Problem Chosen	2021 MCM/ICM Summary Sheet	Team Control Number
C		2103994

Data-Driven Model for Asian Hornet Detection with imbalanced and limited dataset

Summary

As Asian Hornet invadeds Washington State, the public voluntarily provide information about the insects they have seen or captured which looked like the hornet. And to analysis the spread of the hornet and distinguish the insect, we use **hypothesis testing theory** to prove that the spread of hornet is random and the pest has been eradicated gradually in Washington States.

The sighting reports contains images, videos, note texts, spatial-temporal information so it is a multimodal classification. To fully exploit the data set, we propose a **two-stage multimodal classifier**.

1. In the first stage, we propose different methods to process multimodal inputs and transform them into scaled features. For videos, we convert them to images by **key frames extractions** with FFmprobe and FFmpeg tools.

   For images whose backgrounds differ significantly, we crop them into sub-images containing potential hornet only and established an **Inception v4** network based on images in the reports.The network outputs predictions of class probability, which represents a feature of images. Croping the images require annotated object boxs in each picture, we annotate some samples manually and then train a **YOLOv5** object detection network to automate the process.

   For spatial-temporal information, we use inverse and exponential function to describe the decay of hornet spread over space and time.

   For texts, we use **n-gram** and **ridge regression** to classify the report note and also output the class prediction probablity as a feature for the second stage.

2. In the second stage, we combine all the features together to use **logistic regression** as the final classifier. Grid search with cross validation is introduced to tune the model.

We note that the sample set is quite imbalance for the amount of positive sample is much smaller than positive sample, so we use **data augmentation** when deal with images, and use **focal loss** as loss function to train Inception net. We also enhance sample by using **over-sampling** algorithm **ADASNY**.

Also we keep the model to be **online trainable**, so once a new sample comes, we can adjust the model at once. This is realized by **fine-tune** and **FTRL**.

We also test every part of the model is neccessary and propriate by **Alation Experiment** along with hypothesis test methods.

In conclusion, a robust, transferable, multimodal, online learning model is established to classify the reports. And test the situation of hornet invasion in statistics methods.

Keywords: Multimodal Classification, Hornet Detection, Imbalanced sample, Online learning

Contents

1 Introduction

1.1 Problem Background

1.1.1 Asian Giant Hornet Invading

Vespa mandarinia is the largest species of hornet in the world, and the occurrence of the nest was alarming. Additionally, the giant hornet is a predator of European honeybees, invading and destroying their nests. A small number of the hornets are capable of destroying a whole colony of European honeybees in a short time. At the same time, they are voracious predators of other insects that are considered agricultural pests.

In September 2019, a colony of Vespa mandarinia (also known as the Asian giant hornet) was discovered on Vancouver Island in British Columbia, Canada. The nest was quickly destroyed, but the news of the event spread rapidly throughout the area.

From the first report of Asian giant hornets received, the public has played a critical role in the detection of Asian giant hornets in the Pacific Northwest. Washington State Department of Agriculture(WSDA) has set up platforms[1] to collect informations from the public and set traps to get hornets.

1.1.2 Asian Giant Hornet

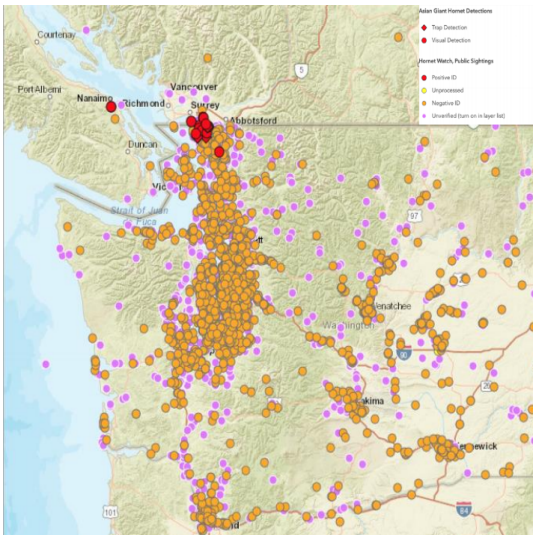

Figure 1: Asian Giant Hornet Detections[2]

Figure 2: *Vespa mandarinia*[3]

Name

Common name: Asian giant hornet, sparrow wasp, murder hornet

Scientific name: Vespa mandarinia Smith, 1852

Range of Activity

Usually to find food, the Asian giant hornet goes only 0.5 to 1.25 miles (1-2 kilometers) away from the nest (and never more than 5 miles (8 kilometers)) .

Life Period

Species	Similarity	Distinguish
Asian Giant Hornet	around 1.5 inches, yellow head, black belly, live on the ground or lower than 6 feet	
VespaCabro	size, shape, color are simlilar	small dierences on face and chest, live higher than 6 feet
VespulaSquamosa	only the queen may be confused in spring	much smaller in size
cicada killers	simliar size	no yellow in head and belly
Dolichovespula Maculata		small size, in the color of black and white, live on the tree or reef

Table 1: Comparison of Similar Species

Like other social wasps, Asian giant hornets are an annual species that build new nests every year. When winter arrives, the current seasons' nests die out and the only individuals that survive are overwintering queens. When overwintering queens emerge in the spring, they seek out protected areas in the ground to begin building a nests.

Comparision with Other Species

While Asian giant hornets do not occur in eastern North America, there are a number of other large wasps that may be confused for them.Here introduce some species alike and there distinctions.

1.2 Our work

- Firstly, we try to predict the spread of hornet, but just find that the spread is not regulated and can be regarded as random walk by examine the independency of report locations.
- Secondly, from all the information WSDA recieved, several confirmed sightings of the pest have occurred in neighboring Washington State, as well as a multitude of mistaken sightings. So we construct a model to distinguish whether or not the observed object in one sighting record is Asian Giant Hornet.
- Then, we use this model to classify unverified records.
- After that, we discuss methods to update the model as new data comes in.
- Finally, we use the result to verify that the hornet in Washington State has been readicated.

2 Assumptions

To simplify our model and eliminate the complexity, we make the following main assumptions in this literature. All assumptions will be re-emphasized once they are used in the construction of our model:

1. The first arrival of Asian Giant Hornet in the United States is the first found in Washington State (time: 2019/9/19; latitude: 49.149394; longitude: -123.943134)

2. Under the control of authoritied concerned, the distance and time of the spread of hornet are limited.

3. For unverified, unprocessed and potential sightings, some of their features are similar to the verified ones', so we can build a data-driven model.

