Master thesis project: ChimpRec

Théodore Cousin & Julien Demeure

Uclouvain

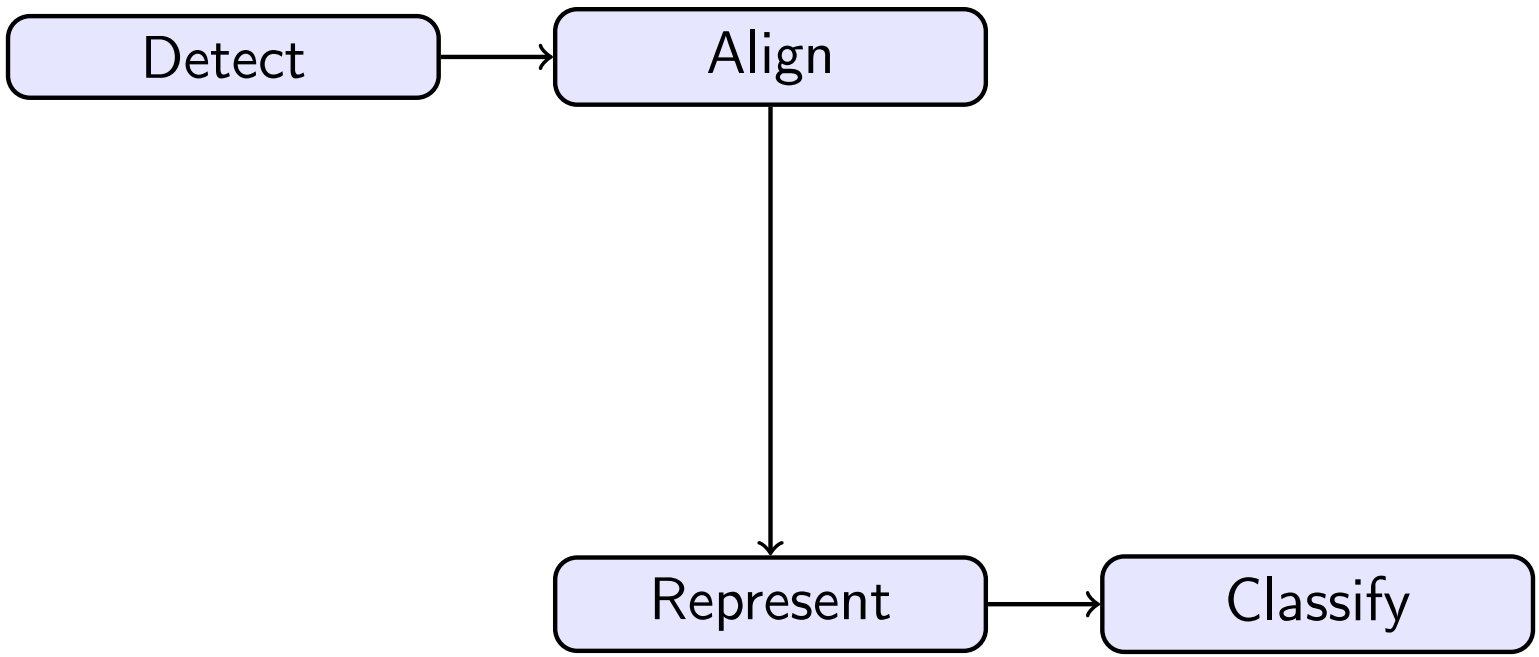
30th of October

Table of Contents

Link

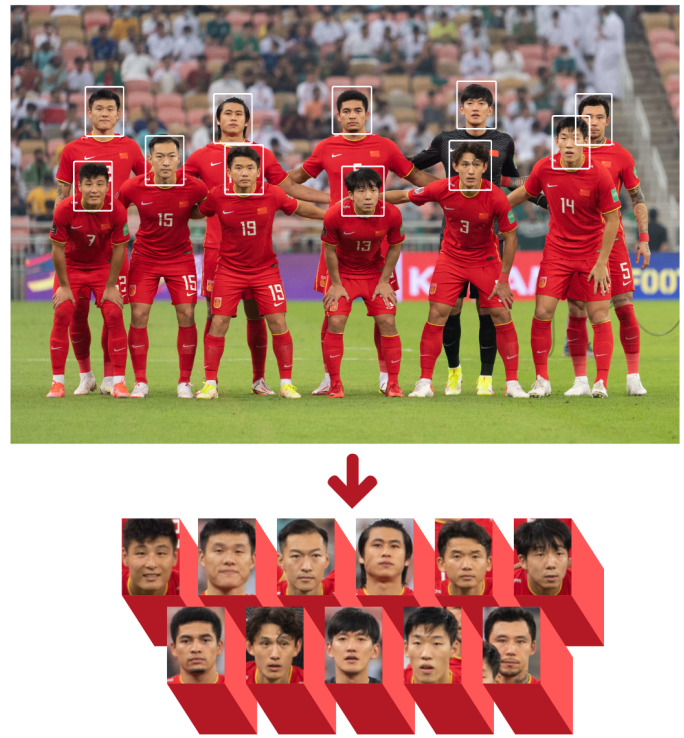
Figure 1: Github repository

Facial Recognition Pipeline



Detection

- Detect which areas in the image correspond to a face
- Crop the image and eventually save it in a file



Alignment

- **Position Standardization**: Align key facial landmarks (eyes, nose, mouth) to a consistent position, reducing variation across images.
- **Normalization of Conditions**: Adjusts for lighting, camera angle, and orientation to ensure that external factors do not affect recognition performance.
- **Pre-Extraction Consistency**: By normalizing the face, alignment helps produce a standardized input for feature extraction, improving accuracy and reliability in subsequent steps.

Representation (Feature Extraction)

- **Feature Vector**: Converts the aligned face into a compact vector of key facial characteristics.
- **Machine-Readable**: A numeric vector that captures essential features, not directly interpretable by humans.
- **Efficient Encoding**: Stores critical information in a small, fixed-size vector, reducing storage needs.
- **Trained Model Output**: The model learns to produce vectors that capture unique and distinctive facial features.

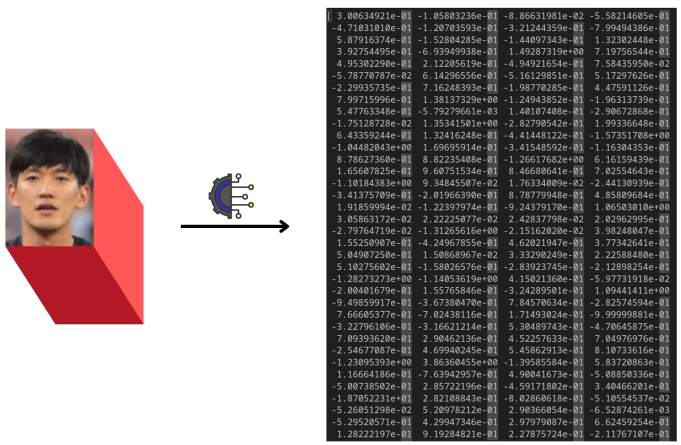


Figure 2: Embedding process: illustration

Classification

The purpose of this task is to predict the class of the input based on the feature vector the model has produced.

