LAKE FOREST COLLEGE

EMOTION CLASSIFICATION BASED ON EXPRESSIONS AND BODY LANGUAGE USING CONVOLUTIONAL NEURAL NETWORKS

AASIMAH TANVEER

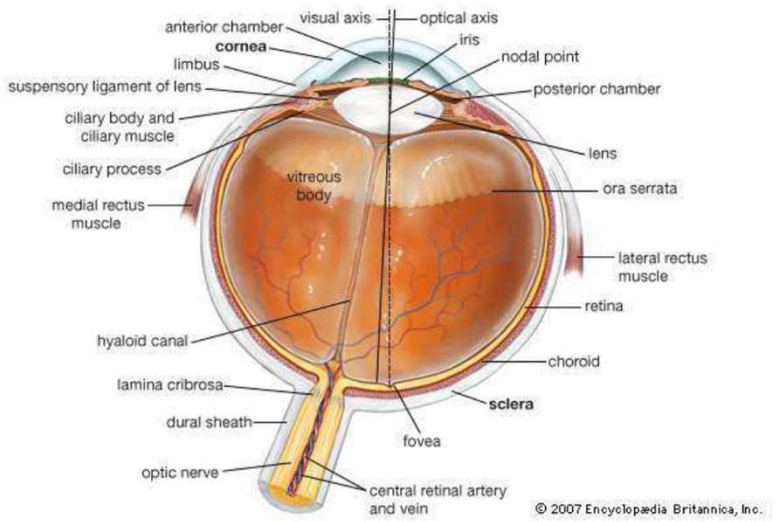
PI: DR. SUGATA BANERJI

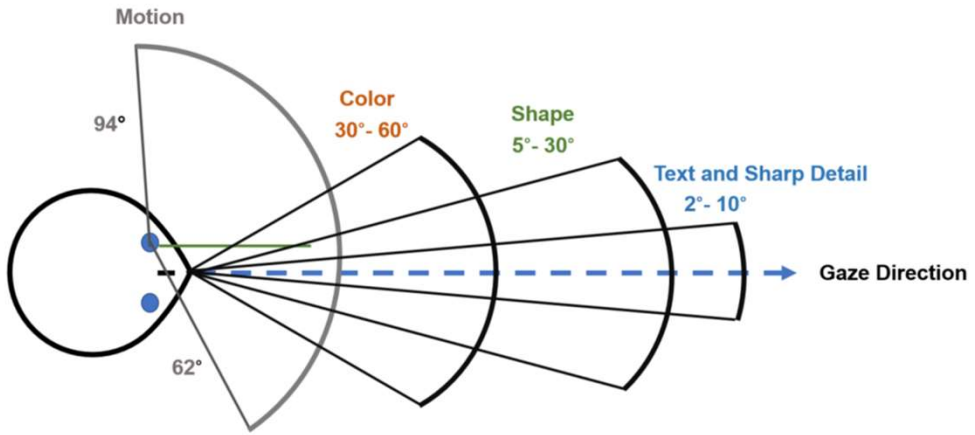
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, NEUROSCIENCE PROGRAM

INTRODUCTION

INTRODUCTION

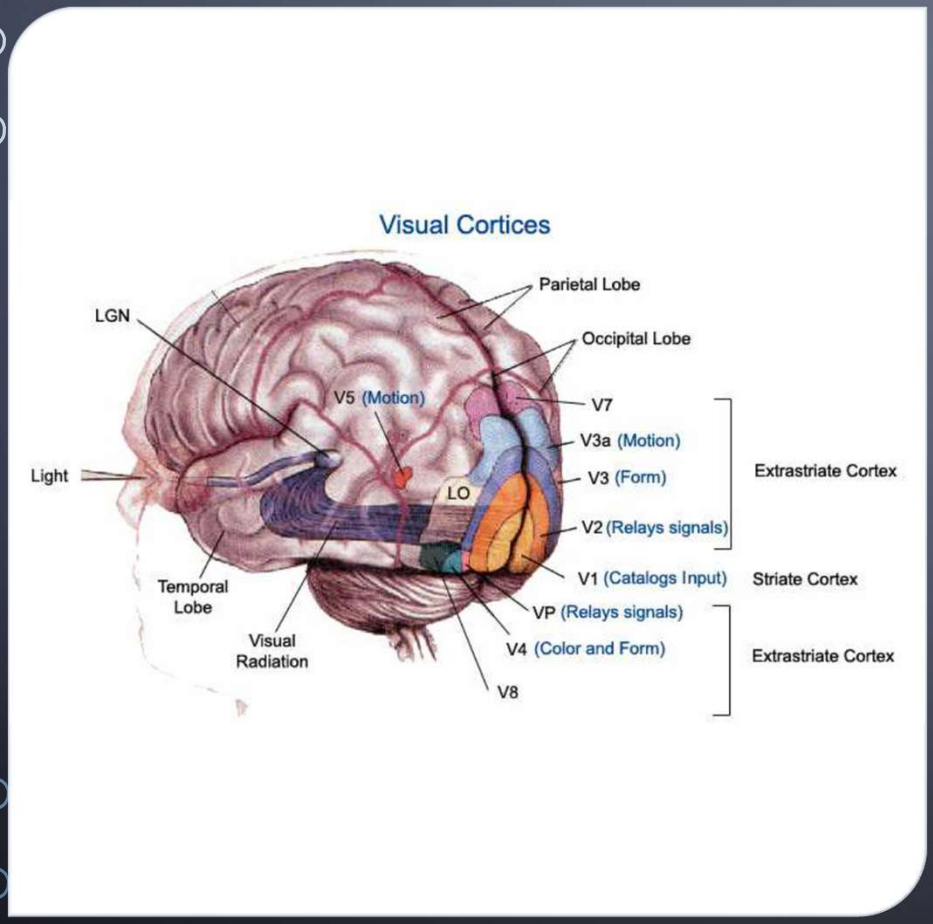
HOW WE SEE





EYES ARE THE FIRST FILTERS TO THE BRAIN

THINKING AND SEEING



- Lateral Geniculate Nucleus (LGN)
- Superior Colliculus
- Primary Visual Cortex - V1

