iWildCam 2019 - FGVC6

Categorize animals in the wild

阿不都维力　陶星辰　吴 欣

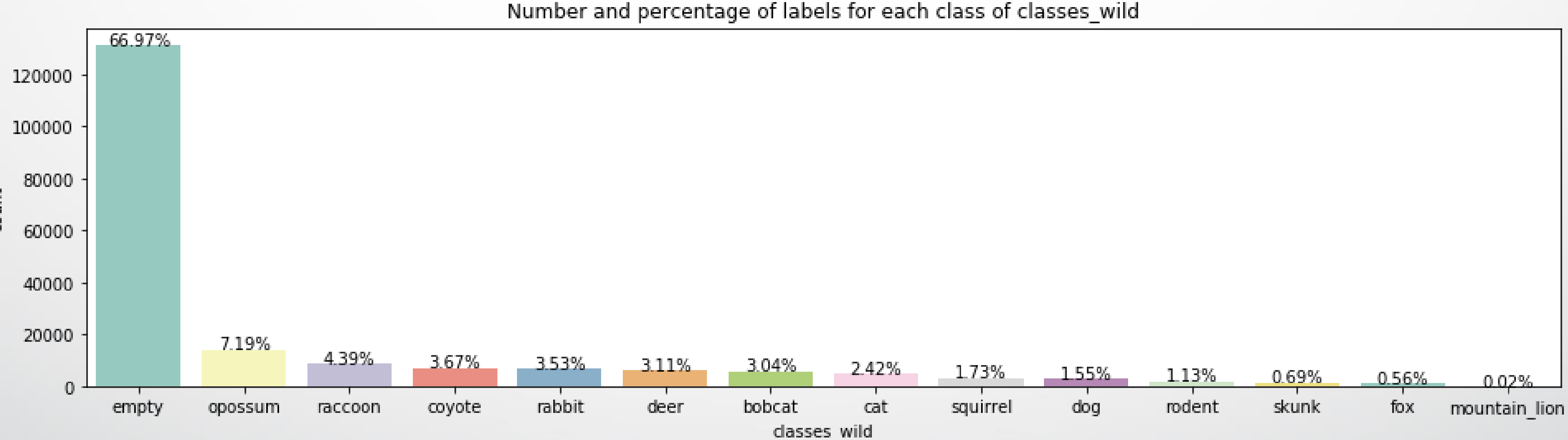
2019-06-10

CONTENTS

Challenges

Challenges on categories

- the training data and test data are from different regions. The American Southwest the American Northwest.
- The species seen in each region overlap, but are not identical. The total number of label is 23， the number of training data label is 14.



Challenges on images

- **Illumination:** Images can be poorly illuminated, especially at night. The example below contains a skunk to the center left of the frame.
- **Motion Blur:** The shutter speed of the camera is not fast enough to eliminate motion blur, so animals are sometimes blurry. The example contains a blurred coyote.
- **Small ROI:** Some animals are small or far from the camera, and can be difficult to spot even for humans. The example image has a mouse on a brance to the center right of the frame.
- **Occlusion:** Animals can be occluded by vegetation or the edge of the frame. This example shows a location where weeds grew in front of the camera, obscuring the view.
- **Perspective:** Sometimes animals come very close to the camera, causing a forced perspective.
- **Weather Conditions:** Poor weather, including rain, snow, or dust, can obstruct the lens and cause false triggers.
- **Camera Malfunctions:** Sometimes the camera malfunctions, causing strange discolorations.
- **Temporal Changes:** At any given location, the background changes over time as the seasons change. Below, you can see a single loction at three different points in time.
- **Non-Animal Variability:** What causes the non-animal images to trigger varies based on location.

Challenges on images

Camera trap data provides several challenges that can make it difficult to achieve accurate results.

Dataset

Smaller CCT images 27GB

iNat Idaho images and annotations 7.2GB zipped （supplementary dataset）

Method

- Animal Detection
- Image Augmentation
- Apply transfer learning on pre-trained models
- Advanced Regularization Strategy
- Ensemble learning

1. Detection

- The Camera Trap Animal Detection Model is a tensorflow Faster-RCNN model with Inception-Resnet-v2 backbone and atrous convolution.