- Compute the similarities between the feature vector obtained and our classes
- The highest similarity indicates to which class the observed face is the most likely to be part of

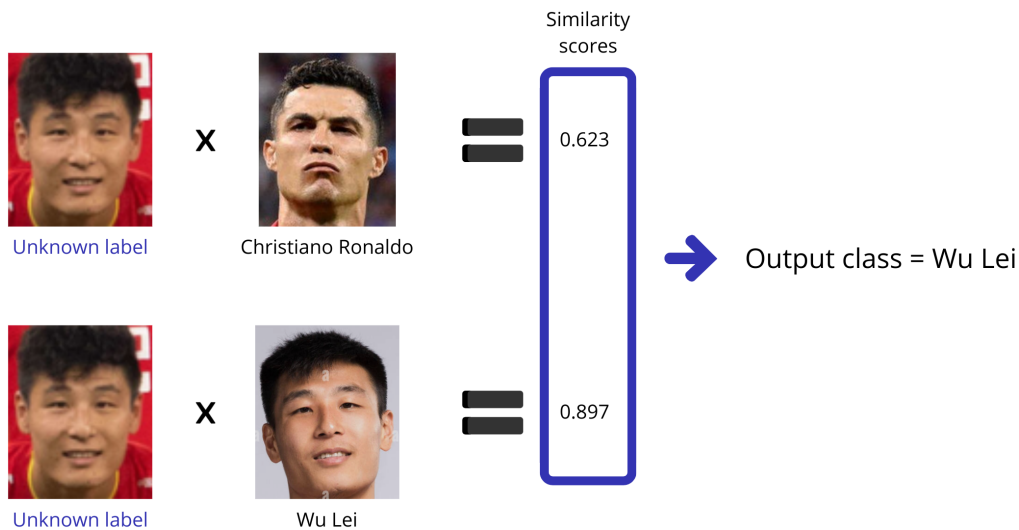


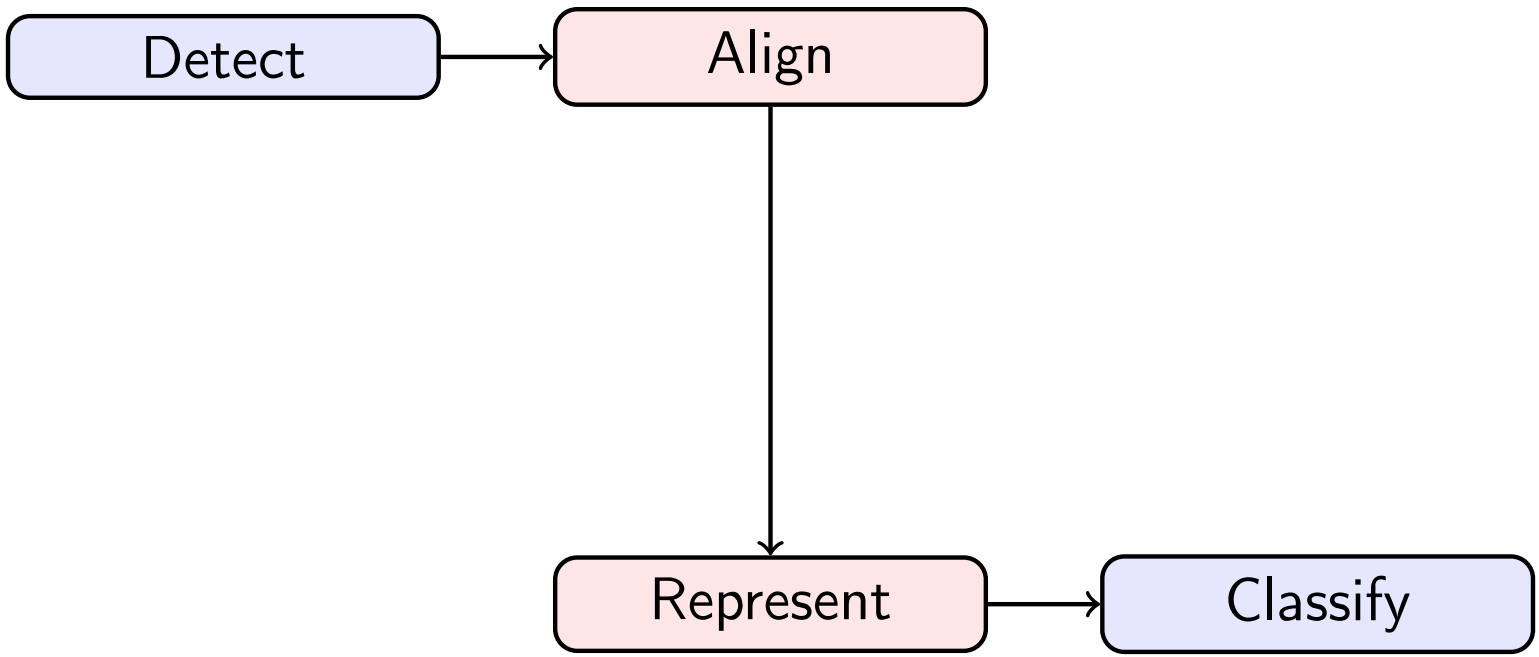
Figure 3: Class prediction: illustration

DeepFace

Figure 4: Taigman et al. (2014). [3]

- **High Accuracy**: Achieves 97.5% accuracy on human faces, making it one of the most accurate facial recognition models.
- **Human-Level Performance**: The model's performance is close to that of human recognition abilities, bridging the gap between machine and human facial recognition.
- **Compact Face Representation**: Uses a compact face representation through a feature vector, which captures essential facial features in a low-dimensional format.

Deepface: modified pipeline



Advancements by DeepFace

- **Innovations Introduced by DeepFace:**
  - **3D Face Alignment:** DeepFace uses a 3D model-based alignment to "frontalize" faces, improving accuracy in challenging conditions.
  - **Deep Neural Network Architecture:** Employs a deep neural network with over 120 million parameters, optimized for face recognition.
  - **Large Training Dataset:** Trained on 4 million images from 4,000 identities, enhancing its generalization to diverse faces.
  - **Compact Face Representation:** Generates a compact feature vector that efficiently represents a face, reducing the need for large descriptors.

- **Results:** Achieved 97.5% accuracy on the Labeled Faces in the Wild (LFW) benchmark, approaching human-level performance.

Deepface: Alignment

- **Goal of Alignment:** Normalize the face to a frontal view, reducing the effects of pose variation and ensuring consistent positioning of facial features.
- **2D and 3D Alignment:**
  - **Initial 2D Alignment:** Detects key facial landmarks (e.g., eyes, nose, mouth) and applies a 2D similarity transformation to roughly align the face.
  - **3D Alignment (Frontalization):** Uses a 3D face model to map the face to a frontal pose, addressing out-of-plane rotations.
- **Fiducial Points and Frontalization:**
  - Detects 67 fiducial points on the face for detailed alignment.
  - Applies a piecewise affine transformation using the 3D model to achieve a frontal view, preserving identity-related features.
- **Result:** The aligned face is in a standardized frontal position, making it more robust to variations in pose, which significantly improves the recognition accuracy.

DeepFace: 2D Alignment

- **Fiducial Point:** A specific point on the face that serves as a reference for alignment, typically located at key facial landmarks such as the eyes, nose, and mouth.
- 6 fiducial points are used
- Used to perform **basic 2D alignment**
- **Scaling** and **rotating** the image to a standardized view
- **Similar** to what is used in other face recognition systems

Figure 5: Fiducial points disposition on Sylvester Stallone's face

DeepFace: 3D Alignment

- **Mapping 2D to 3D:** The 2D aligned face is mapped to a generic 3D face model, allowing for a more robust representation of facial features. This mapping accounts for depth and structure, enabling the system to recognize faces with varying angles and orientations.
- **Frontalization of the Face:** By using the 3D model, DeepFace performs "frontalization," transforming the face so it appears as if it's viewed head-on. This process compensates for variations in pose and improves the consistency of facial representations.

