Master's Thesis

Detecting phenological transition dates of vegetation based on multiple deep learning models

Zhaoyang Cheng
zyclark.cheng@gmail.com

Thesis committee:

Prof. dr. ir. M.J.T. Reinders
Dr. J.C. van Gemert
Dr. S. Picek
Dr. S. Khademi

August 2018

Abstract

Vegetation phenology is the interaction between vegetation activities and ecosystem. Accurate monitoring of vegetation phenology is required to build models and enhance the understanding of the relationship between creatures and climate-environment. PhenoCam is a ground-level, webcam based images database recording the growing of various vegetations, PhenoCam and multiple modeling methods have been utilized to study vegetation phenology since 2000s. In this paper, it first time the deep learning models are applied to detect the phenological transition dates of vegetation. Four different deep learning models: Convolution Neural Network (CNN), Siamese Network, 3-D Fully Convolution Neural Network (FCN) and Regression Network are used to study the vegetation phenology, based on these approaches, the transition dates of vegetation activities within annual time can be determined from webcam-based images, some of these deep learning methods are more accurate than traditional modeling method in detecting the transition dates.

1. Introduction

The vegetation dynamics in ecosystems largely reflect the response of the biosphere to dynamics of the climate [1][2][3]. Analysis on vegetation dynamics have provided important record of how vegetations have responded to climate change, however, observation by human on long-term vegetation dynamics is laborious and time-consuming. Accurate and automatic monitoring of vegetation dynamics is therefore an important work for researchers to investigate.
PhenoCam is a database consisting of inter-annual digital images of various vegetations throughout North America. Because of the consistent record of phenology of different vegetations, PhenoCam has played an important role in building model to monitor vegetation dynamics at regional scales. In the study of building models based on PhenoCam, a key challenge is determining how the phenology derived from webcam based images relate to biological events that a human observer would recognize. In the last decade, a number of different methods have been developed to determine the timing of vegetation's greenup and senescence(i.e. the start and end of growing season of vegetation) which are called transition dates of vegetations, most methods extract the vegetation index (VI) that calculates a specific feature of vegetation, e.g. green chromatic coordinate (GCC) and red chromatic coordinate (RCC), and build sigmoid-based model to fit the vegetation index, the curve of fitted model can largely reflect the growing of vegetation. Instead of choosing a specfic vegetation index, utilizing deep learning to automatically extract the most distinguishable features is more convenient and accurate.

In this thesis, my research topic is how to apply deep learning methods to detect transition dates of vegetation based on images of vegetations. The research data includes the data of four sites: (1) Queens; (2) Bartlettir; (3) Umichbiological; (4) Harvard, in PhenoCam database.

1.1. Monitoring phenology using PhenoCam

Vegetations have many key phenological phases, e.g., the date when leaves start becoming green, the date when flowers come out, etc.. In this thesis, four basic phenological activities are investigated: (1) the dates when the majority of trees started leafing out, its phenological meaning is the onset of photosynthetic [4]; (2) the dates when green leaf area is maximum, it phenological meaning relate to the end of spring; (3) the dates when the canopy first started to change color in the fall, which means the onset of senescence; (4) the dates when the majority of trees had lost all leaves, its phenological meaning corresponds to the start of dormancy. Field based ecological studies have demonstrated that vegetation phenology tends to follow relatively well-defined temporal patterns [5]. The dates when leaves green up tend to be followed by a period of rapid growth, followed by a relatively stable period of maximum leaf area, after the brightest color phase in fall, leaf area decreases dramatically. The vegetation index(VI) used to quantify phenological status of the vegetation over time is usually green chromatic coordinate(GCC), researchers can fit logistic function to VI from PhenoCam and find key transition points (usually the extremas) which are identified as transition dates, however, the simple logistic function is not accurate enough in detecting the transition dates because of the complexity of temporal and spatial information of PhenoCam data. They therefore propose some modified sigmoid methods based on empirical equations.

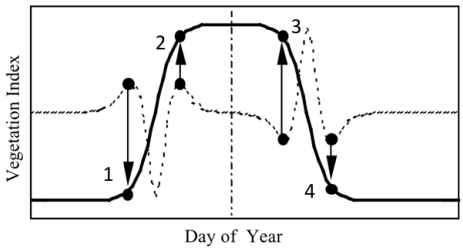

Figure 1. A synoptic figure [5] of how transition dates are calculated by the extremas in the rate of change of curvature, the solid line is an idealized curve of VI with respect to day of year, the dashed line is the rate of change of VI curve.