2. Image Preprocessing and Augmentation

◆ Image preprocessing

- White-balance：To solve color reduction and tone processing problems
- Histogram equalization: This method is useful for images that are too bright or too dark in the background and foreground, but it also may increase the contrast of background noise and reduce the contrast of useful signals
- Image denoising: reducing noise in digital image
- Image resizing: reducing images size

◆ Data enhancement

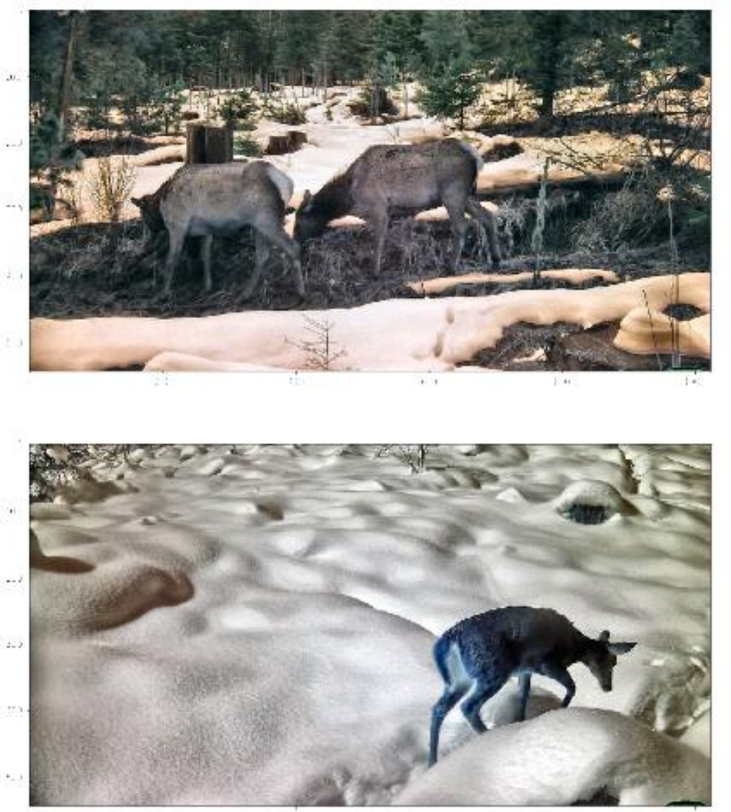
- Traditional methods: random cropping, rotation, translation, increasing brightness, increasing gaussian noise
- Histogram equalization
- Grayscale

