Deep Learning for Leopard Individual Identification: An Adaptive Angular Margin Approach

David Colomer Matachana

The Challenge: Leopard Individual Identification

The Problem

- Researchers need to accurately **count individual** leopards in the wild.

Current Approaches

- **Traditional** Computer Vision techniques: Hotspotter.
- Recent research on deep learning applications: **Triplet Network** applied to tigers.

The objective

- Develop a deep learning solution based on current **state-of-the-art** techniques in facial recognition.
- Develop a novel **edge detection** preprocessing technique.
- Determine if the new method is a **valid** technique and its **comparability** to existing techniques.

Dataset

- The Nature Conservation Foundation, provided 8900 images of leopards from trail cameras.

- High quality and diverse.

Cleaning

- 623 removed due to impossibility of identifying a leopard.

Train-test split

Open-set problem: Different classes in train and test/application phase.

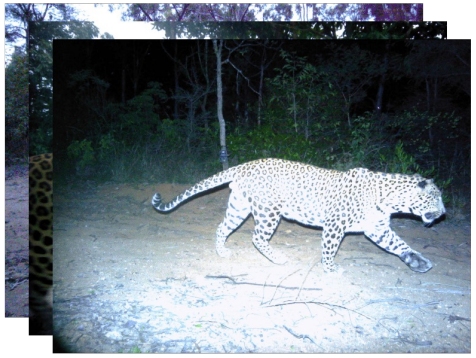
TRAINING

1191 ID's, 7757 images

Train ID 1

Train ID 2

Train ID 3



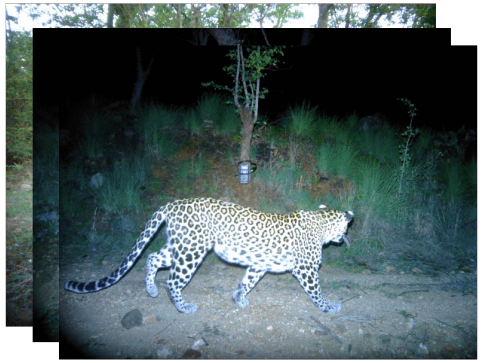
Train ID 4

...

