# Detection of Sun Direction by Convolutional Neural Network Using Shadow from Images


By

**Itthipat Prangthong 5905158**

**Advisor**

**Dr.Petchara Pattarakijwanich**

**Faculty of Science Department of Physics**
**Mahidol university**

# Contents

# Acknowledgement

I would like to thank Dr.Petchara Pattarakijwanich as an advisor for this project. All advice and suggestion are helpful during this project work, which keep my motivation in developing this project. All the staff and Friends in the Physic department are also important in giving suggestion. Finally, special thanks to many user-friendly python package developers such as numpy, OpenCV or scikit-learn in assisting calculation and processing data.

Itthipat Prangthong 5905158

# Chapter 1
# Introduction

## 1.1 Abstract

Convolutional Neural Network has been used to predict data including interpretation from image. By using SBU (Stony Brook University) Shadow Dataset of images. We are able to predict shadow region from using K-mean clustering from SV data plots, then use the shadow mask image to train the network to predict up to 8 directions in cardinal directions and 1 for uncertain direction. Our result achieves 11% accuracy and 6 validation loss from test data, which occur form overfitting.

## 1.2 Introduction

Shadow are commonly found in everyday image such as people or aerial photo. When an opaque object blocked light from light source, a shadow is created. The sun direction can provide usable information as well as shadow region. In visual odometry, the sun is an example of known directional landmark, which have been used in robot sensor to assist with the determination of direction[1].

Convolutional Neural Network have been used in multiple classification in computer vision. By using feature extraction of image, the network can learn specific detail to predict result. In this project we use CNN to predict sun direction from shadow mask image created by K-mean clustering method in SV data plots with gap statistic to determine K.

The images are provided by SBU (Stony Brook University) Shadow Dataset [2] which is a dataset of images containing shadow. In this project we used a total of 3,715 images, split to 3,000 for training and 715 for testing. The CNN framework is based on Keras and Tensorflow using Python. Opencv2, scikit-learn, matplotlib and pandas python package were also use in processing image.

## 1.3 Objective

1.Shadow region detection from images
2.Sun direction prediction from shadow mask images

## 1.4 Expected Result

1.Acceptable clustering with k-mean method to detect shadow region
2.Rough Estimation of sun direction from shadow mask images

## 1.5 Project Duration Table

| Task | 2019 | | | | 2020 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sep | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May |
| 1.Study and preparing the required method | ■ | ■ | ■ | | | | | | |
| 2.Coding shadow detection | | | ■ | ■ | ■ | ■ | | | |
| 3.Coding CNN model | | | | | ■ | ■ | ■ | | |
| 4.Code testing with dataset | | | | | | | ■ | ■ | ■ |
| 5.Evaluate result | | | | | | | | | ■ |

# Chapter 2
# Theory and Methodology

In this project we use multiple theories and methods for creating CNN to predict sun direction as listed below.

1.Color space and shadow

2.K-mean clustering and gap statistic

3.Convolutional neural network for image

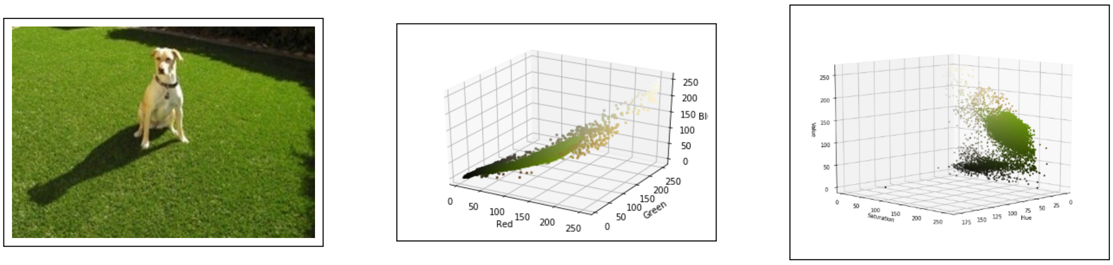
## 2.1 Color space and shadow

Computer images compose of 2d array pixels of different values and channel, typically in RGB color space. Difference feature on image have difference of these values, so we can use this to find range of value that is shadow region. But due to the nature of RGB color space, differentiating between shadow and non-shadow can be difficult. HSV color space can provide more discrete value shown in [Figure 2.1]. In [Table 2.1] shadow region has only lower saturation and value (brightness) but not hue. We select SV color space to find shadow region and discard hue value because uncorrelation between shadow and non-shadow. This method reduces the complexity and dimensions of dataset into 2d instead.


