Introduction

Catch & Release (C&R) is a collection of procedures that allow one to apply machine learning classification onto field videos. C&R's goal is to facilitate the creation of under-represented knowledge in machine learning in general, and experimental datasets for neural network image classification in particular. C&R allows anyone with a mobile phone and a laptop to create viable datasets for image classification (and to train state of the art convolutional neural networks with these datasets).

Furthermore, C&R can extract text from video. It can extract labels from video and use them as labels to generate image categories associated with the utterance.

This software and the bali-26 dataset are the basis for the 'Return to Bali' project that explores machine learning to support the representation of ethnobotanical knowledge and practices in Central Bali.
http://www.realtechsupport.org/new_works/return2bali.html

C&R can help create machine learning datasets or can serve as a companion while studying convolutional neural network systems. If you would like to contribute toward building a rich dataset for machine learning training, contact the repository owner (marcbohlen@protonmail.com).

License

Cite this software project as follows: 'Catch&Release version1'

Platform Information

C&R runs on Linux and macOS under Python3 and Flask with Chromium / Chrome or Firefox.

C&R uses the PyTorch framework to train and test image classifiers and connects to the Google Speech API (free) for speech processing. Library versions and dependencies are given in the requirements file.
C&R has been tested on a desktop (i7-4770 CPU with 16GB of memory) and a laptop (i7-3667 CPU with 8GB of memory) under Ubuntu (18.04 TLS under kernels 5.2.8 and 5.3.0) and under macOS (Catalina) with images sourced from .mp4 and .webm video (HD [1920 x 1080] at 30f/s; .mp4 H.264 encoded) from multiple (android OS) mobile phones and GoPro Hero 6 action cameras.

Recommended browser: Chromium on Ubuntu, Chrome on Mac.

Install Chromium on Ubuntu:

```
sudo apt install -y chromium-browser
```

Install the free Classic Cache Killer:

https://chrome.google.com/webstore/detail/classic-cache-killer/kkmknnnjliniefekpicbaaobdnjjikfp?hl=en

Software Installation

Clone the C&R repository on GitHub
Open a terminal window and type:

git clone https://github.com/realtechsupport/c-plus-r.git

Cd to the c-plus-r directory and run the following commands to update your basic python environment:

chmod +x basics.sh
sudo sh basics.sh
(This script just updates your Ubuntu installation and requires sudo to do so.)

Create a virtual environment:

python3 -m venv env

Activate the environment:

source ./env/bin/activate

Cd to to the c-plus-r directory again. Install Requirements and Dependencies.
(This may take about 30 minutes.)

pip3 install -r requirements.txt

Generate an STT key (optional)

While there are multiple providers of Speech to Text services, the most effective offering with the widest range of languages is at this moment provided by Google. If you want to make use of the text from video extraction you should obtain an access key to the Google Speech API. Creation of this key is free of charge and you can use it in this software at no cost as C&R operates within free limits of the API. However, you do require a google account in order to create the key. If that is not palatable, skip the sections that make use of the Speech API.

Instructions to generate a key (https://cloud.google.com/text-to-speech/docs/quickstart-protocol):

1. In the Cloud Console, go to the **Create service account key** page.
2. From the **Service account** list, select **New service account**.
3. In the **Service account name** field, enter a name.
4. Don't select a value from the **Role** list. No role is required to access this service.
5. Click **Create**.
6. Click **Create without role**. A JSON file that contains your key downloads to your computer.
7. Save the JSON file to the C+R project.

Other Speech to Text engines are available, but are not yet integrated into C+R. The key barrier at this point is limited availability of high quality language corpora in less popular languages. This is a major concern beyond C&R.

Launch C&R

Activate the virtual environment:
source ./env/bin/activate

Start C&R (in the c-plus-r directory):

on ubuntu: *python3 main.py ubuntu chromium no-debug*
on mac: *python3 main.py mac chrome no-debug*

Specify all three items: OS, browser and debug mode. Supported OS: Ubuntu and Mac OS. Supported browsers: Chromium and Firefox (less stable). To run in debug mode replace ’no-debug’ with ‘debug’.

crtl + / ctrl – increase / decrease zoom factor.

Stop C&R

Stop the app from the terminal:	ctrl-c
Exit environment at the terminal:	ctrl-d
If you see browser errors .. clear the browsing history:	ctrl-H clear browsing data clear data

Description of modules in C&R

Overview

The modules in C&R - with the exception of the last test case – are applicable to all field videos (.mp4 H264 as mentioned above).

When you run C&R for the first time, video samples, image samples and the trained models will be downloaded from pCloud automatically. You need these data files to run the examples.