TESTING

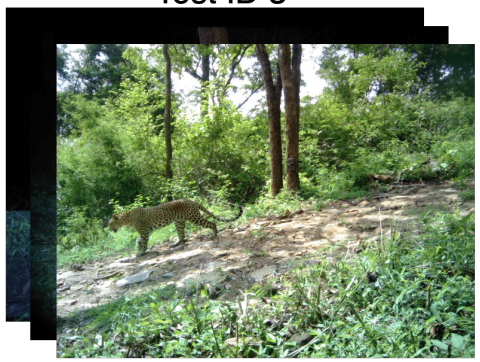
88 ID's, 520 images

Test ID 1



Test ID 2

Test ID 3



Test ID 4

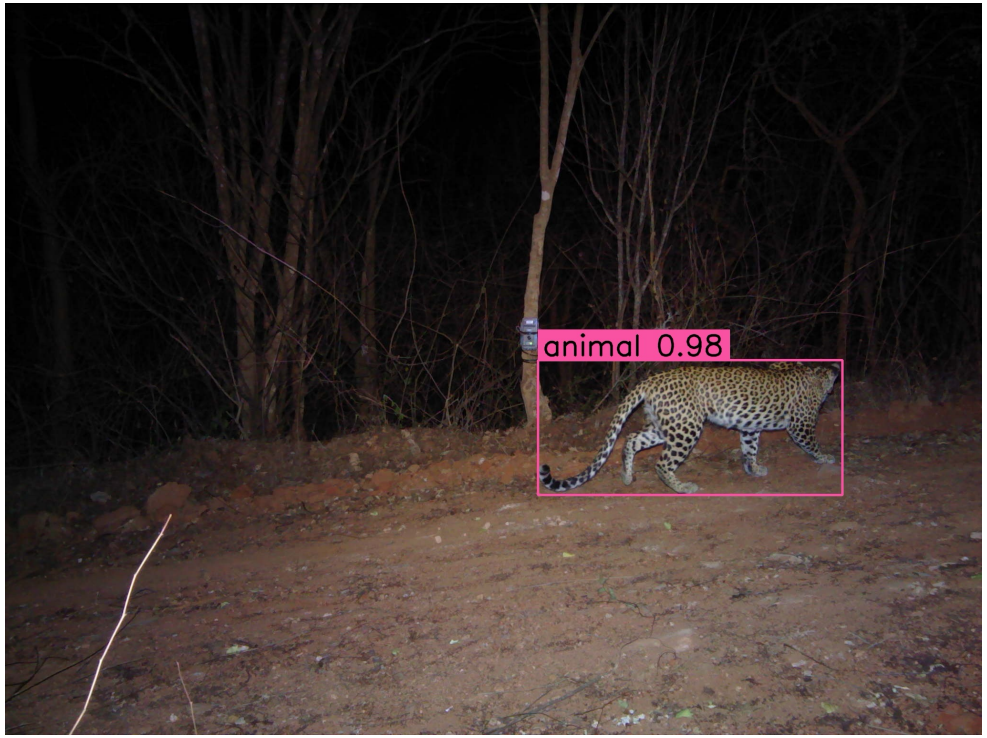
Preprocessing

- Essential in this context, as images come from fixed cameras, and are thus variable in distance from camera, lighting, etc.

