Facial Expression Detection

AIM97

June 14, 2019

Contents

1 CH1 - Introduction 5

1.1 Preface 5

1.1.1 Preface 5

1.2 Image Processing 5

1.2.1 Brief 5

1.2.2 How important is image processing? 6

1.2.3 How is image processing important for our project 7

1.3 Neural Networks 8

1.3.1 Breif 8

1.3.2 Basics of Neural Networks. 8

1.3.3 Why Neural networks are important? 9

1.3.4 Types of neural network 9

1.3.5 How Neural network is useful for Facial Emotion Detection 11

1.3.6 A Gentle Introduction to Convolutional Neural Network Layers 11

2 CH2 - Review 15

2.1 Preprocessing 15

2.1.1 Data sets 15

2.1.2 Face detection 16

2.1.3 Image enhancements 17

2.1.4 Landmark extraction 17

2.1.5 HOG 17

2.1.6 Data augmentation 17

2.1.7 Under sampling 18

2.2 model 19

3 CH3 - Proposed System 21

3.1 System Architecture 21

3.2 Preprocessing 21

3.3 Model 21

3.4 Library 21

3.5 Summary 21

4 CH4 - Results 23

5 CH5 - Conclusion 25

6 References 27

Chapter 1

CH1 - Introduction

1.1 Preface

1.1.1 Preface

<You expect employees to have high levels of emotional intelligence when interacting with customers. Now, thanks to advances in Deep Learning, you'll soon expect your software to do the same.>

<Companies have also been taking advantage of emotion recognition to drive business outcomes. For the upcoming release of Toy Story 5, Disney plans to use facial recognition to judge the emotional responses of the audience. Apple even released a new feature on the iPhone X called Animoji, where you can get a computer simulated emoji to mimic your facial expressions. It's not so far off to assume they'll use those capabilities in other applications soon.>

1.2 Image Processing

1.2.1 Brief

definition Image processing is the process of applying some operations to an image to reach an enhanced image that satisfies a certain goal depending on the application in hand. for example if we need to make an application that detects edges within an image we use an image processing technique that is capable of highlighting those edges and make them stand out. the result image is not necessarily a beautiful one from the perspective of a human, but it has to highlight the features of interest within the image that would be used for further processing.

impact Apart from the rule image processing plays in graphics enhancement to make image more visually appealing, Image processing is very important tool that is used for the specially preparation for computer vision and Machine learning, image processing a key preprocessing step to be taken before start in any of the two fields. the key difference between those two purposes is that when we want the image to be more visually appealing our target is a human, a human is the one who should view that image in the end. but when it comes to fields like computer vision or Machine Learning, the target is a computer that is programmed to act based on the content of input image. for this computer to do that it must be able to clearly extract feature of interest from the image, in order to make use of the image, we must have 2 main tasks for image processing:

1. noise removal: to remove the noise (like salt and pepper, or gauessian blur, ...etc) that we estimate to exist in the image so as to refine the features to be extracted from the image.
2. feature extraction: to highlight and evaluate features of interest that exist in the image to be used as input data for computer vision or Machine learning algorithms e.g. neural networks.

these two processes are the most common use cases for image processing, and we will go over them with more detail in next section.

1.2.2 How important is image processing?

The applications of image processing are many we will catch on some applications for noise removal and feature extraction.

Noise removal

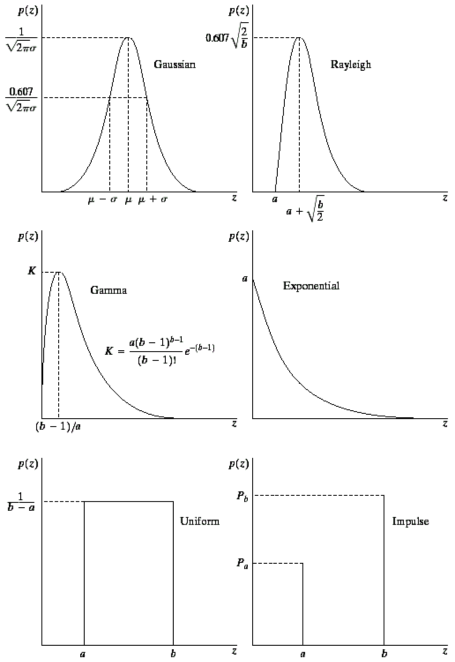
this process is done to make features more clear to refine the quality of extracted features by removing different types of noise (see figure 1.1).
multiple filters exists to restore original image by removing noise as much as possible.