◆ Data balancing

2. Image Preprocessing and Augmentation

Original Image

White Balance
+ Histogram equalization



2. Image Preprocessing and Augmentation

Image Augmentaion

Original Image

Blur

Rotate & Shift

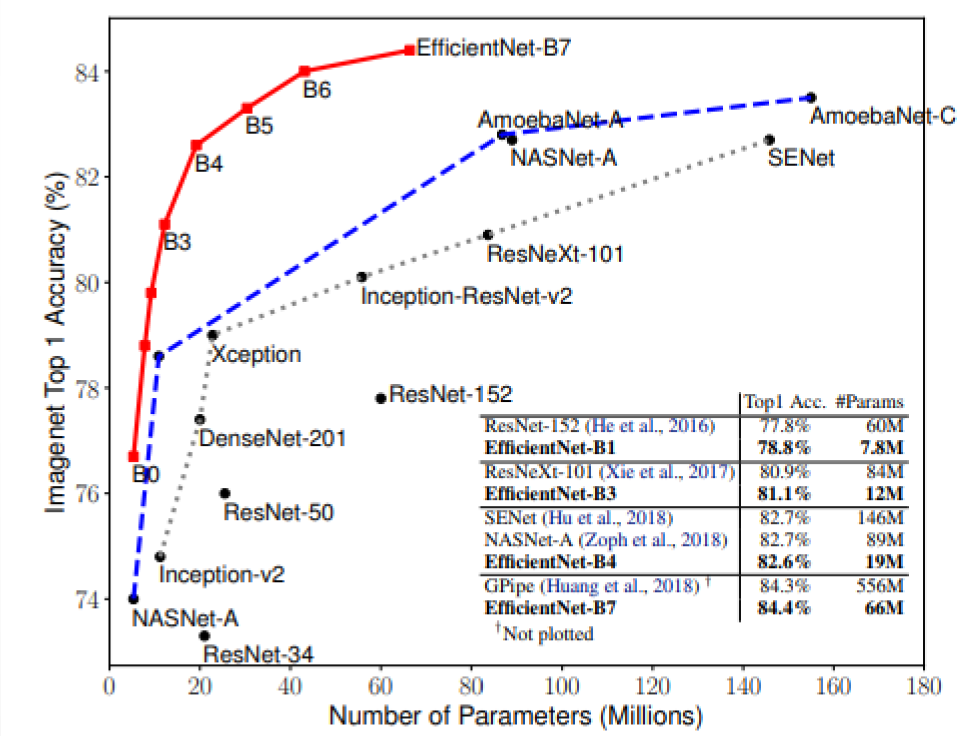
To Gray

3. Pretrained Models for transfer learning

- Resnet50
- Resnet101
- Efficientnet-b0
- Efficientnet-b3
- Vgg16
- Vgg19
- Densenet121
- Xception

3. Pretrained Models for transfer learning

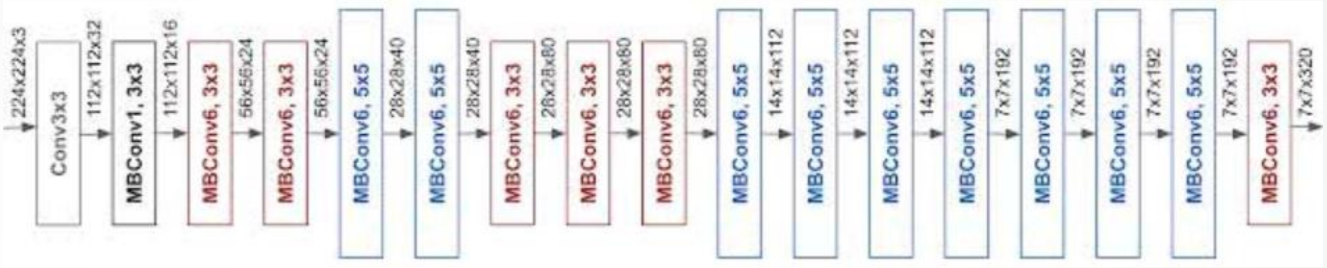
Efficientnet



Tan, Mingxing , and Q. V. Le . "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." (2019).

3. Pretrained Models for transfer learning

Efficientnet-b0



Model Scaling

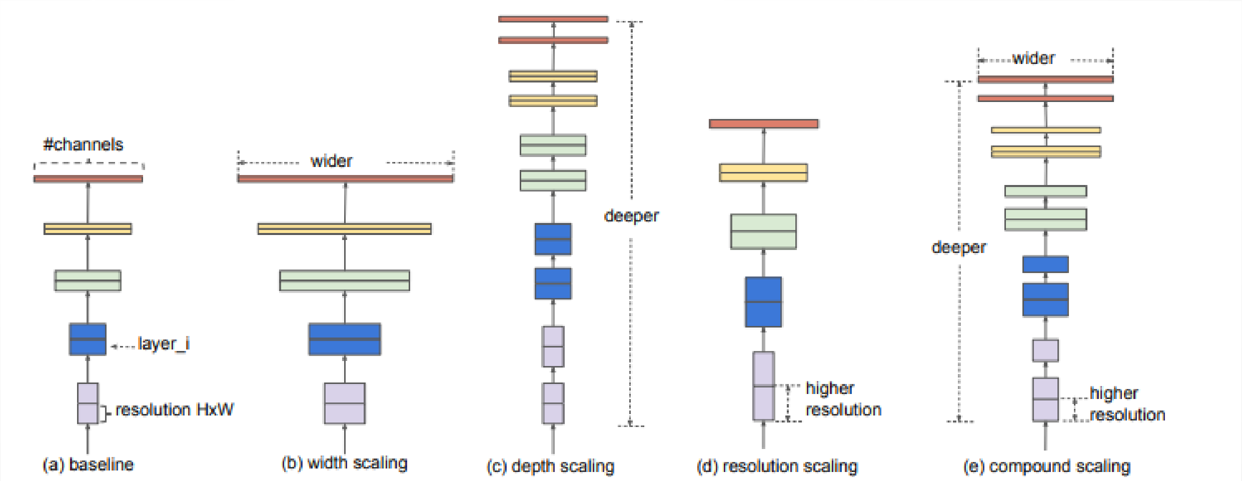


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Tan, Mingxing , and Q. V. Le . "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." (2019).

