Facial Emotion Detection with AI

Anuva, Allen, Tanya, Ritvik, Yonish

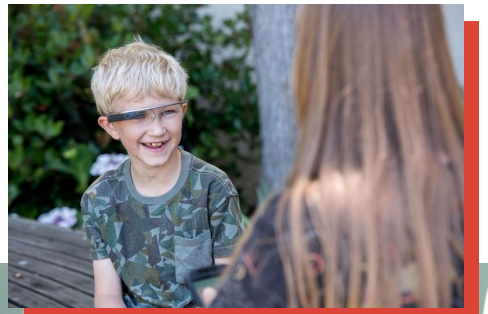
Introduction - What is facial detection?

- Getting computers to understand human expressions through images
- Even humans can only accurately predict ~60% of the time
- Computers use landmarks to identify where features are located
- Based on the identification and classification of landmarks, the computer learns what the result is

Audience Question:
Where is facial emotion detection used?

An Example of Facial Emotion Detection

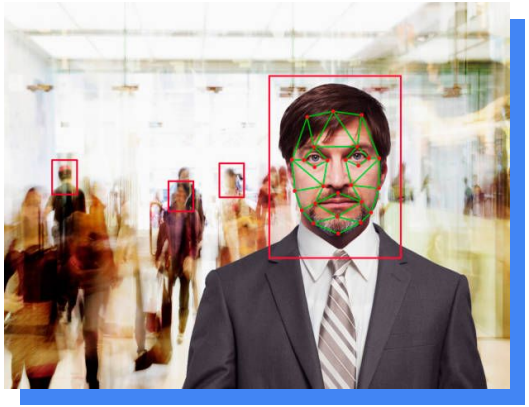
- Stanford Medicine: Google Glass helps kids with autism read facial expressions
- Screen and speaker to give video and audio information
- Distinguished between happiness, sadness, anger, disgust, surprise, fear, neutral and contempt
- “After one to three months of regular use, parents reported that children with autism made more eye contact and related better to others.”



Data Preprocessing

Feature extraction in a few steps:

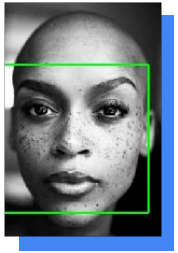
- Detect faces
- Estimate facial landmarks
- Extract landmarks coordinates
- Calculate euclidean distances



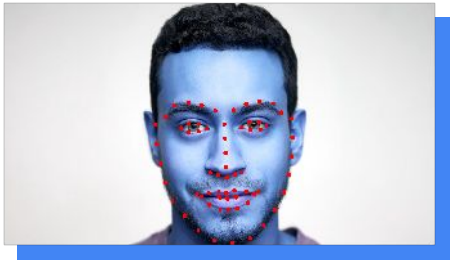
Detect faces & estimate landmarks

Using DLib pre-trained model to identify faces

```
dlib.get_frontal_face_detector()
```