- Three steps:
  - Crop image
  - Remove background
  - Edge detection

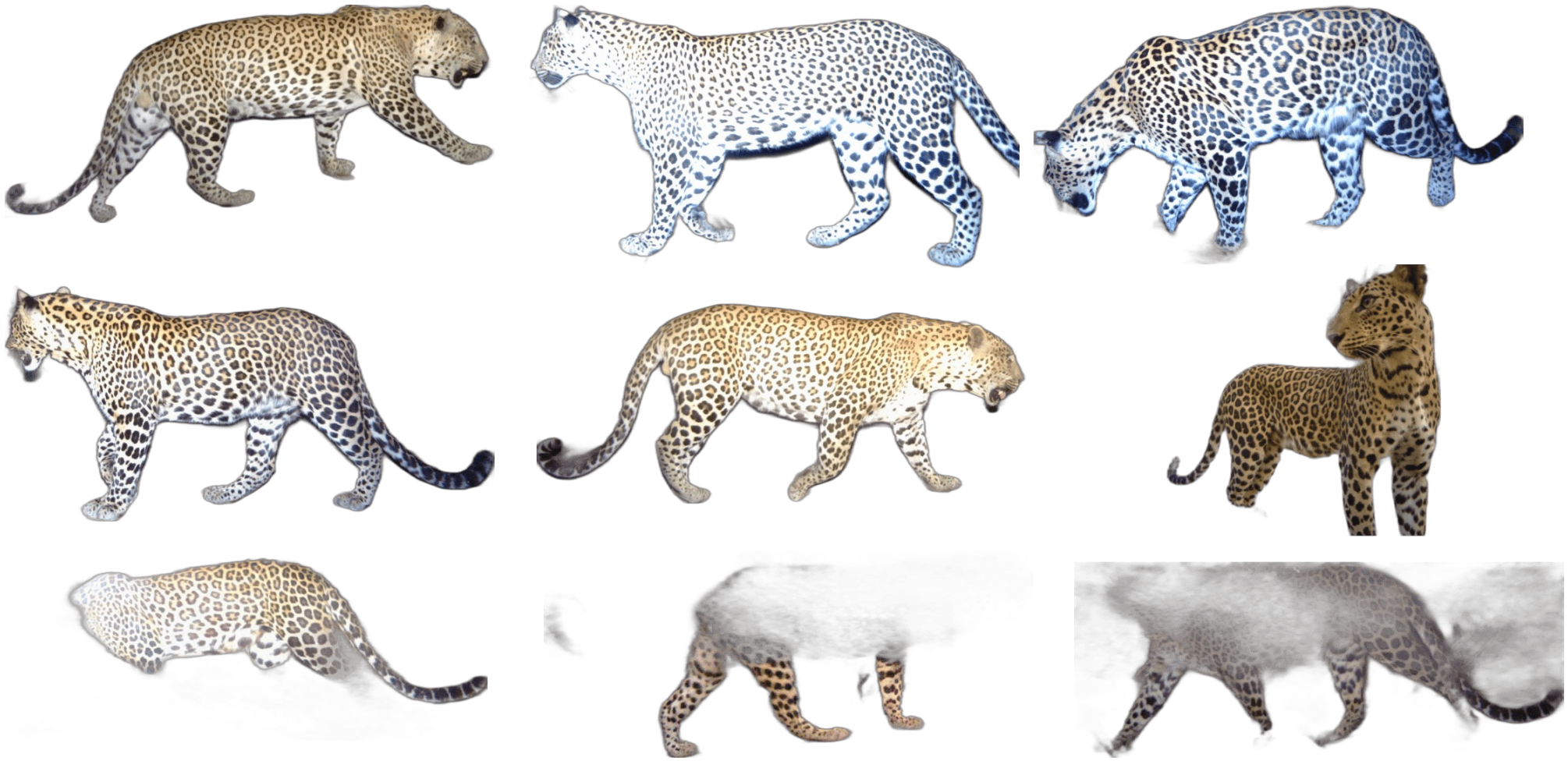
Crop Image

Perfect accuracy in finding usable leopard flanks.



