

# Final Report

*7 May 2020*

Caribou Cams

Sponsor: Kathleen Orndahl

Mentor: Mahsa Keshavarz

Members: Samantha Muellner, Shuyue Qiao, Keenan Swanson, and  
Dongyang Yu



# TABLE OF CONTENTS

<b>1 INTRODUCTION</b>	<b>2</b>
<b>2 PROCESS OVERVIEW</b>	<b>5</b>
<b>3 REQUIREMENTS</b>	<b>6</b>
<b>4 ARCHITECTURE AND IMPLEMENTATION</b>	<b>8</b>
4.1 CNN	9
4.1.1 Splitting Videos Into Frames	
4.1.2 Training the CNN	10
4.1.3 Predicting Unclassified Videos	
4.2 Website	
4.2.1 Website navigational flow	
<b>5 TESTING</b>	<b>12</b>
5.1 CNN	
5.2 Website	
<b>6 PROJECT TIMELINE</b>	<b>13</b>
<b>7 FUTURE WORK</b>	<b>14</b>
7.1 Improve accuracy	
7.2 CNN and web page maintenance	
7.3 Add more categories	
<b>8 CONCLUSION</b>	<b>15</b>
<b>9 GLOSSARY</b>	<b>16</b>
<b>APPENDIX A: DEVELOPMENT ENVIRONMENT AND TOOLCHAIN</b>	<b>17</b>
A.1 HARDWARE	
A.2 TOOLCHAIN	
A.3 SETUP	
A.4 PRODUCTION CYCLE	19

# 1 INTRODUCTION

Caribou are a major species that is important for environmental, cultural, and humans is approaching numbers close to endangerment. In order to help this species thrive, we need to understand their daily and herd habits in order better help them by providing more to or interfering less with their lives. Unfortunately, despite how important they are to the environment, cultures, and us humans, we still know very little about them. Caribou are the largest, northern-ranging terrestrial mammal that provide indigenous people and predatory animals with food while also contributing to soil nutrients by defecating and the decomposition of their bodies once they die. A loss of them would have a devastating effect on our environment.

One of these species is caribou. While not in danger of extinction yet, caribou have declined in numbers by more than 56% in the past two decades. Despite the efforts of scientists and ecologists to reestablish the numbers of these herds, there's a lot that we still don't know about caribou and thus not a lot we can do to help stop their numbers from plummeting.

Our sponsor, Kathleen Orndahl, has currently teamed up with a group of Ph.D. students from the University of Montana (UoM) in order to try and help solve this problem by gathering large amounts of data directly from these caribou herds by using collar cameras to record their day-to-day lives. A Ph.D student at Northern Arizona University, Katie works in the Goetz lab as an Ecoinformatics student who specializes in wildlife and plant ecology. It is because of these fields of research that she has teamed up with the UoM group to study the daily habitats, eating patterns, and calf numbers of caribou.

Our client and her team are studying caribous' everyday lives by the use of collar cameras that have been attached by scientists to caribou in specific herds. The cameras that are attached to these collars are set with a timer that makes it so they take a 9 second video every 20 minutes for a year. After this time, the cameras fall off and are able to be located and obtained by scientists due to a GPS chip installed in the camera. The videos on these cameras capture the everyday life of caribou, recording anything from a calf being birthed to what the caribou are eating. The collars are analyzed by volunteers, who our client and her team have to manually send a folder of videos to in order for the volunteers to manually go through and extract the necessary data. This entire process is extremely time consuming for both our client and the volunteers. The data in these collars, after being analyzed, proves helpful to our client and her team

because it can provide them with insight on what the caribous daily habits are, such as what they are eating, how many calves there are, and where they like to roam, without inhibiting the lives of the caribou.

These videos are a helpful resource for gathering all the unknown data about caribou. However, our client now faces the problem of having too much data for it to be analyzed in a reasonable amount of time. With over 3 terabytes worth of videos that need to be analyzed, and only human volunteers currently available to analyze it, Katie and her team estimated that the amount of time it would take would be a minimum of 1 year and 1 month before getting all results back. Ideally, she would like to have this information sooner, but there are many problems in our client's method of analyzing the data that is keeping this timeline so long. The major problems are as follows:

- *The sheer volume of videos will take a lot of time to analyze*
- *Volunteers have to repeatedly rewatch videos in order to find input for all categories*
- *Volunteers have to spend time inputting data that could have already been previously determined*
- *The transfer of information between the volunteers and the organizers is disjointed and inefficient*