Using DLib pre-trained model to estimate facial landmarks

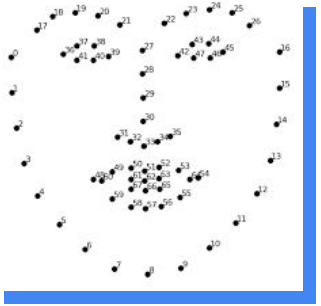


Extracting coordinates to calculate euclidean distances

Calculating euclidean distances from landmarks coordinates

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

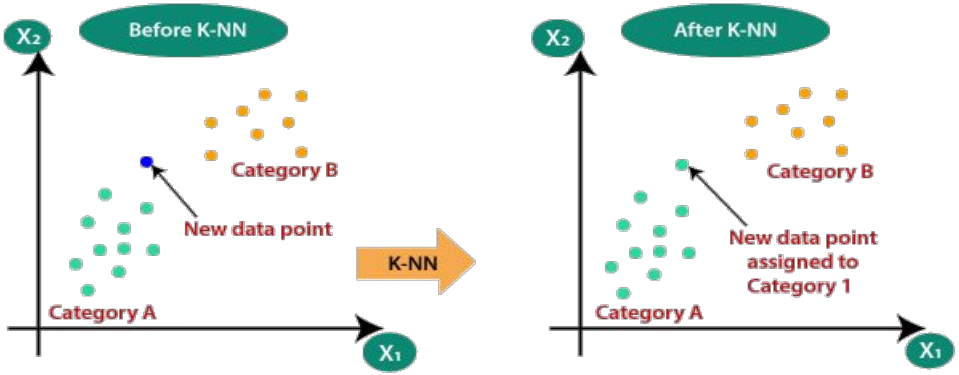
```
math.sqrt((p2[0]-p1[0])**2 + (p2[1]-p1[1])**2)
```