3 Task 1: The Spread of Hornet

To decide whether the spread of hornets can be predicted based the collected. We introduce the hypothesis test theory in statistics to test whether the time series of the longitude and the latitude is a white noise sequence (random sequence) or not. In **theory of time series analysis**, the Ljung-Box test is a test designed for judge the randomness of a time series data. It rejects the null analysis if the data isnt random. In that case, we can perform some methods of time series analysis for prediction. If it accepts the null analysis, then we can draw the conclusion that this sequence is random and can not be used for prediction.

We divide our process into two steps:

First, preprocess the data and perform visualization of the data to draw some qualitative conclusion.

Secondly, we perform the Ljung-Box test and draw the Quantitative conclusion.

3.1 Data Preprocessing

Firstly we want to know about the basic information of these sighting records.

Table 2: Basic Information of records

Positive number	14
Negative number	2096
Unverified number	2342
Unprocessed	15
Date range	1980 to 2020
Latitude range	45.4886 – 49.5480
Longitude range	-124.6650 – 116.8736

According to the table above, we decide to choose process data in these way:

- Select positive ID and unverified sample to anlysis. That's because the sample of Positive ID is too small and random.
- Use the average position of each month's sighting site as the center of hornet, and search the tend of the species by the change of center point.
- Use the data after March, 2019. For it's when the first confirmed sighting appeared.

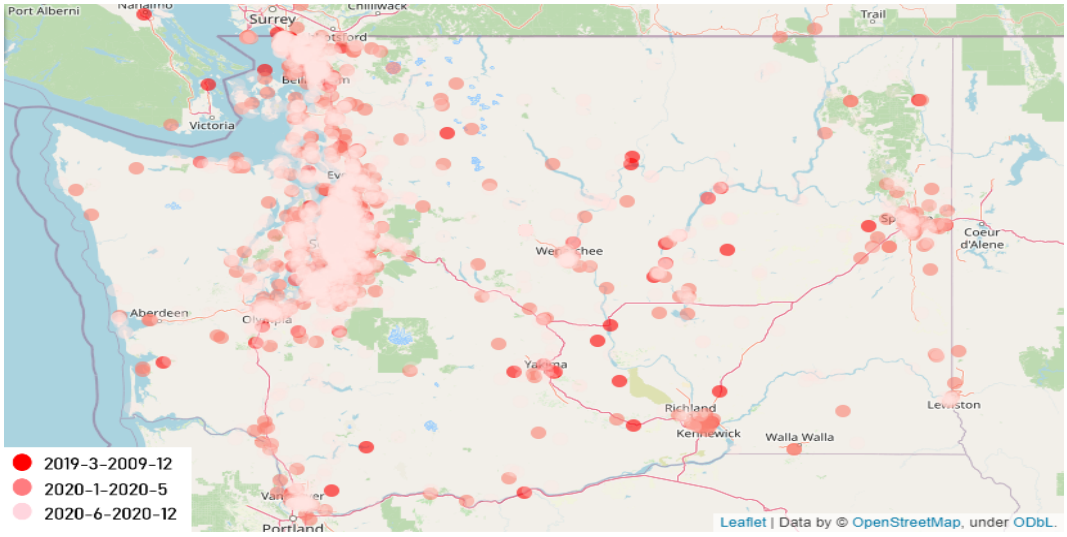

Figure 3: sighting sites

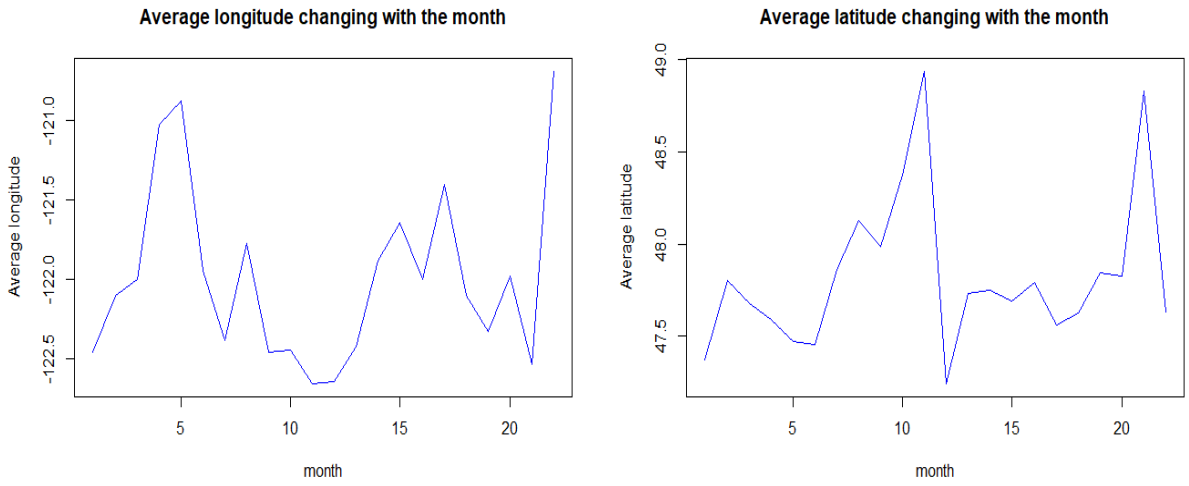

Figure 4: The change of center point

From the pictures above, we find out that the movement of hornet is quite random, so we need to use some statistic methods to approve that.

3.2 Hypothesis Testing

3.2.1 Model

We want to test whether the center points of hornet are independent or the route of the colony has some regulation.So we hypothesis that:

$H0$: **These center points are independent** $\leftrightarrow$ $H1$: **These center points are not independent**

And for it's two dimension data, we test the regulation in longitude and latitude apparently.

As we use ρ_i to represent the k-order lag correlation coefficient of the sample's longtitude. The hypothesis can be writen as:

$$H0 : \rho_1^2 = \rho_2^2 = ...\rho_h^2 = 0 \leftrightarrow H1 : \exists k, s.t.\rho_k^2 \neq 0$$

And choose pivot quantity

$$Q = n(n+1)\sum_{k=1}^{h}\frac{\rho_k^2}{n-k}, \qquad ,h : degree\,of\,randomness \tag{1}$$

For $Q \sim \chi^2(h)$, as significance level is $alpha$,the rejection region is $Q > \chi^2_{\alpha-1}(h)$.

In the same way, we test the randomness of the latitude data by using latitude information to calculate $\bar{\rho}_i$ *and* $\bar{Q}$.

3.2.2 Conclusion

According to calculating result,neither Q and $\bar{Q}$ doesn't fall in the the rejection region even in the significance level of 0.5 (P-value are above 0.5 in whatever the randomness is). **That means we can't say the spread of hornet can be predicted even the accuracy is 0.5**

4 Task 2:Classification Model

4.1 Model Introduction

From all the records we get information in quite a lot of forms including **word, image, video and so on**.