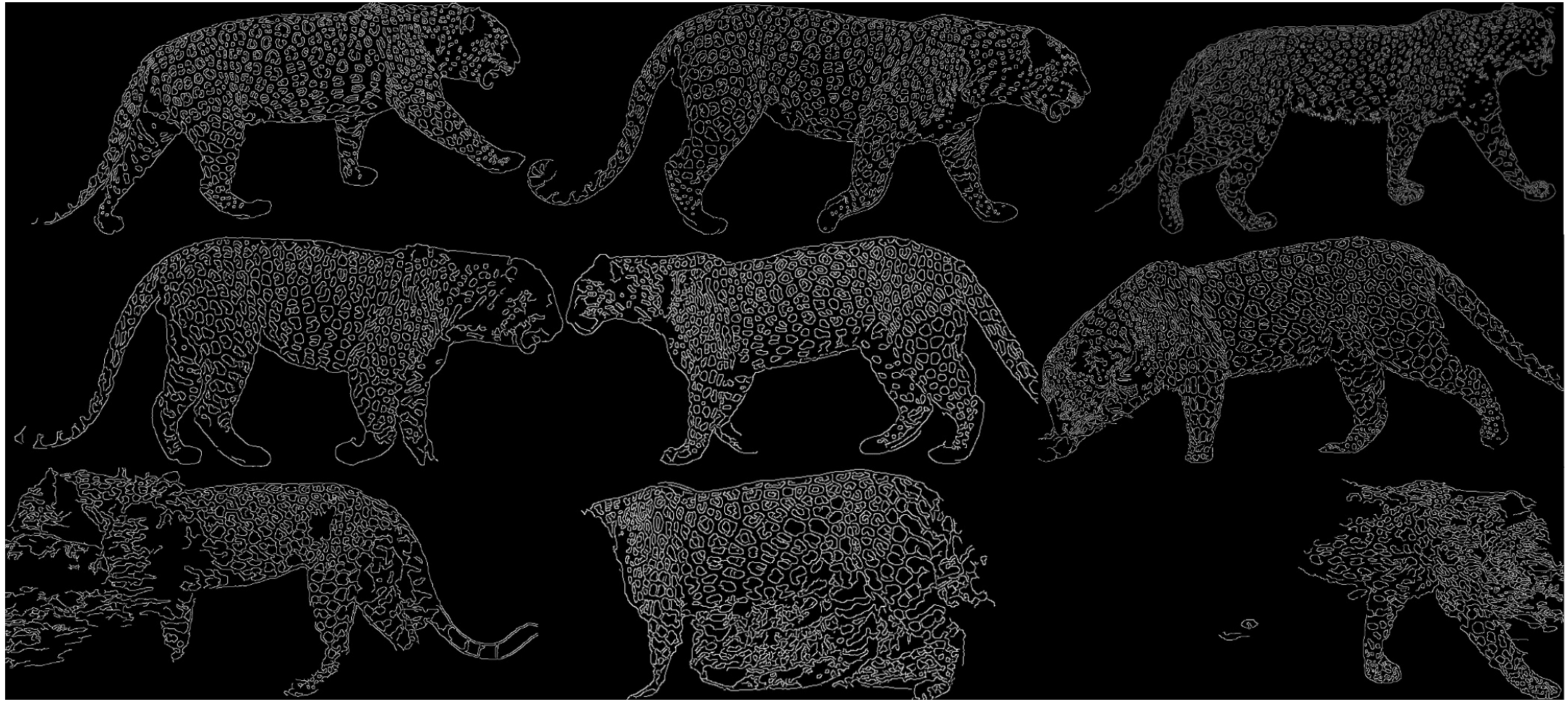
Background removal

Remove background to focus model on the subject. A small subset of flanks are removed incorrectly.



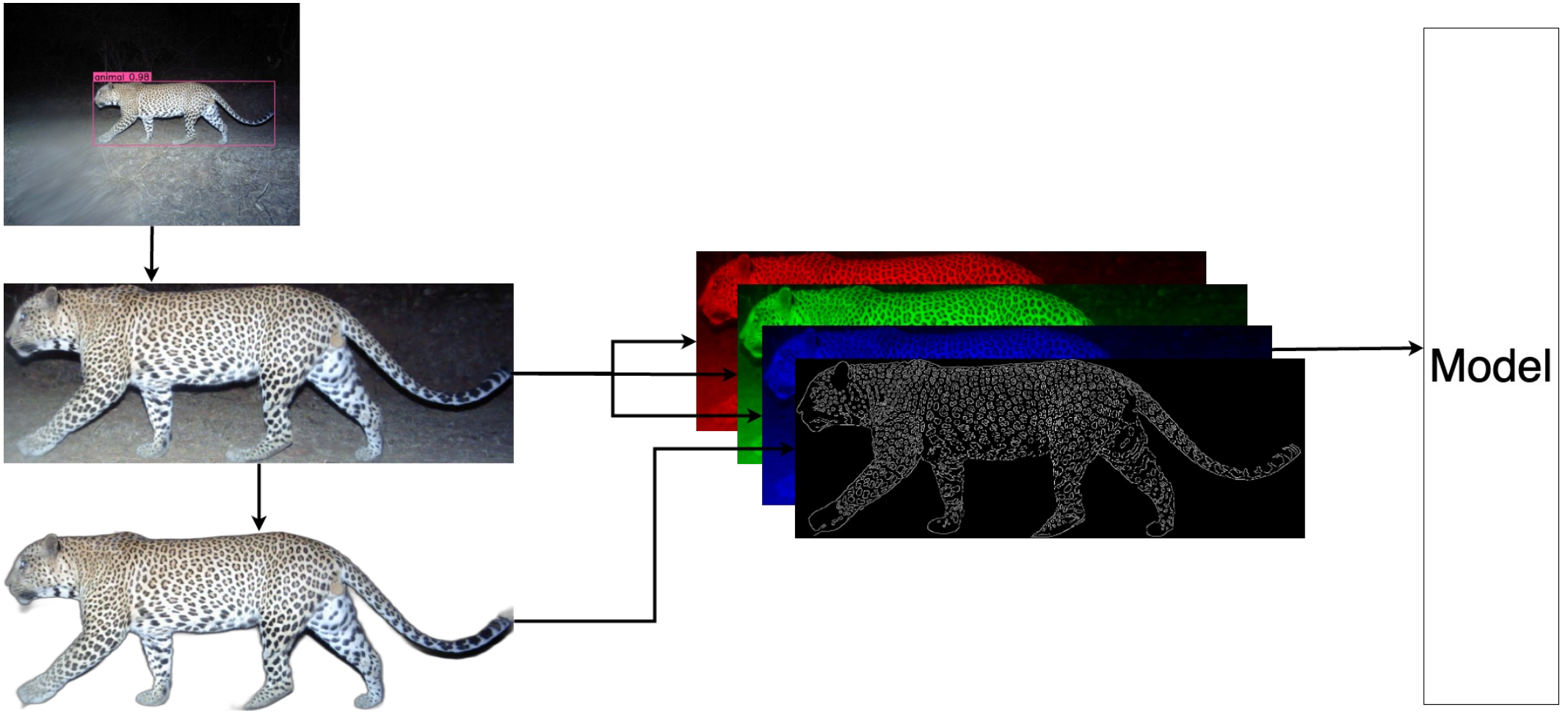
Edge Detection

Highlights rosettes irrespective of lighting. A small subset of flanks have inaccurate edge detection.