This will help us define features such as whether the eyes are closed or opened

Our First 3 Models

KNN - calculating the number of nearest neighbours

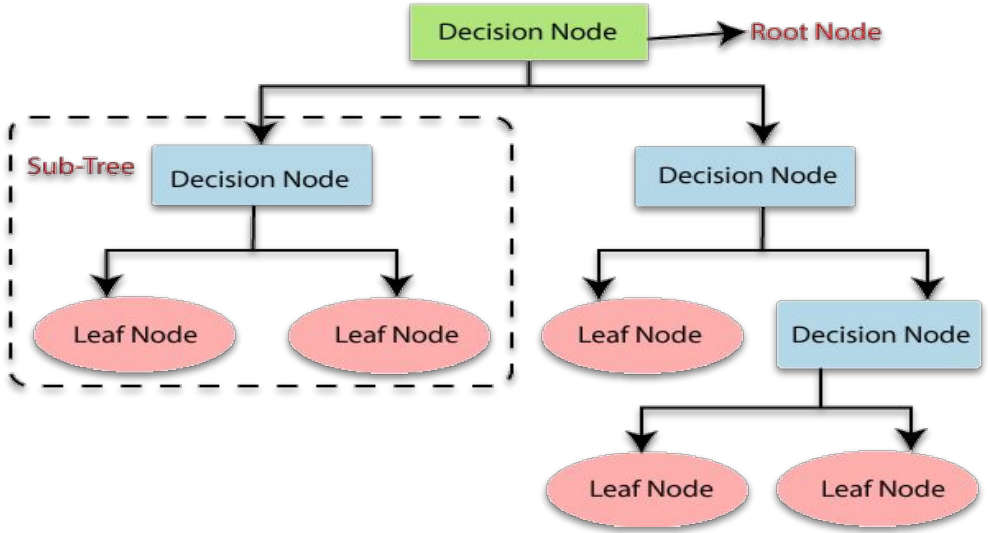



```
accuracy- 0.4715
```

Our First 3 Models

Decision Tree Classifier

- branching out different options
- making decisions at each step



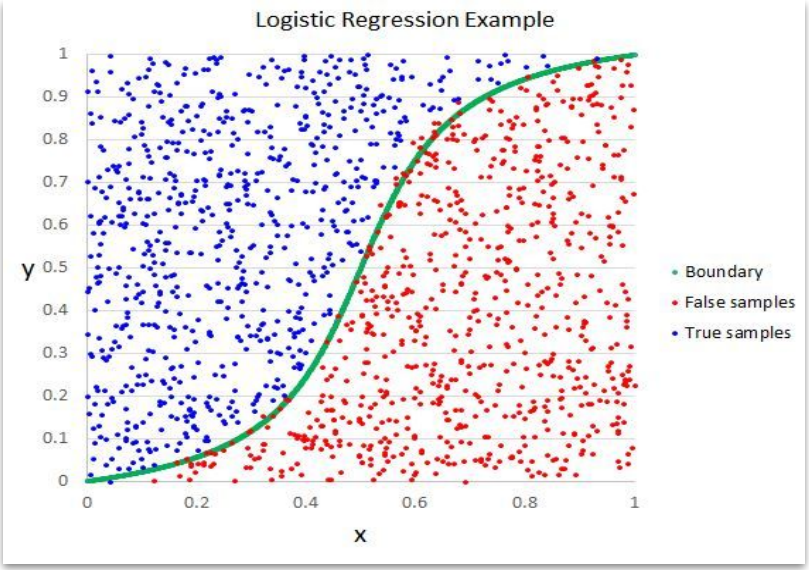

```
accuracy- 0.4315
```

Our First 3 Models

Logistic Regression - classification model

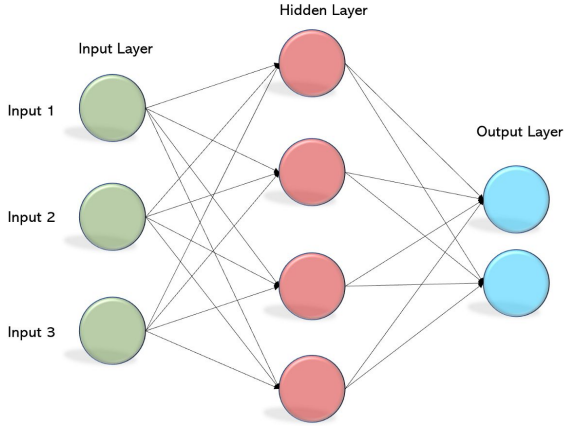
- Predicting an outcome
- Based on probability

```
accuracy- 0.503
```



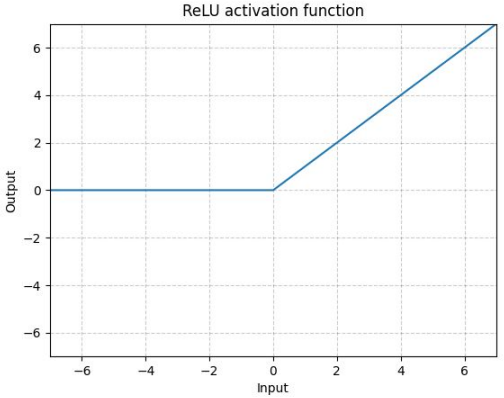
Neural Networks & Multilayer Perceptron

- A neural network is a series of algorithms that send and receive information to recognize relationships in a set of data
  - Similar to how neurons function in the body
- MLPs contain 3 types of layers:
  - Input Layer:
    - Inputs are inserted here to start processing
  - Hidden Layer(s):
    - Neurons receive and can modify the data before sending it to the another neuron
  - Output Layer:
    - Uses the data to classify the input

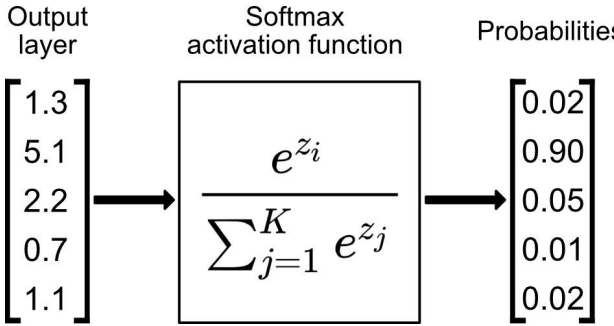


Neural Network Processing

- Our project uses a Feedforward Neural Network
  - Outputs are sent in one direction and will NEVER go back to a previous layer
- Activation function:
  - Defines the output of a neuron with a given input
    - Rectified Linear Unit (ReLU)
      - Piecewise linear function that outputs 0 if the value is negative, otherwise the value itself
      - Makes the model easier to train
    - Softmax
      - Creates a vector of probabilities for each emotion the input can be
      - Used for multi-class classification, often implemented in the output layer