With the problem being that there is too much data to analyze by human-hands within a short amount of time, our team has devised a two stage solution that is designed to shorten the amount of time it will take for our client to get the information of the videos. The first stage of our solution consists of a CNN (Convolutional Neural Network) that will analyze the videos and sort them by video quality, the category of which consist of 'Good', 'Fair', 'Poor', and 'Extremely Obstructed.' Once we have completed the first stage, we will move onto the second phase which consists both expanding the CNN to cover more categories such as 'foraging' and 'movement' as well as an easy-to-use website that will provide the volunteers analyzing the videos with an easier way to input data from the videos as well as prefill in data that is already known about the videos.

Each stage of the above stated solution will solve the previously stated problems with our clients workflow as follows:

- *The sheer volume of videos will take a lot of time to analyze will be fixed by implementing an AI that will lower the amount of videos and information that needs to be analyzed*

- *Volunteers having to repeatedly rewatch videos in order to find input for all categories will be partially fixed by creating an AI that will already pre-categorize videos, which will reduce the amount that needs to be analyzed in a video*
- *Volunteers having to spend time inputting data which could have already been previously determined will be partially fixed by creating a website that already prefills all the data found for that video*
- *The disjointed and inefficient transfer of information between the volunteers and the organizers will be fixed by creating a website that will automatically upload the videos instead of the organizers having to look through and send the data*

In this paper, our team will go into further detail about the process we used to create this solution, the requirements needed for this solution, the architecture of the solution, and our testing of the solution. We will also touch briefly on the timeline of the solution. Included at the end of this paper is also the *Appendix A* section, which discusses the development environment and toolchains used for our solution.

## 2 PROCESS OVERVIEW

In the previous section we introduced our project, our client's problem, and our solution to that problem. In this section we will discuss the process we took in order to develop and achieve that solution.

Our team developed the above mentioned solution by using a number of means to communicate efficiently, organize our work, and stay on top of our tasks. Weekly task reports to our mentors helped us organize our tasks and remind us what we had coming. The use of weekly client meetings with our team, client, and mentor allowed us to further delve into the specifics of our project and papers so that we could fully understand what we were developing. These meetings also gave us time to ask questions and discuss any setbacks caused to our team.

Along with the weekly team meetings, our team used discord and email to keep in contact with each other. These allowed us not only ask clarifying questions about documents or sections of our solution, but also allowed us to keep things organized. Specific email chains and discord channels were designated to the different sections of our project so that questions and comments regarding those sections could be easily found.

Many of these sections were based on the different documents we had to write. These documents helped our team plan out our approach and solution in a variety of ways that allowed the entire team and our client to come to an agreement on how we wanted to approach our client's problem. The documents discussed everything from the basic approach of our solution to the specific architecture of our solution. The way we coded our solution was based on the information in these documents, all of which were kept in a single google drive for easy access by our team.

We used version control, specifically the use of git, to keep our code manageable and clean. Repositories were created on GitHub for both the website and the CNN with different branches underneath each that allowed each member of the team to be able to work on their own section of code without fear of messing up someone else's section of code.

Through the use of these methods, our team was able to create a viable solution with relative ease. In the following sections, you will read more about the specifics of the solution this process was able to help us create.

# 3 REQUIREMENTS

In order to accomplish the solutions discussed in the previous sections, our team analyzed our solution and created a list of requirements that would allow it to fully be able to fix our client's previously stated problems. The following is a list of our domain requirements for this project, requirements that are the highest level features that the user needs from a domain perspective:

- *D.R.1* Use of volunteer analysis to train CNN
- *D.R.2* Secure video data accessible to volunteers

Based on the above domain requirements, our team has also come up with the following functional requirements for our project. These requirements detail exactly what the end product should be able to accomplish:

- For *D.R.1 The analysis result data for CNN training*
  - *F.R.1* Train CNN using classified videos
  - *F.R.2* CNN analyzes quality of video
- For *D.R.2 Secure video data accessible to volunteers*
  - *F.R.1* Train volunteers before their signup
  - *F.R.2* Maintain user account
  - *F.R.3* Different video materials are provided based on users wants
  - *F.R.4* Volunteer classifies videos

Not only should our product meet the above functional requirements, but it should also perform to the standards of our client, Katie Orndahl. In order to do this, we have developed the below performance requirements:

- For *D.R.1 The analysis result data for CNN training*
  - *P.R.1 for F.R.2* Accurate CNN analysis - over 80%
- For *D.R.2 Secure video data accessible to volunteers*
  - *P.R.1 for F.R.2* Secure Website
  - *P.R.3 for F.R.3* Reliability
  - *P.R.4 for F.R.3* Convenience
  - *P.R.2 for F.R.4* Response Time

Lastly, we have the following environmental constraints for our project that must be met:

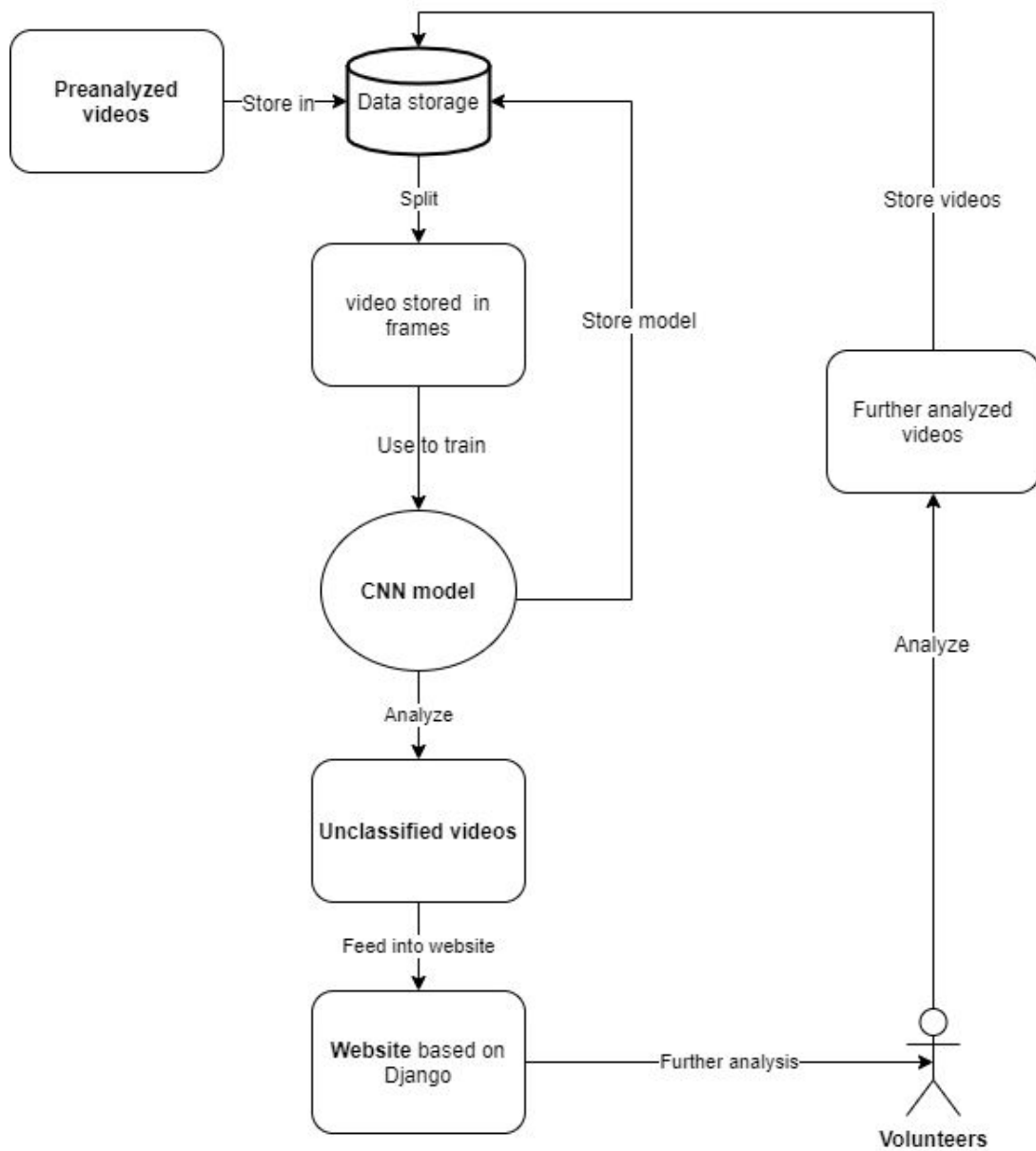
- Technologies used must be free
- Fully supported by all browsers
- Confidentiality of user information

For a more detailed version of our requirements and specifications for this project, please reference our Requirements Specification Documentation document. In this document, our team will be discussing the steps we will take to implement these above requirements into our product.