INTRODUCTION

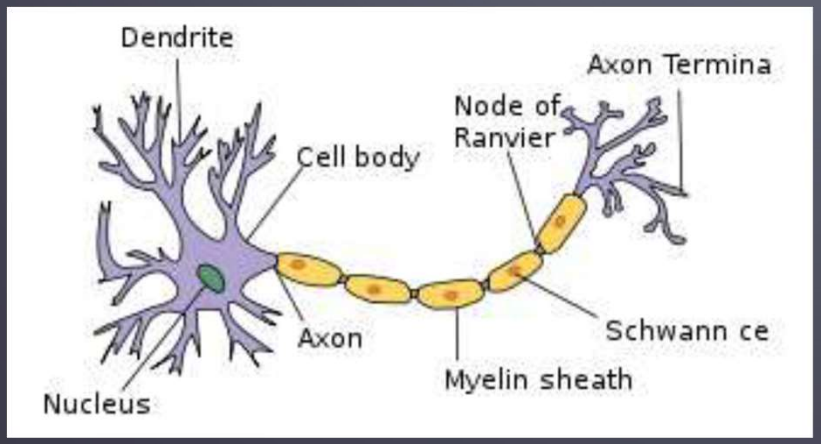
CONNECTING THE BRAIN AND THE MACHINE

NUERAL OBJECT RECOGNITION

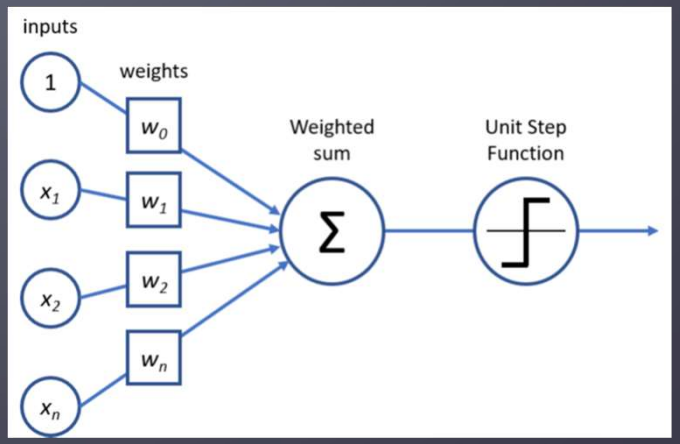
Stage 1	Processing of basic object components - color, depth, and form
Stage 2	Grouping information on distinct edges to the visual form
Stage 3	The visual representation is matched with structural descriptions in memory
Stage 4	Semantic attributes are applied to the visual representation - meaning and recognition