2. Related Work

In this section, I first introduce some sigmoid-based models that have been successfully applied in detecting

the transtion dates of vegetation, then I introduce previous works that study phenology through deep learning.

$$GCC = \frac{Green}{Green + Red + Blue} \quad (1)$$

The vegetaion index used in forest phenology is green chromatic coordinate(GCC), equation 1 is a non-linear transformation of the camera's measured green digital numbers to values representing the proportion of the greenness measured, GCC measures the changing levels of green pigmentation in vegetation. An advantage of using GCC is to reduce the influence of differences in scene illumination between images, literature [6] found that weather-induced changes in scene illumination are largely suppressed when using GCC, it is especially helpful when applying GCC index to PhenoCam data because illumination is often different in morning and afternoon, illumination also changes quickly with different weather conditions in PhenCam datasets.

2.1. Simple sigmoid-based method

$$f_s(t) = \frac{c}{1 + exp(a + bt)} + d \quad (2)$$

Equantion 2 is the Simple Sigmoid moedl, which is widely used in phenology community [5][6][7]. $f_s(t)$ represents the model value of vegetation index, parameter a decides the time of decrease or increase, parameter b controls the rate of increase or decrease, parameter c is the amplitude of increase or decrease in vegetation index, parameter d defines the dormant season baseline value of vegetaion index.

2.2. Generalized sigmoid-based method

The Simple Sigmoid model does not work accurately enough in fitting the decreasing greenness in summer time, so Elmore *et al*. [8] propose a Double Sigmoid model,

$$f_{pv}(t) = m_1 + (m_2 - m_7 * t) * V \quad (3)$$

$$V = \frac{1}{1+exp((m_3-t)/m_4)} - \frac{1}{1+exp((m_5-t)/m_6)} \quad (4)$$

in equation 3, $f_{pv}(t)$ is photosynthetic vegetation fraction (FPV) stacked by day of year, $f_{pv}(t)$ is related to the pattern of leaf development and growing season stability. m_1 is the average FPV measured in winter, m_2 is the difference between FPV measured in summer and m_1, m_3 and m_4 control the shape of growth curve in summer time, m_5 and m_6 control the shape of growth curve in winter time, the parameter m_7 tunes the seasonal vegetation index, hereby makes this model more accurate. Recently a more flexible model is presented by [9], they introduce two additional parameters v_i, q_i in equation 6 , which allow more flexible rates of increase near the lower and upper asymptotes of the sigmoid-based function. This Generalized Sigmoid model can not only control the baseline value of vegetation index via a_1, but fit the greenness decrease in summer time which is a common phenomena in many sites via a_2 and b_2.

$$f(t) = a_1 * t + b_1 + (a_2 * t^2 + b_2 * t + c) * V \quad (5)$$

$$V = \frac{1}{[1+q_1 exp(-h_1(t-n_1))]^{v_1}} - \frac{1}{[1+q_2 exp(-h_2(t-n_2))]^{v_2}} \quad (6)$$

2.3. Local extremas and transition dates

For each sigmoid-based method, the local extrema in the rate of change of curvature is estimated as phenological transition date,

$$k = \frac{f''(t)}{(1 + (f'(t))^2)^{\frac{3}{2}}} \quad (7)$$

equation 7 [10] can be used to calculate the rate of change of curvature k. In Simple and Double sigmoid model, points where the k is largest and smallest in the seasonal transition phase are identified as transition dates, for example, in Figure 1, point 1 and 2 are estimated as the transition dates for event 1 (majority of trees leafing out) and event 2 (green leaves area reaches maximum) respectively, point 3 and 4 are identified as event 3 (leaves start changing color in fall) and event 4 (majority of trees lose leaves) respectively.
For Generalized Sigmoid model, the third extreme in the curvature change rate is used to detect the transition date of event 2. the transition date of event 1 is identified as the date corresponding to 10% amplitude between the dormant season and the values of vegetation index at event 2, the detection approach is similar in fall.