Figure 2.1 RGB and HSV color space

ref: 1.https://en.wikipedia.org/wiki/RGB_color_model

2.https://en.wikipedia.org/wiki/HSL_and_HSV




(a) (b) (c)

Figure 2.2 An example image of (a).original image (b).RGB data plot and (c).HSV data plot

| | H | S | V |
|---|---|---|---|
| Shadow | 56 | 26 | 46 |
| Non-shadow | 54 | 84 | 112 |

Table 2.1 Comparing HSV value between shadow and non-shadow in grass area from figure 2

## 2.2 K-mean and gap statistic

K-mean clustering is an unsupervised machine learning method. Parameter K is input cluster centroid of each unique group. Given n number of clusters C with $\mu_j$ be the centroid. The algorithm tries to group data $x_i$ into K groups by minimize within-cluster sum-of-squares function with iterative method (Eq.2.1) [3].

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(\|x_i - \mu_j\|^2) \tag{2.1}$$

K-mean clustering method can be helpful to find correlation or pattern. In this case we use to find shadow region in SV data plot. By selecting range around lowest value (Y axis) centroid, we are able to create shadow mask that mark the shadow region in original color image.

The gap statistic [4] is also used to find optimal value of K. Since K number of clusters depend on different color in image. Gap statistic is defined by difference of dispersion of original dataset and dispersion of reference dataset (Eq.2.2, 2.3, 2.4). As a result, we use the highest gap statistic within set range of K cluster to determine the optimal K value for k-mean clustering.

$$D_r = \sum_{x,x' \in C_r} d_{xx'} \tag{2.2}$$

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{2.3}$$

$D_r$ is the sum of pairwise distance $d_{xx'}$ or cluster $r$.

$$Gap(k) = \frac{1}{B} \sum_{b=1}^{B} \log(W^*_{kb}) - \log(W_k) \tag{2.4}$$

With B being reference dataset with random uniform distribution.

## 2.3 Convolutional neural network for image

Convolutional neural network is a type of artificial neural networks primarily use in image pattern recognition. CNN typically contain 3 types of layers: convolution, pooling, and fully connected. These layers are mathematical operations to extract feature from input, in this case shadow mask and give probability of listed output (prediction) [5].

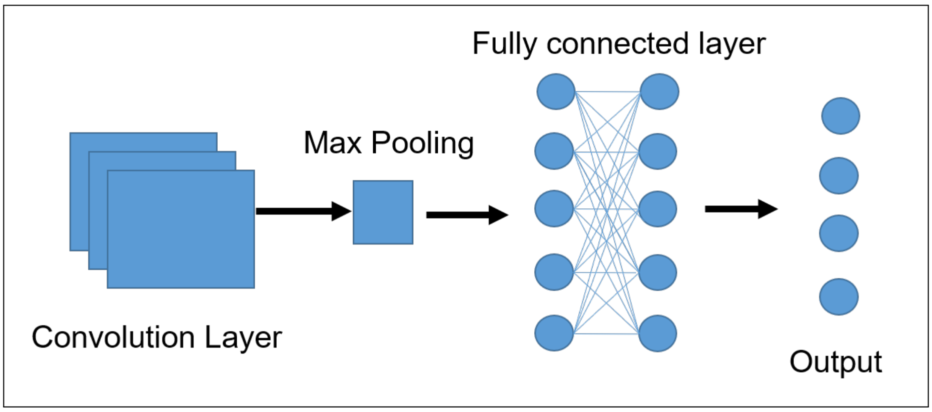



Figure 2.3 An example of basic CNN structure.

# Chapter 3
# Project Procedure

This section consists of list of procedure in the following order.