BIOLOGICAL VS. ARTIFICIAL NEURON

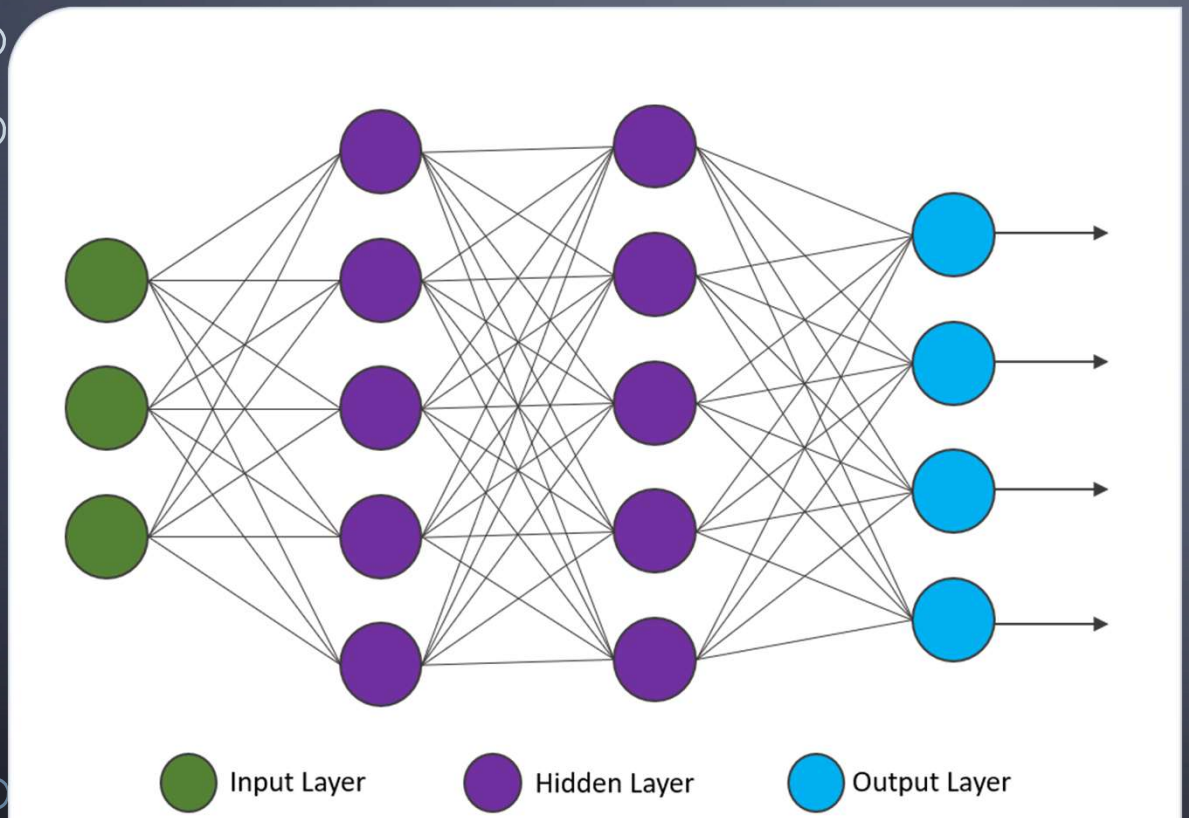
NEURON



PERCEPTRON

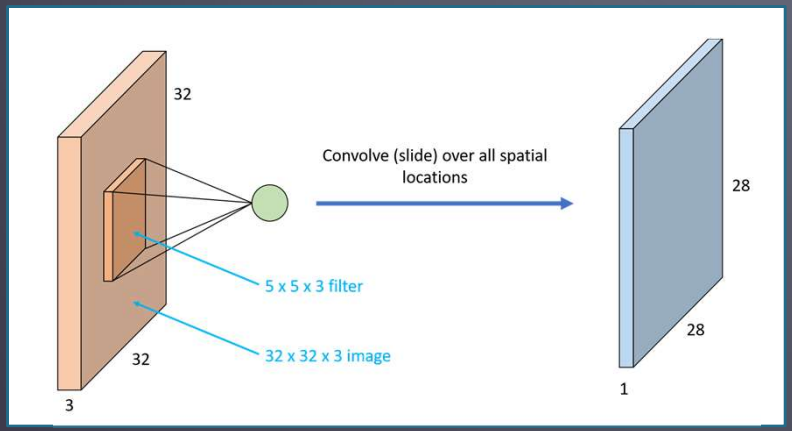


NEURAL NETWORK LAYERS

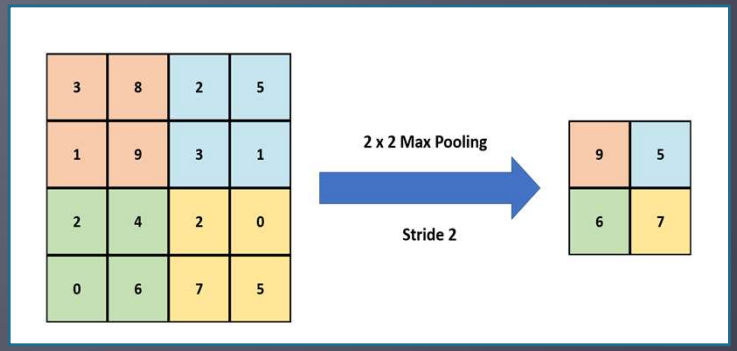


CONVULTIONAL NEURAL NETWORK

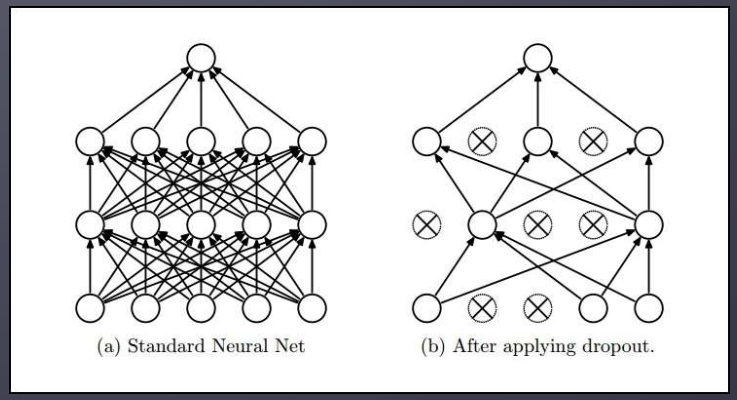
CONVOLUTION



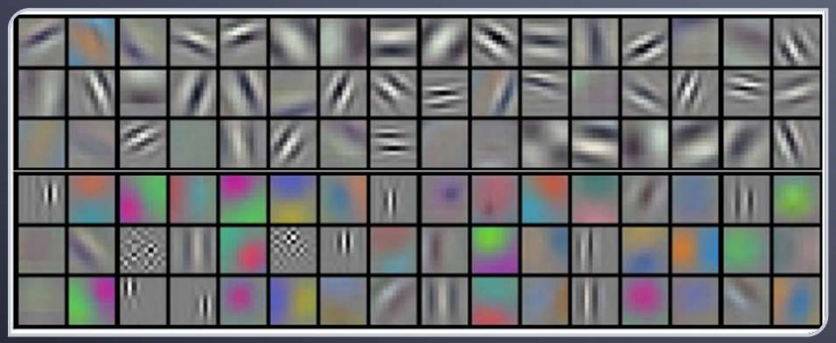
MAX POOLING



DROPOUT LAYER



ALEXNET



- Image recognition database
- Basis for our network
- Also pulled from other ImageNet sources