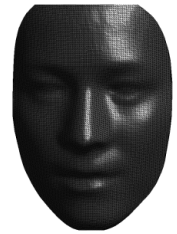


Figure 6: The reference 3D shape transformed to the 2D-aligned crop image-plane

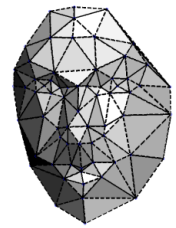
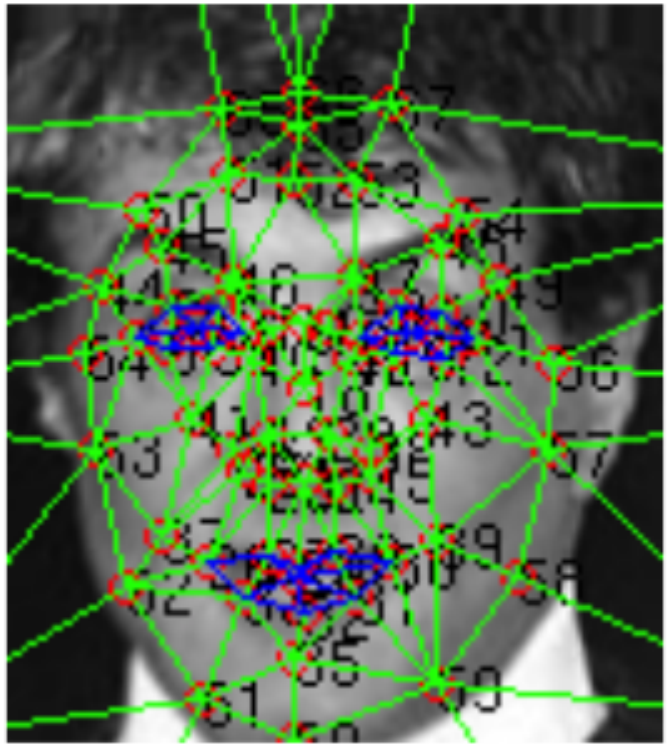


Figure 7: Triangle visibility w.r.t. the fitted 3D-2D camera; darker triangles are less visible

DeepFace: Fiducial Points and Frontalization

- **67 Additional Fiducial Points:** These are key points (such as eyes, nose, mouth) across the face used to improve alignment accuracy, especially in diverse poses and angles.
- **Consistent Alignment:** By precisely aligning based on these points, the system creates a standardized frontal view, reducing the impact of rotation and camera angles, which ensures reliable face recognition across different conditions.



DeepFace: Result

- **Frontalized Crop**: The final output is a standardized, frontal view of the face, which helps maintain a consistent perspective across different inputs.
- **Reduced Variability**: This process minimizes variations in pose, lighting, and expression, resulting in a more reliable representation for recognition. It enables better matching by focusing on identity-relevant features rather than environmental conditions.

Figure 8: Illustration of the frontalization process: initial face (left) and frontalized result (right)

Does DeepFace work with chimps?

- Attempting to use DeepFace for chimpanzee facial recognition
- Encountered an AttributeError when running the code

```
img_1 = DeepFace.extract_faces(path_1)
img_2 = DeepFace.extract_faces(path_2)
```

```
AttributeError: module 'deepface.modules.modeling' has no attribute 'build_model'
```

Figure 9: Error we encountered

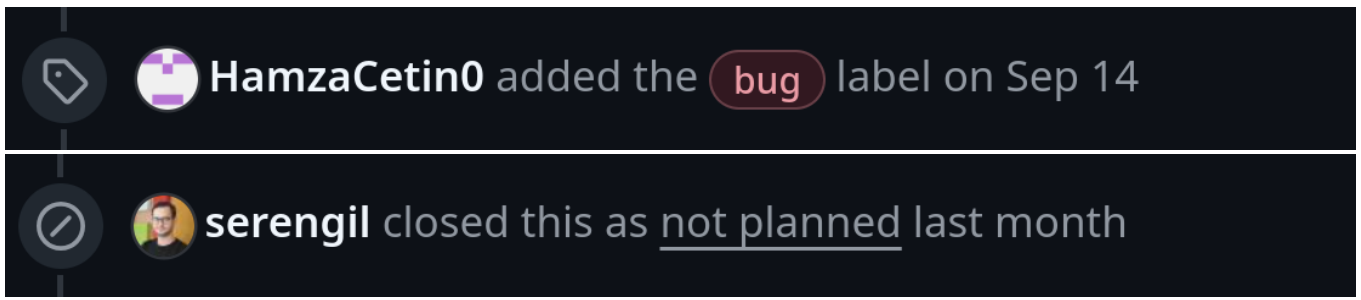


Figure 10: *Link to GitHub Issue*

SLIC Superpixel

- **What is a Superpixel?** - A superpixel is a cluster of pixels that groups regions with similar properties. - Commonly used in computer vision for image segmentation.
- **SLIC Superpixel Benefits:** - SLIC (Simple Linear Iterative Clustering) produces compact superpixels, preserving important features while simplifying the image.
- **Use in Chimpanzee Facial Recognition:** - Helps focus on meaningful facial regions, enhancing recognition accuracy. - Reduces computational cost by processing superpixels instead of raw pixels.

Achanta et al. (2010) [5]

SLIC Superpixel: examples


n_segments=20


n_segments=100


n_segments=1,000


n_segments=10,000

Figure 11: Visualisations of the application of the SLIC algorithm on shimpanzees.

Mask R-CNN

- **Mask R-CNN:** An advanced framework for instance segmentation that detects objects and generates high-quality masks for each instance. It builds upon Faster R-CNN by adding a branch for pixel-level mask prediction.
- **Use of Superpixels:** By leveraging SLIC superpixels, Mask R-CNN can process compact regions with similar properties, reducing computational complexity while retaining essential image features.
- **Chimpanzee Detection in Video:** Applied Mask R-CNN to accurately detect and segment chimpanzees in video footage, isolating each individual for further analysis.

He et al. (2017).[7]

Results and Limitations of Mask R-CNN

- **Interesting Results**: The Mask R-CNN model produced promising detections and segmentations of chimpanzees in the video. (See video for examples)
- **High Processing Time**: Processing a 2-minute video took approximately 15 minutes.
- **Recognition Errors**:
  - Occasionally, Mask R-CNN misidentifies non-chimpanzee objects as chimpanzees.
  - Some true chimpanzees in the video go undetected

Figure 12: Sample detections from the video

Motion detection: faster and more accurate

- **New Approach for Detection:** We plan to explore an alternative method that leverages temporal changes between frames to detect chimpanzees.
- **Using Pixel Changes:** By identifying pixels that change from one frame to the next, we can determine areas where chimpanzees are moving in the image.
- **Assumption of a Steady Camera:** This approach relies on the camera remaining steady, allowing us to detect only movement within the frame (i.e., chimpanzees) rather than changes in background or camera angle.
- **Benefits of Temporal Detection:** This technique is expected to be less time-consuming and more accurate by focusing on areas of motion rather than processing every object in each frame.

First Prototype of Chimpanzee Detection

- **Prototype Status:** We have developed an initial prototype for chimpanzee detection in images using mask R-CNN (pre-trained).
- **Flaws:** Slow, not accurate.

