets to revise the learned clusters. Furthering this work, the system can be improved by the inclusion of both the right as well as the left mystacial areas of the lion's muzzle and studies may be conducted to test for correlation between these patterns for each lion. Furthermore, the analysis of the closeness of various clusters may reveal other correlations that may suggest similarity of these whisker patterns between close relatives of lion individuals. This work thus provides for an efficient non-invasive identification mechanism for lions that will aid the various efforts towards the monitoring and hence towards the conservation of this beloved species.

## REFERENCES

- [1] Nguyen, H., Maclagan, S.J., Nguyen, T.D., Nguyen, T., Flemons, P., Andrews, K., Phung, D. (2017). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, pp. 40-49. <https://doi.org/10.1109/DSAA.2017.31>
- [2] Pennycuick, C.J., Rudnai, J. (1970). A method of identifying individual lions *Panthera leo* with an analysis of the reliability of identification. *Journal of Zoology*, 160(4): 497-508. <https://doi.org/10.1111/j.1469-7998.1970.tb03093.x>
- [3] Miththapala, S., Seidensticker, J., Phillips, L.G., Fernando, S.B.U., Smallwood, J.A. (1989). Identification of individual leopards (*Panthera pardus kotiya*) using spot pattern variation. *Journal of Zoology*, 218(4): 527-536. <https://doi.org/10.1111/j.1469-7998.1989.tb04996.x>
- [4] Friday, N., Smith, T.D., Stevick, P.T., Allen, J. (2000). Measurement of photographic quality and individual distinctiveness for the photographic identification of humpback whales, *Megaptera novaeangliae*. *Marine Mammal Science*, 16(2): 355-374. <https://doi.org/10.1111/j.1748-7692.2000.tb00930.x>
- [5] Burghardt, T., Thomas, B., Barham, P.J., Calic, J. (2004). Automated visual recognition of individual African penguins. Fifth International Penguin Conference, Ushuaia, Tierra del Fuego, Argentina.
- [6] Arzoumanian, Z., Holmberg, J., Norman, B. (2005). An astronomical pattern-matching algorithm for computer-aided identification of whale sharks *Rhincodon typus*. *Journal of Applied Ecology*, 42(6): 999-1011. <https://doi.org/10.1111/j.1365-2664.2005.01117.x>
- [7] Loos, A., Ernst, A. (2013). An automated chimpanzee identification system using face detection and recognition. *EURASIP Journal on Image and Video Processing*, 3(1): 49. <https://doi.org/10.1186/1687-5281-2013-49>
- [8] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278-2324. <https://doi.org/10.1109/5.726791>
- [9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [10] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 779-788. <https://doi.org/10.1109/10.1109/CVPR.2016.91>
- [11] Anderson, C.J., Da Vitoria Lobo, N., Roth, J.D.,

- Waterman, J.M. (2010). Computer-aided photo-identification system with an application to polar bears based on whisker spot patterns. *Journal of Mammalogy*, 91(6): 1350-1359. <https://doi.org/10.1644/09-MAMM-A-425.1>
- [12] Osterrieder, S.K., Salgado Kent, C., Anderson, C.J., Parnum, I.M., Robinson, R.W. (2015). Whisker spot patterns: A noninvasive method of individual identification of Australian sea lions (*Neophoca cinerea*). *Journal of Mammalogy*, 96(5): 988-997. <https://doi.org/10.1093/jmammal/gyv102>
- [13] Mara Predator Project: <http://www.livingwithlions.org/mara/>, accessed on 16-8-2020.
- [14] Schroff, F., Kalenichenko, D., Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-823. <https://doi.org/10.1109/CVPR.2015.7298682>
- [15] Zhang, X., Yu, F.X., Kumar, S., Chang, S.F. (2017). Learning spread-out local feature descriptors. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, pp. 4605-461. <https://doi.org/10.1109/ICCV.2017.492>

## NOMENCLATURE

<i>YOLO</i>	You Only Look Once (Neural Network)
$x$	x-coordinate
$y$	y-coordinate
$\hat{x}$	predicted x-coordinate
$\hat{y}$	predicted y-coordinate
$w$	Anchor box width

$h$	Anchor box height
$C$	Object presence score
$p(c)$	Classification loss
$\mathbb{1}_{ij}^{obj}$	Object presence parameter
$\mathbb{1}_{ij}^{noobj}$	Object absence parameter
$B$	Number of anchor boxes
$S^2$	Total number of grid cells
$A_{(i)}$	Anchor image vector
$P_{(i)}$	Positive match vector
$N_{(i)}$	Negative match vector
$LFlab$	Dataset of labelled lion faces
$CLWlab$	Dataset of cropped lion face with labelled whisker region
$CLwin$	Dataset of cropped whickers tagged by individual identity
$Y1$	First YOLO net
$Y2$	Second YOLO net
<i>Incep</i>	Inception net
<i>_BackProp</i>	Back Propagation (postfix)
<i>_ForwardProp</i>	Forward Propagation (postfix)
<i>LiDb</i>	Dataset for vectors of known lions
$I$	Input image
<i>crop_img</i>	Image cropping function
<i>Square_diff</i>	Calculate sum of the squared differences
<i>Threshold_diff</i>	Threshold variability for a match

## Greek symbols

$\alpha$	Margin parameter in embedding space
$\lambda$	Focus hyperparameter