INTRODUCTION

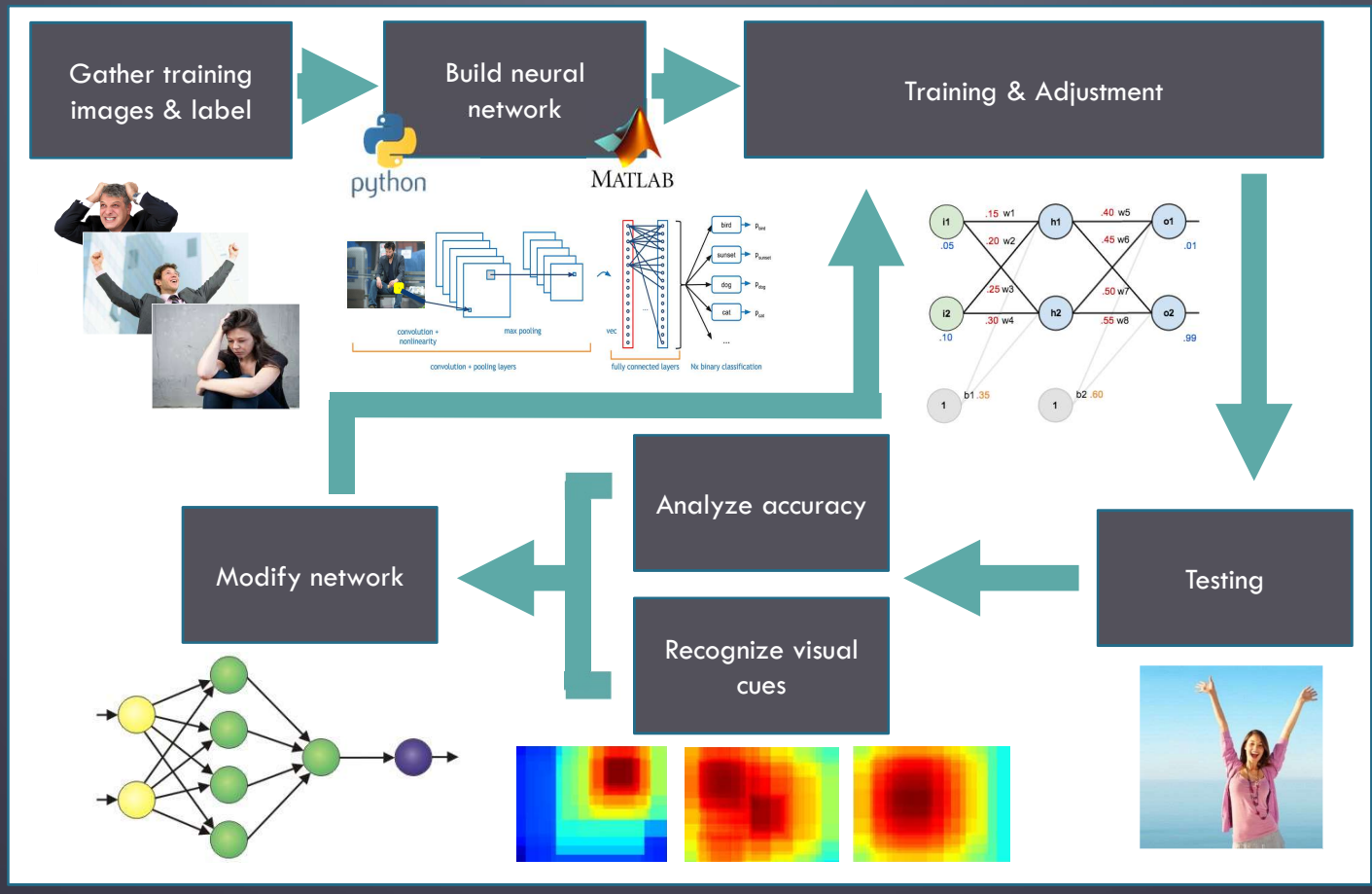
OBJECTIVE

CENTRAL QUESTION

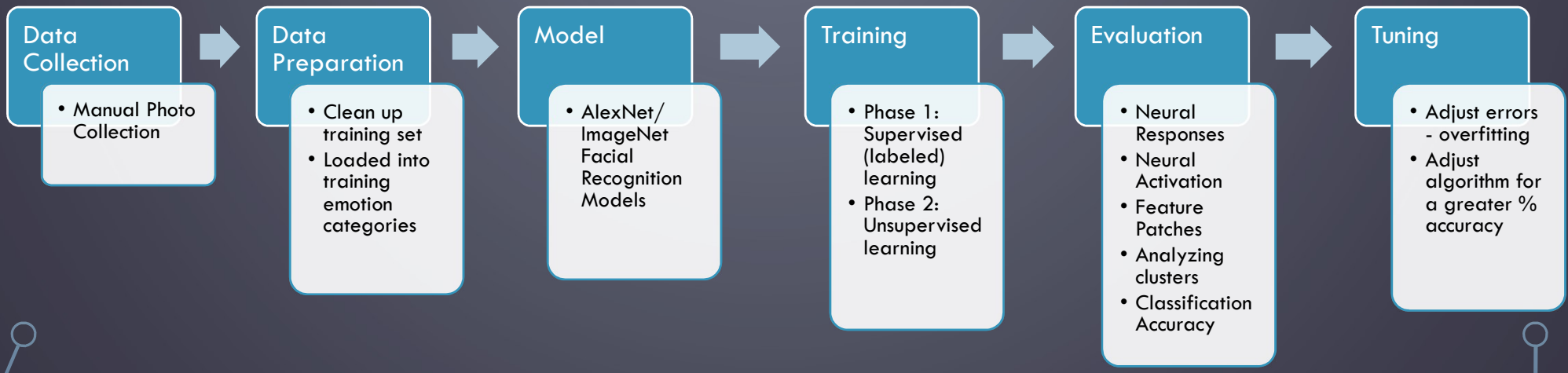
- CNNs are can be designed to be similar to the human vision system in some respects, but are they able to solve this very human problem of emotion recognition by evaluating holistic body language with facial recognition from images?