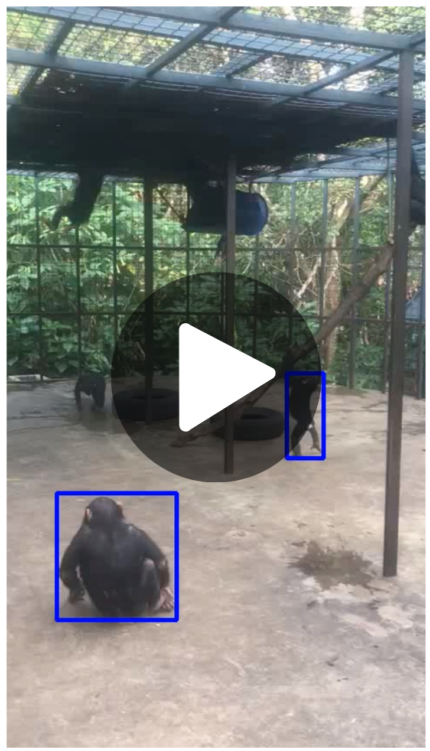


Figure 13: Chimpanzee detection: first prototype

Research on an automated chimpanzee identification system: general approach

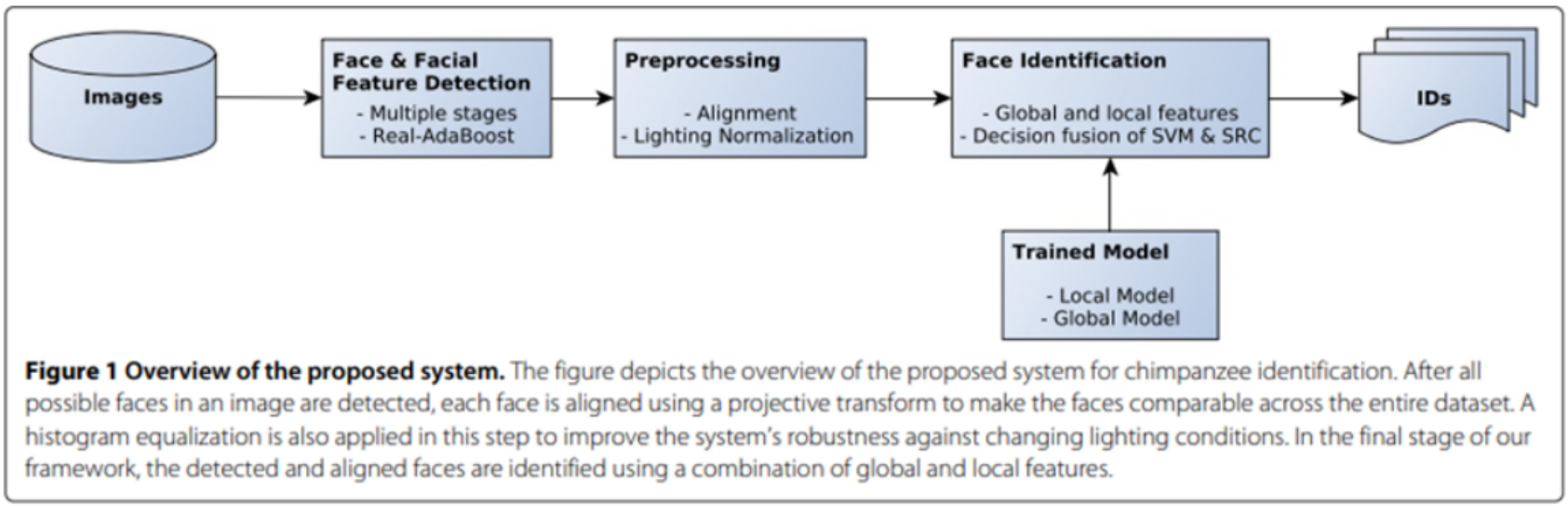


Figure 1 Overview of the proposed system. The figure depicts the overview of the proposed system for chimpanzee identification. After all possible faces in an image are detected, each face is aligned using a projective transform to make the faces comparable across the entire dataset. A histogram equalization is also applied in this step to improve the system's robustness against changing lighting conditions. In the final stage of our framework, the detected and aligned faces are identified using a combination of global and local features.

Figure 14: General approach for the face recognition of chimpanzee.

Research on an automated chimpanzee identification system: face and facial features detection

- **Goal**: Detect faces in the image and locate the characteristic points that will be used as landmarks to better understand the structure of the face.
- **Size of the model**: 24x24 (common for face detection)
- The face detection model has 9 stages has described below that use 3 different types of features :

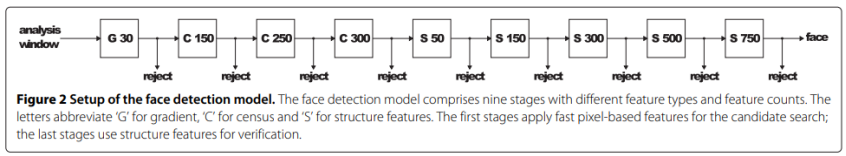


Figure 2 Setup of the face detection model. The face detection model comprises nine stages with different feature types and feature counts. The letters abbreviate 'G' for gradient, 'C' for census and 'S' for structure features. The first stages apply fast pixel-based features for the candidate search; the last stages use structure features for verification.

At each step, if it is detected that the current window has no face in it, it is rejected (the first step are quicker to compute but less accurate than the last ones)

Research on an automated chimpanzee identification system: 3 types of features

Here are the 3 types of features use for the detection:

- **Gradient direction**: We use the 3x3 Sobel kernel to extract the x (= horizontal) and y (= vertical) gradients. These gradients measure how brightness intensity changes locally in the image. If the gradient values of x and y are equal to zero, then the feature is encoded as 0 (region with no change in brightness). Otherwise, the gradient direction = atan2(sx,sy).
- **Census features**: We take a 3x3 neighborhood of pixels and compare the intensity of the central pixel with its 8 neighbors. If a neighbor is darker than the central pixel, it is assigned a 1, otherwise a 0, giving us an 8-bit string in which each bit represents a comparison between the central pixel and one of its neighbors.
- **Extended structures**: Takes into account a larger area. Enlarged versions of census features are calculated over areas of 3u x 3v pixels.

Research on an automated chimpanzee identification system: further detection

- **3x3 mean filter** : replaces a pixel with the average of the values of its immediate neighbors to reduce the noise in the image.
- **Image pyramid**: Resize the filtered image with different scaling factors and generate an image pyramid to detect faces of arbitrary size (24x24 in this case). We have different version of the image but with different resolution such that we could find a face in the image on a window of 24 by 24 even if the face is farther or closer.
- **Eyes detection**: After the face detection, we make a eyes detection wich requires 5 steps and less feature than face detection

Research on an automated chimpanzee identification system: face alignment, and lighting normalization

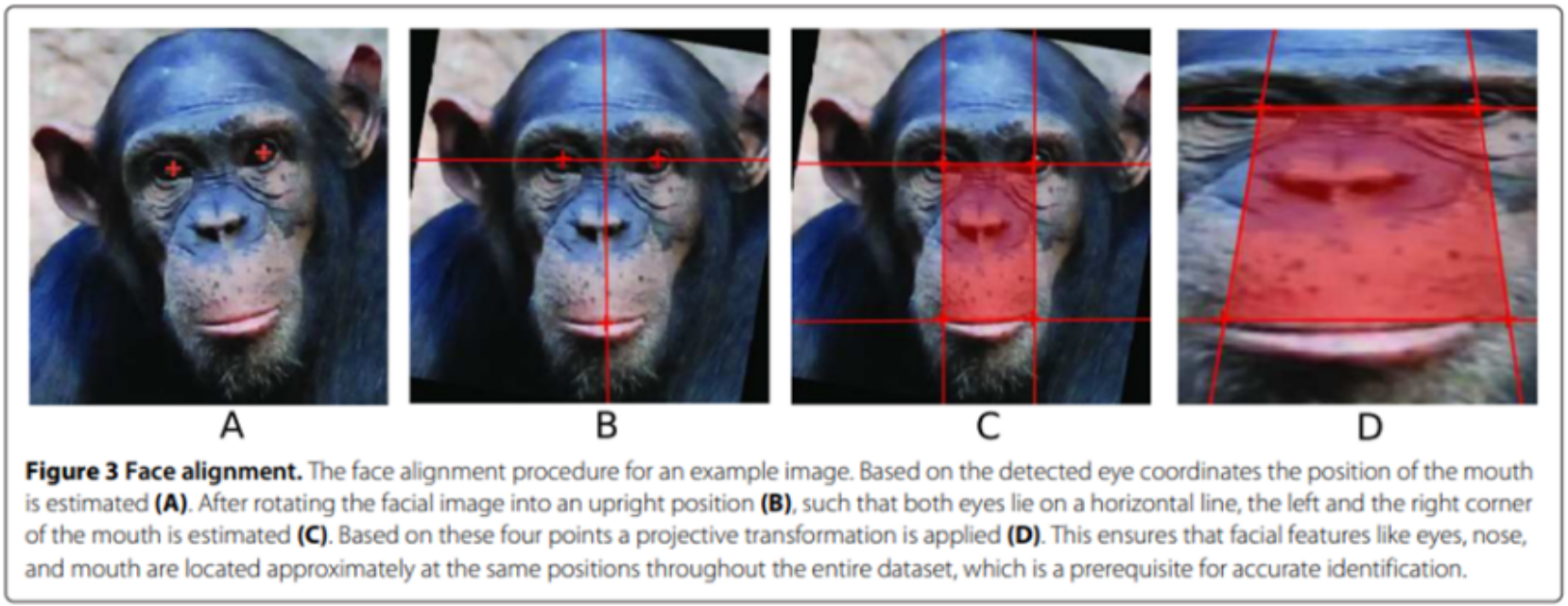


Figure 3 Face alignment. The face alignment procedure for an example image. Based on the detected eye coordinates the position of the mouth is estimated **(A)**. After rotating the facial image into an upright position **(B)**, such that both eyes lie on a horizontal line, the left and the right corner of the mouth is estimated **(C)**. Based on these four points a projective transformation is applied **(D)**. This ensures that facial features like eyes, nose, and mouth are located approximately at the same positions throughout the entire dataset, which is a prerequisite for accurate identification.