Model Input

Cropped RGB + Edge Detection as model input



Architecture

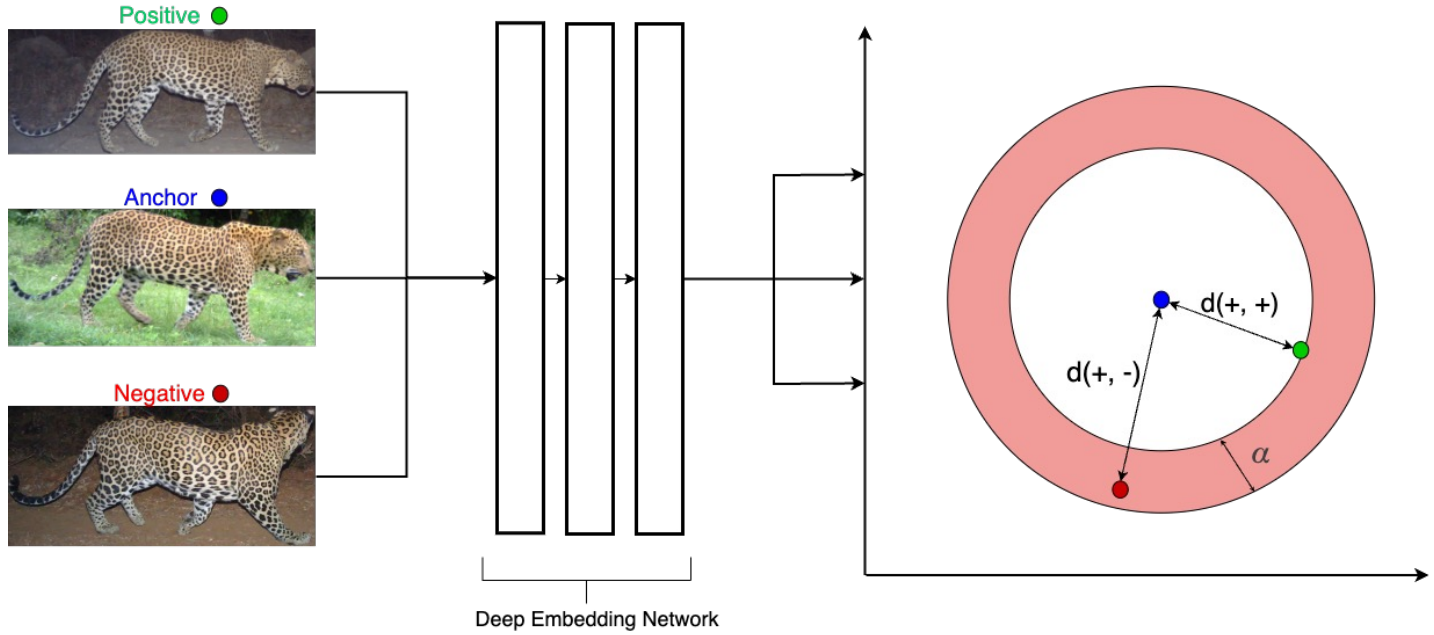
Open-Set: A classifier is not a valid solution. It must learn to discriminate between classes in general.