2.4. Deep learning and phenology

There is no previous work that applies deep learning to datasets from PhenoCam, while research usually apply deep learning model to plants classification by convolution neural network (CNN) and long short term memory network (LSTM)[11], there are also applications: classification of vegetation growing stages by CNN [12], corp yield estimation by CNN [13]. In these studies, every training image has a corresponding label, and test images are assigned labels at prediction stage, which is same with classification tasks that have been solved by CNN in other fields.
In the detection of transition date problem, transition date is only one day of year, thus, the visually annotated label of transition date is also one day of year, which means many samples do not have labels, it is difficult to perform classification with CNN, literature [14] inspires me to solve the transition detection problem by video shot boundary detection methods.
Considering the scale of PhenoCam database, deep learning has potential to solve phenological problems with PhenoCam, however, shortage of label and the format of label

could be problems if I want to apply deep learning to PhenoCam dataset, as unsupervised learning is unsuitable in solving phenological problems at present. To tackle above problems, I present several deep learning models in following sections.

3. Methods

Looking through the aforementioned sigmoid-based algorithms in a deep learning worker's perspective, I find that they all belong to unsupervised learning style, to acquire more accurate estimation of transition dates, I bring the human-annotated label in and build different neural network models to detect transition date in a supervised learning way, morever, once the trained model is built, it can be applied to data in coming years.

To define the phenological problem with deep learning methodologies , training data, label and test data should be defined at the beginning, I therefore regard data in year 2008 and 2009 as training data, and data in year 2010 as test data. The label is the day of year that transition event happens. The data is continuous time-series images. I propose four different methods to solve this problem in different perspectives, the 3-D fully convolution neural network and regression network exploit the temporal information of data, while convolution neural network and siamese network focus on using spatial information of images.

3.1. Classification by convolution neural network

In conventional classification task, the sample size of each class is basically same, and every traning sample has a corresponding label, however, in PhenoCam dataset, labeled samples is too few to allow us perform a balanced classification, I therefore utilize data augmentation to make the class size roughly balanced.

3.1.1 Data augmentation

I find the transition date is the most representive day during the transition period, but the transition period usually lasts few days, so at first stage I decide to 'expand' the transition dates label in a small scale, in other words, I assign same label to the former and later E days of labeled transition date, these days are also regarded as transition dates. By visually inspect, the candidates of E are [3, 4, 5, 6, 7], how to choose the best 'expanded index' E is another task to do, a naive classification model consisting of convolutional layers and fully connected layers is used to find the best E, the expanded four events are regarded as four classes, for the rest of images, they are 5-th class, let us call it the noise class hereafter, however, even having expanded the labeled four event classes, the images in noise class is still much more than those in other four classes, to make class size balanced, I use data augmentation to increase the number of images of four event classes, the first measure I take is horizontal flipping of image, horizontal flipping reserves the characteristics of vegetation images and increase the number of images of four classes by the factor of two. I also notice that image's brightness changes over time in each day due to different illumination and weather condition, for example, the images taken in the morning and images taken in the afternoon may look different because of different brightness, to suppress the weather-induced change of illumination, I use gamma correction[15] to generate one darker and one brighter image from every original image,

$$I_{out} = I_{in}^{\gamma} \tag{8}$$

in equation 8,I_{in} and I_{out} is the image before and after gamma correction, γ is the parameter to control the brightness of image, by setting γ 1.1 and 0.9, a darker and a brighter image is generated based on original image. I therefore have a new dataset three times the size of previous dataset. When using image fliping and gamma correction together, the new dataset is six times the size of original dataset.

The data augmentation is also applied to the noise class, by sampling images from the noise class, there are five class with balanced size. I use data in 2008 as training data, data in year 2009 as validation data and data in 2010 as test data to select the best 'expand index' N, in my experiments, N is chosen as 5, which means the labeled transition dates are not 1 day but 11 days.

After augmentation, the only problem is how to design the architecture of deep learning model, for this common classification task, a convolutional neural network (CNN) is used, CNN has achieved great success since the AlexNet in ImageNet Challenge 2012, more and more nets with improved architecture are developed [16][17][18], they are widely applied in classification, object detection, semantic segmentation, etc.. CNN-based solutions usually have much higher accuracy , they are replacing traditional machine learning algorithms in many fields. In my scenario, for each site there are about 600 images per class (300 for site Bartlettir), advance network, e.g. ResNet [18] or VGG [17] is not used here otherwise serious overfitting will occur and the accuracy would not be high.

