Term Project Plan - DS801, Data Engineering for Big Data Analytics

Description of Data

Target Data: A dataset of 31,848 animal images with 10 classes, which will be used to train an AI model for 10-class animal image classification. There is no Validation and Test data provided.
Link: https://drive.google.com/file/d/1B0TqxV0bsQrmZYF1IHdNZnbmth5HivoJ/view?usp=drive_link (Also, can be available on KLMS platform)

- **Brief Data Statistics**:

Cat.	Cat	Lynx	Wolf	Coyote	Cheetah	Jaguar	Chimpanzee	Orangutan	Hamster	Guinea Pig
ID	0	1	2	3	4	5	6	7	8	9
# Img.	4698	1542	1949	5361	4911	2681	2292	3472	2520	2422

- **File Format**
  - All the images were stored 64x64 pixel raw images in JGP.
  - Every image has the same file name format: **img_{*random-image-key*}.jpg**
    - *random-image-key*: a unique key to designate each image
  - Data labels are saved in JSON format in the **data-labels.json** file
    - This is a dictionary with key "img-id" and value "one-hot vector" of the data label
    - e.g., {"img_786a2bec-78a7-4fd1-9274-515487390f0b": [0, 0, 1, 0, 0, 0, 0, 0, 0, 0], …}

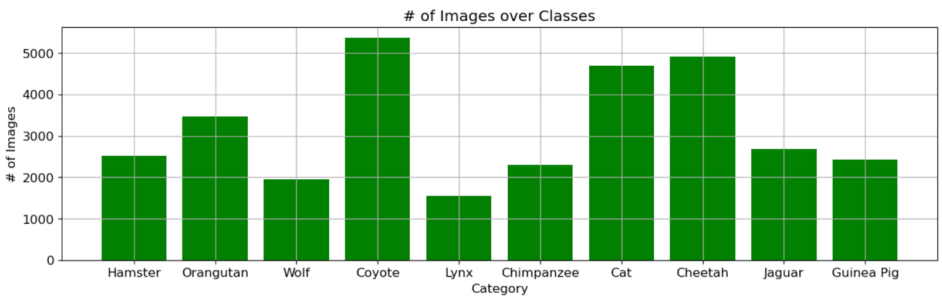
- **Data Quality Challenge**

This dataset suffers from multiple data quality challenges:

- **1) Out-of-distribution Images (Outliers)**: There are many images that are not animals belong to the pre-defined 10 classes above. For examples,

- **2) Noisy Labels:** There are many images that were mislabeled by human and corrupted by an adversary (= Professor). The noise ratio is unknown to make the project challenging. For example, you can see the mislabeled images mislabeled as Cat:

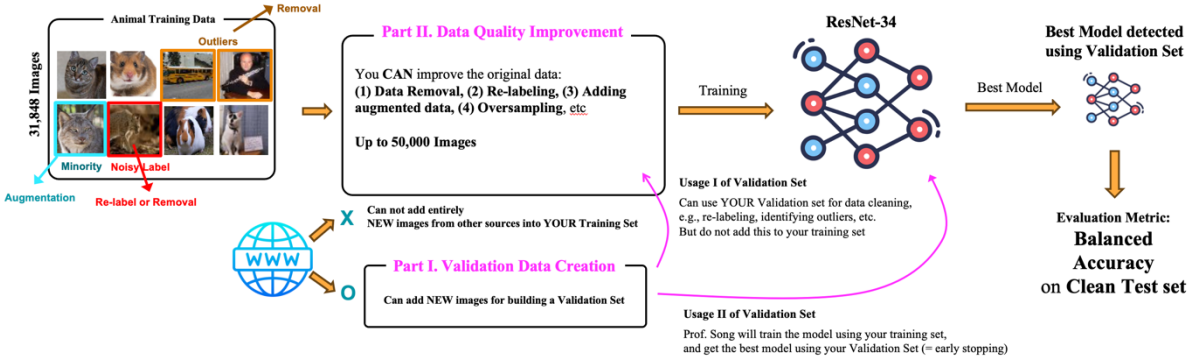
- **3) Class Imbalance:** The data has the class imbalance problem. The distribution of the number of images over classes are summarized as:



Objective of Term Project

Objective: This term project involves ***(Part I) creation of your own clean validation dataset***, and ***(Part II) improvement of the data quality of the given data***.

The give data is a training set to learn the ResNet-34 (in Pytorch) for animal image classification. We train the dataset using the ResNet-34 model from scratch (not using the model pre-trained for ImageNet-1K). Therefore, the objective is to **achieve optimal performance on clean test data** through these efforts. The figure below outlines your term project, highlighting Part I and Part II.



Part I. Validation Data Creation

Your **first goal** is **creating your own validation data** used for two purposes: **(Usage I)** improving your training set by yourselves and **(Usage II)** checking the performance of the model trained on your training data to determine whether its quality has been improved enough or not, since there will be no clean data provided to you. Furthermore, your validation set will be utilized on my side for detecting the best model during training, which involves early stopping by Prof. Song.

You can use any approach to collect the **CLEAN validation data** for the 10 animal classes, e.g., crawling with web search, human labeling, etc. There is no rule for this validation data as long as there are more than 1 images per class. Also, in Part II, it is not mandatory to use a validation set; if there is a better method, it can be used. Refer to the details in Part 2.

A basic way of creating CLEAN validation data is crawling Images with animal category names as keywords using Google Image Search, and then verifying the label quality on your own. You can select any approach to build this data.