## 4 ARCHITECTURE AND IMPLEMENTATION

The general architecture of the project includes a Convolution Neural Network (CNN) and a simple website design. In general, the CNN would use pre classified videos, that are stored into their corresponding folders and divided into frames, for training. The model will then make predictions on unclassified videos that will be exported and stored onto the website. Here the volunteers can make further, and more detailed, classifications. These results will ultimately be stored back into the database and can be used to retrain the CNN for better accuracy and more features when implemented. This can be seen by the diagram below.



## 4.1 CNN

### 4.1.1 Splitting Videos Into Frames

The first key component of the CNN is to split videos into images. Videos are split then section off to their corresponding folders in either the training set or the validation set. This is what allows the CNN to train and make predictions.

### 4.1.2 Training the CNN

The training set is used to train the CNN and the validation set is used to predict the accuracy of the CNN. Once that has been finished, the CNN creates a model and its weights.

### 4.1.3 Predicting Unclassified Videos

The model and its weights are passed into the prediction model. This model then classifies a set of unknown videos in a folder by breaking them down into images. After the classification, it assigns the label to the video and moves them to the output corresponding classification folder.

## 4.2 Website

We have one major component that implements the classification function (Figure 4.2-1). After login, users will choose categories according to their preference. Then, a random video with the preference will pop up. Users can watch the video and complete the analysis. When they click the submit button, the data will be saved to the database.

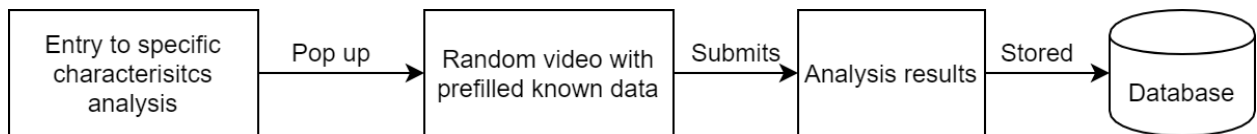


Figure 4.2-1 Classification component

### 4.2.1 Website navigational flow

The navigational flow shows how our website works. For new volunteers, they need to go to the tutorial page to read all instructions about analysis and take a test to evaluate their analytical ability. After that, they can register a new account. When they log in, different categories will be displayed, and users can choose categories of videos they prefer. Then, a random video with their preference will pop up, and they can complete the analysis form.

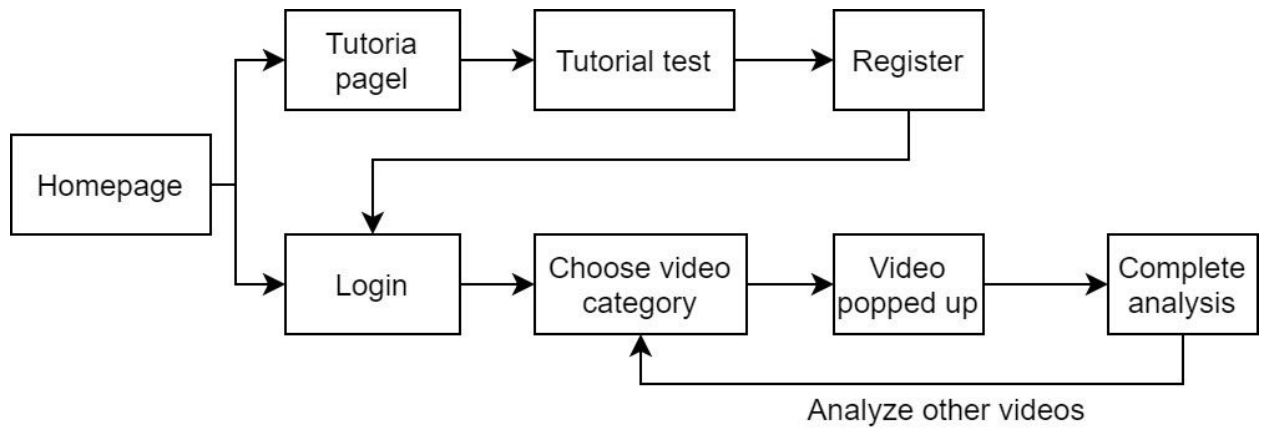


Figure 4.2-2 Website navigational flow

# 5 TESTING

## 5.1 CNN

In the CNN, there are a few important functions that are critical to running the CNN. To test these functions we used Unit Testing. We tested the VideoToFrames.py file as it is the most essential when it comes to making a model possible. Therefore, we tested the following methods:

- getVidDuration: ensures the file is a video

- getFrameRate: produces the amount of frames wanted for training the model

- getFrame: Splits the video into images for training and predicting.