The architecture is shown in Table 1. When designing the architecture of network, I follow the principle of designing VGG net [17], with a given receptive field (the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because more non-linear layers increase the depth of the network which enables it to learn more complex features, and at a lower cost. The literature [19] found that compared to the CNN structure having same number of feature maps per layer , the pyramid architecture (the number of feature maps of this structure increases by a

Layer	Input size	Kernel size
data	N 256 256 3	3 3 3 8
layer 1	N 256 256 8	8 3 3 8
pool 1	N 128 128 8	
layer 2	N 128 128 8	8 3 3 16
pool 2	N 64 64 16	
layer 3	N 64 64 16	16 3 3 16
pool 3	N 32 32 16	
layer 4	N 32 32 16	16 3 3 32
pool 4	N 16 16 32	
flaten layer	N 16×16×32	
fc layer1	N 50	
fc layer2	N 5	
softmax layer	N 5	

Table 1. Architecture of our classification net, fc layer means fully connected layer, we use max pooling [16] to retain the important image information and reduce the number of parameters of model.

factor of multiple) can exploit computation resources more efficiently. At the beginning of designing the network, I use seven convolution layers, when number of convolution layers is reduced to four, the model still yields good result, I therefore only use model with four convolution layers. The number of parameters of the net is more than the number of samples, to suppress overfitting, I use dropout [20] in fully connnected layer. At the softmax layer, each test image is assigned to a class, to find the transition date, I find all the dates of test images in one class, after discarding the earliest date and latest date, the averaged date is the predicted transition date for test dataset.

3.2. 3-D fully convolution neural network

In CNN-based classification method, the temporal information of PhenoCam dataset is ignored. Inspired by the methods used in video shot boundary detection [14], I treat the images dataset as a continuous video, such that the transition date can be found in the same way that shot boundary in video is found. 3-D means dimension width, height, and time, 3-D convolution neural network (CNN) takes video snippets as input, and the convolution kernel is a 3-D cube rather than a 2-D window. To reduce number of parameters, I use fully convolution network (FCN) [21], in other words, there is no fully connected layers in FCN. Figure 2 shows how the input frames relate to the prediction by 3-D FCN when batch size is 1. If the frame 6 of input 10 frames is a shot boundary, the input snippet will be annotated as 1, otherwise this snippet is labeled as 0.

The architecture of my FCN is presented in table 2. In deep learning, data is divided into many batches having same length which is called batch size, every batch is fed to neural network and neural network minimizes the loss based on batch. N denotes the batch size in table 2, the 3-D FCN is

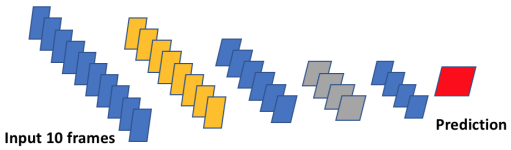

Figure 2. Illustration of 3-D FCN for shot boundary detection. each frame prediction is based on the context of N frames, in my setting, N is 10, the frame 6 of input 10 frames is predicted as shot boundary or not.

Layer	Input size	Kernel size	stride
data	N 10 64 64 3	3 5 5 3 16	1 1 2 2 1
layer 1	N 8 30 30 16	3 3 3 16 24	1 1 2 2 1
layer 2	N 6 14 14 24	3 3 3 24 32	1 1 2 2 1
layer 3	N 4 6 6 32	1 6 6 32 16	1 1 1 1 1
layer 4	N 4 1 1 16	4 1 1 16 2	1 1 1 1 1
layer 5	N 1 1 1 2		
flatten layer	N 2		

Table 2. Architecture of 3-D FCN. 'Valid' padding [22] and 3-D convolution are used in the model, N is the batch size

trained to predict if the 6-th frame of 10 frames is a shot boundary or not, if I change the parameter setting of 3-D FCN, it can also accept input of other length, e.g. input having 20 frames, FCN will predict if frame 6 to 16 are shot boundaries.