Figure 15: Summary of the face alignment.

Putting all faces in this form ensures that all facial features are comparable for all faces. We then set our aligned face image to gray scale and apply a simple histogram equalization to normalize brightness and adjust contrast.

Research on an automated chimpanzee identification system: individual identification

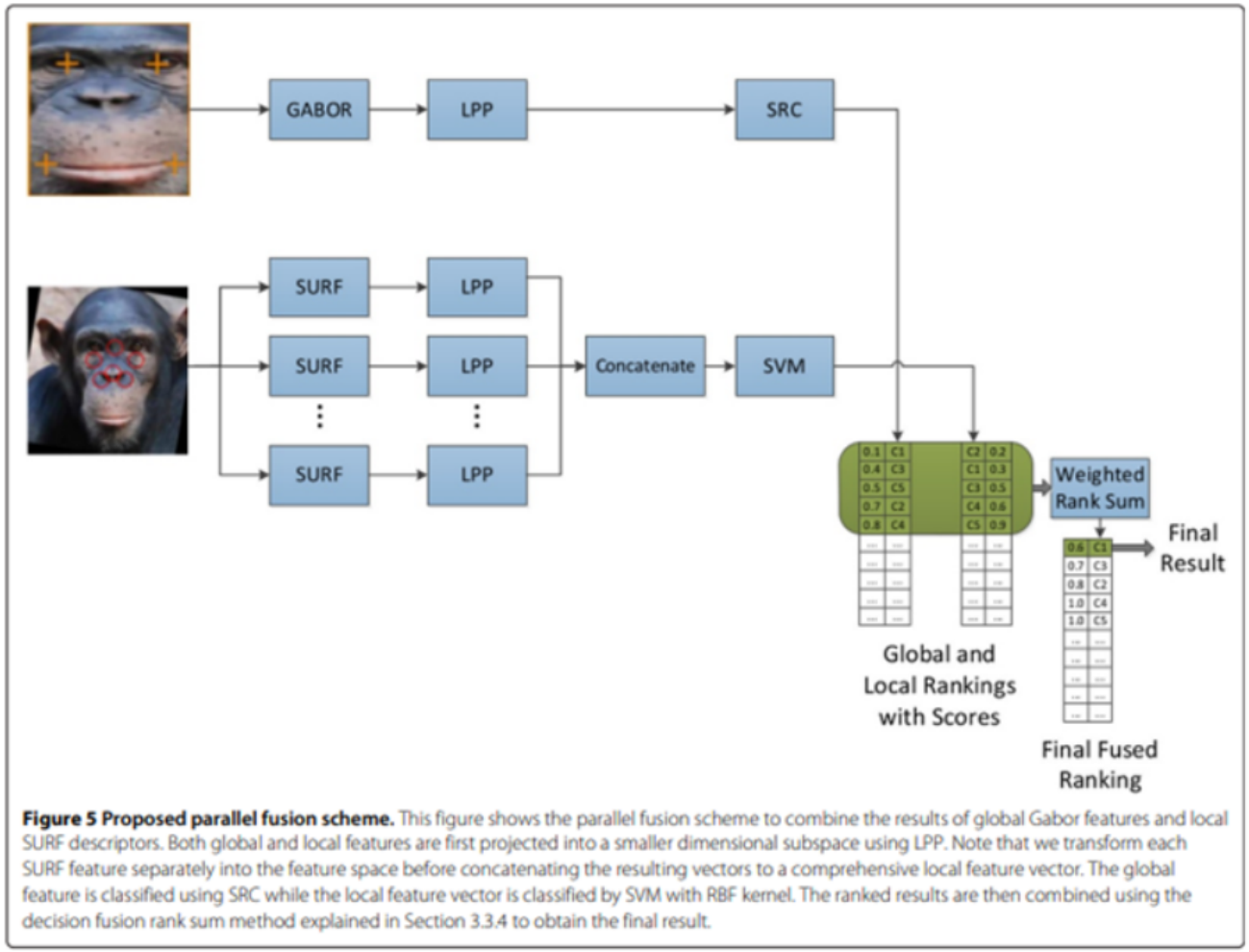


Figure 5 Proposed parallel fusion scheme. This figure shows the parallel fusion scheme to combine the results of global Gabor features and local SURF descriptors. Both global and local features are first projected into a smaller dimensional subspace using LPP. Note that we transform each SURF feature separately into the feature space before concatenating the resulting vectors to a comprehensive local feature vector. The global feature is classified using SRC while the local feature vector is classified by SVM with RBF kernel. The ranked results are then combined using the decision fusion rank sum method explained in Section 3.3.4 to obtain the final result.

Figure 16: Summary of the individual detection.

Research on an automated chimpanzee identification system: individual identification

3 steps:

- **Feature extraction**: global features with Gabor features and local features thanks to SURF (Speeded-up robust features)
- **Feature space transformation**: project the N-dimensional feature vector into a smaller subspace of size with LPP (locality preserving projections)
- **Classification**: fusion of 2 classificators (SRC for global features and SVM for local features). The fusion takes into account the degree of confidence of each classifier in its results. Each classifier assigns a score or probability to each class, and these classes are then ranked according to these scores. The rank thus indicates the priority or probability of each class being correct according to the classifier. The final score for each class is obtained by summing the two weighting functions associated with the SRC and SVM results.

Research on an automated chimpanzee identification system: global and local features

- **Global features** $\rightarrow$ Gabor features : convolve the grayscale input image I(z) with a set of kernels of Gabor
- **Local features** $\rightarrow$ SURF : Fast, robust point-of-interest descriptor and detector that doesn't depend on the scale and rotation of the image. SURF descriptors are extracted from 6 fiduciare points on the chimpanzee's face, based on where the eyes have been detected :

(The mouth region is not used for feature extraction because they noticed that this region is often subject to occlusion and facial expressions)