1.Data Preparation

2.Creating shadow mask by K-mean clustering

3.Labelling and Training CNN with acquired shadow mask

## 3.1 Data Preparation

We use SBU Shadow Dataset for total of 3,715 images which consisted of a variety of scenario [Figure 3.1], divided to 3,000 for training and 715 for testing in approximately 80:20 ratio. All images have also been resized to 50x50 pixels.

Figure 3.1 An example of original image dataset.

## 3.2 Creating shadow mask by K-mean clustering

K-mean clustering was done by scikit-learn package to find the centroid of shadow region with appropriate K value determined from gap statistic. We selected a range of $\pm$ 25 saturation and $\pm$ 25 value around the centroid creating a box shape region. The data in this region can be considered shadow region pixel in image. Then by selecting data in selected range, shadow mask can be created in a form of binary shadow silhouette shape.

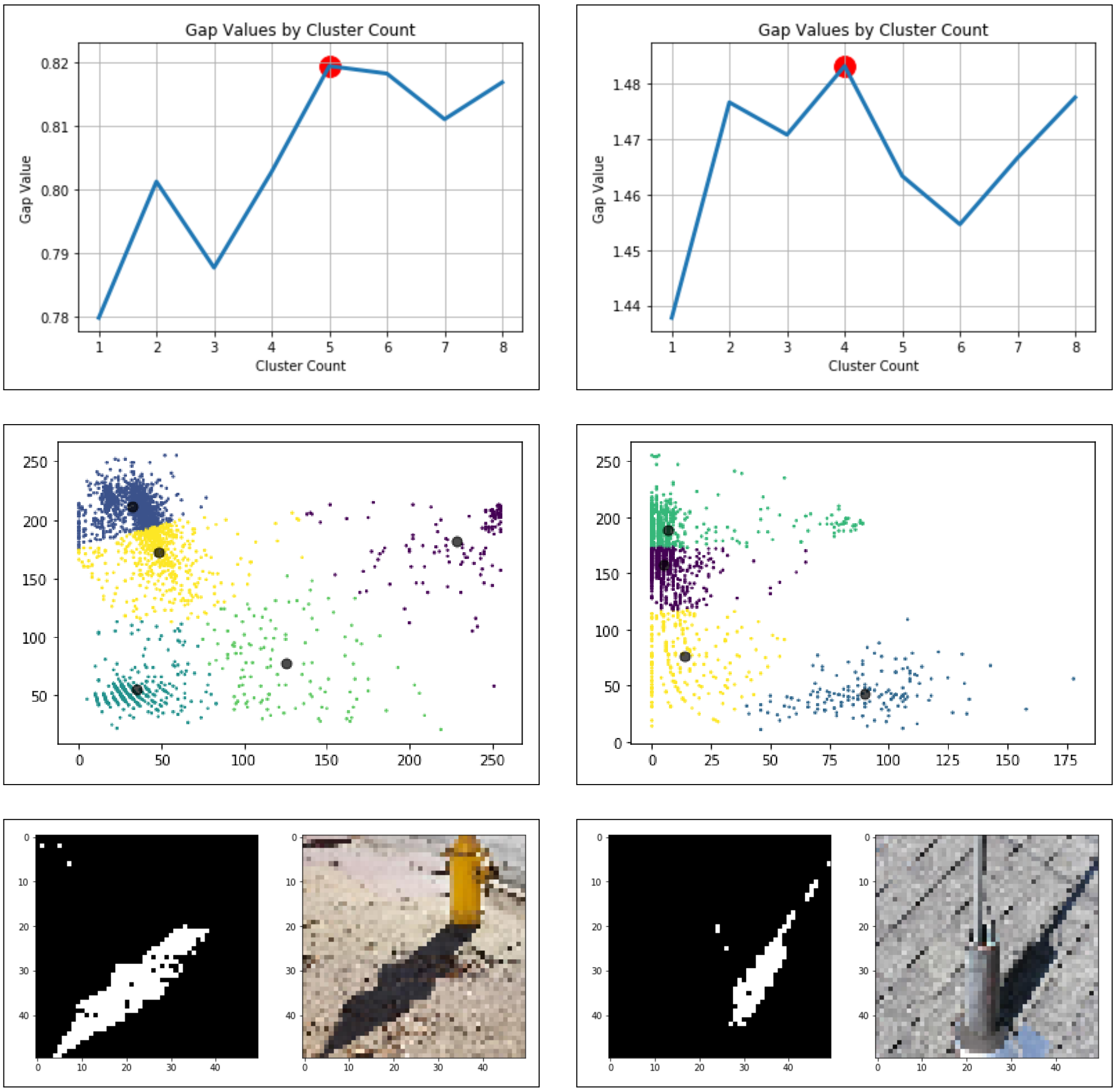



Figure 3.2 Shadow mask procedure each left and right image. *Top row* : Graphs shown optimal K number for K-mean. *Middle row* : Clustering data into K number of groups with centroid. *Bottom row* : By selecting area around lowest value centroid, a shadow mask is created with highlighted shadow region.