At the prediction stage, a snippet can only be predicted as shot boundary or not, I use the central frame to represent the snippet and find the corresponding day of year of that frame, as a result, there are a set of dates, I use the k-means clustering [23] to group these dates into four classes. After discarding the earliest and latest date in each class, the averaged date of rest date is the predicted transition date for each class.

3.3. Siamese Network

In the CNN-based classification, I have spent much time in preprocessing and data augmentation to make the class size balanced, is it possible to use limited unbalanced class to learn the representation of dataset and estimate the transition dates? The answer is using siamese network [24][25][26], aforetioned CNN needs to ensure which class each sample belongs to, however, when number of samples is few or there are too many classes, conventional CNN-based classification does not work well because it can not learn a good representation from few samples. Siamese network learns a similarity metric between during training, and compare or match training samples to test samples based on this learned similarity metric. Siamese network is suitable for scenarios when there are too many

classes or too few samples per class. In a siamese network

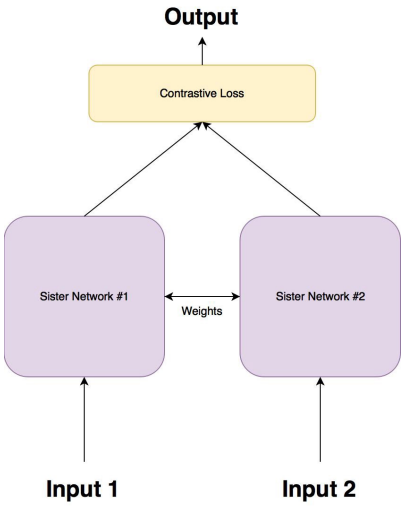

Figure 3. Architecture of siamese network [27], two sister networks share the same weights.

there are two identical sister networks, each taking one of a pair of images, the last layers of two neural networks are fed to contrastive loss function, which is the most special componet of the siamese network, instead of classifying input to exact class, contrastive loss function enables siamese network to measure how different a pair of images are.

$$D = \sqrt{\sum_j^n (f(I_1^j) - f(I_2^j))^2} \tag{9}$$

$$Loss = \frac{1}{2} y D^2 + \frac{1}{2}(1-y)(max\{0, m - D\})^2 \tag{10}$$

In Equation 9, f is the function performed by sister network, D is the euclidean distance between the two outputs of a pair of inputs. Equation 10 is originally invented by Yann leCun *et al.* in [26], $y = 1$ if a pair of inputs are from same class, otherwise $y = 0$. When $y = 1$, the two inputs are similar, if their euclidean distance is large, then the contrastive loss will also be large; when $y = 0$, $Loss = \frac{1}{2}(max\{0, m - D\})^2$, this is a hinge loss function [28], m is the margin value, if the euclidean distance of outputs from a actually dissimilar pair is beyond this margin, the hinge loss is zero and does not contribute to the contrastive loss, because I only want to optimize the network based on the pairs that are actually dissimilar but the network thinks they are fairly similar.

A CNN plays the role as sister network , which outputs a vector of shape [N 256] where N is the batch size, this vector is fed to contrastive loss function to calculate the similarity of the input pair.

In siamese network there are only the similarity between samples, to predict the transition date, I borrow the idea of image retrieval [29]. Let us call the images from training dataset taken in transition date the transition images hereafter, I feed all test images and transition images to the trained model and get output features of them respectively, then average the output features of transition images to get the transition vector. By calculating the euclidean distance vector between transition vector and output features of test images, I can therefore find the most similar n test images to the transition images, the n is set as 10. After discarding the earliest and latest date of n test images, the averaged date of rest date is the predicted transition date.

3.4. Regression

Besides neural networks mentioned above, I also build a regression network that direcly uses the day of year of the input image as label. The regression network is built by removing the softmax layer from CNN and setting the number of class as one, the rest part of regression network is same with prementioned CNN. The output of regression network is not the class number anymore, but a scalar S, the label I feed to regression network is the corresponding day of year (called D hereafter) of the input image, the regression network is optimized by minimizing the difference between D and S.

$$L_\sigma(a) = \begin{cases} \frac{1}{2}a^2 & \text{if} |a| < \sigma \\ \sigma(|a| - \frac{1}{2}\sigma) & \text{otherwise} \end{cases} \tag{11}$$

Regression problem usually utilizes huber loss function which is defined in equation 11 [30]. σ is a parameter to be set in huber loss, a is difference between output and ground truth, this function is quadratic for small values of a, and linear for large values, compared with least square loss, huber loss lower the penalty for outliers, make our regression network more robust to outliers. The prediction stage of regression network is similar to that of siamese network, each test image get an ouput from trained model in the format of day of year. I find those test images whose outputs are identical or close to transition dates on training dataset, the day of year of those test images are the predicted transition dates of test dataset.