A > prepare videos
> capture text
> video annotation
> label images from video > check results
> bulk > remove outliers
> by audio label > *add labeled images to collection*

B > test classifiers trained on bali-26

> display images and choose input
> *classify input; show best guess with confidence level and top 3 choices*

The ‘Context’ button gives you pertinent information on what the individual functions perform.

The next sections describe the individual modules on C&R.

Prepare Field Videos

This module allows one to chunk a long field video into smaller parts for processing. Supported formats are .mp4 and .webm. Select the video and choose the chunk size. It is suggested that segments do not exceed 3 minutes.

Chunked segments are saved to the C+R /tmp directory, and are deleted if you go back to start or exit the program. You can also download some sample videos to process and experiment with.

When you have some data ready, continue either onto the text extraction or the video voice-over modules.

Capture Text from Field Video

Use this module to extract text from a field video.
First load a field video to locate a section you want to extract text from, then reload to capture the text.
This module requires an access code for the Google Speech API (key for capture text).

If you add a search term, text from video sections that contain that term will be listed separately.
If the video is shorter than 1 minute, reduce chunk length.

Reducing the confidence level to below 0.9 increases detection chances and false positives.
This module may take several minutes to complete. It is best to limit the difference between start and end times to a few minutes.

Add Voice-over to Field Video

Caveat – Audio recording can be tricky. This module may need to be adjusted for your particular hardware.

Use this module to replace the field audio with a new audio file. This can be helpful in the event an expert wishes to annotate a field video.

'Remove-old-audio' will delete old recording assets from previous recording sessions.

Select your field video and choose the microphone input.
The default mic is set as "cardn = 0, devicen = 0". This is typically the default setting on a Linux computer.
If you want to change this, modify the defaults for cardn and devicen in 'av_helper.py' (lines 65 ff) .

'Check-Audio' will record three seconds of audio on the selected microphone and play it back right after the three second recording event. If you do not hear your recording, something is amiss with either the microphone input, your microphone choice or the speakers. Do not continue until you hear a good test recording with your selected input device.

You can add you favorite external microphone to the list of available microphones in C&R. After the USB microphone is connected to your computer, enter at the terminal the following command:

arecord –list-devices

You will see something like this (instead of 'Kmic' you will see your microphone name):

card 3: Audio **[Kmic USB],** device 0: USB Audio [USB Audio]

Add this info (i**nside of the [ ]**) into the AnotateInputs class of the 'inputs.py' file (line 54ff) that contains the recording parameters.

Before:
mic = SelectField(label='select microphone', choices = [('0,0', 'default'),('AT2020', 'AT2020'),('UC02', 'UC02')], default=('default'), validators=[validators.InputRequired()])

After:
mic = SelectField(label='select microphone', choices = [('0,0', 'default'),('AT2020', 'AT2020'),('UC02', 'UC02'), **('Kmic', 'Kmic')**], default=('default'), validators=[validators.InputRequired()])

Save the file (ctr s) and restart C&R (or just reload the annotate audio page if you are in debug mode). You should be able to select the USB microphone you just added.

Then proceed as follows. Load the video (from samples, tmp or annotate folders) to identify the spot on which you want to add voice-over. Set start and end times to define the interval in which you will add a voice-over. Make sure the end time is greater than the start time and less than the length of the video.

Then segment the video. This can take a few minutes as the .mp4 file will be re-encoded. The terminal window shows the progress.

When the segmentation is complete, the silent segmented video will auto-play. Use the video controls to stop the video and rewind to the start (move the dot in the display timeline to the left).

Turn off your speakers.

Then click voice-over and comment on the video with appropriate key terms – these are the terms that the subsequent module will search for in order to associate text with image.

When you finished the voice-over (it can only be as long as the segmented video), go back to the start page. You can process this voice-over video (stored in the annotate folder) to labeled images with the next module.

Troubleshooting Video Voice-over

Speech to text:
Make sure your selected video has a good audio track. Load the video to check. If you can not clearly hear the audio or voice over, the speech to text system will not be effective.
The current speech to text system has substantial limitations. Some words will not be detected. (Results are printed to the terminal window, and if you see ‘ wordcollection: []’, then the key term was not detected (and the subsequent quality control page will show no images). Moreover, the system can be fooled if you use compound words (such as ‘banana tree’)into creating two categories instead of one, for example.
Use simple terms. If you want to detect more than one category via this module, run it multiple times, each time with a different key term.
You can rename incorrectly or inadequately labeled categories/folders manually in the images folder after the labeling, if required. If the speech system does not pick up the voice in the field video, you can go to the ‘add voice-over’ module and post-annotate the video. Alternatively, you can set the category manually with any ASCII name, select ‘bulk label’ a selected video file and then manually remove inappropriate images from the resultant folder.

USB microphones:
Most any microphone will work, but a good microphone will produce superior results. Recommended: AT2020 (Audio-Technica AT2020 Cardioid Condenser).