Running these unit tests we found one error of producing one too many frames. We caught the error and were able to fix it quickly.

Also during Unit Testing, we wanted to make sure the model was working as expected. To check we looked at the following functions. Predict and filesForPrediction.

Lastly, is the Usability Testing for our application. This was done by explaining how to use the application to a group of people that generally don't know how to execute a python file and with our clients. After the explanation we let them go through the process of using the application themselves. We received feedback from them about it and made some changes.

## 5.2 Website

For the website, another important part is the database. Because the database records the data of the website, they do testing together.

Since our website and database are all under the structure of django, the code of each part will be monitored by Django, so the code of each part will be no problem. So Unit Testing is not required.

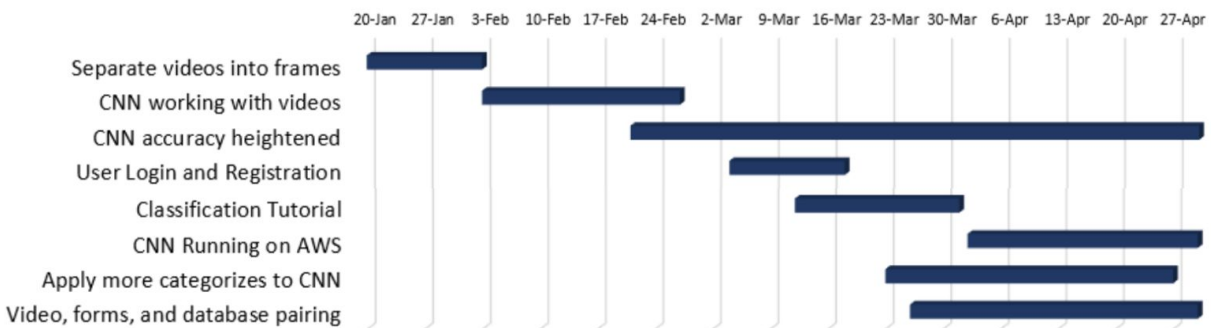
For Integration Testing, the website needs to be connected to the database. So it is necessary to test whether the website is connected to the database very well. What needs to be tested is whether the user registered on the webpage and the video analysis filled in the webpage are displayed in the database. We conducted hundreds of tests, and the test results were free of any errors.

For Usability Testing, because the users who can log in to the website and the database are not the same, the website and the database are tested separately. For the website, we found some volunteers who ranging from people with no computer science skills to people with some

computer science skills to run the website, and then got feedback. Through feedback, we changed the style of the fonts on the website to keep all text in the same style and capitalization. We also changed the login and registration pages to make them look clearer. For the database, we let customers run the database and get feedback. The database has not been changed.

## 6 PROJECT TIMELINE

Our group completed all planned work well. Our group joined the website when 60% of CNN was completed. Then CNN continued to improve accuracy. The final result of our group is a website and a CNN.



For CNN, the two most important stages are to make CNN work with videos and improve the accuracy of CNN. Because the first stage is to complete CNN, the second stage requires a lot of data and improved algorithms to achieve, which is also our team in the completion process. Two parts that encountered difficulties. In the first phase, our team was led by our team leader Samantha, and each of the four people was assigned different tasks. In the second stage, only Samantha and Keenan participated in this part of accuracy improvement because of the webpages.

For web pages, the more important stage is login, registration interface and database connection. First of all, in order to ensure that the videos on our web pages are not downloaded

by strangers, we need to restrict who can log in to the web page, which is why the registration and login interface are important. Then, the connection of the database ensures that the entire webpage can run normally. Only Shuyue and Dongyang participated in this part.

## 7 FUTURE WORK

### 7.1 Improve accuracy

Now the accuracy of our CNN is 75% -78%. This is because we do not have enough data. After the new data is obtained, we can add new videos to the training. In this way, the accuracy of CNN will be further improved.