- **Video**: video of the recorded object, including video after capture the insect, surveillance video contains the insect, video of outdoor observation and so on.
- **Image**: image with the recorded insect held on hand, captured by bottle, on a web,in flowers and so on.
- **Word**: notes recorded by the observer; comments replied by the officer
- **Other information**: longitude and latitude of the sighting site and the date.
- **Label Status**: whether or not the object is Asian Giant Hornet , including unverified.

So, we choose **Multimodal machine learning** to classify the records. That means we nearly use all the given information to construct the classification model.

Another key porblem of this issue is: the positive sample is too small (14 compared with the negative sample number:2096). So the **sample imbalance** is great. We use **sample augmentation, focal loss, oversampling** to deal with this problem.

A more detailed explaination of the model can be seen in the picture below:

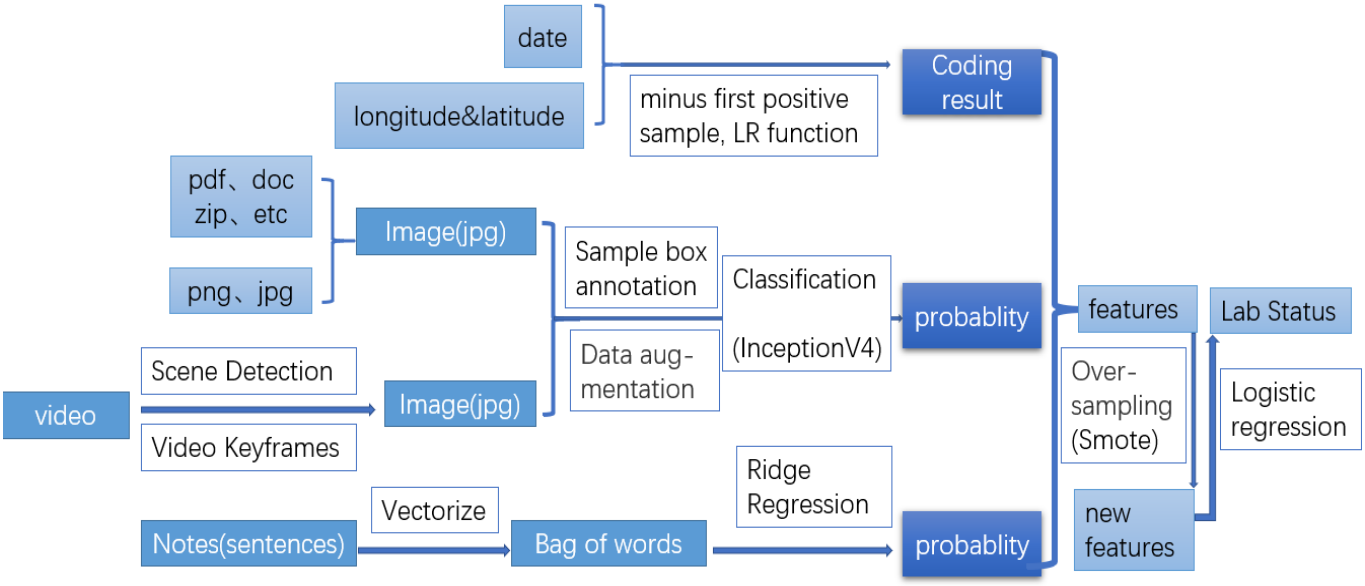

Figure 5: Multimodal Classification Model

4.2 Date, Longitude and Latitude Coding

To deal with these three index, we do two steps:

1. choose the first positive sample as reference(for date we counts the days between two samples) to eminate the influence of magnitude.
2. **LR function** for date, longitude and latitude:$f(x) = e^{-x}$.

The LR function can limit the number of these indexes in 0-1. And when new data comes in, the previous number don't have to change (compared with typical standardized).

4.3 Video Processing

In the given reports there are 92 vedio. Some videos almost keep quiet and can get quite clear image of the object, while others may contains great movement and can not recognize things clearly. In order to use all these information completely, we use **Scene Detection**to recognize **Scene Boundaries** and get**Video Keyframes** from video and porcess them with images together.

4.3.1 Scene Detection

Video scene detection algorithms generally use the degree of similarity difference between frames to measure, if the video frame is greater than a certain threshold, it is considered a new scene, otherwise it is not a new scene.

To do this there are mainly two ways:

- **scikit-video**:it uses the similarity between color and fringe to detect.

- **ffmpeg**[4]:it chooses the frames considering the compression degree.

Compared these two methods we use ffmpeg for it's quicker and moere efficient.

4.3.2 Video Keyframes

Video Keyframes are frames used for Video compression and Video codec and contain complete information. Other non-key frames will be compressed by the difference between them. Video frames can be divided into I,B,P three types.

And we firstly use **ffprobe** to get IBP frames, and than combine them to construct jpg image.

```
#ues ffprobe get IPB time:
ffprobe -i 666051400.mp4 -v quiet -select_streams v -show_entries frame=pkt_pts_time,pict_type
#transform IPB to jpg:
ffmpeg -i 666051400.mp4 -vf "select=eq(pict_type I)" -vsync vfr -qscale:v 2 -f image2 ./%08d.jpg
```

4.4 Image Processing

After combine the image provided and the image produced from the video, we cope with them in two ways:

1. **Sample box annotation**
2. **Data augmentation**

to make the information for efficient thus improve classification's accuracy.

4.4.1 Sample Box Annotation

In many pictures, the insect is hidden in the environment such as flowers and woods and the size of the insect is small.

So it's hard to distinguish the insect through these pictures.To deal with these we try to annotate the object box .For the amount of pictures is large, we do this by **YoloV5**[5], an auto-annotate algorithm:

- First we annotate the sample box by hand to get 750 picture as trian data.
- Then we use YoloV5 to deal with more than 2000 pictures.

The annotation result is pretty good as we can see in the picture below.

And as we can see in 8, the sizes of sample boxes are not uniform, so it's neccessary to do **Sample Box Annotation** to extract information correcetly.