General troubleshooting:
Exit C&R and check audio settings. You can type ‘alsamixer’ at a terminal prompt to check if your microphone inputs are muted.

Label images from Field Video

Use this module to create labels from field videos or from your voice-over additions. Creating labels directly from voice input is based on an invention (030-7278 Expertise collection with action cameras) by the author. It makes use of the synchronicity between image and audio streams in video formats and uses the synchronicity to associate an image with a label.

Load the video to check, just in case.
There are two options:

A - Label images with key terms from audio track (label by audio).
In this case you will select a single key term and the number of images per utterance. Then set a confidence level for the speech to text API and the language spoken in the video. Select your key file to access the Speech API. If you want to search for multiple key terms, repeat the process above with a new term.

B - Label all images in the video with a given term (bulk label).
This option is the easiest to use and does not require speech recognition. In this case, all images generated from a chosen video will be bulk labeled with a given category / folder name. Set the frame rate (number of images to be extracted per second).

When the process has completed, click 'check the results' to open the subsequent module for quality control and collection creation.

Quality Control, Archiving, Sharing

This module allows you to control the quality of the images created from the field videos (both images created by bulk labeling and by audio label). The purpose of this module is to combine automated and human quality control, to remove out of context and low quality images and retain only high quality images for subsequent classification. High quality images will enable better classifier training and performance. The degree to which aesthetics matter for the classifier is not entirely clear. Sharpness is important, but poorly chosen backgrounds and offensive content might not matter for the classifier. The human image organizer plays a significant role in the compilation of these image sets. This is a new field of design.

The following options are available in this module:
Remove-selected:
Manually select images for removal. Select (multiple) images with a left click, confirm and then click ‘remove-selected’

Remove-divergent:
Select a single image as reference with left click, confirm, and then click ‘remove-selected’.
Hit <enter> to update the page after the removal process has completed.

Images that deviate from this selected reference in luminosity beyond the set min /max levels (under and overexposure) will be deleted. Other images that deviate structurally (different visual contexts or blurry images) beyond the set similarity measure will be deleted.

Once the image set has been reviewed you can add the resultant set to the collection. Once you have several collections / categories, you can archive the collection (compress the data sets) or, if something is amiss, delete everything and start again.

The final archived (compressed) collection will be the input to the classification procedures as described in the next module.

The image categories you build in this manner can become part of a larger collection. More on this opportunity forthcoming.

Aside -
Each image category should have at least 1500 viable images in order to offer enough information to neural net classifiers. Visually complex categories require substantially more than that. Collections very many categories have higher collection size requirements. More information on this issue forthcoming.

Test Case - Evaluate Sample Images From The Bali-26 Dataset On Multiple Neural Networks

This module loads one of several deep neural net classifiers trained on the bali-26 dataset and allows you to select an image from the validation set to test the selected classifier performance.

Left click to select a single image from the image gallery, then click 'classify image' to obtain the results. The first time you run C&R, the selected trained model will be downloaded from pCloud automatically.
The gallery will only load once. To inspect additional images, left click to select a new image, then click 'classify image' again, and repeat.

Three state of the art architectures are included: Resnext50, Resnet152 and AlexNet, all trained on bali-26 (where each of the 26 categories is limited to 2500 images) over 50 epochs, with data distributed evenly (50% each) between training and evaluation. Normalization factors were calculated directly from the bali-26 image set. The models are offered *with and without* pre-training. Caveat – Pytorch uses ImageNet to pre-train neural nets! The improved performance of the pre-trained models shows that exposure to ImageNet can improve classification accuracy and is all the more reason to work towards an unbiased alternative with similar network performance improvement.
(https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html)

The best prediction and the confidence levels are listed as are the top three results (two alternates to the first estimate). These results are calculated in real time on your local computer.

Train a Deep Neural Net

You can in fact train a deep network on a large data set with the code inside C&R if you move the internal routines to a remote GPU computer. The GPU version requires the installation of Nvidia drivers. See the v*env+pytorch+nvidia.txt* file in the 'remote' folder for step-by-step instructions.

Copy the following 4 files from the 'remote' directory of the repository to a directory on the server:

pyt_trainandsave.py *pyt_loadandeval.py* *pyt_utilities.py* *create_norms.py*

Place your data collection produced by C&R on the server as well and adjust the paths in the 4 python files. The *pyt_trainandsave* program will train your chosen network on the data collection; *pyt_loadandeval* will evaluate its performance. If you want to adjust the image normalization to your own image set, run 'create_norms.py' and replace the results with the bali-norms used here.

These routines were used to train the AlexNet and the deep Resnet152 and Resnext50 classifiers on the 50'000 images of the bali-26 collection.