## 3.3 Labeling and Training CNN with acquired shadow mask

We classify shadow direction into fixed 8 cardinal directions independent of the ground in image. This means sun direction is opposite of shadow direction, so we chose shadow direction for easier procedure. Labeling each direction in numbers 1 to 8 and also label 0 for image that has uncertain shadow direction [Figure 3.3]. Then we use the shadow mask and label of each image to train CNN using following model in pic [Figure 3.4]. Result in model 1.Training accuracy, 2.Training loss, 3.Validation accuracy, 4.Validation loss. Accuracy show the percentage of same (or correct) prediction given by the label and loss show a value of loss function.

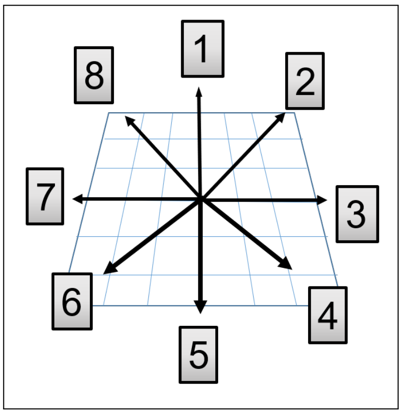



Figure 3.3 Label directions of 8 direction. For example, the right image would have label 2

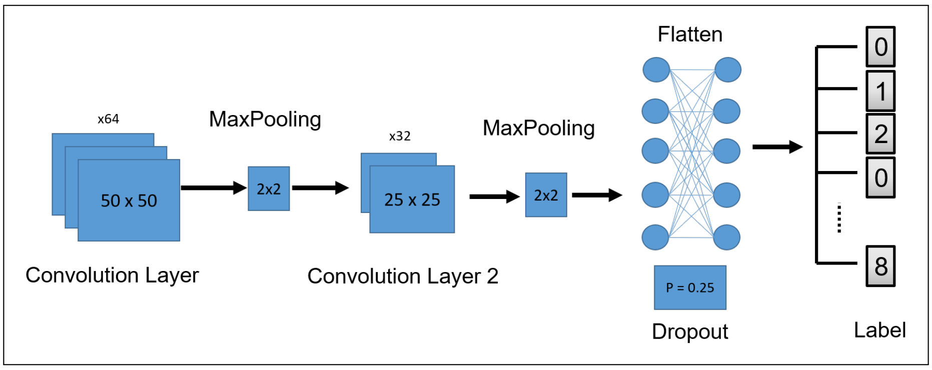



Figure 3.4 Our current CNN model structure.

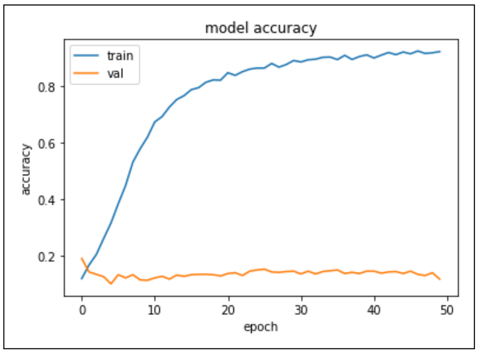
# Chapter 4
# Project Result

## 4.1 Model Result

The result in training CNN for 50 epochs gives 4 value and following [Figure 4.1].