Part II. Data Quality Improvement

Your **second goal** is **improving the given original training dataset**. Initially, the given data size is 31,848 (see the Table on page 1), but you are permitted to (1) remove some data images (if they are irrelevant; out-of-distribution images), (2) to add more images (if you want to add augmented images from original images in the given data), and (3) to fix the incorrect labels into correct ones (by replacing the {class-id} of the file name. To summarize:

- **1) Outlier Detection / Removal**: You can remove images in the dataset if you want. Just delete the image from the data folder. For example, you can use a pre-trained image model to get the embedding of the image, and then delete the images if their embedding is very distant from the average embedding of the pre-defined classes (i.e., outlier detection and removal).

- **2) Data Augmentation / Oversampling**: You can add additional images into the data folder by applying any data augmentation from the original images. Note that it is not allowed to add new images from other data sources (e.g., web search, crawling, or other datasets). You must keep the {random-image-key} of the original image to specify which images are utilized to generate a new image.

For example: using Mixup using img_0a2ca0be-f2db-4b59-9309-5f1cfff4a95b" and "img_13609fd2-173c-4b4c-827d-d0d691d58b1c. Then, the file name of the new image must be either img_0a2ca0be-f2db-4b59-9309-5f1cfff4a95b_{technique-name}.jpg or img_13609fd2-173c-4b4c-827d-d0d691d58b1c_{technique-name}.jpg where {technique-name} is mixup here since we use it.

For example: oversampling the image "img_0a2ca0be-f2db-4b59-9309-5f1cfff4a95b". Then, the file name of the new image must be img_0a2ca0be-f2db-4b59-9309-5f1cfff4a95b_{technique-name}.jpg where {technique-name} is oversampling. If there are images oversampled multiple times, then use the technique-name like "oversampling-1", "oversampling-2", etc.

For the augmentation, {technique-name} can be many other techniques. So, you can freely define the name of the augmentation technique and change the corresponding token. **Note that the new labels for the augmented img must be added to the data-labels.json file. Also, if there are {random-image-key}s that are not in the original dataset, all those images will be automatically removed.**

-

- **3) Data Cleaning / Re-labeling**: You can fix the label of original images if you can identify noisy labels from the original data. You can use any AI models or any techniques to do this. For example, you can use the relabeling method we studied using Colab to fix the original labels. The relabeling can be done simply by replacing the labels in the data-labels.json file with a new label you decide.

In addition to the three techniques, you can apply any other methods you think of, as long as entirely new images are not added to the training data and the filename format is maintained.

Team Matching

You can find your team members yourselves. Each team can have four to five students taking DS801 course. If those who cannot find the team within the team matching period, seats will be randomly assigned to an empty team.

Team Building Period: 2024. May 3rd
Excel Link:
https://docs.google.com/spreadsheets/d/12bHScpny4wT5pGv05Qmwg5Wz8xNI3JeWb53LCQgIsxk/edit?usp=sharing

What to Submit and Prepare

As the **deliverables**, you need to prepare the followings:

- 1) An **improved dataset** that has **at most 50,000 images** (+ updated data-labels.json)
- 2) A **new** created **validation dataset** that has at least 10 images (1 image per class) and at most 1,000 images (+ data-labels.json for validation set)
- 3) Presentation **slides for 20 mins** presentation for Part I and Part II. The final presentation will be done in the last week of our course (Not yet determined whether it is live or recorded video).

In your slides, you must cover the followings: (1) Data Creation, (2) Data Cleaning, (3) Analysis (e.g., Visualization), and etc.

Format of the training and validation datasets: Each set should be a folder with name "{team-name}-training-set" and "{team-name}-validation-set". For example, the hierarchy of your directory must be:

- {team-name}-training-set # folder
  - imgs # folder
    - 50K images

    - data-labels.json
- {team-name}-validation-set # folder
    - imgs # folder
        - Your validation data images
    - data-labels.json

Note that the **submission deadline for 1) and 2) is 2024 June 9 (upload your all deliverables on KLMS,** https://klms.kaist.ac.kr/mod/assign/view.php?id=960436 **),** the **3) presentation** will be proceeded **during 10-14 on June**
For 1) and 2), the final balance accuracy of your model submitted will be evaluated and announced in the last week. You can check the performance of your data using your own Validation Data as an alternative way.

Evaluation

The assessment of your score for this term project will be conducted on a team-based basis. There are multiple aspects that will affect your grading:

- The **novelty** and **completeness** of your methodology in constructing clean validation data and enhancing the quality of the training data will be assessed.
- The **balanced accuracy on clean test data** (which is not accessible from your side) will be considered.
- The **quality of your presentation**, including content, originality, and clarity, will be evaluated by Prof. Song and your peers.

Reference Codes

You can use the **Colab environments** or **Lab Server** of your Lab (or our **Department GPUs** if you major GDSD or IE: https://ie.kaist.ac.kr/0308)

- **Base code** to train ResNet-34 on the animal training data: https://drive.google.com/file/d/1ZNgsrmI8oA_05wkVDMkc1jPZN07Xeznb/view?usp=sharing
- You can use **any models** and **training codes** when refining your dirty training data, but note that this training code will be used to get the final balanced accuracy using your training and validation data submitted.

For your reference, the balanced accuracy of the model on clean test set using Colab was as follows (this is the minimum performance of the model without doing Part I and II):

Final Balanced Accuracy: 36.22%
class 1 : 0.177319587628866
class 2 : 0.19148936170212766
class 8 : 0.2863636363636364
class 9 : 0.30736842105263157
class 0 : 0.3087971274685817
class 5 : 0.3244274809160305
class 6 : 0.4064516129032258
class 3 : 0.5073170731707317
class 7 : 0.5547576301615799
class 4 : 0.5579567779960707