METHODOLOGY

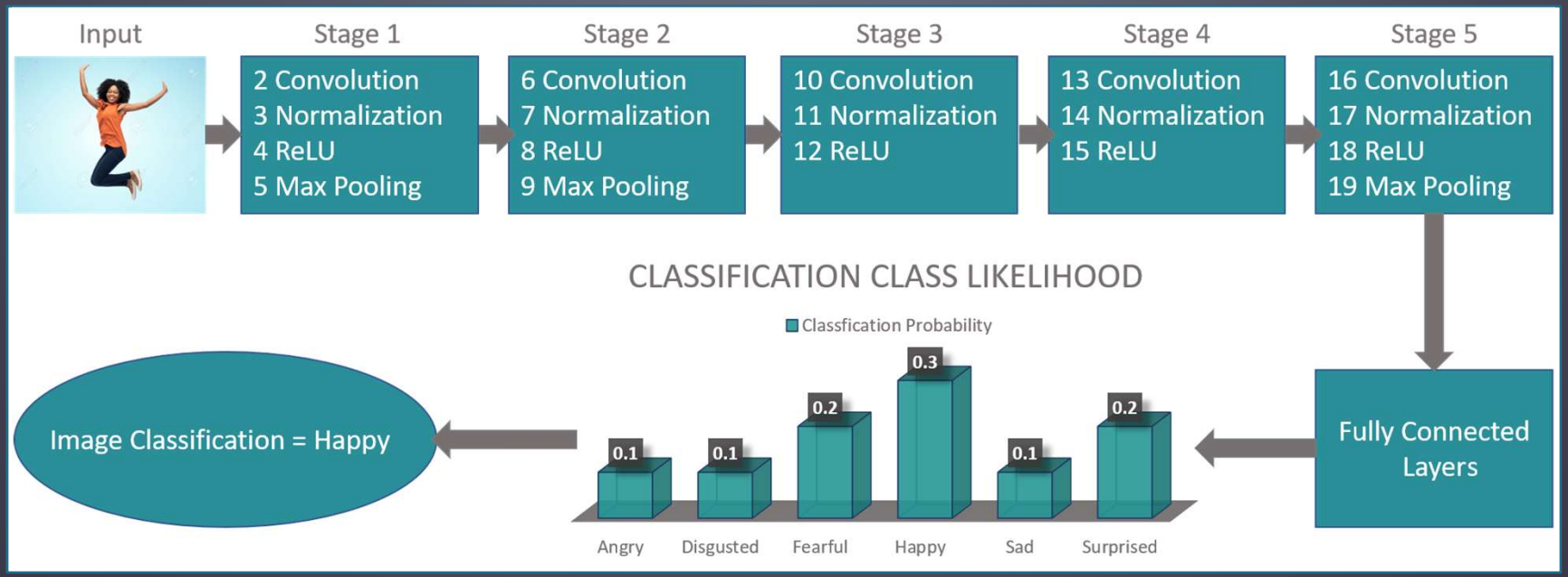
OVERALL PROCESS



LEARNING PROCESS EXPANDED



SCHEMATIC REPRESENTATION OF THE CNN

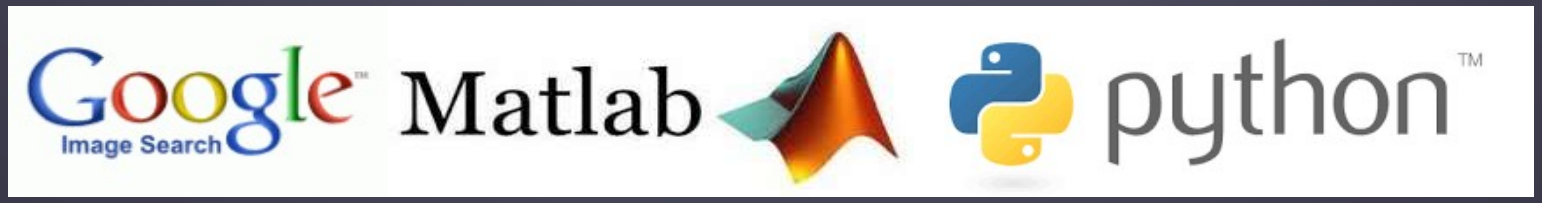


CNN ARCHITECTURE

Layer	Type	Description
1	Image Input	224x224x3 images with 'zerocenter' normalization
2	Convolution	96 11x11 convolutions, stride [4 4], padding [0 0 0 0]
3	Normalization	Batch normalization
4	ReLU	ReLU
5	Max Pooling	3x3 max pooling, stride [2 2], padding [0 0 0 0]
6	Convolution	256 5x5 convolutions, stride [1 1], padding [0 0 0 0]
7	Normalization	Batch normalization
8	ReLU	ReLU
9	Max Pooling	3x3 max pooling, stride [2 2], padding [0 0 0 0]
10	Convolution	384 3x3 convolutions, stride [1 1] and padding [0 0 0 0]
11	Normalization	Batch normalization
12	ReLU	ReLU
13	Convolution	384 3x3 convolutions, stride [1 1], padding [0 0 0 0]
14	Normalization	Batch normalization
15	ReLU	ReLU
16	Convolution	256 3x3 convolutions, stride [1 1], padding [0 0 0 0]
17	Normalization	Batch normalization
18	ReLU	ReLU
19	Max Pooling	3x3 max pooling, stride [2 2], padding [0 0 0 0]
20	Fully Connected	4096-sized fully connected layer
21	Normalization	Batch normalization
22	ReLU	ReLU
23	Dropout	50% dropout
24	Fully Connected	1024-sized fully connected layer
25	Normalization	Batch normalization
26	ReLU	ReLU
27	Dropout	50% dropout
28	Fully Connected	6-sized fully connected layer
29	Normalization	Batch normalization
30	ReLU	ReLU
31	Softmax	Softmax
32	Output	6 numbers representing the probabilities of the image being in the 6 emotion classes

TOOLS

- Images: Google Images
- Software:
  - MATLAB
    - CNN – Classification % Accuracy compared to predicted
  - Python
    - Data Visualization – Feature Maps, Neural Activation, Neural Response



DATASET

Angry

Disgusted

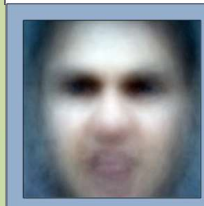
Fearful

Happy

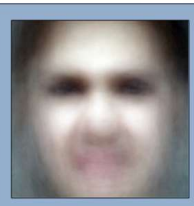
Sad

Surprised

Class Name	Image Count	Training Set	Test Set
Angry	172	150	22
Disgusted	165	150	15
Fearful	205	150	55
Happy	275	150	125
Sad	294	150	144
Surprised	354	150	204