- Training accuracy : 92.10 %
- Training loss : 0.2313
- Validation accuracy : 11.89 %
- Validation loss : 5.7105

Above values are shown at 50 epochs.



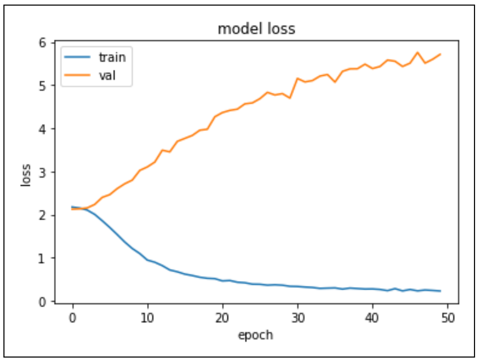




Figure 4.1 Plots of model accuracy and loss for 50 epochs of shadow direction prediction.

The result indicated that increasing validation loss mean the model is overfitting, also shown by unchanging low validation accuracy. Overfitting in CNN is commonly caused by high number of parameters in data in this case giving shadow silhouette shape to predict shadow direction [5]. We also tried the same model with 2 benchmark datasets which is MNIST handwritten digit dataset and MNIST fashion dataset to check if the model is usable [Figure 4.2, 4.3].

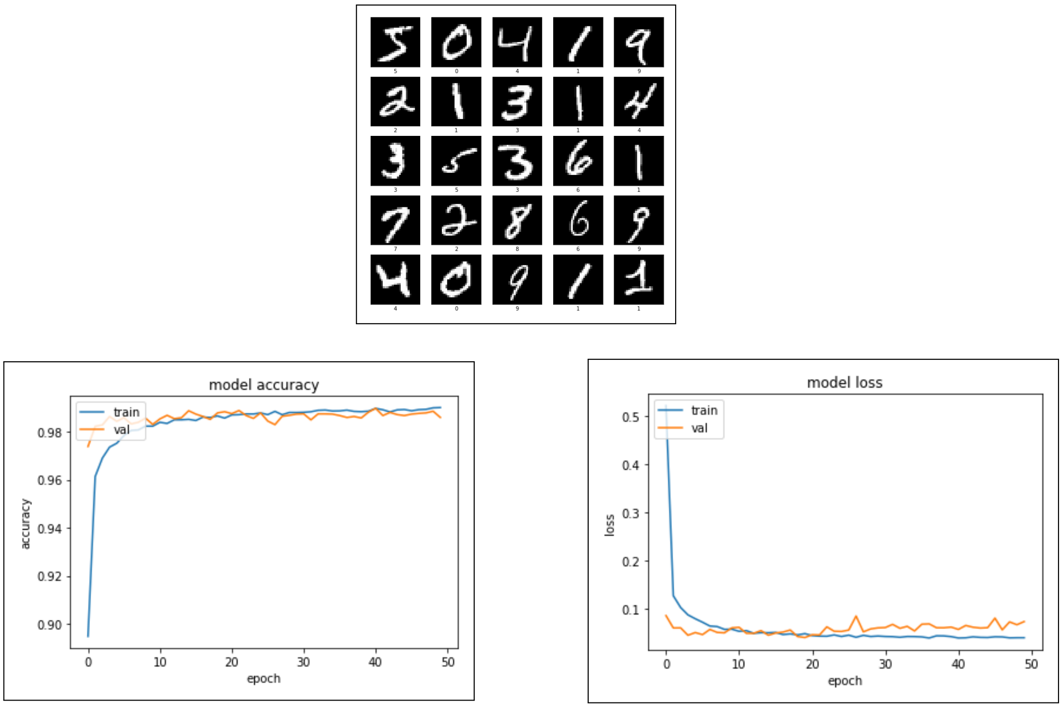



Figure 4.2 Plot of model accuracy and loss for 50 epochs of MNIST handwritten digit dataset with example above.