1. Max filter: the output at one pixel is the **maximum** value of the pixels around it.
2. Min filter: the output at one pixel is the **minimum** value of the pixels around it.
3. Median filter: the output at one pixel is the **median** value of the pixels around it.
4. Mean filter: the output at one pixel is the **mean** value of the pixels around it.
5. Gauessian filter: it applies a matrix with values with gaussian distribution(highest weight in the center and weight decreases as we go away from the center) to the current window and the result is assigned to current pixel.

these filters make use of the values of pixels around current pixel in order to detect abnormal changes within the image which is probably noise and based on the values of surrounding pixel a new value is assigned to current pixel which is estimated to be closest to the original value.
Some other filter are used for enhancement can be sharpening filter and.

Feature extraction

this process is done to highlight features of interest in an image



1

Figure 1.1: examples for different types of noise

1.2.3 How is image processing important for our project

The purposes of our project is to recognize the facial expression from face image, for this task multiple image processing techniques have been applied to the input image before extracting features like face landmarks and HOG from the image.

Noise Removal

since we don't expect the input image to be particularly corrupted we only use median filter to remove white noise and salt and pepper noise a gaussian filter was being used at as well but it was inefficient in terms of time so the median filter took its place without a problem. we also use sharpening filter to make face features more clear for the landmark extraction process.

Feature Extraction

unlike CNN (convolutional neural network) model which extracts the features it needs from the image directly, one of the approaches we took requires pre-processing to extract some features from input image we needed two types of features in particular:

1. HOG (Histogram of Oriented Gradients) : feature descriptor which means that it generalize the object in a way that the same object (in this case a person) produces as close as possible to the same feature descriptor when viewed under different conditions.

2. Face Landmarks : those are points in the face that represent the face main features and we need those to estimate the emotion as well.

1.3 Neural Networks

1.3.1 Breif

What is Neural Networks? Artificial neural networks are one of the main tools used in machine learning. As the "neural" part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

While neural networks (also called "perceptrons") have been around since the 1940s, it is only in the last several decades where they have become a major part of artificial intelligence. This is due to the arrival of a technique called "backpropagation," which allows networks to adjust their hidden layers of neurons in situations where the outcome doesn't match what the creator is hoping for — like a network designed to recognize dogs, which misidentifies a cat, for example.

Another important advance has been the arrival of deep learning neural networks, in which different layers of a multilayer network extract different features until it can recognize what it is looking for.

1.3.2 Basics of Neural Networks.

a basic idea of how a deep learning neural network learns, imagine a factory line. After the raw materials (the data set) are input, they are then passed down the conveyer belt, with each subsequent stop or layer extracting a different set of high-level features. If the network is intended to recognize an object, the first layer might analyze the brightness of its pixels. see figure 1.2

The next layer could then identify any edges in the image, based on lines of similar pixels. After this, another layer may recognize textures and shapes, and so on. By the time the fourth or fifth layer is reached, the deep learning net will have created complex feature detectors. It can figure out that certain image elements (such as a pair of eyes, a nose, and a mouth) are commonly found together.

Once this is done, the researchers who have trained the network can give labels to the output, and then use backpropagation to correct any mistakes which have been made. After a while, the network can carry out its own classification tasks without needing humans to help every time.

1

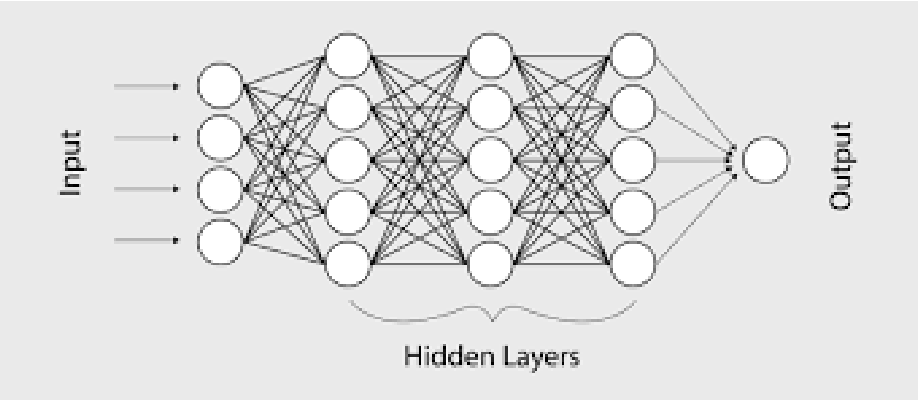


Figure 1.2: example of neural networks

Beyond this, there are different types of learning, such as supervised or unsupervised learning or reinforcement learning, in which the network learns for itself by trying to maximize its score

1.3.3 Why Neural networks are important?

ANNs (Artificial Neural Networks) have some key advantages that make them most suitable for certain problems and situations:

1. ANNs have the ability to learn and model non-linear and complex relationships, which is really important because in real-life, many of the relationships between inputs and outputs are non-linear as well as complex.