4. Advanced Regularization Strategy

- Label smoothing： it can effectively suppress the over-fitting phenomenon when calculating the loss value by “softening” the traditional one-hot type label
- Cut out： a simple regularization technique of randomly masking out square regions of input during training, can be used to improve the robustness and to avoid overfitting
- Mixup： a large deep neural network may be sensitivity to adversarial and memorization, mixup is a simple learning principle to alleviate these issues.

4. Advanced Regularization Strategy

◆ Label smoothing

- Modify your one-hot targets with equation (original notation from paper):

that

$$q'(k) = (1 - \epsilon)\delta_{k,y} + \frac{\epsilon}{K}.$$

We refer to this change in ground-truth label distribution as *label-smoothing regularization*, or LSR.

K - is the number of classes
epsilon is a hyperparameter, usually 0.1

Szegedy, Christian , et al. " [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Rethinking the Inception Architecture for Computer Vision." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* IEEE, 2016:2818-2826.

4. Advanced Regularization Strategy

◆ Cut out

Randomly masking out square regions of input during training

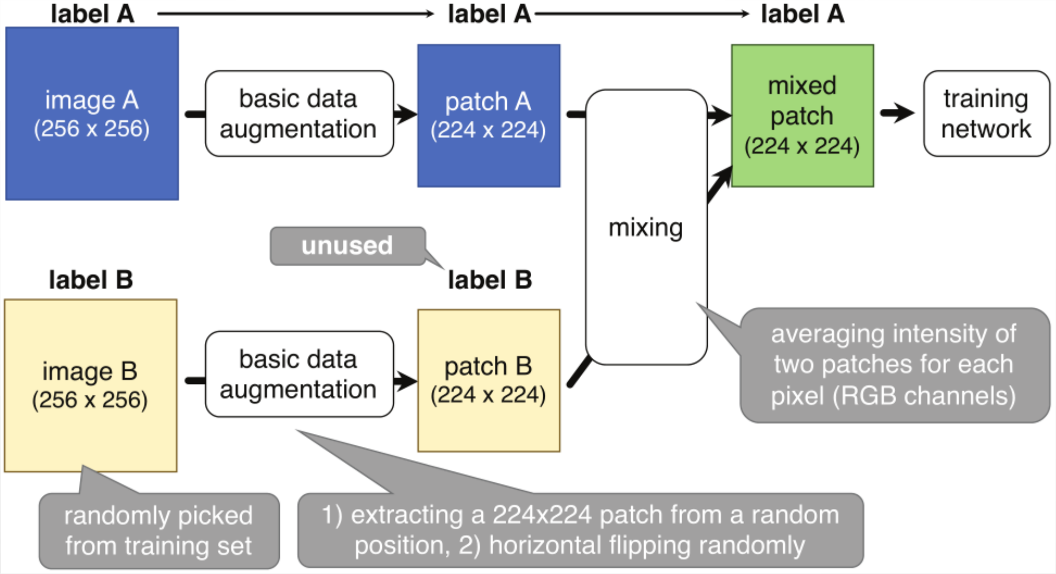
Devries, Terrance , and G. W. Taylor . "Improved Regularization of Convolutional Neural Networks with Cutout." (2017).

4. Advanced Regularization Strategy

◆ Mixup

Contribution Motivated by these issues, we introduce a simple and data-agnostic data augmentation routine, termed *mixup* (Section 2). In a nutshell, *mixup* constructs virtual training examples

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$



Inoue, Hiroshi . "Data Augmentation by Pairing Samples for Images Classification." (2018).

5. Ensemble learning

◆ Averaging the probability of following 9 predictions:

3 models * 3 different test image preprocessing = 9 prediction results

3 models :

(1) Resnet 101, (2) Efficientnet-b0, (3) Efficientnet-b3

3 data pre-processing for test image:

(1) original test image, (2) CLAHE, (3) to Gray

5. Ensemble learning

◆ Training

- Data enhancement
- 3 pretrained models: resnet101, efficientnet-b0, efficientnet-b3
- Optimizing strategies

◆ Testing

- Test data resized to 128*128
- Test data based on Histogram equalization and resized to 128*128
- Test data based on grayscale and resized to 128*128