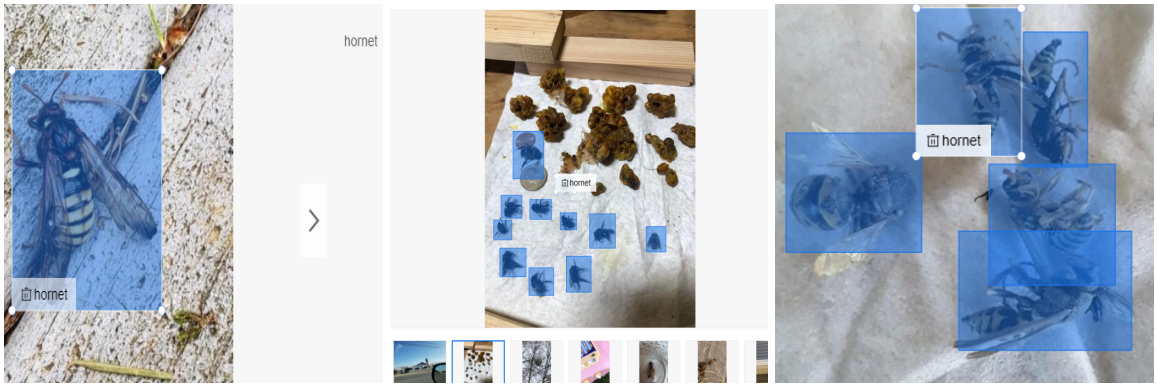

Figure 6: Sample box annotation by hand

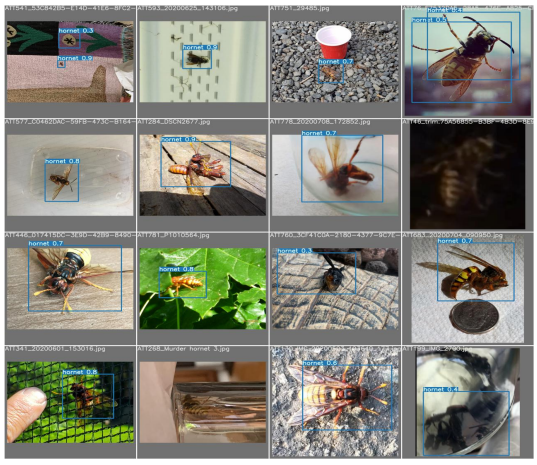

Figure 7: Sample box annotation by yolo

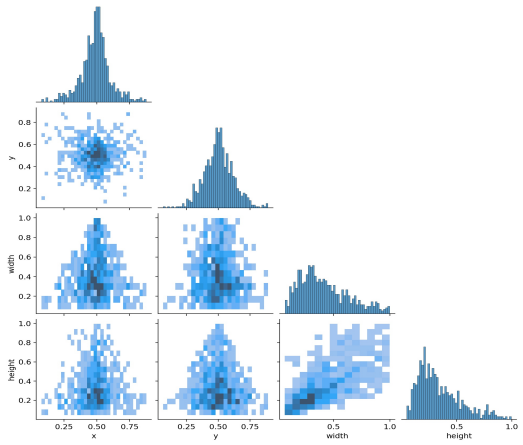

Figure 8: the size and position of object boxs[1]

4.4.2 Data Augmentation

For the number of positive sample is too small, we use **data augmentation** to create more pictures from one positive sample through photo change. Thus we can get more information about how Asian Giant Hornet looks like.

As 8 shows, the size and shape of the object box is not regular(The center of the sample frame is normally distributed and the height and width are skewed), so we choose to do the change randomly, that's to say the angle we rotate or the size we cut and other parameter are choosen randomly, so we can get more useful information.

Through this, we augment the positive sample(combined with the pictures provided by WSDA 26) 20 times and get 520 pictures.

More detailed operation can be seen in the table3, and a result can be seen in the picture9.

[1]x, y reprsent middle point of the object box,width and hight describe the shape.

Table 3: Data augmentation operation

Operation	python code[2]
Flip	RandomHorizontalFlip(), RandomVerticalFlip()
Rotation	RandomRotation(180, expand=True), RandomRotation(180, expand=False),
Affine	andomAffine(180, (0.3, 0.3), (0.4, 1), 3)
Crop	RandomResizedCrop(size), transforms.RandomCrop(size, pad_if_needed=True), transforms.CenterCrop(size), transforms.RandomPerspective(distortion_scale=0.3)
Color	transforms.RandomGrayscale(0.2), transforms.GaussianBlur(9), transforms.ColorJitter(0.8, 0.8, 0.8, 0.4)

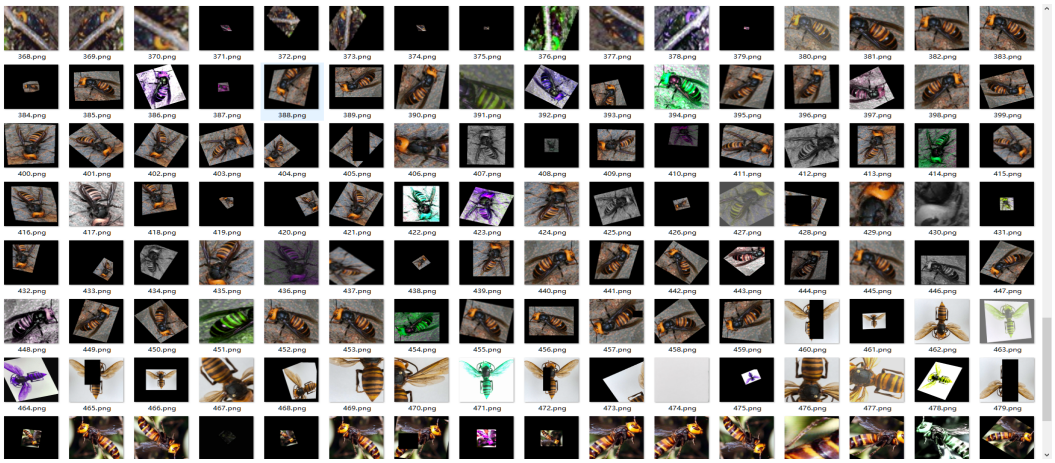

Figure 9: Some data augmentation result

4.5 Image Feature Extraction

4.5.1 Inception V4

After we process all the image and video, we get dozens of pictures. Then we need to extract features from these samples for further classification.

To do this, we choose a **CNN model** which is an improvised version of GoogLeNet: **Inception V4**[6]. For it is a typical model of image classification with many 1 dimension kernels which make it spends less time and space. And it is the **SOTA(state-of-the-art) model** in this area. Following is the structure:

Input–> Stem–>4× Inception A–>Reduction-A–>7× Inception-B–>Reduction-B–> 3×Inception-C–>Average Pooling–>Dropout–>Softmax

The end of the structure is a softmax layer, in these layer we can get the probability of being a hornet. And we use the probability as a feature.

[2]Python code using torchvision.transforms package.

