project-proposal

June 24, 2019

1 Capstone proposal - Deep Learning for LEPUS 💬

1.1 The problem to be solved

1.1.1 (a) What are the main project idea and goals?

By thinking about the capstone project, one emerged when discussing with prof. Yves Hausser from the nature management department at hepia (HES-SO Geneva) about a deep learning project of natural images. As follows, an introduction of this project is presented:

The Lepus software [1] was designed to help scientists analyze wildlife images acquired using photographic traps. At present, species recognition and individual identification are carried out manually, which is very time-consuming.

The objective of this project is to test Deep Learning technology to automate certain tasks such as

1. detecting the presence or not of an animal in the image
2. locating animals using bounding boxes
3. identifying certain species or more generally its type or family
4. ideally, identify each individual animal of a specific specie with respect to physical characteristics (e.g. to help the Wild Life Conservation Society)

Each of these problematics can be independent and with another Extension EPFL school learner (*Julien Smets/Blerim Arslani*), we decided to share this project. Here is the chosen configurations for our specific capstone projects:

Project 1 *Blerim Arslani : Detection of the presence of an animal in the image (binomial classification)*
Project 2 *Julien Smets : Identification of the type of an animal in the image (multnomial classification)*

Please let us shortly motivate our choice. By solving these two problematics, the saved time for nature management scientists could be very high (days of work), especially for the presence detection problem because only a small amount of images contains animals (this will be more deepely detailed below). Moreover, the labelled data do not include the bounding boxes which excludes the second project (here we consider only supervised learning to ensure a validation metric) and the fourth one is limited to the significant inspection variance of the manual identification and the very low amount of data available.

Note that these two chosen distinct projects are individual (will not be team-based but only the dataset will) and can be combined together by performing the second work right after the first one in order to classify the detected animals (cf. diagram as follow).

As follows, you can see the project cloud for more details:

- [1] http://ec2-35-157-78-9.eu-central-1.compute.amazonaws.com/ 💬
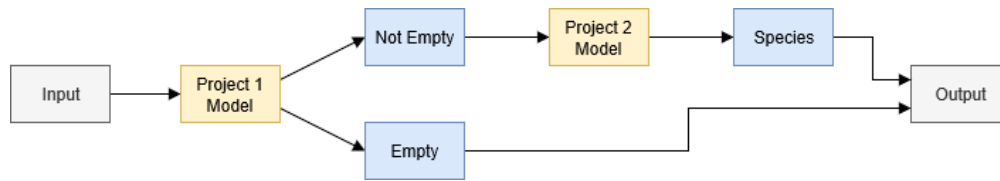
1.1.2 (b) What story you would like to tell with the data and what would you like to achieve at the end?

Project 1 This part of the project would be to detect with a relatively high accuracy the presence or absence of an animal.

Project 2 In this second project, the aim it to classify correctly a group of animal species with the assuption that there is always an animal in the image.

1.1.3 (c) What is the main motivation behind your project?

The main idea following Lepus project is to reduce the high time consuming manual detection, localization and identification (and ideally with better accuracy that humans but it will not be experimented here) of animal species in photographic traps.

1.2 The data set

1.2.1 (a) What is the size and format of the data that you plan to use?

Data details The data set consists of **750'000 pictures** containing about **57 different species** including about **80% of images without any animal** on it.

These images come from **hundred of cameras** placed in different nature spots in Tanzania and represent a covered area of **10'000 km^2**. This data has been obtained since 2008 and until 2017 (more recent data remains unlabelled).

The data set is given as image files separated in several folders. The labels and other images information are stored in a **CSV** file.

Data information The given species can be very small depending on the animal size and its distance to the camera or very large taking a large part of the image. Animals can be **occluded** by background objects (trees) of even be **partially viewed** (especially for large animals such as giraffes or elephants). The amount of species are in proportion irregular depending on the rarity of these species. Moreover, a majority of the given images doesn't contain any animals due to trees movement, dust tornado, butterflies, etc causing false positive captures. The proportion of empty pictures are **~60-85%** depending on the device environment.

In addition, cameras have many different fields of view (FOV) and two resolutions (medium: 800x600 and high: 1024x768) including RGB and graylevel images. The latter comes from the difference between day and night acquisition devices. Note that this results to **highly non uniform representation of species** w.r.t. different situations (day/night, background situation, etc.). As an example some species are only nocturne. This will need robust data preprocessing in order to avoid biases.

Some of the images are time correlated due to animals running and get captured several times, i.e. **small timelaps images**. These set of images are already grouped by the Lepus software in

specific folders ready to process. These grouped images are **called evenement independant capture (EIC)** and numbered from 3 up to ~300 if an animal stays in front of the camera the whole day/night. This can be an important information and will be discussed after.

Others Note that the data needs to be confidential with a DNA (standard Non-Disclosure Agreement) due to some very rare species which are often hunted for money.

In addition, for pre-training purposes, several dataset from Kaggle can be used. To this aim, the following data set could be used: - iNaturalist 2019 at FGVC6, Kaggle, Paper2018, Github, presentation2018 - iWildCam 2019 at FGVC6, Kaggle, Github, Top3 private score, Top7 private score (Top7 report) - Oregon wildlife, Kaggle - CaltechCameraTraps, website - Snapshot Serengeti, website

1.2.2 (b) How do you expect to get, manage and process the data?

Recieving the data The data is shared by physical transfer in order to conserve privacy.