Draw parallels from Facial Recognition, use embedding network:

- Triplet Network Baseline
- CosFace
- Novel Modified CosFace

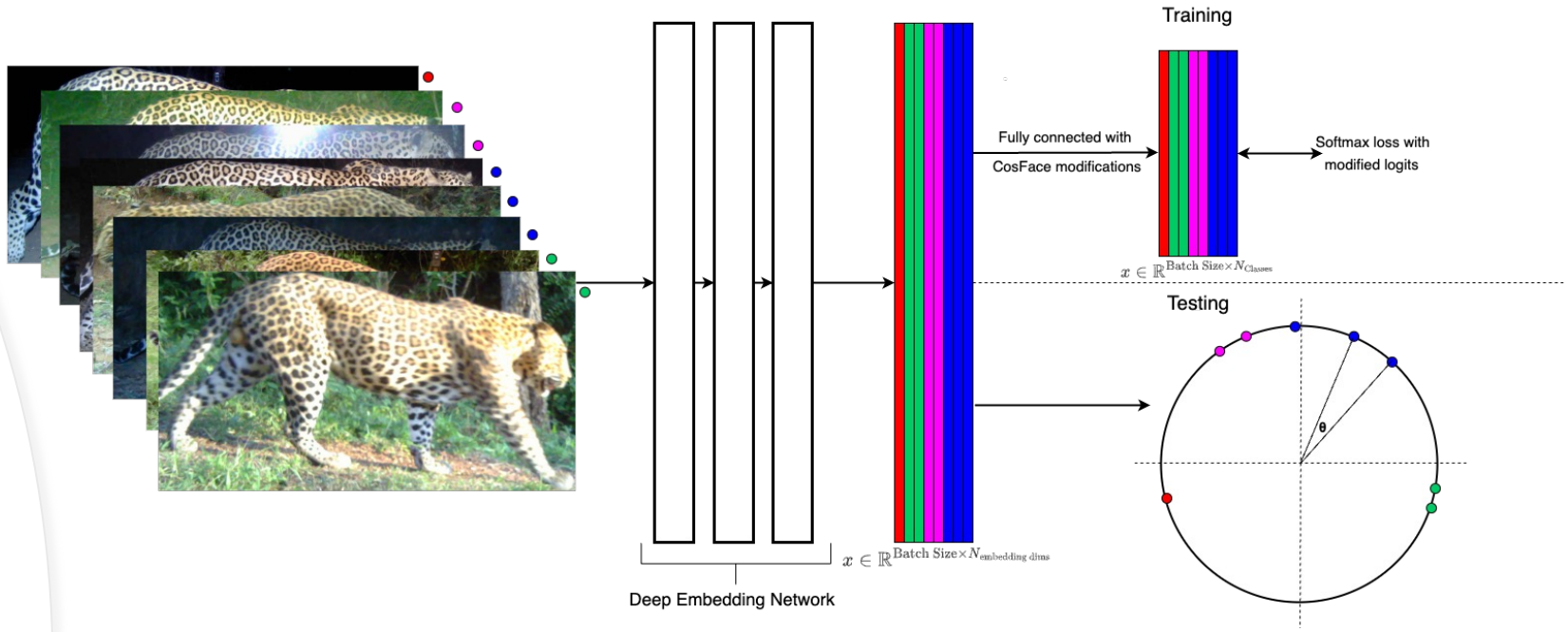
Triplet Network

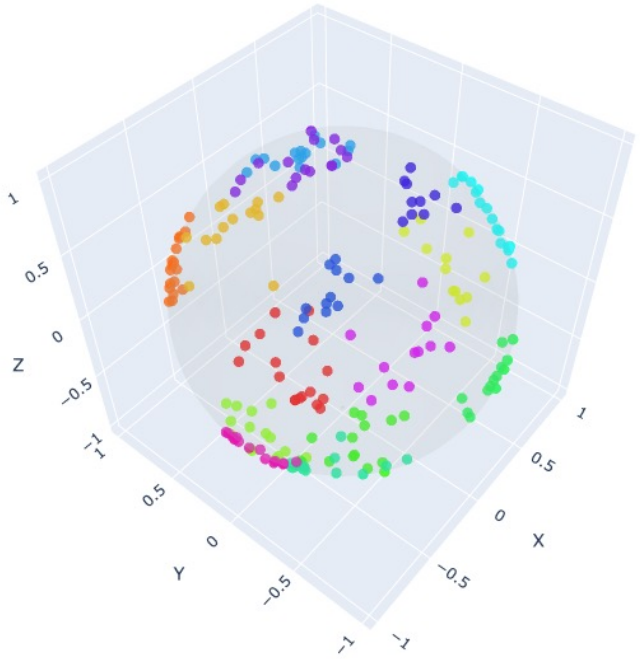
- Compares 3 elements, an anchor, a positive, and a negative. Tries to minimise the anchor-positive distance and maximise the anchor-negative distance.
- No longer state-of-the-art.

