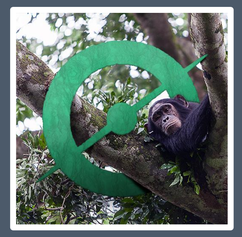
Pri-Matrix Factorization: Automated Species Tagging in Video Clips

Niranjan Balachandar (niranja9@stanford.edu)
Andrew Chen (awchen@cs.stanford.edu)
Jiwoo Lee (jlee29@stanford.edu)

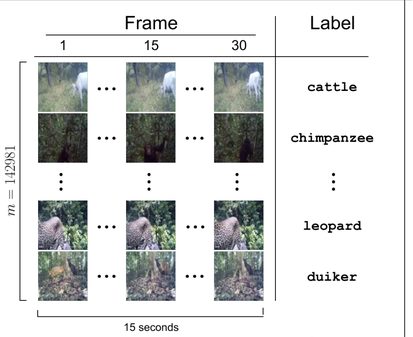


Introduction

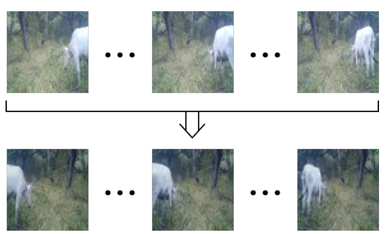
Cameras with heat sensors have made observation of the African jungle without human interference possible. However, thousands of experts have to skim through the videos to find and manually label the species present in videos captured by these cameras. We adapt models from machine learning and deep learning to create a classification pipeline that takes in videos and returns predicted probabilities of the presence of various animals. Our algorithm automates the process of video labeling, which has applications in numerous other domains as well.

Data

Our dataset is from the drivendata.org competition "Pri-matrix Factorization." Raw videos consist of 15 second RGB (0-255) video clips from camera traps, downsampled to 30 frames and 64px × 64px resolution, with crowd-sourced binary labels indicating the presence of a species. Examples of the data are shown in the figure to the right. During training we augmented with random horizontal flipping, random shading, and random contrast adjustment. Below is an example of a video pre and post augmentation. If x is the original augmented video, the x''' will be fed into the models:



$$x' = \text{flip}(x) \text{ if } P > 0.5 \text{ else } x$$
$$x'' = x'^{a}$$
$$x''' = x'' + b$$
where
$$P \sim \text{uniform}(0, 1)$$
$$a \sim \text{normal}(1, 0.05)$$
$$b \sim \text{normal}(0, 16)$$



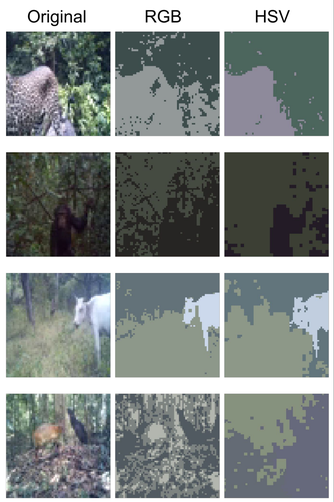
We use the competition's training, validation, and test set (we do not have access to test set labels). The training set consists of 142,891 labeled vids, the validation set consists of 61,239 labeled videos, and test set consists of 87,484 unlabeled videos. Labels indicate presence of 1 of 23 animals, or blank.

Features

The feature representation of the videos for K-means will be [x, y, H, S, V], which is described in more detail in the K-means section. The feature representation for the deep learning models was an array of size (30, 64, 64, 3) containing pixel intensities for each pixel position. For logistic regression, the input was vectorized.

K-Means for Segmentation

We noticed that many videos contained visually-distinct animals. We therefore sought to evaluate the performance of K-means for segmentation, with the intention of using the output as additional classification features. We initially naively used a vector [x, y, R, G, B], where R, G, B are the 0-255 RGB color values at pixel (x, y). However, this resulted in highly noisy segmentations (column 2). Therefore, we converted the feature vector to [x, y, H, S, V], where HSV (Hue, Saturation, Value) is a color model designed to reflect human color perception. This resulted in more visually-coherent segmentations (column 3). We optimized both the number of clusters as well as the scaling between the HSV/RGB and pixel location values.