Chimpanzee face recognition from videos: general approach

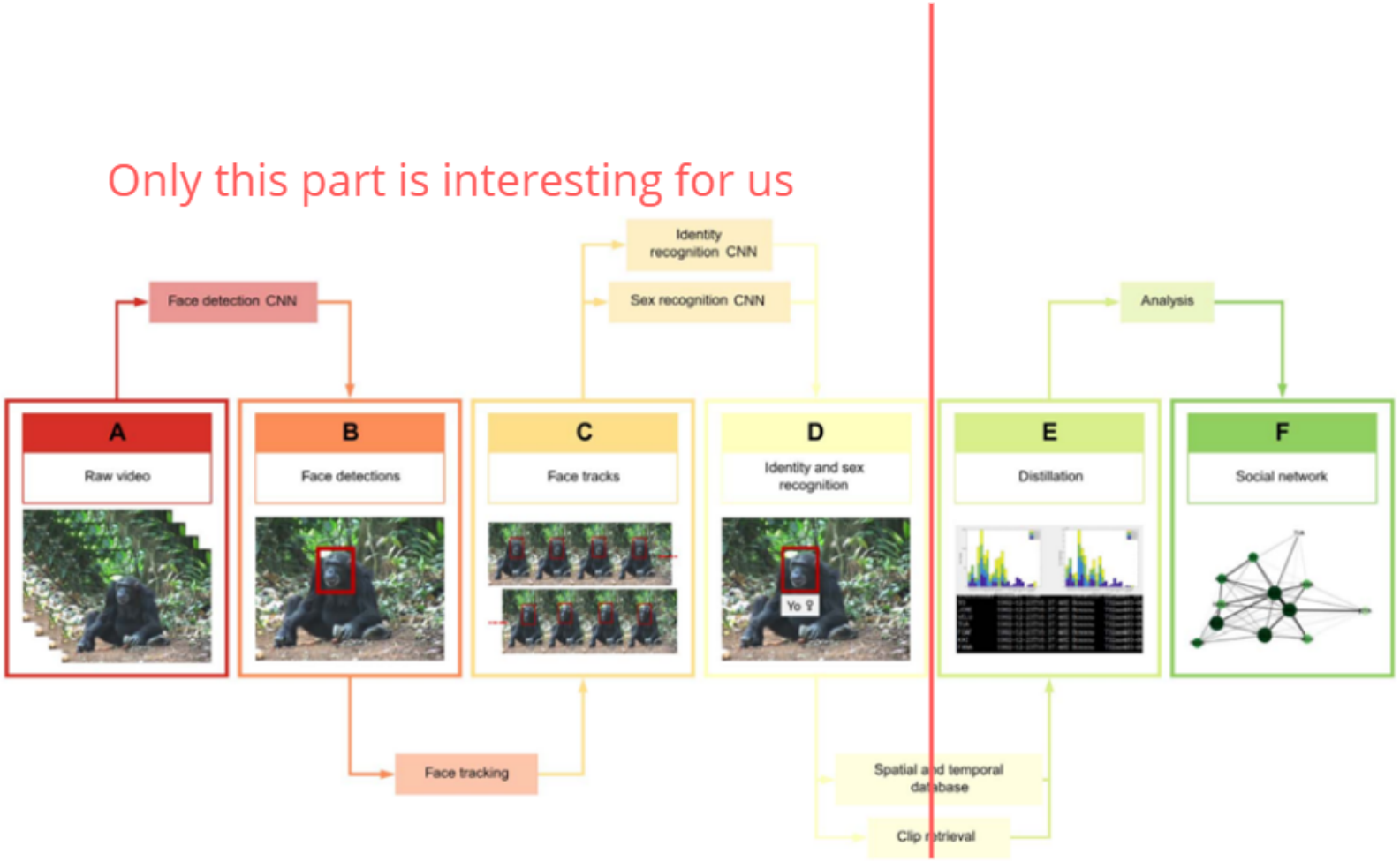


Fig. 1. Fully unified pipeline for wild chimpanzee face tracking and recognition from raw video footage. The pipeline consists of the following stages: (**A**) Frames are extracted from raw video. (**B**) Detection of faces is performed using a deep CNN single-shot detector (SSD) model. (**C**) Face tracking, which is implemented using a Kanade-Lucas-Tomasi (KLT) tracker (*25*) to group detections into face tracks. (**D**) Facial identity and sex recognition, which are achieved through the training of deep CNN models. (**E**) The system only requires the raw video as input and produces labeled face tracks and metadata as temporal and spatial information. (**F**) This output from the pipeline can then be used to support, for example, social network analysis. (Photo credit: Kyoto University, Primate Research Institute)

Chimpanzee face recognition from videos: general approach

3 steps :

- **Face detection**
- **Face tracking**
- **Identity recognition** (and sex recognition but not interseting in our case)

Little pre-processing (face alignment, brightness control, etc.), so the model is quite sensitive to certain parameters such as brightness, resolution, head orientation, ...

Chimpanzee face recognition from videos: Face detection

Pipeline :
1) Building a dataset of annotated frames from videos. Annotation consists of drawing a tight bounding box around each head.
2) Training our detector with this dataset.
3) The detector is executed on all the frames of the video, which makes it possible to detect the faces on each frame.
Evaluation protocol on a held-out test set (dataset not use during training) $\rightarrow$ precision/recall curve :

- **recall**: proportion of positive examples (= chimpanzee faces) that are correctly detected
- **precision**: proportion of examples classified as positive that are actually positive

$\rightarrow$ Priority to recall, because it's crucial not to miss any chimpanzee faces. For example, if the detector confuses the back of a chimpanzee with a face, these errors can be automatically corrected by a face tracking algorithm.

Chimpanzee face recognition from videos: Face detection, implementation details

The detector is an SSD detector (Single Shot MultiBox Detector) implemented thanks to MatConvNet: An SSD (Single Shot Multibox Detector) is a type of object detection model. It is designed to locate and classify several objects in an image in a single step (single shot).

Various data augmentation techniques are applied during the pre-processing phase to create more diversity in the training images. This helps to improve the model's robustness to variations in the images:

- **flip**: images are flipped to help the model learn image variants from different angles.
- **zoom**: images are enlarged or reduced to help the model learn to detect faces at different scales.
- **distortion**: geometric image distortions to simulate variations in outlook.

Chimpanzee face recognition from videos: Face tracking

Pipeline :

- **Shot detection**: consists in dividing each video into shots (= a continuous segment of video without any significant change of scene).
- **Face tracking**: the face detections made in each shot are grouped together to form face tracks. These tracks represent the trajectory of a face through several consecutive images in the same shot. This is done using KLT (Kanade-Lucas-Tomasi) tracking (= algorithm for tracking points of interest in images, often used to track moving objects in video).
- **Post-processing**: False positives (detection errors) are rarely tracked over several successive images, so tracks lasting less than 10 frames are eliminated.

Face tracking takes place within each shot, which means that the same face is not followed through several different sequences.

Chimpanzee face recognition from videos: Face recognition

Pipeline :

- **Dataset stage**: Face tracks extracted from chimpanzee videos are tagged with the identity of the corresponding chimpanzee.
- **Training stage**: After annotation, these data are used to train a chimpanzee facial recognition model. This recognition is first performed frame by frame (or "frame level"). The results for each frame within a face track are then aggregated to produce a single label for the entire track.
- **Recognition**: The recognizer is then applied to all face tracks and assigned a label.

Chimpanzee face recognition from videos: More details about recognition stage

For aggregation, the model predictions for each frame are combined by averaging the pre-softmax predictions. The model aims to classify faces into 23 different identity classes, corresponding to 23 individual chimpanzees. An additional class is added for face tracks that are false positives, and another for individuals that are not present in the dataset (called "negative class"). The network is trained to minimize the softmax log loss (cross-entropy) over 25 classes:

$$L = -\sum_{n=1}^{N}\left(y_c(x_n) - \log\sum_{j=1}^{C_i} e^{y_j(x_n)}\right)$$

where x_n is a single face input to the network, y_j is the pre-softmax activation for class j of size C_i, and c is the true class of x_n. Since the classes were heavily unbalanced (the number of training examples for each individual varies greatly; see Fig. 3A), the loss was weighted according to the frequency distribution of the classes. These weights for each class c are specified as

$$a_c = \frac{n_{\text{total}} - n_c}{n_c}$$

where n_{total} is the total number of training examples, and n_c is the number of training examples for class c. The weighted loss can then be expressed as

$$L_w = \sum_c a_c l(c)$$

where $l(c)$ is the component of the loss for class c.

Count, Crop and Recognise: Fine-Grained Recognition in the Wild

- **Goal**: Automatically label individual chimpanzees in every frame of a video, aiming to go beyond traditional face and body recognition.
- **Challenge**: High variability in pose, lighting, and occlusion makes detection difficult, especially in wild environments.
- **Approach**: Uses a multi-stage pipeline to improve recognition accuracy.
- **Results**: Achieves a balance between high precision (typical in face detection) and high recall (typical in full-frame approaches).

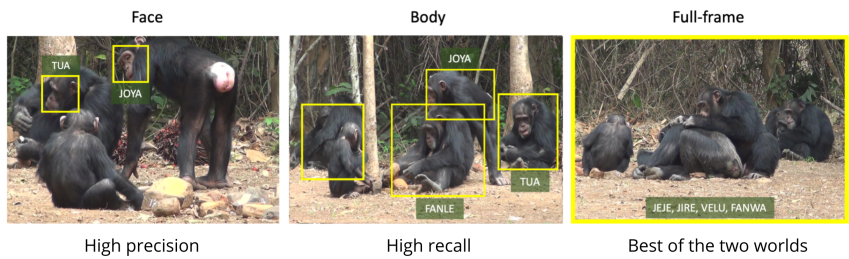


Figure 17: Example of recognition levels: Face (high precision), Body (high recall), and Full-Frame (best of both).

Count, Crop and Recognise: A Modified Pipeline

Detect $\longrightarrow$ Align $\longrightarrow$ Represent $\longrightarrow$ Classify

Count $\longrightarrow$ Crop $\longrightarrow$ Recognise

Bain et al. (2019).[11]

CCR Pipeline: Count, Crop, and Recognise

Count and Crop

Goal: Find the area in the image where all the individuals are located.

- A **Coarse-grained Counting CNN** detects areas with potential individuals.
- The **Region Proposal** step generates bounding boxes around detected areas.

Recognise

Goal: Use a multi-label classifier to identify all individuals within each cropped frame.