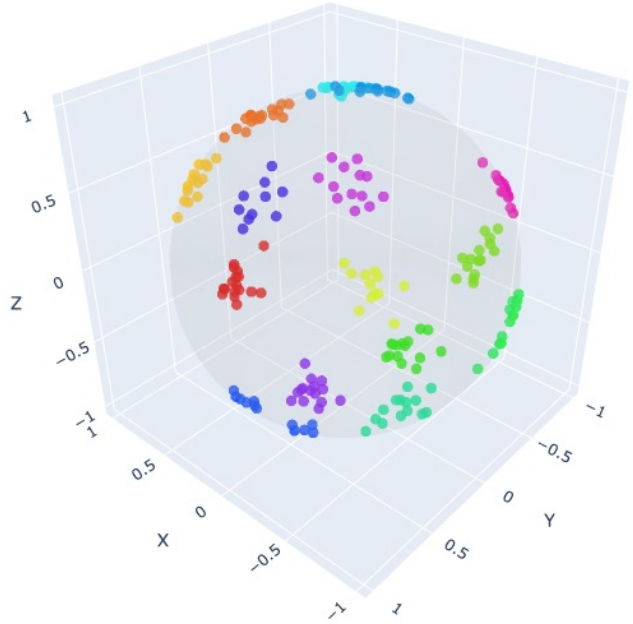


CosFace

- Full batch comparisons.
- Encodes embeddings into a hypersphere ⟹ Comparisons made through angular separations.
- Adds extra FC layer after embedding, and applies CosFace loss to it, which enforces an angular margin **m** between classes.