4.5.2 Focal Loss

To deal with the imbalance of sample and make the positive sample easier to be detected, we use **focal loss**[7] in Inception model **instead of cross entropy** to give more weight on the loss of small sample loss. Thus the postive model can be distinguished better.

$$Focal\ Loss = -\alpha(1-p)^{\gamma} log(p) \tag{2}$$

p is the probability produced by the net, $\gamma = 2, \alpha = 0.25, 0.75$

4.5.3 K-folds

For the lack of positive sample in trianing data may lead to great loss in model. We use **k-folds** to deal with the problem:

1. Combine all the images and the augmented image together and shuffle them.
2. Use 5-folds: every time choose $\frac{4}{5}$ of the data to train the model and leave the other $\frac{1}{5}$ for test.
3. Choose the maxium of the loss as total loss, considering one sample may have many images(like video sample), some image is powerful while others maybe not be useful, so we should rely on the most powerful image to classify the sample.
4. After parameter adjust, we use all the positive and negative data to **retrain** the model.

4.6 Text Processing

In the given infomation, there are notes and comments with each record,to deal with the words we need to code them first.We use **N-grams** to **vectorize** the sentence. This takes two steps:

1. take adjacent words to construct combinations.
2. represent the combinations with their frequency in whole sample.

Through then we change a sentence into a string of numbers.

Example: 'One dead wasp seen in Blaine, and suspect flying nearby'
$\Rightarrow$ [0 1 2 ... 0 0 0], dimension is 3000

4.7 Text Feature Extraction

After coding the notes, we use **ridge classification** to classify the notes and get the probability of a sample being a hornet. Ridge classification add penalty term to ordinary model, thus the effect of **multicollinearity** due to the high dimension can be eliminate.

$$x = [x_1, x_2, ..., x_{3000}], \qquad y = 1(positive)\ or\ 0(negative)$$

$$y = \sum_{i=1}^{3000} \beta_i x_i + \beta_0 \tag{3}$$

$$\beta^{ridge} = argmin_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{3000}\beta_j x_j)^2 + \lambda \sum_{i=1}^{3000}\beta_j^2, \qquad N: sample\ size \tag{4}$$

4.8 Overall Classification

4.8.1 Feature Combination

We combine all the features below to be a 5 dimensional vector as **independent variable**

X=[longitude coding,latitude coding,date coding,image probability,
language probability,if sample has photo/video]

And use "Lab Status": Y=1 (if Postive),0(if Negative) as **dependent variable**

4.8.2 SMOTE

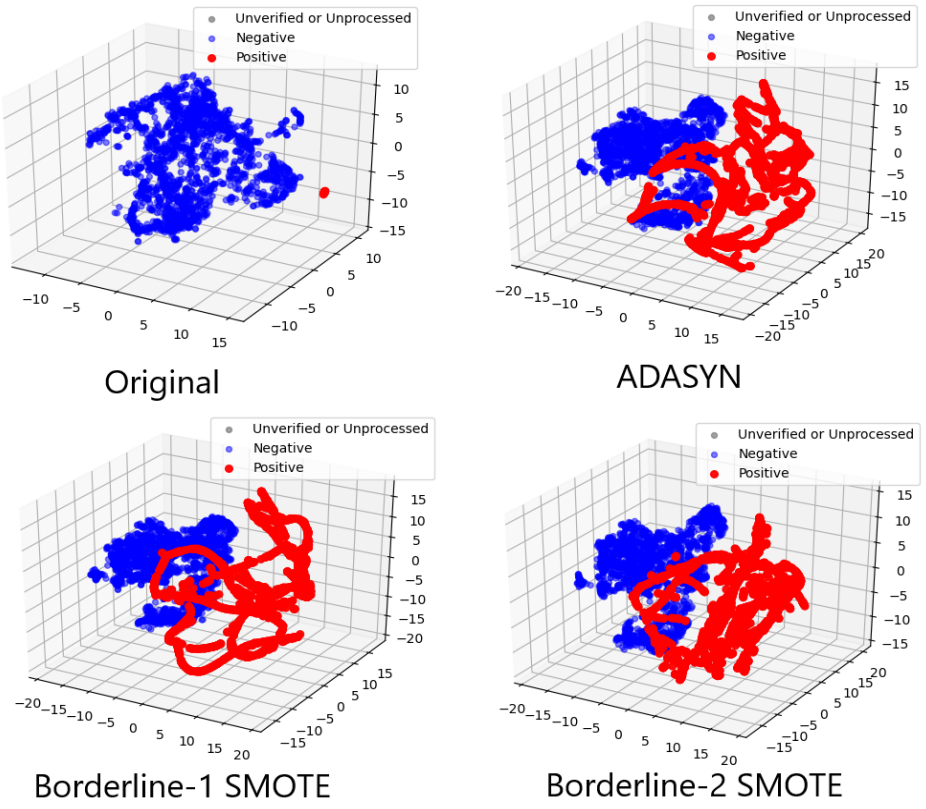

Figure 10: t-SNE T-Distributed of Original Sample and Samples through Three Over-Sampling Method [1]

As is shown in Figure4.8.2,all these features satisfied **Manifold assumption** so the classification is reasonal and the features are usable.

The only problem is that the positive samples are too scarce, so we need to over-sample them. The most intuitive method is randomly repeat those positive samples with techniques like Bootstrap

[1]t-SNE [8] is a tool to visualize high-dimensional data. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. We use PCA result as the original number and iterate 500 times.

and this will not change the space distribution of them, but actually they are too concentrated on one point and our classifier may lay down the border anywhere between two clusters, which is not desired. What's more, slight noices will prevent over-fitting and help increase the robustness of our model.

we introduce SMOTE(Synthetic Minority Oversampling Technique)[9] and **ADASYN**(Adaptive Synthetic Sampling)for over-sampling for we want more noise to train the model. It focuses on generating minority class sample near the border of the minority class and majority class. In this problem, it can strengthen the border near positive and negative samples.

SMOTE algorithm contains three steps:

1. For everyone in the positive sample, get its k-neighbor positive points(by Euclidean distance)
2. Choose sampling ratio n. For each positive sample x, selected n points from the k points in the first step for $x_k, k = 1, 2, ..., n$.
3. : For each x_n, create new point: $x_{new} = x + \epsilon(x - x_n), where \epsilon \sim U(0, 1)$

Then, we get new points added to our samples. ADASYN is an improvised algorithm based on SMOTE. More details of this method are accessible in [10]

4.8.3 Logistic Regression

We conduct **logistic regression** to evaluate all the features and set up a model to classify the records. And the model of logistic regression is showed below:

$$y = \frac{1}{1 + e^{-z}}, \qquad z = \omega^T x + b \tag{5}$$

$$\omega, b : parameter, \qquad y : the\ probability\ to\ be\ positive$$

5 Task 3: Classify Unverified Reports

Using the model in problem 2, we conduct the same process to calculate the features and take all the verified reports as the training dataset to fit the two-stage model. We use the model to **predict the probabilty of the unverified points to be positive**. Output of each stage can be adjusted to a fraction between 0 and 1, indicating the likelihood of the current predicting report is an Asian Giant Hornet sighting.