- A **Fine-grained Recognition CNN** is applied to the cropped regions to label each individual.

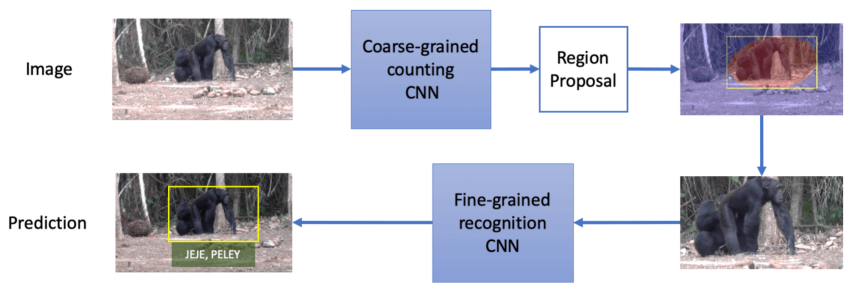


Figure 18: CCR pipeline

Challenges in Body Recognition and Multi-Instance Detection

Body Recognition Challenges

- **High Deformability**: Animal bodies, unlike faces, are highly flexible and vary greatly in shape, making recognition harder.
- **Lack of Clear Features**: It's challenging to identify consistent, discriminative features across different body poses and angles.

Multi-Instance Recognition with Weak Supervision

- **First in the Field**: They address the unique challenge of identifying multiple individuals in a single frame using limited supervision.
- **Fine-Grained Classification**: Classifying each individual within the same frame, especially under weak supervision, is critical for accurate, detailed recognition.

CCR: mathematical definitions

$\mathrm{x} \in \mathbb{R}^{C \times H \times W}$ $\rightarrow$ a single frame of the video

$\mathrm{x}' \in \mathbb{R}^{C \times H' \times W'}$ $\rightarrow$ an image cropped in x

$Y \in \{0,1\}^k$ $\rightarrow$
a finite vector indicating which of the k individuals are visible

$Y[i] = 1$ $\rightarrow$ if the i-th individual is visible

$Y[i] = 0$ $\rightarrow$ otherwise

Mathematically: Count (Definition)

What they need

$$c_\theta(x')$$

A function that counts the number of individuals within the cropped image.

Goal

They aim to predict the count as a class $n \in \{0, 1, \ldots, N\}$, where N is a hyperparameter.

- $c_\theta(x')$ is instantiated as a deep convolutional neural network (CNN).
- CNN structure: convolutional layers followed by fully connected layers.

Mathematically: Count (Approach)

How they get it

- The problem can be solved using:
  - A **regression model** (predicting a number within a range).
  - A **classification model** (predicting a class among a set of classes).
- Given the limited number of individuals per frame, they opted for a classification model.
- The function to minimize is the cross-entropy loss (standard for classification tasks).

- The input resolution H', W' is typically lower than the original H, W to align with pretrained CNNs.

Mathematically: Crop

Goal: Use Class Activation Maps (CAMs) to localize and crop regions containing chimpanzees in the frame.

- **Generate CAMs:** From the counting model $c_\theta(x')$, compute CAM $M_n(i,j)$ for each spatial location to highlight discriminative regions.
- **Formula:**

$$M_n(i,j) = \sum_k w_k^n f_k(i,j)$$

  where $f_k(i,j)$ is the activation of unit k in the last convolutional layer and w_k^n the weight for count n.
- **Identify Regions:** Normalize and threshold the CAM to isolate areas of interest.
- **Crop Image:** Bounding boxes around the largest connected components in the thresholded CAM define the cropped area, enhancing focus on chimpanzees.

Class Activation Map: illustration

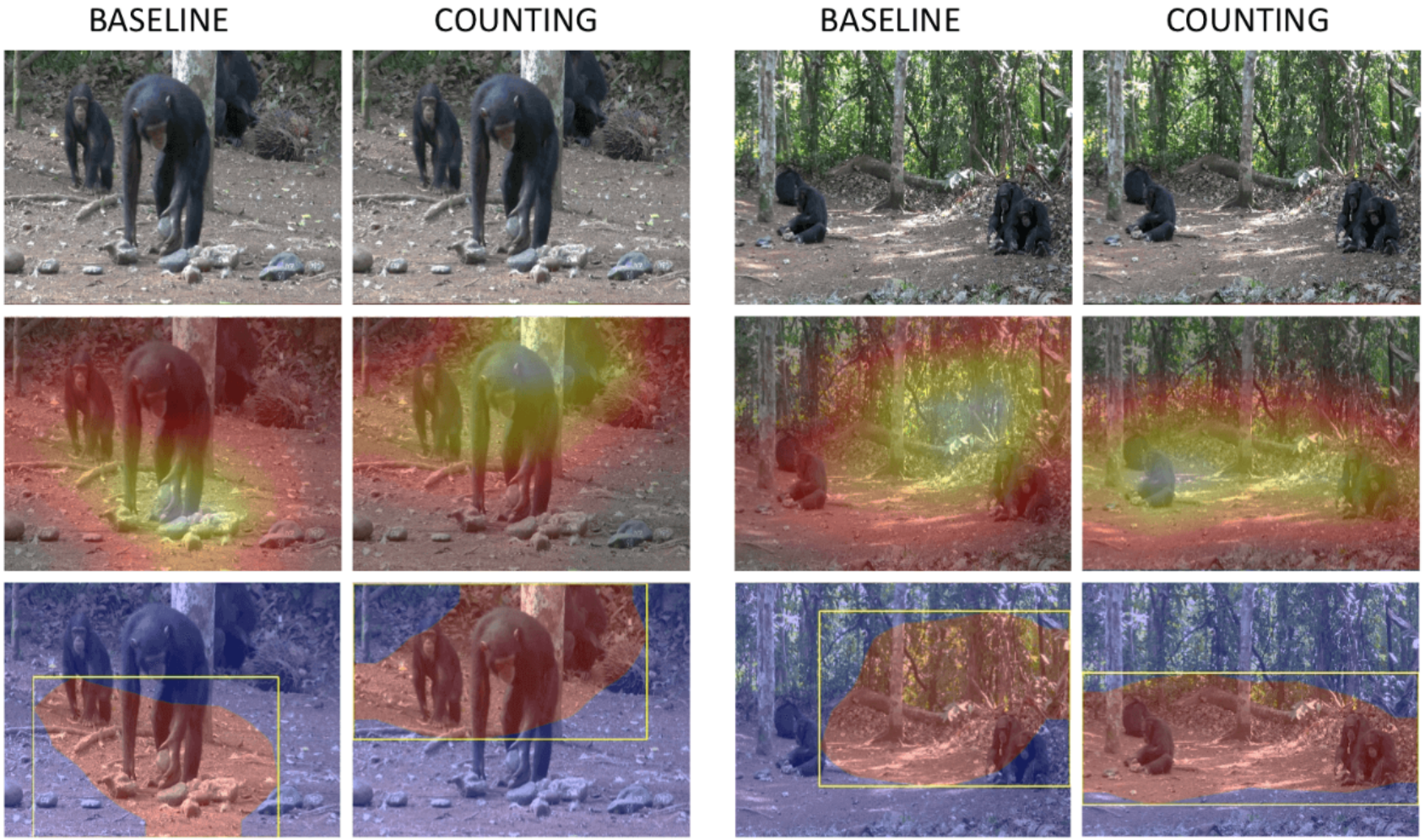


Figure 19: Region proposals via CAMs for the Chimpanzee Bossou dataset. Top: Original frame; Middle: CAM; Bottom: Region Proposal.

