

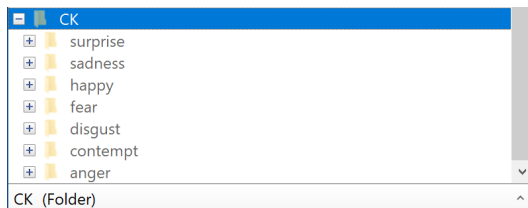
Emotion Recognition using CNN

1. ใช้ dataset CK+ (Extended Cohn-Kanade dataset) <https://doi.org/10.1109/CVPRW.2010.5543262>

The Extended Cohn-Kanade (CK+) dataset contains 593 video sequences from a total of 123 different subjects, ranging from 18 to 50 years of age with a variety of genders and heritage. Each video shows a facial shift from the neutral expression to a targeted peak expression, recorded at 30 frames per second (FPS) with a resolution of either 640x490 or 640x480 pixels. Out of these videos, 327 are labelled with one of seven expression classes: anger, contempt, disgust, fear, happiness, sadness, and surprise. The CK+ database is widely regarded as the most extensively used laboratory-controlled facial expression classification database available, and is used in the majority of facial expression classification methods.



ให้สร้าง datastore ในโฟลเดอร์ Matlab



2. ใช้คำสั่งสร้าง datastore

```
imds =  
imageDatastore('CK','IncludeSubfolders',true,'LabelSource','foldernames');
```

3. แบ่ง data ทำ training และ validate

```
[imds_train, imds_valid]=splitEachLabel(imds,.8,'randomize');  
CountLabel = imds.countEachLabel  
aa=read(imds);  
size(aa)
```

4. สร้าง CNN layers ประกอบด้วย

- Image Input Layer
- Convolutional Layer
- Batch Normalization Layer
- ReLU Layer
- Max-Pooling Layer
- Fully Connected Layer
- Softmax Layer
- Classification Layer

```
layers = [  
    imageInputLayer([48 48 1])  
    convolution2dLayer(3,8,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
    maxPooling2dLayer(2,'Stride',2)  
    convolution2dLayer(3,16,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
    maxPooling2dLayer(2,'Stride',2)  
    convolution2dLayer(3,32,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
    fullyConnectedLayer(7)  
    softmaxLayer  
    classificationLayer];
```

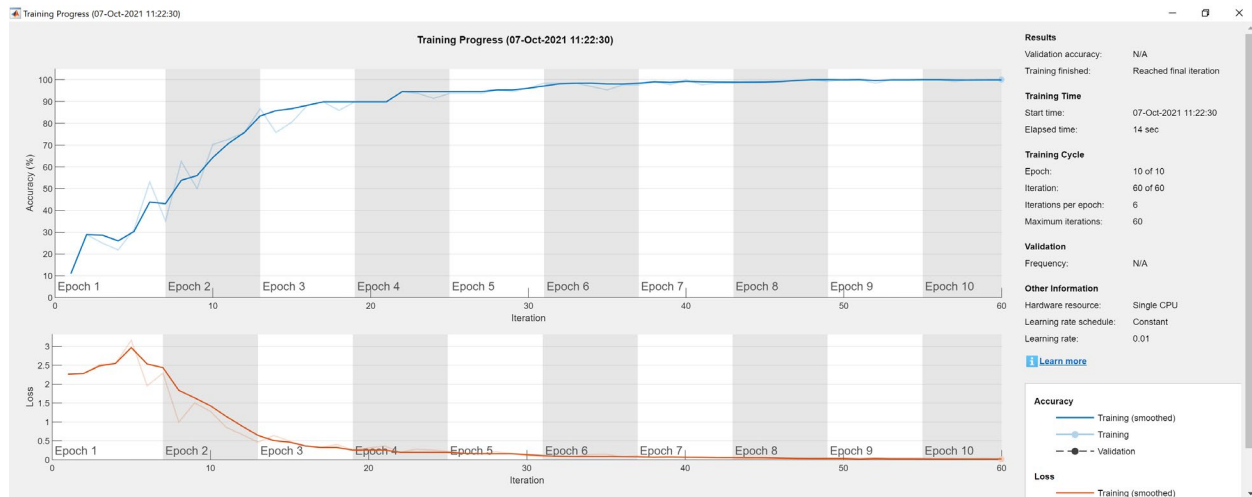
5. กำหนดค่าพารามิเตอร์สำหรับ training

```
options = trainingOptions('sgdm', ...  
    'InitialLearnRate',0.01, ...  
    'MaxEpochs',10, ...
```

```
'Shuffle','every-epoch', ...
'ValidationFrequency',10, ...
'Verbose',false, ...
'Plots','training-progress');
```

6. Train model

```
convnet = trainNetwork(imdstrain, layers, options);
```



7. คำนวณความถูกต้องของโมเดลด้วย validate dataset

```
YPred = classify(convnet, imdsvalid);
YValidation = imdsvalid.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation);
```

8. สร้าง confusion matrix

```
plotconfusion(YValidation, YPred)
```

Confusion Matrix									
Output Class	anger	26 13.3%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	96.3% 3.7%
	contempt	0 0.0%	11 5.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	91.7% 8.3%
	disgust	0 0.0%	0 0.0%	35 17.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	fear	0 0.0%	0 0.0%	0 0.0%	14 7.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	happy	0 0.0%	0 0.0%	0 0.0%	0 0.0%	41 20.9%	0 0.0%	0 0.0%	100% 0.0%
	sadness	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 8.7%	0 0.0%	94.4% 5.6%
	surprise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 25.0%	100% 0.0%
		96.3% 3.7%	100% 0.0%	100% 0.0%	93.3% 6.7%	100% 0.0%	100% 0.0%	98.0% 2.0%	98.5% 1.5%
		Target Class							
		anger	contempt	disgust	fear	happy	sadness	surprise	

9. ทดลองอ่านไฟล์ภาพจาก data store และ classify

```
a=read(imdsvalid );
class=classify(convnet,a);
imshow(a)
title(string(class))
```

10. classify แบบ real time โดยใช้ web cam (จะต้องลง Matlab support package for USB webcam ก่อน)

```
clear camera
figure
camera = webcam(1);
while true
    im = camera.snapshot;
    picture=rgb2gray(im);
    picture = imresize(picture,[48,48]);
    label = classify(convnet, picture);
    image(im);
    title(char(label));
    drawnow;
end
```

% Take a picture
% Resize the picture
% Classify the picture
% Show the picture
% Show the label



การบ้าน

ทดลองปรับปรุง CNN เพื่อใช้กับ data set อื่นๆ เช่น

- FER2013 (Facial Expression Recognition 2013 Dataset)

<https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>



- FER+ (Face Expression Recognition Plus dataset)

<https://github.com/Microsoft/FERPlus>