5.1 Predict Result

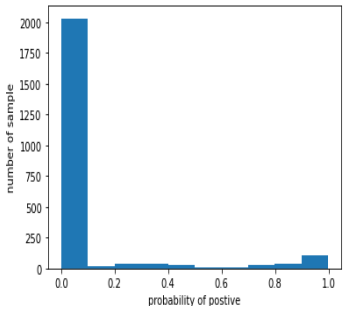

Figure 11: The distribution of positive probability of unverified sample

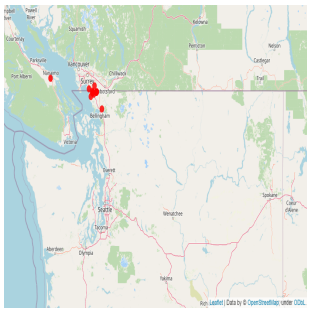

Figure 12: Distribute of original positive samples

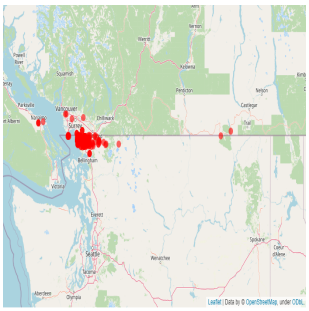

Figure 13: Positive samples and unverified with probability>0.75

Figure 8.2 has shown the probability we get through the model in task two.And the other two pictures show the distribution of positive sample after add the unverified sample whose has more than 0.75 probability to be positive.

6 Task 4: Model Update

Considering the model is complex and the training data is huge, it would spend great long time and quite large space to retrain the model. So when construct the model when have considered this problem to guarantee **online learning**,that means we can update our model given a new sample in a quite short time. So **we can tune the model as soon as new sample comes in**.

The adjustment of model can be divided into two parts:

- **Feature Engineering Part** :process new sample
- **Model Fine-Tuning Part** :adjuest the parameters in learning model.

6.1 Feature Engineering Part

- **Longitude ,Latitude and Date**:For these three index we only need to do: $\hat{x} = e^{-(x-x_{1st\,positive})}, x = longitude/latitude/date$. Then, we can get coding result.
- **Video and Image**: For video,we only need to do the same: scene detection and video keyframes to transfer a video into some images. Then, for
- **Yolo** has trained well and can get object box automatically, and when the "Lab Status" is postive, it will be augmented as well.
- **Text**To deal with the notes,when we do **n-grams**, we have set up dictionary to the coding to the most frequent 3000 words and phrases. So when new sentence comes in, we only need to code the sentence by the coding dictionary.

6.2 Model Fine-Tuning

In this part, we adjust the parameter of the model as new sample comes in,and we need to make sure the model is **robust, time-saving, space-saving**,thus it can update on time.

Considering all these we use **FTRL** and **Fine-Tune** to do this part.

6.2.1 Fine-Tune

There are mainly three classification model in the overall process:

- Ridge Classification for language coding
- Inception V4 for image
- Logistic Regression for all features

Between these Inception V4 is much more complicated and spends much more time to train. So for this part, we choose to keep most of the net fixed and just adjust weights of the end of the net like Inception C module, also with a small learning rate decayed exponetially. The **fine-tune** can also make it easier to converge and save more time.

6.2.2 Online optimizer: FTRL

Table 4: Notations

Symbol	Definition
T	Total iterationround
t	The round of iteration
l	Loss function
w	Parameters
W	The range of parameters
g	The gradient of losss function by parameters
η	Learning rate
λ_1	Regularization coefficient

To deal with the robust of model tuning and optimize the model as we have new report, we choose **FTRL** algorithm and pack it as an optimizer.

And use it in updating ridge classification, Inception V4 and logistic regression.

For this model can decrease regret($Regret = \sum_{t=1}^{T} l_t(w_t) - min_W \sum_{t=1}^{T} l_t(w)$)and increase sparsity. So we take less space and the model is more accurate. Here is the iterate formula.

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left(\sum_{s=1}^{t} \mathbf{g}_s \cdot \mathbf{w} + \frac{1}{2} \sum_{s=1}^{t} \sigma_s ||\mathbf{w} - \mathbf{w}_s||_2^2 + \lambda_1 ||\mathbf{w}||_1 \right), \qquad \sum_{s=1}^{t} \sigma_s = \frac{1}{\eta_t} \tag{6}$$

6.2.3 Updating Process

After we treat the new report as feature engineering part shows, we send these data to tuning these model and save the model as 4.1 shows. And repeat this once a new report comes.

7 Task 5:Hornet in Washington State

We need to find evidence that prove the pest has been eradicated.

Intuitively, when the pest has been eradicated in the area, the probability of the occurences of positive sample in Washington State equals to zero. To estimate the probability, we calculate the frequency of the positive sample in a year. If this value is small, than we can approximately think the probability of the occurences of positive sample equals to zero.

Utilizing the online learning method incorporated in our framework, We can easily get the result when new data comes in. We set the threshold 0.01, if the frequency calculated by new data is less than 0.01, then we draw the conclusion that the pest has been eradicated in Washington State.

8 Model Evaluation and Testing

8.1 Ablation Experiment for Focal Loss

To test whether focal loss is efficient, we do an **Ablation Experiemt** as below:

8.1.1 Model

$$The\ accuracy\ of\ model: y = \mu + \epsilon, \qquad \epsilon \sim N(0, \sigma^2)$$

Hypothesis:Image classification is more efficient by focal loss instead of BCE(Binary Cross Entropy)

$$H1: \mu_1 < \mu 2 \leftrightarrow H2: \mu_1 \geq \mu_2$$

$y_1 = \mu_1 + \epsilon_1, y_2 = \mu_2 + \epsilon_2, y_1, y_2$represent the accuracy by focal loss and BCE respectively.

We use **T test** to verify the hypothesis.

8.1.2 Test

As we use 5-fold method to go through the model, so we can get 5 samples in each case: $y_{11}, y_{12}, y_{13}, y_{14}, y_{15}$ and $y_{21}, y_{22}, y_{23}, y_{24}, y_{25}$.