### 7.2 CNN and web page maintenance

Our group has made this product, it is necessary to carry out later maintenance. We will give all codes and database permissions to customers. In this way, they can monitor the situation of each user. If CNN and the website are damaged, customers can also download the code again and reset it.

### 7.3 Add more categories

Now, our group has completed the analysis of the video quality using CNN according to the customer's requirements. Because the video not only needs to analyze this item, there are many other categories. Therefore, we can change part of our CNN code and let it analyze other categories and get the results.

For the website, if CNN analyzes other categories and draws results, then the website will also be improved and this category will be added to the category page.

## 8 CONCLUSION

With caribou being such an important part of our ecosystem and their numbers dwindling, information about them has risen in importance. Our client, Katie Orndahl, has teamed up with a group of Ph.D students from the University of Montana in order to obtain more information about caribou with the help of collar cameras. These collar cameras show videos of the caribous everyday life, allowing people like Katie and her team to gather more information about them, however the information still needs to be obtained from these videos.

Currently, our client's only way of obtaining this information is manually through volunteers analyzing the videos. The problem with this current way of analyzing videos is that it is time consuming, both for our client to manually send certain videos to these volunteers and for the volunteers to analyze all of their designated videos, and the analysis of videos by volunteers prone to human error.

Our group has come up with a solution that can help our client obtain this information faster for current and future videos. We have designed a CNN, a Convolutional Neural Network, that will be able to analyze videos and sort them based on video quality. Since a great deal of the information our client is interested in is determined from visual evidence, this will be a great help as it will allow our client to know what might be visible in each video and thus the videos can be assigned to the proper specialist. Our team has also designed an easy-to-use website that will make the analyzing process easier for volunteers by providing the videos and forms on a single page, with previously analyzed data from the videos prefilled into the forms.

In this document, our team discusses the overall architecture of these two solutions. We go into detail about how our CNN will work structurally, such as how it will train itself



using frames from videos and how it will save its data in a model, and how it will work with our website. We also talk about how our website will function in the background as well as the foreground. Finally, we discussed the testing that was conducted on our solution to make sure that it fit our client's needs. By following the plan in this document, our team was able to create a fully functioning product for our sponsor.

## 9 GLOSSARY

**CNN:** Convolutional Neural Network, a class of deep neural networks, most commonly applied to analyzing visual imagery.

**Django:** a Python-based free and open-source web framework.

# APPENDIX A: DEVELOPMENT ENVIRONMENT AND TOOLCHAIN

## A.1 HARDWARE

CNN and website were developed on Windows 10. The main development machine is a laptop with 2.5 GHz Intel Core i5 CPU and 8 GB RAM. The development process can be done with any platform with a Python environment, and there are no minimum hardware requirements for this project.

## A.2 TOOLCHAIN

### A.2.1 Python Environment

- Python 3.8 (Python is an interpreted, high-level, general-purpose programming language.)
- Pip (Package manager for Python. Use it to install other python components or libraries like django)
- Django 3.0 (Django is a high-level Python Web framework. This is the basic of the web application.)

### A.2.2 Code Editor

- Sublime Text 3 (Sublime Text is a proprietary cross-platform source code editor with a Python API. It's easy to develop code with Sublime Text 3)
- Pycharm (PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language.)

## A.3 SETUP

### A.3.1 CNN Installation

A.3.1.1 To install the CNN the first thing to do is install git.

A: For linux Fedora systems this can be done by the following command into the terminal:

```
sudo dnf install git-all
```

B: For Debian Linux systems use the following command into the terminal:

```
sudo apt install git-all
```

C: For macOS type the following command into the terminal:

```
git --version
```

D: For windows systems you should download from the following link:

<https://git-scm.com/download/win>

A.3.1.2 The next step is cloning the repository.

A. Go to our repository

<https://github.com/Sam-the-Unwise/Caribou-Classification-AI>

B. Click the green “clone or download” button and copy the link

C. On your machine, open your file explorer and choose a location you want the application to be

D. In the desired location of your file explorer now right click on an empty area within the file explorer and open terminal for Linux/Mac systems or click git bash for Windows.