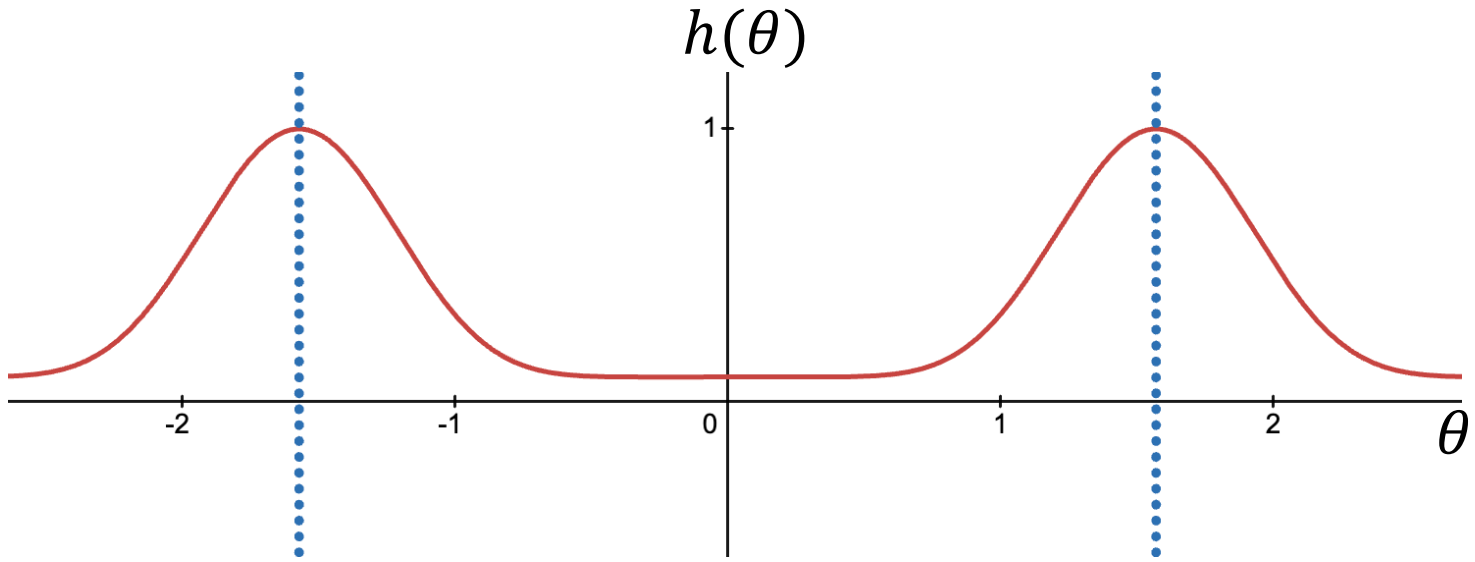






Modified CosFace

- Novel method: Instead of constant margin **m** for every exemplar, adaptive margin **h(θ)*m.**
- Higher margin for harder exemplars ⇨ model learns more robust features.



Results: Preprocessing pipeline

- Best preprocessing pipeline: Crop RGB + Edge Detection.
- Edge detection shown to largely guide model to accurate learning.

Model	Dynamic top 5 average precision	Top 5 rank match detection
Cropped RGB	0.7815	0.9148
Background removed	0.8486	0.9198
Cropped RGB + Edge Detection	**0.8814**	**0.9533**
Background removed + edge detection	0.8284	0.9160

Results: Models

- Modified CosFace outperforms other models, including baseline.
- Hotspotter is still the best.

Model	Dynamic top 5 average precision	Top 5 rank match detection
Naïve model	0.01976	0.09438
Triplet network	0.6907	0.8824
CosFace	0.8413	0.9331
Modified CosFace	**0.8814**	**0.9533**
Hotspotter	0.9475	0.9654

Next Steps?

- Apply to species that don’t see such a high accuracy with Hotspotter.
- Understand how model “thinks” with accurate embedding inversion of class centroids.
- Repeat experiments with larger dataset.

Conclusion

1

Confirm edge detection as a valid technique for preprocessing.

2

Largely surpass the triplet network baseline both with current state of the art and a novel adaptation of it.

3

Confirm Hotspotter to still be the best performing algorithm. Potential to surpass it with applications to other species.

Appendix: Derivation of CosFace loss:

- Derivation of CosFace loss:
- From Softmax loss:
- Normalized softmax loss (angular form):
- CosFace loss:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

$$L_{softmax} = -\frac{1}{N}\sum_i^N \log(f(s)_i)$$

$$s_i = W_{y_i}^T x_i = \|W_{y_i}\|\|x_i\|\cos(\theta_i)$$

$$L_{N\_softmax} = -\frac{1}{N}\sum_i^N \log \frac{e^{s\cos\theta_{y_i,i}}}{\sum_j e^{s\cos\theta_{j,i}}}$$

$$L_{CosFace} = -\frac{1}{N}\sum_i^N \log \frac{e^{s(\cos\theta_{y_i,i}-m)}}{e^{s(\cos\theta_{y_i,i}-m)} + \sum_{j\neq y_i} e^{s\cos\theta_{j,i}}}$$

Appendix: Interesting matches

Anchor Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/MML67_Left/0_0_MM-413A-2018-07-27_00-27-31.jpg

Comparison Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/MML67_Left/0_0_MM-104A-2016-04-19_23-38-36.jpg

Confidence: 68.50%

Anchor Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CUL24_Left/0_0_CU-98A-2018-09-14_18-38-21.jpg

Comparison Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CUL24_Left/0_0_CU-298A-2016-02-17_20-20-21.jpg

Confidence: 85.01%

Anchor Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CUL63_Left/0_0_CU-413B-2018-11-22_03-44-25.jpg

Comparison Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CUL63_Left/0_0_CU-405A-2018-11-25_05-47-41.jpg

Confidence: 61.46%

Anchor Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CML07_Left/0_0_CM32-A-2017-08-20_09-58-16.jpg

Comparison Image

/Users/dc1823/Documents/IRP/irp-dc1823/leopard_id/data/crop_test/CML07_Left/0_0_CU-339A-2018-11-12_05-55-28.jpg

Confidence: 69.59%