By using the two strings data to do T-test, And find out that focal loss is efficient when **significance is 0.1**.

8.2 Classification Model Evaluation

Before classify the unverified reports, we need to find a best model with the lowest generalized loss by changing the hyper-parameters. Since the data set is still small even after data augment, we

use K-fold cross-validation to evaluate the performance of our classfiers. After listing out all the candidate hyper-parameters, we use grid search to select best conbination of hyper-parameters. The cross-validation grid search is implemented with Sciki-learn Python Package[11].

We use precision, recall and ROC-AUC(area under receiver operating characteristic curve) as the scoring function for grid search. Recall is used when refitting the model so as to retrieve the best model and hyper-parameter combination from Pareto Frontier.

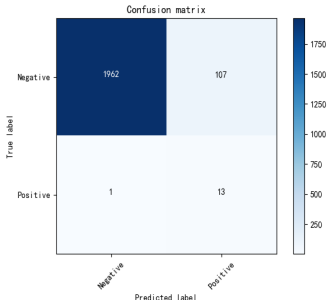

Figure 14: Confusion matrix of the original dataset.

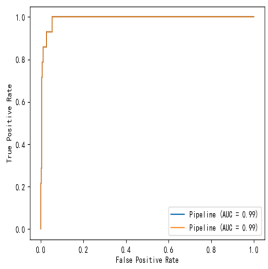

Figure 15: ROC Curve

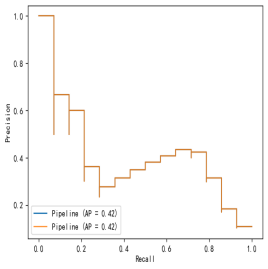

Figure 16: Precision-Recall Curve

As we can see in the Confusion matrix, "True Negative"(TN) decreases apparently, thus make the model more efficient. For it is of great importance to have low TN,or we would be more likely to miss positive sample and can't alarm the public on time.

In the meantime,we also gaurantee the FP number is relatively small by using multiple indexes to choose hyper-parameters .

Average Precision(AP) is small as it is shown in precision-recll and that is why we request small "True Negative"(TN) in the model.From confusion matrix abouve,the number of "False Positive"(FP) is higher than "True Positive"(TP) for we use orignal dataset to calculate metrics.

In real situation,the model well be more effective.For the FP sample would face re-check by the expert,and our job decreases 2083 samples into 120 samples as the most probable positive sample.It greatly reduced the burden of manpower check.

9 Strengths, Weaknesses and Improved Method

9.1 Strengths

Strength 1: The classification model is a two-stage, using separate networks for different features in multimodal data sets. So:

- It is flexiable and transferable, (Like object box annotation and image calssification, we can change another model and guarantee the result is end to end as well), also if we use independent outside dataset, like the AGH dataset from Biology Department or introduce industrial-grade model such as GPT-3 in NLP area to handle notes, the model still can work.

- The model is decoupling, so when the image or the text is missing, it can still do classification. So the model is quite robust.

Strength 2: The classification model is an online learning model. Once a new sample comes, the model can be updated quickly. So this great improves time and space efficiency and let the model constantly optimizable.

9.2 Weaknesses

Weakness 1: The testing for the first task is not enough,we just consider the tend of the center of whole the hornet,but ignore the situation for each colony.

9.3 Improved Method

Take the environment of image into consideration In image classification part, we didn't consider the environment(like cave or hornet nest,etc). The attachment of the problem introduces that the nest of Asian Gaint Hornet is special, and they live in the area near the ground and don't like to go around the flowers as well.So we may excavate this with more background information.

Memo

Dear Governors,

We are writing to inform you our achievement after performing data analysis and modeling. Based on the data provided by the public reports, we have developed a data-driven framework that can be used for both prediction and inference and update its model given new reports over time.

Our framework are structured as follows, which can be used for identifying this pest when new data comes in.

1. **Converting different kinds of data**

   We adopted a series of method to handle different types of data:

   video:
   Use the Scene Detection and Video Keyframes to transform the video into images, and use these images for future analysis.
   Image:
   First use Sample box annotation and Data augmentation to use the information of images more efficiently. Use trained CNN model Inception V4 with online learning method FTRL and Fine-Tune to update the model and calculate the the probability of being a hornet for new data (image score).
   text:
   When new data come in, use N-grams to vectorize the sentence and calculate the frequencies of 3000 feature words. Then based on the ridge regression model trained above, we use online learning method FTRL and Fine-Tune to update the model, and get the probability of being a hornet for new data (text score).

2. **Performing the penalized logistic regression model** You have got the text score, image score and some other features of the new sample from above, now use the trained the penalized logistic regression model combined with online learning method FTRL and Fine-Tune to update the model, and get the resulting probability.

Some highlights of our framework:

1. **Strategies to deal with imbalanced data** We found that labelled data are classified into two imbalanced categories (Positive ID: 14, Negative ID: 2096) based on a brief observation of the data. Imbalanced data will make the model tend to classify new data into the Negative since it can result higher accuracy. In that case, the model is possibly incorrect. To handle this problem, we adopt a series of methods:
   (1) Use focal loss to make positive sample easier to be detected.
   (2) Use Synthetic Minority Oversampling Technique (SMOTE) to create artificial positive sample in logistic regression model.

2. **Online learning methods** We use online learning method FTRL and Fine-Tune to update the model when new data comes in, reduce the computation time and make quick classificaiton.

   Moreover, this data-driven framework can be extended to other similar problem, especially those problem with multitype data. You can adopt the framework deal with and train specific model to detect other pests in Washington State. Also, the SMOTE method implemented in the framework can deal with more general classification problem with imbalanced data.

References

[1] Facebook. Asian Giant Hornet Watch. https://www.facebook.com/groups/hornets.

[2] Washington State Department of Agriculture. 2020 Asian Giant Hornet Public Dashboard. https://agr.wa.gov/departments/insects-pests-and-weeds/insects/hornets/data.

[3] Wikipedia.Asian Giant Hornet. https://en.wikipedia.org/wiki/Asian_giant_hornet.

[4] ffmpeg . https://ffmpeg.org/ffmpeg-filters.html#select_002c-aselect.

[5] Kuznetsova, Anna, et al. Detecting Apples in Orchards Using YOLOv3 and YOLOv5 in General and Close-Up Images. International Symposium on Neural Networks, 2020, pp. 233243.

[6] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.

[7] Lin, Tsung-Yi, et al. Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 29993007.

[8] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).

[9] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 13221328. IEEE, 2008.

[10] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, SMOTE: synthetic minority oversampling technique, Journal of artificial intelligence research, 321-357, 2002.