Models

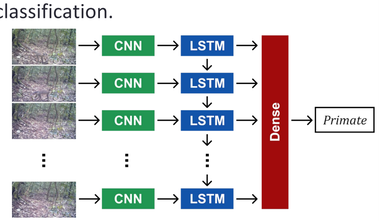
We first zero-centered and scale-normalized all data. As a baseline model, we implemented **logistic regression.**

Vanilla CNN + MLP

Our first main model was a vanilla 2D Convolutional Neural Network (CNN) across frames to extract feature vectors for each frame. These vectors were concatenated and input into an MLP for classification.

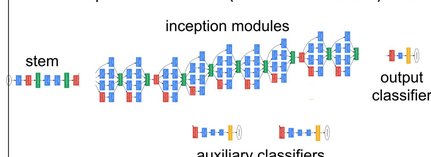
Vanilla CNN + LSTM

Because videos are time dependent, we substituted the MLP for an LSTM. This model is called a Long-term recurrent convolutional network (LRCN), and is summarized to the right.



Inception Net + LSTM

Because we are underfitting we increased the complexity of the CNN architecture. We implemented Inception V4, a high-performing CNN designed for ImageNet. It has inception modules (as shown below) that allow increased network depth.



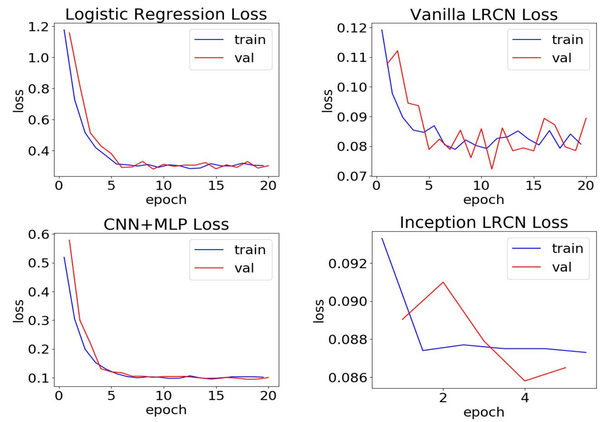
We use the following multilabel binary log loss function for optimization and evaluation:

$$J = -\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(y_j^{(i)}\log(\hat{y}_j^{(i)}) + (1-y_j^{(i)})\log(1-\hat{y}_j^{(i)})\right)$$

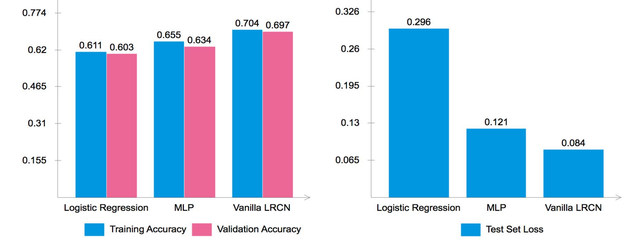
We implemented dropout and L2, but we are underfitting, so we do not apply them. We use Adam for learning and clipped gradients to prevent exploding.

Results

Training and validation losses during training (Inception still training):



The following are training and validation accuracies (Top-1), and test set losses calculated via test set prediction submissions to the competition.



Discussion/Future Directions

In summary, we have implemented a model to automate species labeling in video clips. As can be seen in the figures above, we underfit for all models (inception LRCN is still training). We lowered the regularizations (dropout and L2), removed contrast augmentation and increased model complexity to address the underfitting. As expected, the LRCN achieved the best performance on the validation and test sets. Future work will be to incorporate K-means into the classification pipeline, not use downsampled versions of the data- 64x64 might be too small for the CNNs, and continue increasing model complexity. An accurate video classifier can have important applications in numerous domains.

References

[1] "Pri-matrix Factorization." Max Planck Institute for Evolutionary Anthropology, 2017.
[2] J. Donahue et al., "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 677–691, 2017.
[3] T. Chen, Y. Chen, and S. Chien, "Fast image segmentation based on K-Means clustering with histograms in HSV color space," 2008 IEEE 10th Work. Multimed. Signal Process., pp. 322–325, 2008.
[4] C. Szegedy, et al., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.," Proc. IEEE Int. Conf. Comput. Vis., vol. 115, no. 3, pp. 4278–4284, 2017.