E. Next type the following command:

`git clone`

`https://github.com/Sam-the-Unwise/Caribou-Classification-AI.git`

F. All the files should be located in your desired location, open it and until you see the following contents

`data`

`src`

`CaribouModel.exe`

Note that this system was developed for a windows operated machine and we are unable to predict how it will operate on a Linux or Mac

## A.3.2 Website Installation

A.3.2.1 Follow step 2.1.1 to install git

A.3.2.2 Clone the repository on github

A. Go to our repository

<https://github.com/Dongyang-Yu/Caribou-Classification-Website>

B. Click the green “clone or download” button and copy the link

C. On your machine, open your file explorer and choose a location you want the application to be

D. In the desired location of your file explorer now right click on an empty area within the file explorer and open terminal for Linux/Mac systems or click git bash for Windows.

E. Next type the following command:

`git clone https://github.com/Dongyang-Yu/Caribou-Classification-Website.git`

F. All the files should be located in your desired location

A.3.2.3 Set up the environment on local machine

A. You need to install the following required components and libraries in your local machine first:

- Python
- Pip
- Django (Python web framework)

B. Install Python

1. Since we use the Django version 3.0 framework, the python version you install should be at least 3.6 and we recommend using the latest version of Python 3. The download link is: <https://www.python.org/downloads/>
2. When you have installed it, type `python --version` in terminal to check if it's installed successfully.

C. Install Django

Since pip is already installed if you are using Python 3 >= 3.4, we now install django with pip. Type the following command into the terminal:

```
py -m pip install Django
```

D. Run the website

Go to the folder that you unzipped, which is

Caribou-Classification-Website. Type following command in the terminal

```
python manage.py runserver
```

E. Then, type the following link in your browser:

<http://localhost:8000/>

## A.4 PRODUCTION CYCLE

### A.4.1 CNN

The CNN production cycle is a relatively simple one. To make edits to the CNN, changes have to be made to the src folder, and the “.py” files within it. Once the changes were made, save the file and run main.py in the terminal. To do this, we opened the terminal, navigated to the directory, and type “python main.py”. This will show you the errors if there are any at compile time. The next thing is to follow the process of using the application after running main successfully, to make sure there are no errors in the process. Finally, creating the application. To create the application, open the command line and navigate to the src directory of the project. Now that we are here we run the command following commands:

```
pip install pyinstaller
```

```
Pyinstaller -w -F -n CaribouModel main.py
```

\*\* -w is windowless, so a terminal doesn't appear when executing the application

-F is one file, so all the dependencies are packed into one file to make it nice and clean

-n is naming the file of the application

\*\*

This will take a minute or more to run. Once that is finished, within the dist folder in src, the application created should be located in there. Drag and drop that file in the videoCNN folder,

so it can have access to all the files it has dependencies on. Once all these steps are finished, a new version of the application has been created with the edits made.

### A.4.2 Website (based on Django)

The website application was designed for phase one, which is a general analysis for caribous. In the future, this application needs to be updated. In this section, we will explain the production cycle and mechanism of this application.

In this Django application we have three apps: User authentication app(in the **accounts** folder), video analysis app (in the **analysis** folder) and base layout app (in the **bases** folder).

A common way of explaining Django's architecture is to describe it as Model-Template-View (MTV). In each module, Django will construct three main files:model, views, and templates.

**Models** in Django mainly provide the support of different databases. In the models.py file in the **analysis** folder, we build the Result model. It corresponds to a table in the database. You can change, add or delete the model in it to change the structure of the database. Notice that every time you change the model, you need to run the following commands to apply the models to the database:

```
python manage.py makemigrations  
python manage.py migrate
```

**View** is simply a Python function that takes a web request and returns a web response. This response can be the HTML contents of a Web page, 404 error, XML or image, etc.In the views.py file in **analysis** app, there are 3 main views function:

- **targetVideo(request)** function will fetch a video that satisfies the user's requirements from the local video folder and respond to the web request.
- **category(request)** function will fetch the category data form the category form and send them to the targetVideo function.
- **ResultPrefilled(request, video\_id)** function will fetch the user id and video id and prefill them to the analysis form.

In the views.py in **accounts** app, since we don't have a production environment, the reset password link will be sent to the terminal. If you will use it in the future in a production environment, you can change them in the **forgot\_view(request)** function. At the same time you need to go to the setting.py file in the **website** folder and change the EMAIL\_BACKEND setting.

**Templates** in Django are mainly in charge of the appearance of this application. It also receives the data from view functions and displays them in the browser. Template files are stored in the templates folder, written in HTML file. We also used Bootstrap as our front-end framework here.

Every time you modify the code and run the Django application (through the command: *python manage.py runserver*), Django will compile all the files automatically.