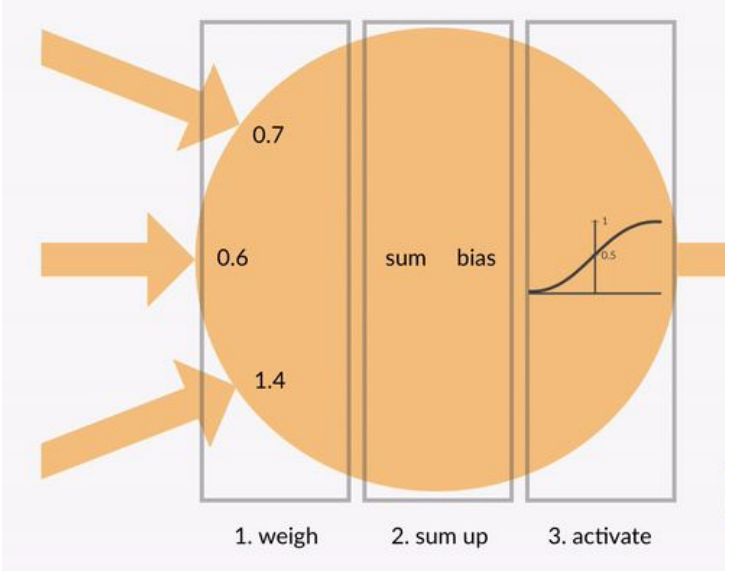
Note: Any x-value < 0 is outputted as 0



Activation Function Factors:

- Weights:
  - Determines how important the output is for the next neuron
    - Can modify the value to be greater or smaller
    - Similar to the slope of a linear function
- Bias:
  - Constant added to the value
  - Shifts the activation function
    - Like the y-intercept

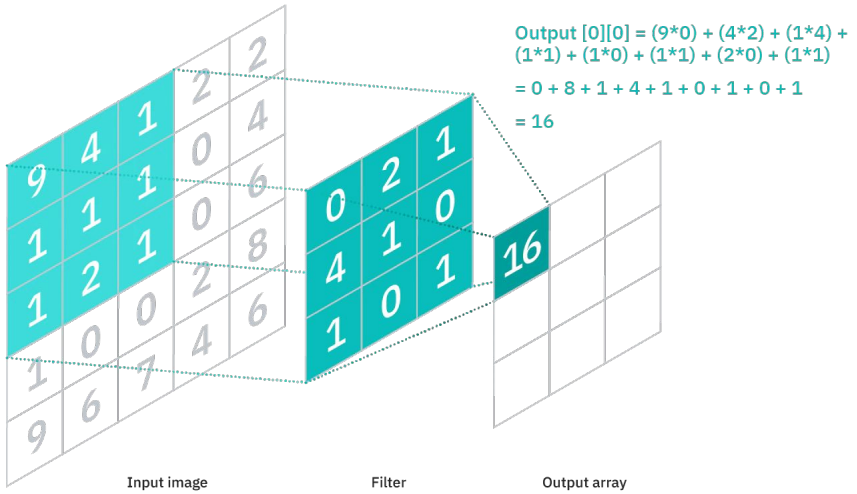
$$y = Mx + C$$



CNNs and Transfer Learning

CONVOLUTIONAL NEURAL NETWORKS

CNNs are simply a system widely used in image classification which helps machines more effectively identify and classify objects in a given picture.