Cropping performance: Baseline model vs Counting approach

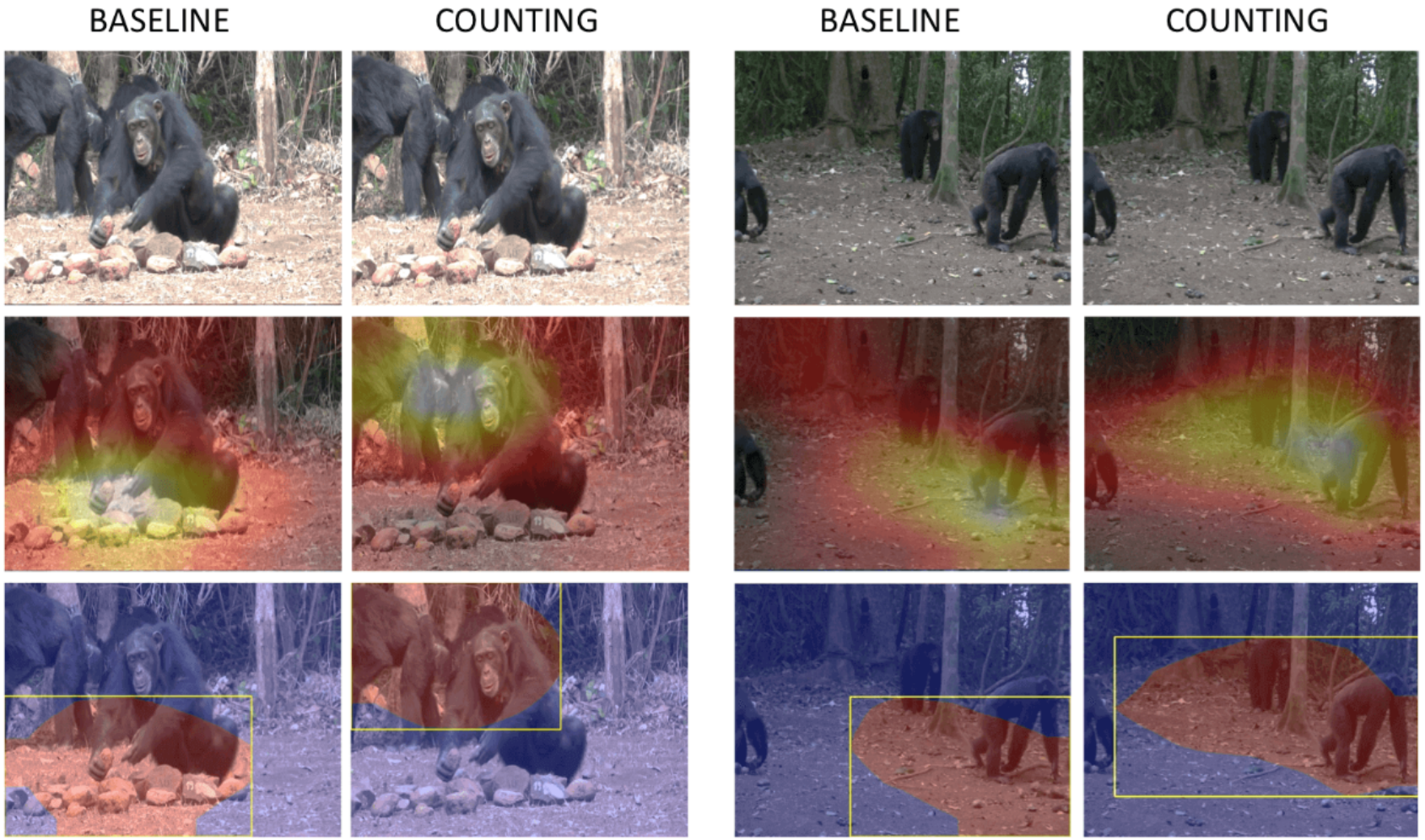


Figure 20: Baseline model and counting approach image cropping comparison.

Recognition

- **Recognition Model:**
  - Uses a fine-grained Convolutional Neural Network (CNN) to identify individual chimpanzees within each cropped region.
  - Adopts a multi-label classification approach, allowing the model to assign multiple identities in a single region if multiple chimpanzees are visible.
  - Trained to differentiate individual chimpanzee features, like fur patterns and facial structures, to achieve accurate recognition.
- **Handling Occlusions & Variability:**
  - Leverages spatial and temporal patterns to maintain identification even when chimpanzees are partially obscured or facing different angles.
  - Utilizes patterns in co-occurrence, as certain chimpanzees are often seen together, aiding in identifying individuals within groups.
  - Robust to pose changes and lighting variability, adapting to challenging real-world wildlife conditions.

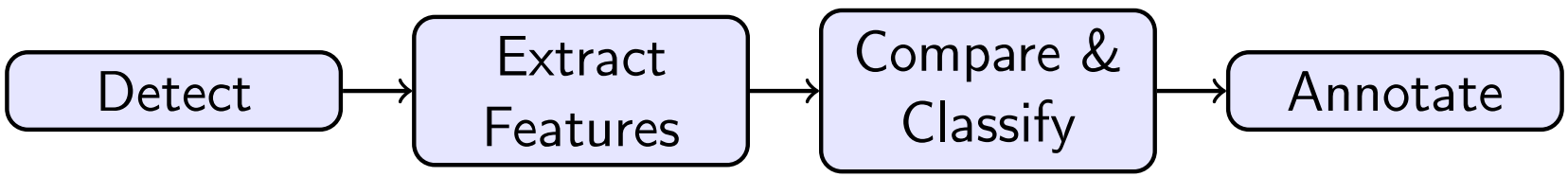
Advantages and Limitations of CCR in our case

Advantages	**Limitations**
Faster processing time, allowing efficient analysis of large video datasets	Does not locate the exact pixels belonging to each chimpanzee, lacking pixel-level segmentation
Identifies individual chimpanzees in each frame, useful for counting and general identification	Limited spatial precision, making it challenging to track exact movements within the frame
Suitable for distinguishing individuals in less complex backgrounds or low-occlusion conditions	Less effective in crowded scenes or with high occlusion, where precise localization is needed

Next Steps: our pipeline

Pipeline for Detecting and Labeling Chimpanzees in Video:

1. **Detect Regions with Chimpanzees:** Identify areas in each frame with chimpanzee presence.
2. **Crop & Feature Extraction:** Crop detected regions and generate feature vectors, using either facial or body recognition (method to be decided).
3. **Compare with Labeled Examples:** Match extracted feature vectors to our labeled dataset for identification.
4. **Annotate Video:** Use recognized identities to annotate each frame in the video.



Next objectives

- **Efficient Motion Detection:** Implement motion detection to quickly locate chimpanzee movement, reducing search areas in each frame.
- **Pre-trained Model for Recognition:** Explore pre-trained models for facial recognition in chimpanzees or humans that may be applicable. We have ideas:
  - **DeepFace:** Taigman et al. (2014). [3]
  - **Automated chimpanzee identification system:** Loos, A., & Ernst, A. (2013). [9]
  - **Chimpanzee face recognition from videos in the wild using deep learning:** Schofield et al. (2019) [8]

Future Questions for Project Development

- **Video Footage Status:**
  - Are the videos already recorded?
  - If not, can we ensure that the camera remains steady during filming?
  - **Rationale:** A steady camera enables us to detect motion accurately, improving chimpanzee localization.
- **Acceptable Execution Time:**
  - What is the maximum acceptable processing time for a video of length x minutes?
  - **Goal:** Determine a realistic time limit for efficient video processing, balancing accuracy and speed.

References I

- Barath, D., & Matas, J. (2018). Graph-cut RANSAC. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6733-6741)
- Turk, M. A., & Pentland, A. P. (1991, January). Face recognition using eigenfaces. In Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 586-587). IEEE Computer Society.
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

References II

- Tao, W., Jin, H., & Zhang, Y. (2007). Color image segmentation based on mean shift and normalized cuts. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(5), 1382-1389.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2010). SLIC superpixels. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
- Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut" interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics (TOG), 23(3), 309-314.
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

References III

- Schofield, D., Nagrani, A., Zisserman, A., Hayashi, M., Matsuzawa, T., Biro, D., & Carvalho, S. (2019). Chimpanzee face recognition from videos in the wild using deep learning. Science Advances, 5(9), eaaw0736.
- Loos, A., & Ernst, A. (2013). An automated chimpanzee identification system using face detection and recognition. EURASIP Journal on Image and Video Processing, 2013, 1-17.
- Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. In Proceedings of the IEEE international conference on computer vision workshops (pp. 142-150).

References IV


Bain, M., Nagrani, A., Schofield, D., & Zisserman, A. (2019). Count, crop and recognise: Fine-grained recognition in the wild. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).