4. Experiments

The objective of my experiments is to find a more accurate algorithm to detect the key phenological transition dates, therefore, I use the human-annotated transition date as baseline, and calculate the gap of days between the transition date detected by algorithms and transition date annotated by human. To evaluate the performance of these algorithms, the smaller the gap is, the better the algorithm is.

4.1. Data used

The images in PhenCam dataset usually not only contain the vegetation but lane or telephone pole, etc., to acquire the region of interest (ROI), images on every site have corresponding mask images, Figure4 and Figure 5 is an example image and corresponding mask image.

Our research data comes from four sites in PheneCam: (1) Harvard; (2) Bartlettir; (3) Queens; 4) Umichbiological. The images of the above sites have two advantages over images from other sites, first: their images have higher resolution; second, in the images from above four sites, the percentage of forest area of the whole image is higher than those from other sites. These two advantages make it possible to extract more representative information from images. For different methods, the format of input data is different, I will introduce them in experiments settings.

I study the phenology of above sites on year 2008, 2009

Figure 4. An example image of umichbiological site in 2010, PhenoCam.

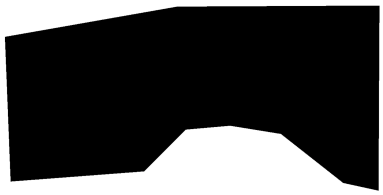

Figure 5. The mask image of site umichbiological, black area is Region of Interest(ROI).

and 2010. For site Harvard, Queens and Umichbiological, images are taken every 30 minutes every day in year 2008, 2009 and 2010, images are taken every one hour at site Bartlettir in year 2008, 2009 and 2010, to keep the high quality of images, I only collect images during 10am to 15pm, so there are 10 images every day for site Harvard, Queens and Umichbiological, 5 images for site Bartlettir.

There are 4 events: (1) majority of trees starts leafing out; (2) leaf area reaches maximum; (3) leaves start to change color in fall; (4) majority of trees lost their leaves. I use the label presented by literature [9]. For every event in each year, six human observers look through data in the year and use a common protocal to annotate the day when the event happens in that year as label. When using labels, to reduce inter-observer variability in visually assessed dates, the earliest and latest annotations of each event are discarded. Data in 2008 and 2009 is used as training data and validation data respectively, data in 2010 is regarded as test data. The models are evaluated on each site data separately.

4.2. Experiments settings

To evaluate the Generalized Sigmoid model, I use the generalized sigmoid function written in matlab[31] by Steve Klosterman to fit the GCC index of year 2010 provided by PhenoCam, the code is available at https://github.com/klostest/PhenoCamAnalysis, the code also contains the function calculating the local extrmas, thus, the returned value is the transition date in format day of year.

The architecture of CNN is shown in Section 3, the input to CNN-based classification model is a batch of RGB images with shape [N 256 256 3], I set N as 64. After the data augmentation, there are about 600 images per class for site Queens, Harvard and Umichbiological respectively, 300 images per class for site Bartlettir. I choose entropy loss function and Adam Optimizer to minimize the loss which are common setting in CNN classification tasks. The condition to stop the training process is when the gap of days between validation data and human-annotated data has reached the minimum.

The input to 3-D FCN is a video snippet, different from other model having image size of [256 256 3], the image size in 3-D FCN is [64 64 3] as 3-D convolution needs more computation, a large image size would make training extremely slow. To cover as much temporal information as possible, I divide the whole dataset into multiple snippets of 10 frames, with an overlap of 5 frames for non-transition frame and overlap of 1 frame for transition frame, there are about 3000 frames for site Queens, Harvard and Umichbiological respectively, 1500 frames for site Bartlettir, thus, there are about 1500 snippets and 40 snippets of them are annotated as transition, for site Queens, Harvard and Umichbiological respectively, 700 snippets for site Bartlettir, and 20 snippets of them are annotated as transition, to make the training data more balanced, I randomly discard half of those non-transition snippets, the rest snippets are fed to 3-D FCN.