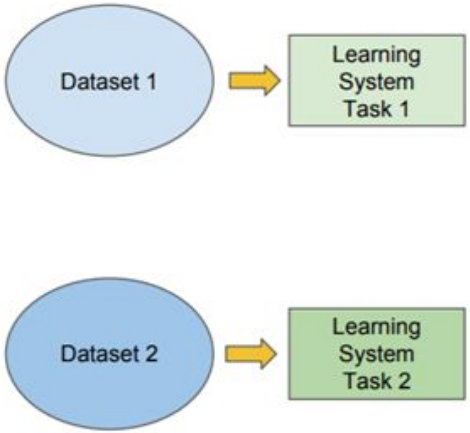


They make use of filters which are user defined pixel arrays that help the machine to classify objects

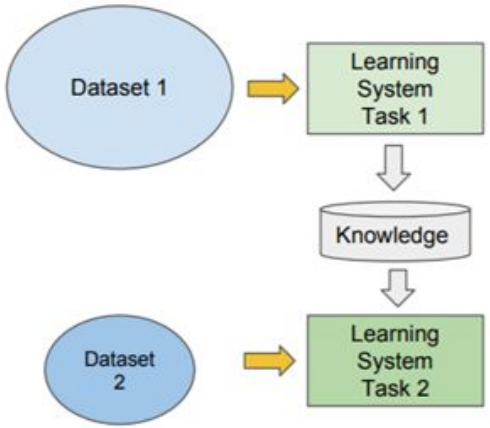
TRANSFER LEARNING

Traditional ML vs Transfer Learning

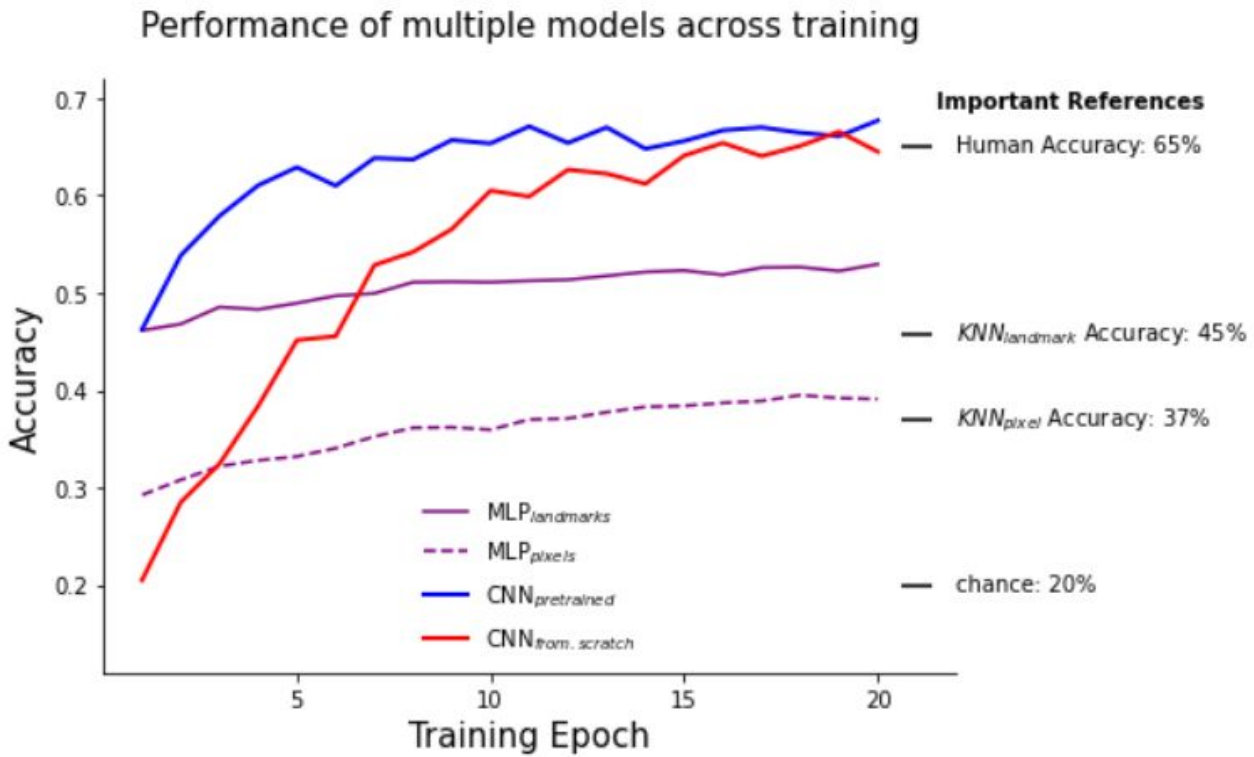
- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Results:


Performance of multiple models across training
Accuracy
Training Epoch
0.7
0.6
0.5
0.4
0.3
0.2
5
10
15
20
MLP landmarks
MLP pixels
CNN pretrained
CNN from. scratch
Important References
Human Accuracy: 65%
KNN landmark Accuracy: 45%
KNN pixel Accuracy: 37%
chance: 20%

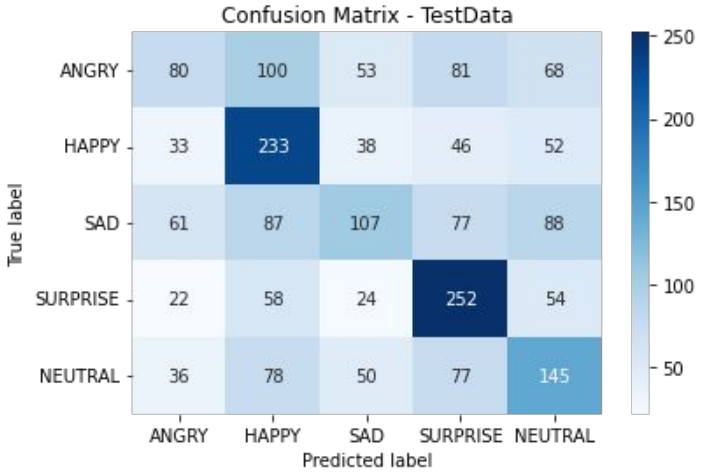
Challenges

Overfitting:

- By overtraining the model, the results may still come out to be far from expected
  - Change dropout to stop overfitting

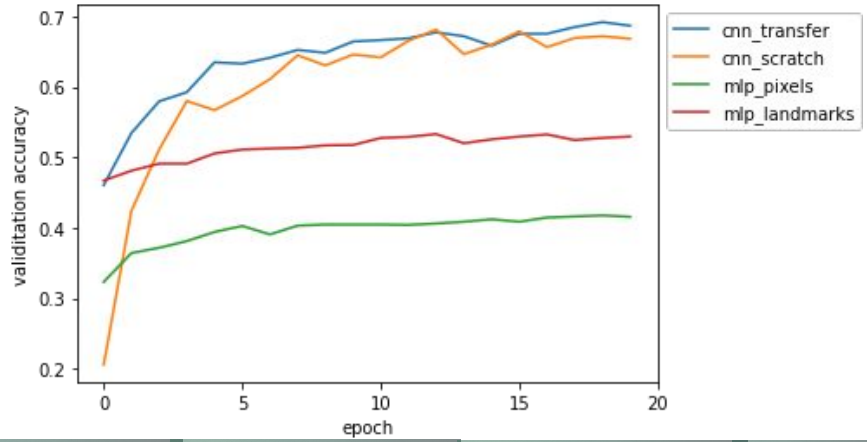
Confusion matrix:

- Compares how well the algorithm performed by comparing it to the correct answers in a visual format
- In this matrix, we can see that the computer did really well when identifying happiness and surprise



Summary

- Goal: To make an algorithm that could detect human emotions based on their expressions
- Types of data we used: Landmark distances, Pixel-based images
  - The computer did far better with the landmarks than with the pixels
- Types of Models:
  - Logistic Regression did better than KNNs
  - KNNs did better than Decision trees



Thank You! Questions?

Facial Emotion Detection with AI

Anuva, Allen, Tanya, Ritvik, Yonish