Managing the data Although we do not already know how exactly the data will be organized, we will need to result in the following data structure.

The dataset will be split in a stratified train/validation/test sets in order to conserve the labels (species) proportions and enviroment proportions. Only the labeled data will be considered. It is possible to consider the train/validation split from 2008 to 2016 and use the most recent 2017 dataset as the testing set.

Data pre-processing This dataset has a large amount properties (temporal, day/night assumptions, etc.) but this variability is also its main drawback. A good understanding of the data management and processing is needed. However, the following steps are planned.

The data will be cleaned and processed a minimum by ensuring the images have normalised sizes and pixel intensities (e.g. Z-Scores) and are all labeled incluling ones without animals (label 0).

The temporal information of the timelaps captures can be used as movement detectors (e.g. difference between frames or background substraction).

1.3 The analysis and methods

1.3.1 (-) State of the art

Transfer Learning: The transfer learning consist of using a knowledge gained while solving one problem and applying it to a different but related problem. This method can highly reduce the training time and improve classification accuracy.

Ensemble methods: The ensembling strategy is a group of methods that aim to combine advantages of multiple different models (instances of the same model, different model architectures, etc.) in order to improve the final accuracy. It is known to result in very accurate solutions as shown by the number of Kaggle competitons top winners using this method. To obtain the top3 private score in iWildCam2019, [1] used a model ensemble of Inceptionv3, ResNet152, and InceptionResnetv2 and majority voting for classification. He first located moving objects using background substraction [2].

Images Preprocessing: In [2], the author used image processing techniques to locate moving objects using the difference of timelaps frames representing short time movements such as moving animals and vegetations. Several other steps allowed to extract animals bounding boxes.

References:

- [1] https://github.com/HayderYousif/iwildcam-2019-fgvc6, Github link.
- [2] Yousif, Hayder, et al. "Animal Scanner: Software for classifying humans, animals, and empty frames in camera trap images." Ecology and Evolution (2019).

1.3.2 (a) What are the main challenges that you envision for completing the project and how do you plan to get around each one?

*List of challenges on images: ***

- **Illumination:** Images can be poorly illuminated, especially at night.
- **Motion Blur:** The shutter speed of the camera is not fast enough to eliminate motion blur, so running animals can be blurry.
- **Small Region Of Interest (ROI):** Some animals are small or far from the camera, and can be difficult to spot even for humans.
- **Occlusion:** Animals can be occluded by vegetation or the edge of the frame.
- **Perspective:** Sometimes animals come very close to the camera, causing a forced perspective.
- **Weather Conditions:** Poor weather, such as rain, snow or dust can obstruct the camera and cause false triggers.
- **Camera Malfunctions:** Sometimes the camera can malfunction and cause discolorations.
- **Temporal Changes:** Given a location, backgorund can change over time as the season change.
- **Non-animal variability:** Depending on the location, false positive triggers can significantly vary.

  *based on iWildCam 2019 challenge Top7 report.

List of technical challenges:

- Adapted **preprocessing** steps without inducing any strong bias.
- Be inspired by accurate existing methods and define a good **baseline**.
- Tune, modify and improve the model as **our own solution**.
- **Analyse** and conlcude about the accuracy of the proposed model.

1.3.3 (b) What the are steps that you plan to take to achieve the end goals?

The steps of the work are given as follow:

1. **Preprocess** the dataset and split it in train, validation and test sets.

2. Select a model and use **transfer learning** for our application.

3. Compute its accuracy measure (F1-score) and define it as the **baseline model**.

4. **Optimize** it, validate it and test the performance of the proposed model.

5. Use **ensemble methods** for additionnal improvments.

1.3.4 (c) Show us that you have a pipeline in place and that you understand the feasibility of your project goals.

*1. Preprocessing: ***

- White balance: Adjust saturation and tone processing problems.

- Histogram equalization: Adjust the brightness of the image, but increase the contrast of noise and reduce the contrast of some useful signals.

- Image denoising: Reduce the noise but also smooth edges.

- Image resizing: Reduce images size.

- Data enhancement: Random crop, rotation, translation, brightness and noise effects applied to the original images.

- Data balancing: Weight the data enhancement w.r.t. labels to uniformize their proportions.

  *based on iWildCam 2019 challenge Top7 report.

2. Transfer Learning:

- Select a state of the art pretrained model (**EfficientNet**, ResNet, DenseNet, Xception, etc.).

- Adapt the structure to our problem (adjust last layers to fit our desired output).

- Train it on our dataset.

- Validate the pre-trained model.

3. Baseline model:

- Set this trained model as the baseline for our future experiments.

4. Optimize the model:

- Tune, modify (re-train first layers) and adapt the model hyperparameters, including some eventual preprocessing (filtering) or postprocessing (ensembling, bagging) steps.

- Validate the new model.

- Test our final model using the test set

5. Advanced improvements:

- Do transfer learning on others pretrained model.

- Train and validate them on our dataset.

- Optimize them separately to our problem.

- Use Ensemble methods to further improve our results.

1.4   The communication

The code sample will be implemented in several python scripts (.py) for convenience and simplicity (e.g. separation of pre-processing, training and testing). An additional notebook (.ipynb) will be included with analysis details and visualization figures as a small report (similarly to this document).

Please note that since both projects are similar in terms of challenges (e.g. preprocessing) and in terms of pipeline, they are distiguished by their objective (binary and multinominal classifications). This will guide us to very specific and thus different solutions as figured in the diagram as above.