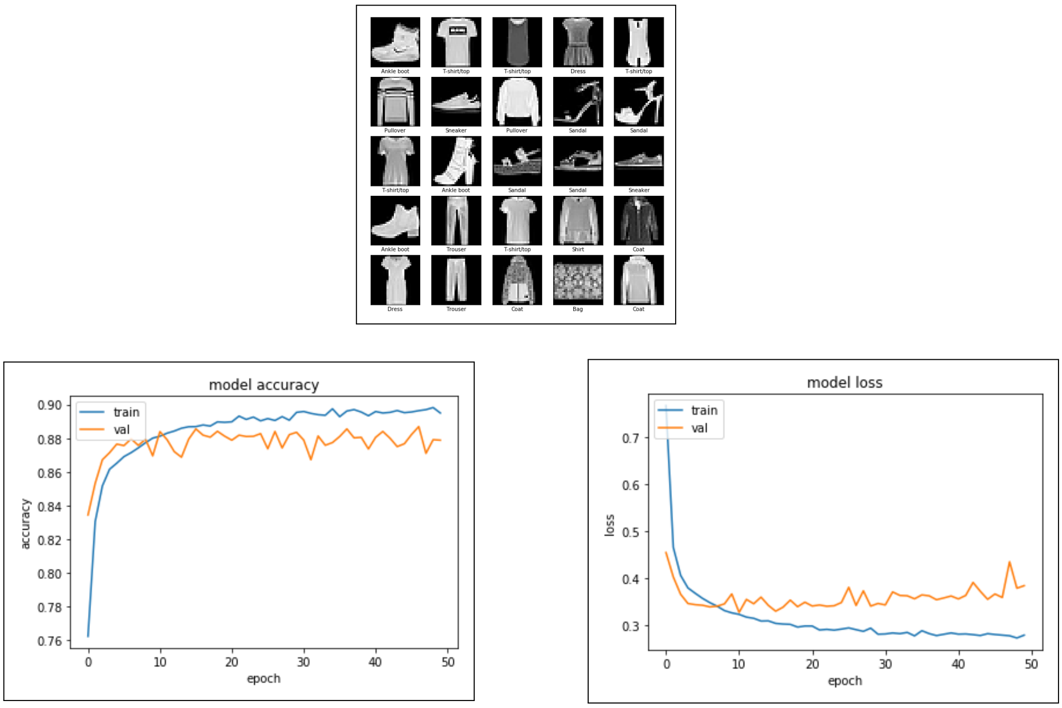



Figure 4.3 Plot of model accuracy and loss for 50 epochs of MNIST fashion dataset with example above.

The result show acceptable result in both datasets. That conclude that this model is not usable with current dataset.

## 4.2 Future work

The main problem is overfitting. A change in model layout or better dataset preprocessing could provide better result. For example, contrast adjustment or background/foreground subtraction. More dataset is also a viable option comparing our current 3,715 images to 60,000 in benchmark dataset. Using other form of image type than shadow mask can be possible due to shadow shape complexity.

# Chapter 5
# Conclusion

We have presented a method to detect sun direction from shadow mask using Convolutional Neural Network. Using k-mean clustering to create shadow mask and then processed with CNN to predict shadow directions, which can infer to sun directions. Our model result appears to overfit the dataset, but not the benchmark dataset. This problem could be resolve by adding preprocessing step of original image to be more suitable for the current model.

# Reference


[1] Valentin Peretroukhin, Lee Clement, and Jonathan Kelly. "Reducing drift in visual odometry by inferring sun direction using a Bayesian Convolutional Neural Network". In: May 2017, pp. 2035–2042. DOI: 10.1109/ICRA.2017.7989235.

[2] Tomas Yago et al. "Large-Scale Training of Shadow Detectors with Noisily-Annotated Shadow Examples". In: vol. 9910. Oct. 2016, pp. 816–832. ISBN: 978-3-319-46465-7. DOI: 10.1007/978-3-319-46466-4_49.

[3] *Scikit-learn K-mean user guide*. URL: https://scikit-learn.org/stable/modules/clustering.html#k-means.

[4] Robert Tibshirani, Guenther Walther, and Trevor Hastie. "Estimating the Number of Clusters in a Data Set Via the Gap Statistic". In: *Journal of the Royal Statistical Society Series B* 63 (Feb. 2001), pp. 411–423. DOI: 10.1111/1467-9868.00293.

[5] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". In: *ArXiv e-prints* (Nov. 2015).