Angry



Disgusted

Fearful

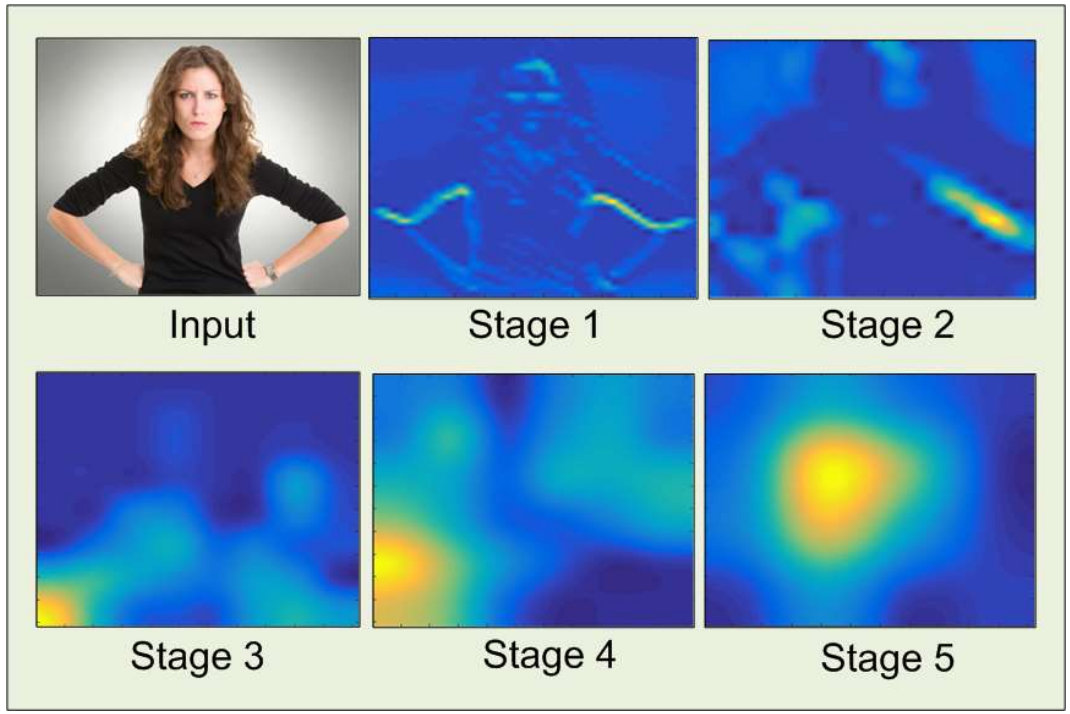
Happy

Sad

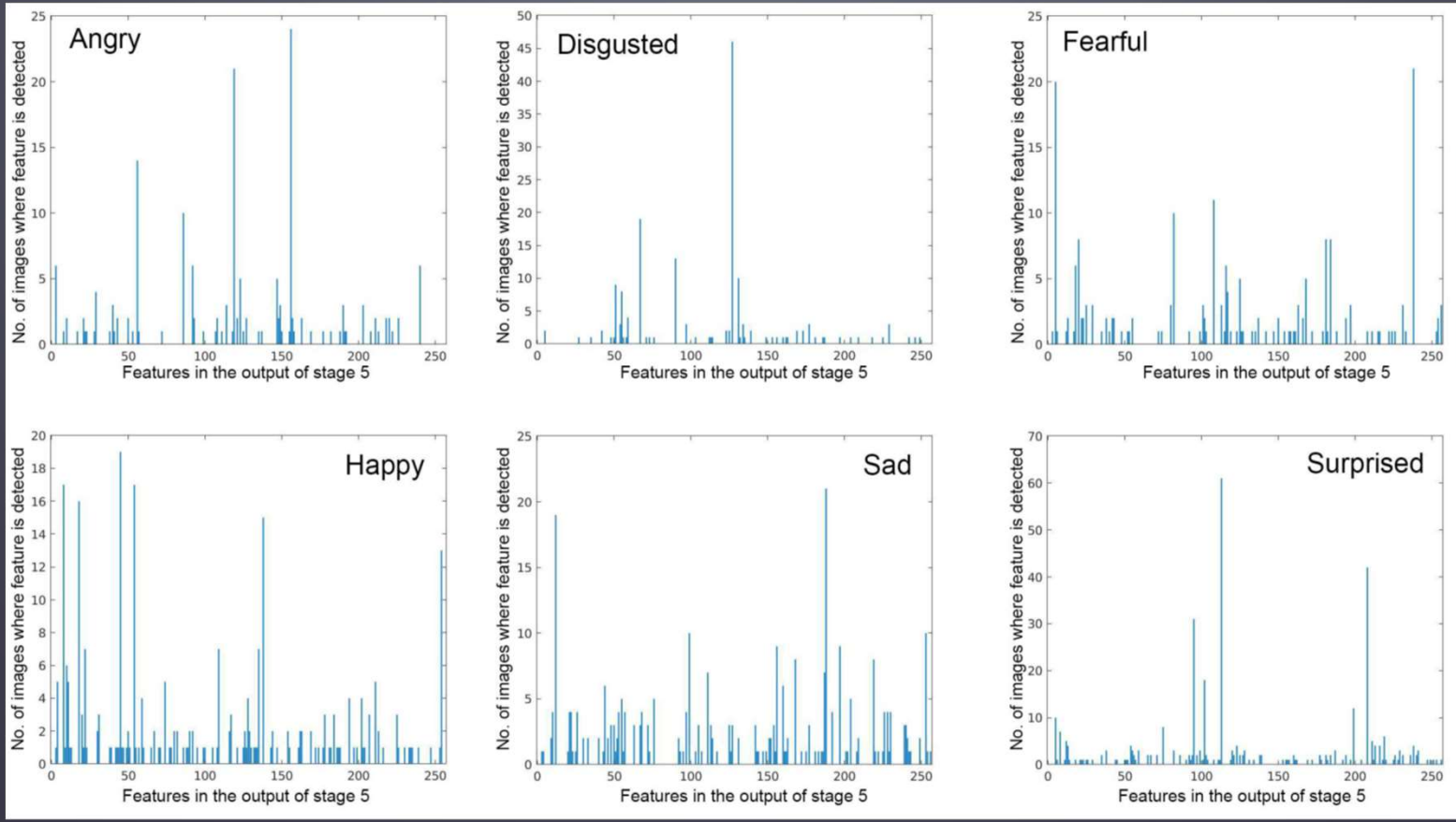
Surprised

RESULTS

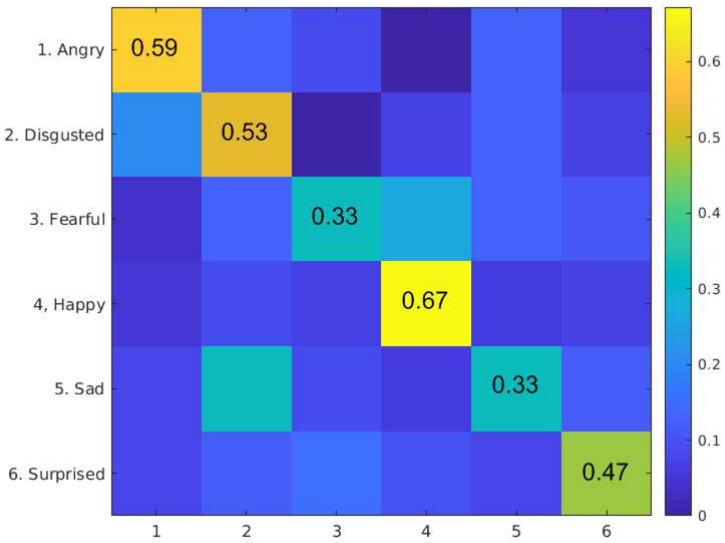
NEURON ACTIVATION

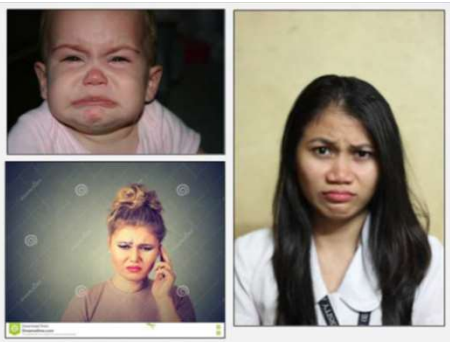


NEURAL RESPONSES



CONFUSION





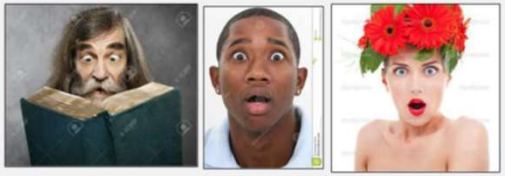
Sad classified as disgusted

Fearful classified as happy

Fearful classified as surprised

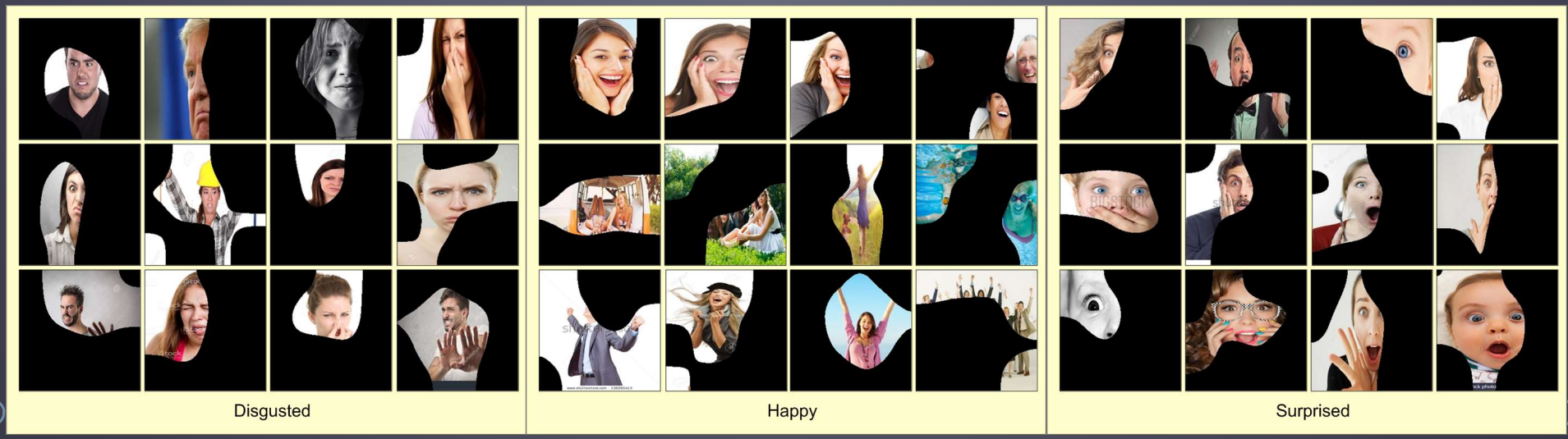
Surprised classified as happy

Fearful classified as sad

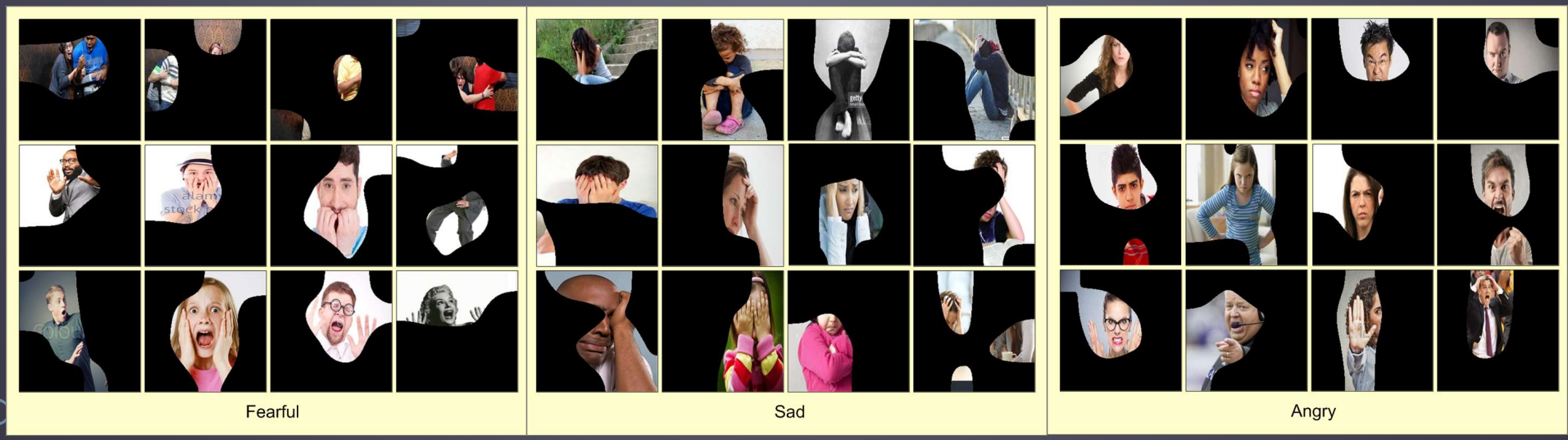


Surprised classified as fearful

NEURON PATCHES



NEURON PATCHES



DISCUSSION

OVERFITTING

Actual Label: angry

Predicted Labels:
angry : 72.31%
sad : 17.85%
disgusted : 2.46%

Actual Label: sad

Predicted Labels:
sad : 98.81%
angry : 0.24%
disgusted : 0.24%

CONCLUSIONS

- Created a successful CNN with a training dataset
- The network is performing a classification task – can identify in 49% cases
- Generate sets of visualizations to understand cues and interpretations from network

FUTURE DIRECTIONS

- Expand dataset
  - Change network as dataset changes – different configurations
  - Multiple emotion labelling
  - Represent a more global dataset
- Blur faces → look at body language more successfully
- Natural images

THANK YOU!