◆ Integrate 9 prediction

- F1 score finally reached 0.228
- Rank the tenth

Result

Private leaderboard: **top 3 % (7/336)**

#	Δpub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	epsilon			0.399	75	4d
2	▲1	Rb			0.362	2	4d
3	▼1	Hayder Yousif			0.340	29	4d
4	▲1	Local Minima			0.319	18	4d
5	▲1	rewolfiluac			0.303	65	5d
6	▼2	xiang			0.300	30	12d
7	▲2	JustVeg			0.228	8	4d
8	▼1	Hugo Touvron			0.227	8	6d
9	▼1	Dave Coates			0.218	21	5d
10	▲2	Patrick Byrnes			0.212	22	5d

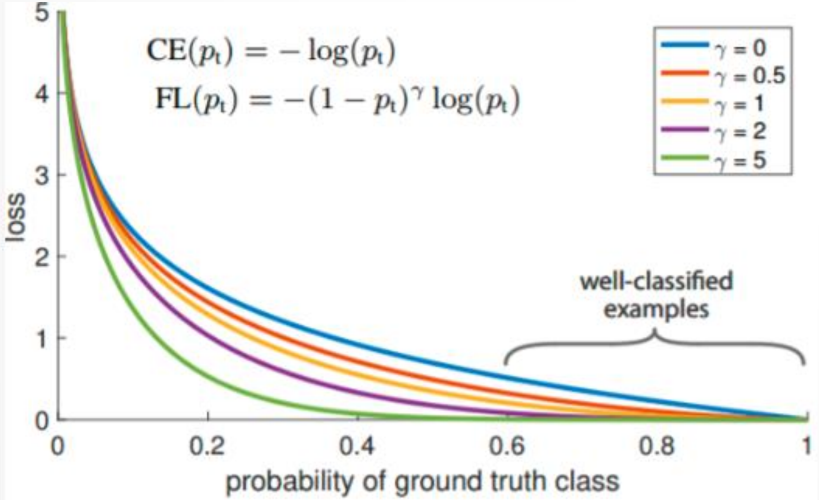
1. Best single model:
Efficientnet_b0 with image augmentation and tricks just mentioned
Efficientnet_b0 Private leaderboard score: F1=0.225

2. Ensemble model:
Private leaderboard score: F1= 0.228
(PS: I've just adopted the ensemble strategy of averaging the results of different models, and if I change to other ensemble methods (e.g. stacking), the performance will be better.)

3. Why are we so much worse than the top 6 teams?
I think, because we cannot consider the data shift (domain shift) between train and test set

Other Attempts

1. Focal Loss

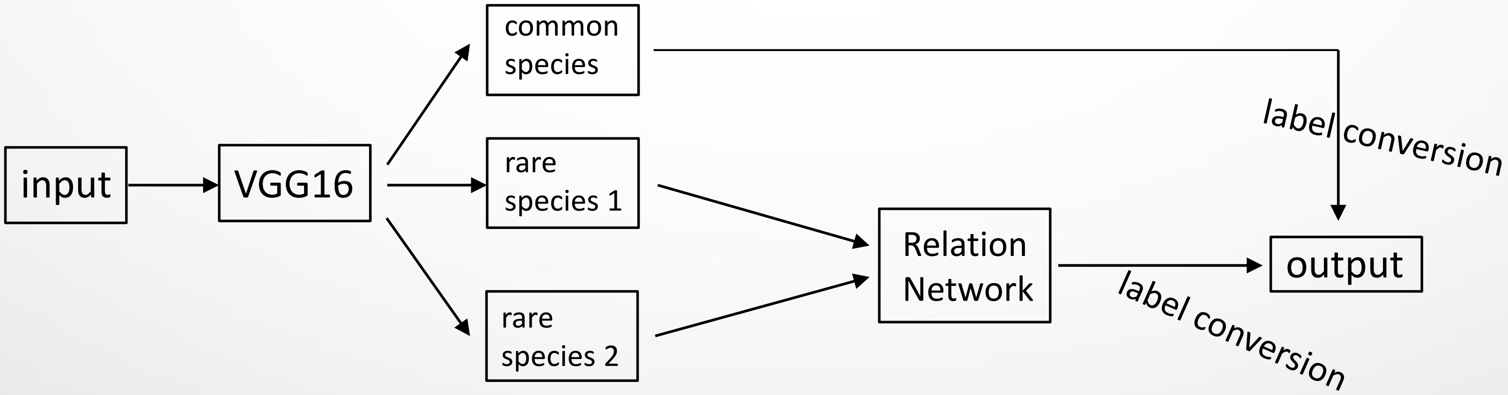


$\mathrm{CE}(p_t) = -\log(p_t)$
$\mathrm{FL}(p_t) = -(1-p_t)^\gamma \log(p_t)$
$\gamma = 0$
$\gamma = 0.5$
$\gamma = 1$
$\gamma = 2$
$\gamma = 5$
well-classified examples
loss
probability of ground truth class
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5

2. Pipelined models

No data augmentation. No ensemble. Rebalance the data distribution a little.