The input to siamese network is a pair of images with shape [2*N 256 256 3] where N is batch size. The expanded labels are also used for siamese network which means in training data the former and later 5 days are also regarded as transition date. There is no need to feed images of noise class to siamese network because the network can learn the dissimilarity between four classes representating four events. After excluding the noise class there are about 400

images for site Queens, Umichbiological and Harvard and 200 images totally for site Bartlettir, these images are randomly paired and the label of pair depends on two images have same class or not, I set the number of pairs is 3000, these pairs are fed to siamese network batch by batch, the batch size N is 64.
The setting for regression network is almost same with that of CNN, the difference is there is no sofmax layer and the number of class is set 1, thus the regression network can be trained to predict the day of year of test images.

4.3. Evaluations

Let us denote GS as generalized sigmoid algorithm, CN as CNN-based classification, SN as siamese network and RN as regression network in the following table.

Method	event 1	event 2	event 3	event 4
GS	2	17	8	22
CN	2	16	2	1
SN	3	20	11	9
RN	11	26	15	19
3D FCN	25	39	60	52

Table 3. Harvard Site 2010: Absolute difference to human-recognized label.

Method	event 1	event 2	event 3	event 4
GS	24	7	2	23
CN	5	5	2	4
SN	7	10	4	8
RN	5	14	13	20
3D FCN	45	42	23	70

Table 4. Umichbiological Site 2010: Absolute difference to human-recognized label.

Method	event 1	event 2	event 3	event 4
GS	1	11	8	18
CN	0	8	3	5
SN	4	10	4	8
RN	3	12	17	27
3D FCN	41	52	12	24

Table 5. Queens Site 2010: Absolute difference to human-recognized label.

Tabel 3, 4, 5 and 6 are the evaulations of four sites. For site harvard, umichbiological and queens, the performance of CNN-based classification is best, it gives the predicted days of event 1 (majority of trees starts leafing out), event 3 (trees first start to change color in the fall), event 4 (majority of trees had lost all leaves) that are close to human

Method	event 1	event 2	event 3	event 4
GS	1	15	11	23
CN	3	11	6	14
SN	2	9	5	10
RN	10	19	16	24
3D FCN	10	37	66	29

Table 6. Bartlettir Site 2010: Absolute difference to human-recognized label.

recognized, for event 2 (green leaf area is maximum), predicted days of all methods have relatively large gap with human annotated results, I conclude that once the green leaf area reaches maximum the area has a decreasing period and reaches maximum again in summer, which makes model hard to learn this process, I also find that observations of six human observers have relatively large variance in recognizing this event. Siamese network is more accurate than other models in detecting the transition dates on site Bartlettir, this is fair because data of site bartlettir only have half the size of other three sites, performance of siamese network decrease more slightly than that of CNN-based classification model when sample size is reduced. The performance of Regression model is not as accurate as that of classification model and siamese net, the 3-D FCN works poor. I think the poor performance of regression model and 3-D FCN is because temporal information has a strong pattern only during growing seasons (April-June, Sep to Nov). The temporal clues out of vegetation's growing season are weak, which makes it hard for regression and 3-D FCN to learn the temporal pattern, another possible resaon is, in common video dataset, the shot boundaries between frames are highly distinguishable, however, in vegetation dataset, the change of frames is slow and not obvious over time, which means it is hard to capture the temporal pattern.
To summarise the performance of different models on detecting the transition dates of vegetations, CNN-based classification model works best, siamese network and generalized sigmoid model have roughly same performance, they are both good, performance of regression model is not bad but 3-D FCN works poor.

5. Discussion

In this thesis, I introduce few deep learning approaches and how they can be applied in detecting the phenological transition dates, I also compare these deep learning methods with traditional sigmoid-based methods, and find CNN-based classification can yield more accurate results.
There are also many limitations in my work, models exploiting the temporal information is of poor performance, the particularity of vegetation dataset and the inappropriate setting of model are both responsible for the poor re-

sult. Actually there is another model which has great potential to achieve success in this task, convolution neural network plus recurrent neural network (RNN), however, it takes much more time to fine-tune the CNN+RNN model, and considering models relying on temporal information have relatively poor performance in experiments, I therefore stop spending time fine tuning the CNN+RNN model, which could be a good approach to perform transition dates detection in future work. I also have to admit, it usually takes much time to train and fine tune the models to earn better performance.

