

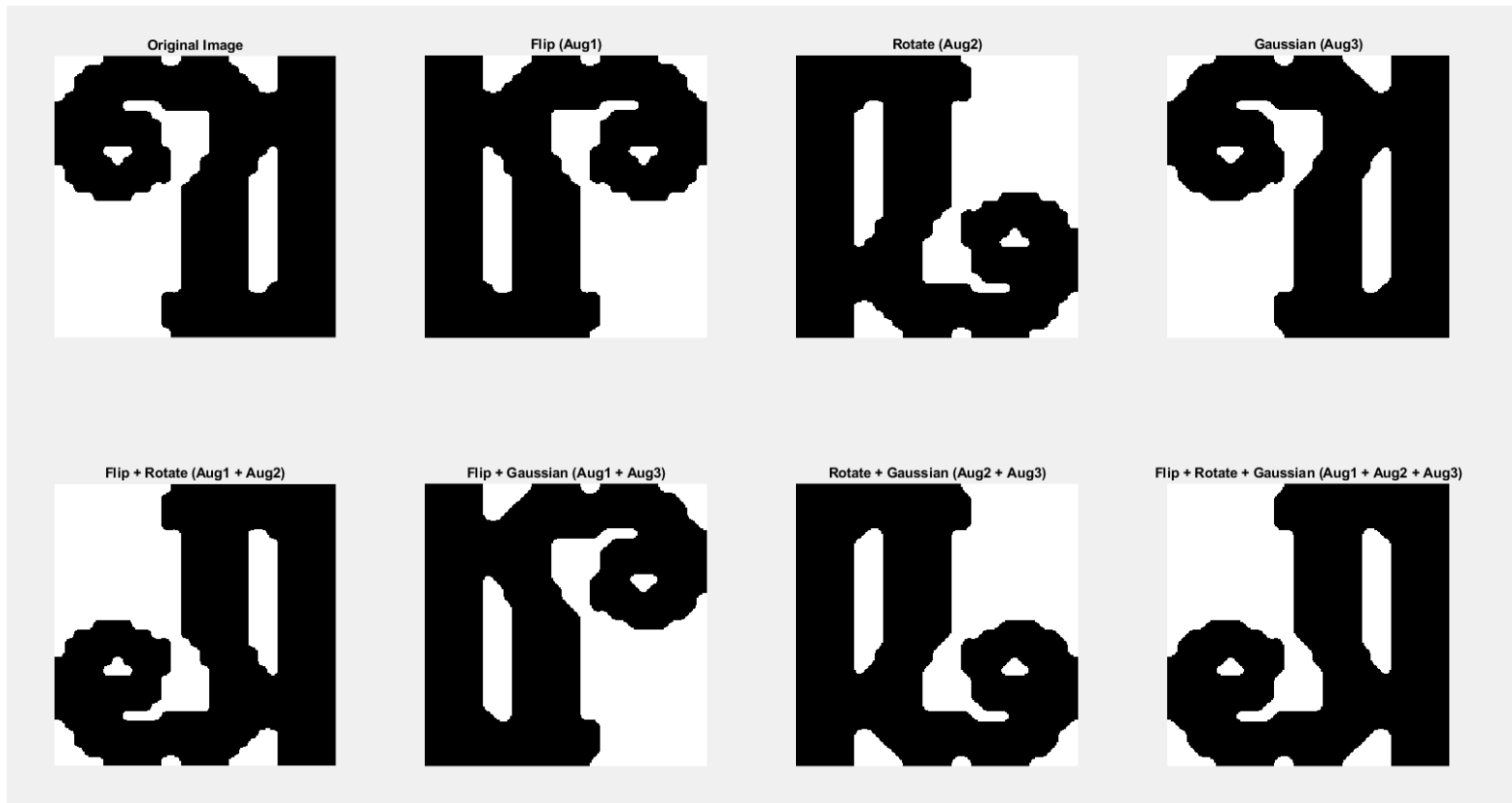
กลุ่ม : คิดไม่ออก

คำอธิบายโค้ดส่วนที่ 1: ตอนที่ 2 โคร่งข่ายประสาทแบบคอนโวลูชัน

- พวกเราทำการเลือกใช้ Augmentation ดังนี้

Augmentation ที่	วิธีการ	เหตุผลที่เลือกใช้
1	Flip ภาพ (แบบกระจก)	เนื่องจากเราคิดว่าการ Flip ภาพนั้นไม่ได้เปลี่ยนธรรมชาติของข้อมูลมากเกินไป (เหมือนเราล๊ากโกเพิ่ม Data ที่ใช้ Train เข้าไป) เพื่อเพิ่มจุดสังเกตของข้อมูลให้ดีขึ้น
2	Rotate ภาพ (180 องศา)	เนื่องจากเราคิดว่าการ Rotate ภาพนั้นไม่ได้เปลี่ยนธรรมชาติของข้อมูลมากเกินไป (เหมือนเราล๊ากโกเพิ่ม Data ที่ใช้ Train เข้าไป) เพื่อเพิ่มจุดสังเกตของข้อมูลให้ดีขึ้น
3	ใช้ Gaussian Blur	เนื่องจากบางตัวอักษรเส้นขอบมีความหยักเล็กน้อย

- ตัวอย่างภาพต้นฉบับและภาพที่ผ่านการ Augmentation รูปแบบต่างๆ



- ผลการทดลองในรูปแบบตาราง

วิธีการ	ผลการทดลอง
ไม่ใช้วิธีการใดๆเลย	93.3333%
Flip ภาพ (Augmentation 1)	98.3333%
Rotate ภาพ (Augmentation 2)	99.1667%
Gaussian Blur (Augmentation 3)	96.6667%
Flip ภาพ (Augmentation 1) + Rotate ภาพ (Augmentation 2)	99.1667%
Flip ภาพ (Augmentation 1) + Gaussian Blur (Augmentation 3)	100%
Rotate ภาพ (Augmentation 2) + Gaussian Blur (Augmentation 3)	100%
Flip ภาพ (Augmentation 1) + Rotate ภาพ (Augmentation 2) + Gaussian Blur (Augmentation 3)	100%

- คำอธิบายโค้ด

เราทำการแบ่งโค้ดออกเป็น 2 ส่วน ก็คือ

1. โค้ดในส่วน Folder Train

โดยโค้ดส่วนที่ทำการ Train จะทำการโหลดรูปจาก Folder Tr ทุกรูปมาป้อนให้กับ CNN

โดยหากไม่มีการทำ Augmentation ก็จะมีการโหลดรูปด้วย function imageDatastore ที่ใช้ ReadFcn ที่เราเขียนขึ้นมาเองที่ชื่อว่า normalReadDataStoreImage

```
function data = normalReadDataStoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = data(:,:,min(1:3, end));
    data = imresize(data,[227 227]);
end
```

โดยหากมีการทำ Augmentation ด้วยวิธีการต่างๆก็จะทำผ่าน ReadFcn ที่เราเขียนขึ้นมาเองดังนี้

Flip ภาพ (Augmentation 1)

```
function data = flipdimReadDataStoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % flip image