[11] Pedregosa, Fabian, et al. Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, vol. 12, no. 85, 2011, pp. 28252830.

Appendices

Appendix1

```
class Inception(nn.Module):
"""
implementation of Inception-v4
"""
def __init__(self, num_classes):
    super(Inception, self).__init__()
    self.stem = Stem_v4()
    self.inception_A = self.__make_inception_A()
    self.Reduction_A = self.__make_reduction_A()
    self.inception_B = self.__make_inception_B()
    self.Reduction_B = self.__make_reduction_B()
    self.inception_C = self.__make_inception_C()
    self.fc = nn.Linear(1536, num_classes)

def __make_inception_A(self):
    layers = []
    for _ in range(4):
        layers.append(Inception_A(384, 96, 96, 64, 96, 64, 96))
    return nn.Sequential(*layers)

def __make_reduction_A(self):
    return Reduction_A(384, 192, 224, 256, 384) # 1024

def __make_inception_B(self):
    layers = []
    for _ in range(7):
        layers.append(Inception_B(1024, 128, 384, 192, 224, 256,
                                  192, 192, 224, 224, 256)) # 1024
    return nn.Sequential(*layers)

def __make_reduction_B(self):
    return Reduction_B_v4(1024, 192, 192, 256, 256, 320, 320) # 1536

def __make_inception_C(self):
    layers = []
    for _ in range(3):
        layers.append(Inception_C(1536, 256, 256,
                                  384, 256, 384, 448, 512, 256))
    return nn.Sequential(*layers)

def forward(self, x):
    out = self.stem(x)
    out = self.inception_A(out)
    out = self.Reduction_A(out)
```

```
        out = self.inception_B(out)
        out = self.Reduction_B(out)
        out = self.inception_C(out)
        out = F.avg_pool2d(out, 8)
        out = F.dropout(out, 0.2, training=self.training)
        out = out.view(out.size(0), -1)
        # print(out.shape)
        out = self.fc(out)
        return F.softmax(out, dim=1)


class FocalLoss(nn.Module):
    def __init__(self, alpha=0.25, gamma=2, size_average=True):
        """
        Focal loss function, -(1-yi)** *ce_loss(xi,yi)
        :param alpha: , class weight
        :param gamma:
        :param size_average:
        """
        super(FocalLoss, self).__init__()
        self.size_average = size_average
        assert alpha < 1
        self.alpha = torch.zeros(2)
        self.alpha[0] += alpha
        self.alpha[1:] += (1-alpha)
        self.gamma = gamma

    def forward(self, preds, labels):
        """
        :param preds: Predict. size: [B,C]
        :param labels: Actual. size: [B]
        :return:
        """
        assert preds.dim() == 2 and labels.dim() == 1
        self.alpha = self.alpha.to(preds.device)
        preds_logsoft = torch.log(preds)
        preds_softmax = preds.gather(1, labels.view(-1, 1))
        preds_logsoft = preds_logsoft.gather(1, labels.view(-1, 1))
        self.alpha = self.alpha.gather(0, labels.view(-1))
        loss = -torch.mul(torch.pow((1-preds), self.gamma), preds_logsoft)
        loss = torch.mul(self.alpha, loss.t())
        loss = loss.mean() if self.size_average else loss.sum()
        return loss
```

```
    def LRGSCV(ds: pd.DataFrame):
    lr = LogisticRegression(max_iter=200, n_jobs=-1)
    ovs = ADASYN(n_jobs=-1)
```

```
    pipe = Pipeline([('ovs', ovs), ('lr', lr)])
    grid = [
        {
            'ovs__sampling_strategy': [0.25, 0.5, 1],
            'lr__penalty': ['elasticnet'],
            'lr__C': [0.001, 0.01, 0.1, 1, 10, 100],
            'lr__fit_intercept': [True, False],
            'lr__class_weight': ['balanced', None],
            'lr__solver': ['saga'],
            'lr__l1_ratio': [0.2, 0.5, 0.8]
        }
    ]
    gsCV = GridSearchCV(
        estimator=pipe, cv=5, n_jobs=-1, param_grid=grid,
        scoring={
            'p': 'precision',
            'r': 'recall',
            'roc': 'roc_auc'
        },
        refit='r' , verbose=2
    )
    from time import strftime, localtime
    log_dir = 'output/' + strftime("%Y_%m_%d_%H_%M_%S", localtime())+'/'
    if not os.path.exists(log_dir):
        os.mkdir(log_dir)
    Xtrain = ds[ds.label != -1]
    ytrain = Xtrain['label']
    Xtrain.drop(axis=1, columns='label', inplace=True)
    gsCV.fit(Xtrain, ytrain)
    from joblib import dump
    dump(gsCV, log_dir+'gsCV')
    file = open(log_dir + 'log.txt', 'w')
    file.write(gsCV.cv_results_.__str__())
    para = gsCV.best_params_
    file.write(para.__str__())
    file.flush()
    file.write('start refit')
    ovs.set_params(sampling_strategy=para['ovs__sampling_strategy'])
    lr = LogisticRegressionCV(
        Cs=[para['lr__C'], 0.002, 0.02, 0.05, 0.2, 0.3, 0.5, 0.75, 1.2, 1.5, 2, 5,
            15, 50]
        + list(np.arange(0.8, 1.2, 0.02)),
        cv=5, n_jobs=-1, penalty=para['lr__penalty'],
        solver=para['lr__solver'], max_iter=200,
        class_weight=para['lr__class_weight']
    )
    if para['lr__penalty'] == 'elasticnet':
```

```
        lr.set_params(l1_ratio=para['lr__l1_ratio'])
    best_model = Pipeline([('ovs', ovs), ('lr', lr)])
    best_model.fit(Xtrain, ytrain)
    ypredict = best_model.predict(Xtrain)
    dump(best_model, log_dir+'best_model')
    file.write('refit done')
    file.write('Metrics on original Dataset: {}\n'.format({
        'acc': accuracy_score(ytrain, ypredict),
        'f1': f1_score(ytrain, ypredict),
        'precision': precision_score(ytrain, ypredict),
        'recall': recall_score(ytrain, ypredict),
        'roc_auc': roc_auc_score(ytrain, ypredict)
    }))
    cm = confusion_matrix(ytrain, ypredict)
    plot_confusion_matrix(
        cm, ['Negative', 'Positive'], log_dir + 'train_cm.png')
    file.write('\ntrain_cm:\n')
    file.write(cm.__str__())
    plot_curve(best_model, Xtrain, ytrain, log_dir + 'roc.png', True)
    plot_curve(best_model, Xtrain, ytrain, log_dir + 'pr.png', False)
    file.close()
    return gsCV
```