The scale of dataset in my experiments is not enough to train the best model, the evaluation of performance will become more convicible if data drom more sites and more years is involved in. Integrating data from many sites and many years could be another good way to train the model.

References

[1] Ranga B Myneni, CD Keeling, Compton J Tucker, Ghassem Asrar, and Ramakrishna R Nemani. Increased plant growth in the northern high latitudes from 1981 to 1991. *Nature*, 386(6626):698, 1997.

[2] Michael A White, Peter E Thornton, and Steven W Running. A continental phenology model for monitoring vegetation responses to interannual climatic variability. *Global biogeochemical cycles*, 11(2):217–234, 1997.

[3] Mark D Schwartz. Advancing to full bloom: planning phenological research for the 21st century. *International Journal of Biometeorology*, 42(3):113–118, 1999.

[4] Christian Körner and David Basler. Phenology under global warming. *Science*, 327(5972):1461–1462, 2010.

[5] Xiaoyang Zhang, Mark A Friedl, Crystal B Schaaf, Alan H Strahler, John CF Hodges, Feng Gao, Bradley C Reed, and Alfredo Huete. Monitoring vegetation phenology using modis. *Remote sensing of environment*, 84(3):471–475, 2003.

[6] Oliver Sonnentag, Koen Hufkens, Cory Teshera-Sterne, Adam M Young, Mark Friedl, Bobby H Braswell, Thomas Milliman, John OKeefe, and Andrew D Richardson. Digital repeat photography for phenological research in forest ecosystems. *Agricultural and Forest Meteorology*, 152:159–177, 2012.

[7] Liang Liang, Mark D Schwartz, and Songlin Fei. Validating satellite phenology through intensive ground observation and landscape scaling in a mixed seasonal forest. *Remote Sensing of Environment*, 115(1):143–157, 2011.

[8] Andrew J Elmore, Steven M Guinn, Burke J Minsley, and Andrew D Richardson. Landscape controls on the timing of spring, autumn, and growing season length in mid-atlantic forests. *Global Change Biology*, 18(2):656–674, 2012.

[9] Stephen Klosterman, Koen Hufkens, JM Gray, E Melaas, O Sonnentag, I Lavine, L Mitchell, R Norman, MA Friedl, and Andrew Richardson. Evaluating remote sensing of deciduous forest phenology at multiple spatial scales using phenocam imagery. 2014.

[10] Morris Kline. *Calculus: an intuitive and physical approach*. Courier Corporation, 1998.

[11] Sarah Taghavi Namin, Mohammad Esmaeilzadeh, Mohammad Najafi, Tim B Brown, and Justin O Borevitz. Deep phenotyping: deep learning for temporal phenotype/genotype classification. *Plant methods*, 14(1):66, 2018.

[12] Hulya Yalcin. Plant phenology recognition using deep learning: Deep-pheno. In *Agro-Geoinformatics, 2017 6th International Conference on*, pages 1–5. IEEE, 2017.

[13] Kentaro Kuwata and Ryosuke Shibasaki. Estimating crop yields with deep learning and remotely sensed data. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 858–861. IEEE, 2015.

[14] Michael Gygli. Ridiculously fast shot boundary detection with fully convolutional neural networks. *arXiv preprint arXiv:1705.08214*, 2017.

[15] Charles Poynton. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Joseph Lin Chu and Adam Krzyżak. Analysis of feature maps selection in supervised learning using convolutional neural networks. In *Canadian Conference on Artificial Intelligence*, pages 59–70. Springer, 2014.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[22] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[23] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[24] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[25] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006.

[27] illustration of siamese network. `https://hackernoon.com`.

[28] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[29] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[30] Peter J Huber et al. Robust estimation of a location parameter. *The annals of mathematical statistics*, 35(1):73–101, 1964.

[31] Users Guide Matlab. The mathworks. *Inc., Natick, MA*, 1992, 1760.