    data = flipdim(data,2);
    data = data(:,:,min(1:3, end));
end
```

Rotate ภาพ (Augmentation 2)

```
function data = imrotateReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % rotate image
    data = imrotate(data, 180);
    data = data(:,:,min(1:3, end));
end
```

Gaussian Blur (Augmentation 3)

```
function data = gaussianReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % Gaussian Blur
    data = imfilter(data, fspecial('gaussian',9,3),'replicate', 'conv');
    data = data(:,:,min(1:3, end));
end
```

Flip ภาพ (Augmentation 1) + Rotate ภาพ (Augmentation 2)

```
function data = flipdim_rotateReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % flip
    data = flipdim(data,2);
    % rotate
    data = imrotate(data, 180);
    data = data(:,:,min(1:3, end));
end
```

Flip ภาพ (Augmentation 1) + Gaussian Blur (Augmentation 3)

```
function data = flipdim_gaussianReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % flip
    data = flipdim(data,2);
    % gaussian
    data = imfilter(data, fspecial('gaussian',9,3),'replicate', 'conv');
    data = data(:,:,min(1:3, end));
end
```

Rotate ภาพ (Augmentation 2) + Gaussian Blur (Augmentation 3)

```
function data = rotate_gaussianReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % rotate
    data = imrotate(data, 180);
    % gaussian
    data = imfilter(data, fspecial('gaussian',9,3),'replicate', 'conv');
    data = data(:,:,min(1:3, end));
end
```

Flip ภาพ (Augmentation 1) + Rotate ภาพ (Augmentation 2) + Gaussian Blur (Augmentation 3)

```
function data = flipdim_rotate_gaussianReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = imresize(data,[227 227]);
    % flip
    data = flipdim(data,2);
    % rotate
    data = imrotate(data, 180);
    % Gaussian Blur
    data = imfilter(data, fspecial('gaussian',9,3),'replicate', 'conv');
    data = data(:,:,min(1:3, end));
end
```