ACKNOWLEDGEMENTS

Thank you to the following for assisting in my thesis work:

- Dr. Sugata Banerji
- Dr. Matt Kelley
- Dr. Craig Knuckles
- Dr. Naomi Wentworth
- Lake Forest College

APPENDIX

FUNCTIONS

- Perceptron Function: $y = sgn(b + < w, x >)$
- Weight Vector: $w_{new} = w_{old} + (d_i - y_i) \times x_i$
- Bias: $b_{new} = b_{old} + (d_i - y_i)$

The binary step function

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases}$$

The Sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}}$$

Rectified Linear Unit (ReLU)

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

MATLAB R2017a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Log In

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

C: ▸ Users ▸ Aasimah ▸ Documents ▸ MATLAB ▸

Current Folder

Name

- caltech256
- ._.DS_Store
- .DS_Store
- cnnAlex_Initial.m

Editor - C:\Users\Aasimah\Documents\MATLAB\cnnAlex_Initial.m

cnnAlex_Initial.m

```
1  close all; clearvars; clc;
2  IMSIZE = 224;
3  INITSIZE = 256;
4  %% Read data
5  imagePath = 'EmotionsDataset/Images';
6  dataPath = 'EmotionsDat/.?aset/data';
7
8  emotions = {'angry','disgusted','fearful','happy','sad','surprised'};
9  trainImages = [];
10 trainLabels = [];
11 count = 0;
12 trainCount = 150;
13 for emo = 1:numel(emotions)
14     path = [imagePath '/' emotions{emo}];
15     fileNames1 = dir([path '/*.jpg']);
16     fileNames2 = dir([path '/*.png']);
17     fileNames = [fileNames1; fileNames2];
18     clear fileNames1 fileNames2;
19     for x = 1:numel(fileNames)
20         count = count + 1;
21         image = imread([path '/' fileNames(x).name]);
22         image = gray2rgb(image);
23         image = imresize(image, [INITSIZE INITSIZE]);
24         trainImages(:,:,:,count) = image;
25         trainLabels(count) = emo;
26         if x == trainCount
27             break;
28         end
```

Workspace

Name | Value

Command Window

New to MATLAB? See resources for Getting Started.

fx >>

Details

Select a file to view deta

Ready

Ln 1 Col 1

```
close all; clearvars; clc;
IMSIZE = 224;
INITSIZE = 256;
%% Read data
imagePath = 'EmotionsDataset/Images';
dataPath = 'EmotionsDat/.?aset/data';

emotions = {'angry','disgusted','fearful','happy','sad','surprised'};
trainImages = [];
trainLabels = [];
count = 0;
trainCount = 150;
for emo = 1:numel(emotions)
    path = [imagePath '/' emotions{emo}];
    fileNames1 = dir([path '/*.jpg']);
    fileNames2 = dir([path '/*.png']);
    fileNames = [fileNames1; fileNames2];
    clear fileNames1 fileNames2;
    for x = 1:numel(fileNames)
        count = count + 1;
        image = imread([path '/' fileNames(x).name]);
        image = gray2rgb(image);
        image = imresize(image, [INITSIZE INITSIZE]);
        trainImages(:,:,:,count) = image;
        trainLabels(count) = emo;
        if x == trainCount
            break;
        end
    end
end
trainLabels = categorical(trainLabels);

aug = imageDataAugmenter('RandXScale',[0.8 1.2],'RandYScale',[0.8 1.2],...
    'RandXReflection',true, 'RandXTranslation',[0 20], 'RandYTranslation',[0 20],...
    'RandRotation',[-5 5]);
imageSource = augmentedImageSource([IMSIZE IMSIZE],trainImages,trainLabels, 'DataAugmentation', aug);
%% Define network
layers = [imageInputLayer([IMSIZE, IMSIZE, 3],'Normalization','zerocenter','Name','inputlayer')
    % ----------Stage 1
    convolution2dLayer(11,96,'Stride',4,'Padding',0, 'Name', 'conv1')
    batchNormalizationLayer
    reluLayer('Name','relu1')
    maxPooling2dLayer(3,'Stride',2,'Name','pool1')
    % ----------Stage 2
    convolution2dLayer(5,256,'Stride',1,'Padding',0, 'Name', 'conv2')
    batchNormalizationLayer
    reluLayer('Name','relu2')
    maxPooling2dLayer(3,'Stride',2,'Name','pool2')
    % ----------Stage 3
    convolution2dLayer(3,384,'Stride',1,'Padding',0, 'Name', 'conv3')
    batchNormalizationLayer
    reluLayer('Name','relu3')
%     maxPooling2dLayer(2,'Stride',2,'Name','pool3')
    % ----------Stage 4
    convolution2dLayer(3,384,'Stride',1,'Padding',0, 'Name', 'conv4')
    batchNormalizationLayer
    reluLayer('Name','relu4')
%     maxPooling2dLayer(2,'Stride',2,'Name','pool4')
    % ----------Stage 5
    convolution2dLayer(3,256,'Stride',1,'Padding',0, 'Name', 'conv5')
    batchNormalizationLayer
    reluLayer('Name','relu5')
    maxPooling2dLayer(3,'Stride',2,'Name','pool5')

    % ----------Stage 6
    fullyConnectedLayer(4096, 'Name', 'fc1')
    batchNormalizationLayer
    reluLayer('Name','relu6')
    dropoutLayer
    % ----------Stage 7
    fullyConnectedLayer(1024, 'Name', 'fc2')
    batchNormalizationLayer
    reluLayer('Name','relu7')
    dropoutLayer
    % ----------Stage 8
    fullyConnectedLayer(6, 'Name', 'fc3')
    batchNormalizationLayer
    reluLayer('Name','relu8')

    softmaxLayer('Name', 'sm1')
    classificationLayer('Name','coutput')];
options = trainingOptions('sgdm','InitialLearnRate',0.001, ...
    'MaxEpochs',150);

%% Train Network
% net = trainNetwork(trainImages,trainLabels,layers,options);
net = trainNetwork(imageSource,layers,options);

%% Testing
testImages = [];
testLabels = [];
count = 0;
for emo = 1:numel(emotions)
    path = [imagePath '/' emotions{emo}];
    fileNames1 = dir([path '/*.jpg']);
    fileNames2 = dir([path '/*.png']);
    fileNames = [fileNames1; fileNames2];
    clear fileNames1 fileNames2;
    for x = trainCount+1:numel(fileNames)
        count = count + 1;
        image = imread([path '/' fileNames(x).name]);
        image = gray2rgb(image);
        image = imresize(image, [IMSIZE IMSIZE]);
        testImages(:,:,:,count) = image;
        testLabels(count) = emo;
    end
end
testLabels = categorical(testLabels');
[guessedLabels, scores] = classify(net, testImages);
accuracy = sum(guessedLabels == testLabels)/numel(testLabels);
fprintf('\nFinal Accuracy (unnormalized) = %.2f', accuracy*100);
```