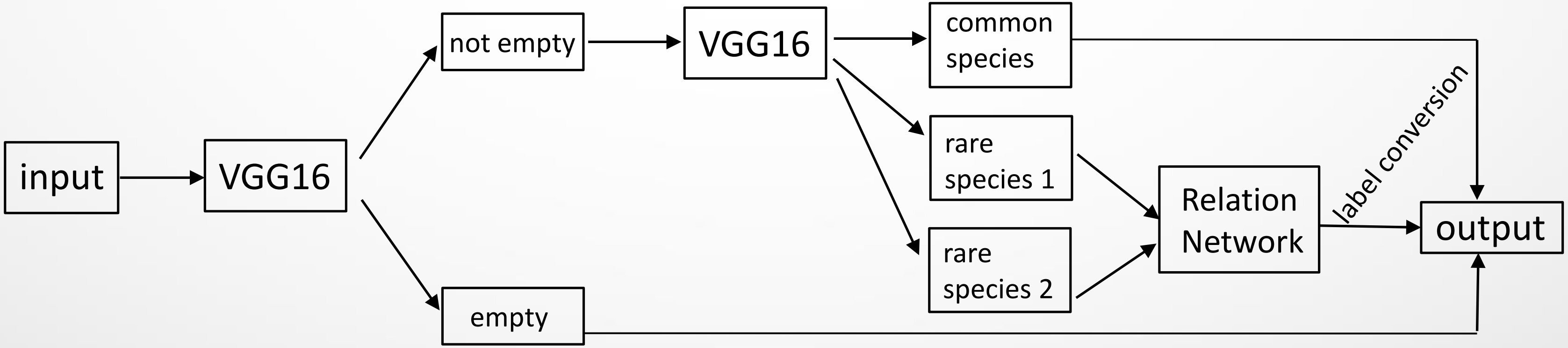
Group those classes with only a few samples into new classes, namely rare species.
(according to the resemblance in shape, there rare species 1 and rare species 2)



2. Pipelined models

This model architecture can’t distinguish between ‘empty’ and ‘something’.
(As a matter of fact, no prediction is ‘empty’ during test. Maybe it’s because the data rebalance…)

With a little modification:



2. Pipelined models

Score: 0.154

We didn't have enough time to do some ablation tests, but we conjecture that data augmentation has played an important role in the final results.

3. Some experiments

◆ data preprocessing + data balance + transfer learning

- To improve brightness and contrast by histogram equalization
- To correct the color by white balance
- To increase the quantity of images which label count is less than 5000 by data enhancement, and to reduce the quantity of images which label count is more than 5000 by step sampling
- Reduce the image size to 224*224
- Transfer learning based on resnet50 of ImageNet

3. Some experiments

◆ data preprocessing + data balance + transfer learning + ensemble

- To improve brightness and contrast by histogram equalization
- To correct the color by white balance
- Only reduce the quantity of label 0 at the rate 0.1
- Reduce the image size to 48*48 and 71*71
- Transfer learning based on vgg16, vgg19, densenet121 and Xception of ImageNet
- Compute the average probability of the sum of model predictions
- Compute the weighted average probability of these models

3. Some experiments

◆ data preprocessing + transfer learning + ensemble

- To improve brightness and contrast by histogram equalization
- To correct the color by white balance
- Reduce the image size to 48*48 and 71*71
- Transfer learning based on vgg16, vgg19, densenet121 and Xception of ImageNet
- Compute the average probability of the sum of model predictions
- Compute the weighted average probability of these models

Todo •••

1. GAN for domain adaptation

the training data and test data are from different regions.
There are domain shift between two dataset.

2. Advanced Transfer learning
Transfer learning with domain shift

3. Advanced Ensemble Strategy
Stacking, Boosting …

THANK YOU

Abuduweili
Xingchen Tao
Xin Wu

https://github.com/Walleclipse/iWildCam_2019_FGVC6