โดยเราจะทำการโหลดรูปที่ผ่าน ReadFcn ต่างๆและเอามารวมกันเพื่อนำไป Train และเก็บไว้เป็นไฟล์ .mat เพื่อให้ง่ายต่อการใช้งาน

**** ทุกๆ ReadFcn จะทำการ Resize รูปภาพให้เป็นขนาด 227 x 227 เนื่องจาก Alexnet รับข้อมูลเป็นรูปภาพขนาดนี้โดยทุกๆไฟล์ที่อยู่ใน Train จะเป็นรูปแบบนี้หมด****

```

clear all, close all, clc

% load alexnet
net = alexnet;
inputSize = net.Layers(1).InputSize;

layersTransfer = net.Layers(1:end-3);
numClasses = 6;
layers = [
    layersTransfer;
    fullyConnectedLayer(numClasses, 'WeightLearnRateFactor', 20, 'BiasLearnRateFactor', 20);
    softmaxLayer;
    classificationLayer
];

% load data
imdsTr = imageDatastore(['../Tr'], 'IncludeSubfolders', true, 'FileExtensions', '.bmp', 'LabelSource', 'foldernames');
imdsTr.ReadFcn = @normalReadDatastoreImage;

% load data (with augmentation (flip))
imdsTr_new = imageDatastore(['../Tr'], 'IncludeSubfolders', true, 'FileExtensions', '.bmp', 'LabelSource', 'foldernames');
imdsTr_new.ReadFcn = @flipdimReadDatastoreImage;

% merge
imdsTr_cat = imageDatastore(cat(1, imdsTr.Files, imdsTr_new.Files));
imdsTr_cat.ReadFcn = @normalReadDatastoreImage;
imdsTr_cat.Labels = cat(1, imdsTr.Labels, imdsTr_new.Labels);

imdsTr = imdsTr_cat;

options = trainingOptions('sgdm', 'MiniBatchSize', 10, 'MaxEpochs', 6, 'InitialLearnRate', ...
    1e-4, 'Shuffle', 'every-epoch', 'ValidationData', imdsTr, 'ValidationFrequency', 3, 'Verbose', false, 'Plots', 'training-progress');

netTransfer = trainNetwork(imdsTr, layers, options);
save('../augmentation_1.mat', 'netTransfer')

```

ตัวอย่างโค้ดไฟล์ augmentation_1.m (ใช้วิธี Flip ภาพ)

**** โดยทุกๆไฟล์ที่เป็นการ Train จะมีรูปแบบโค้ดเหมือนกันต่างกันแค่ตรง ReadFcn ****

2. ส่วนที่ทำการ load model ที่ Train แล้วมาวัดค่าผลการทดลอง

```
clear all, close all, clc

% load training set

load no_augmentation.mat
%load augmentation_1.mat
%load augmentation_2.mat
%load augmentation_1_2.mat
%load augmentation_3.mat
%load augmentation_1_3.mat
%load augmentation_2_3.mat
%load augmentation_1_2_3.mat

% load test
imdsTs = imageDatastore(['Tr'], 'IncludeSubfolders', true, 'FileExtensions', '.bmp', 'LabelSource', 'foldernames');
imdsTs.ReadFcn = @customReadDatastoreImage;

% ทำการ predict
YPred = predict(netTransfer, imdsTs);
[GN, ~, idx] = unique(imdsTs.Labels);

score = 0;

% loop เช็คว่า predict ได้ถูกที่รูปภาพ
for i = 1:numel(idx)
    id = find(max(YPred(i,:)) == YPred(i,:));
    score = score + (id == (idx(i)));
end

% คัดออกมาเป็น %
percent = score/numel(idx) * 100

% function ที่เอาไว้อ่านรูปภาพ
function data = customReadDatastoreImage(filename)
    % code from default function:
    onState = warning('off', 'backtrace');
    c = onCleanup(@() warning(onState));
    data = imread(filename); % added lines:
    data = data(:, :, min(1:3, end));
    data = imresize(data, [227 227]);
end
```

ตัวอย่างโค้ดไฟล์ main.m

**** ในที่นี้คืออ่านตัว model ที่ Train ด้วยการไม่ได้ใช้ Augmentation ใดๆเลย หากต้องการเปลี่ยนก็โหลดตัวอื่นมา ****