2. ANNs can generalize ,After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well,thus making the model generalize and predict on unseen data.

3. Unlike many other prediction techniques, ANN does not impose any restrictions on the input variables (like how they should be distributed). Additionally, many studies have shown that ANNs can better model heteroskedasticity i.e. data with high volatility and non-constant variance, given its ability to learn hidden relationships in the data without imposing any fixed relationships in the data.

1.3.4 Types of neural network

There are multiple types of neural network, each of which come with their own specific use cases and levels of complexity.

feedforward neural network

The most basic type of neural net is something called a feedforward neural network, in which information travels in only one direction from input to output.

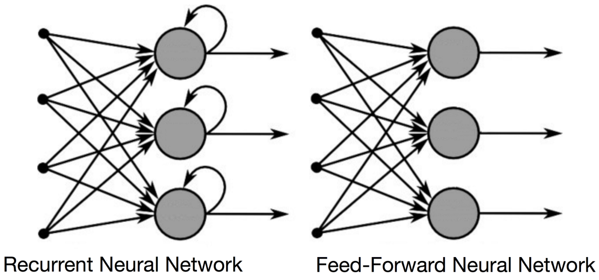


Figure 1.3: Structure of Feedforward and Recurrent Neural networks.

neural network layers are independent of each other; hence, a specific layer can have an arbitrary number of nodes. Typically, the number of hidden nodes must be greater than the number of input nodes. When the neural network is used as a function approximation, the network will generally have one input and one output node. When the neural network is used as a classifier, the input and output nodes will match the input features and output classes.

A neural network must have at least one hidden layer but can have as many as necessary. The bias nodes are always set equal to one. In analogy, the bias nodes are similar to the offset in linear regression i.e. y = mx+b. How does one select the proper number of nodes and hidden number of layers? This is the best part: there are really no rules! The modeler is free to use his or her best judgment on solving a specific problem. Experience has shown that there are best practices such as selecting an adequate number of hidden layers, activation functions, and training methods see figure (1.3).

Recurrent Neural Network

A recurrent neural network (RNN) is a type of artificial neural network commonly used in speech recognition and natural language processing (NLP). RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario. see figure (1.3)

Convolutional Neural Network

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).

A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

1.3.5 How Neural network is useful for Facial Emotion Detection

Analysis and recognition of human facial expressions from images and video form the basis for understanding image content at a higher semantic level. Expression recognition forms the core task of intelligent systems based on human-computer interaction (HCI). So the main task here is "understanding images and video content", this is the core concept of neural networks train over a lot of samples to learn then we can apply the concept of generalization on it by testing it using unseen data and check the output. as I mention above there are different types of learning, here we use Supervised learning as we feed the neural network with labeled data so we lead the training process.

So according to our application, we choose to build our model using the convolutional neural networks (CNN).

Traditional neural networks are not ideal for image processing and must be fed images in reduced-resolution pieces. CNN has their "neurons" arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. The layers of neurons are arranged in such a way as to cover the entire visual field avoiding the piecemeal image processing problem of traditional neural networks.

1.3.6 A Gentle Introduction to Convolutional Neural Network Layers

The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.

A CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

Convolutional Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features

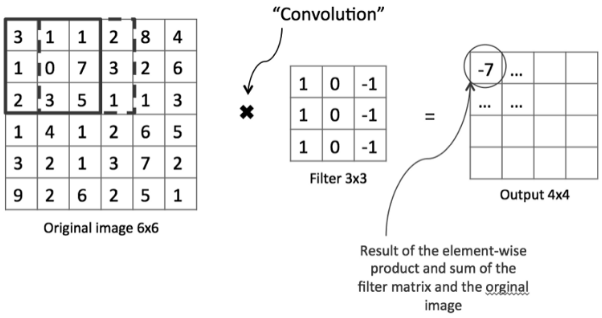


Figure 1.4: example of Convolutional layer.

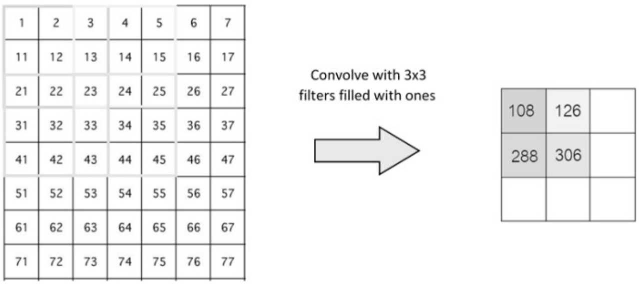


Figure 1.5: example of stride.

using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernal multiply them and get another matrix as a result which is called the feature map.

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters (Kernels).

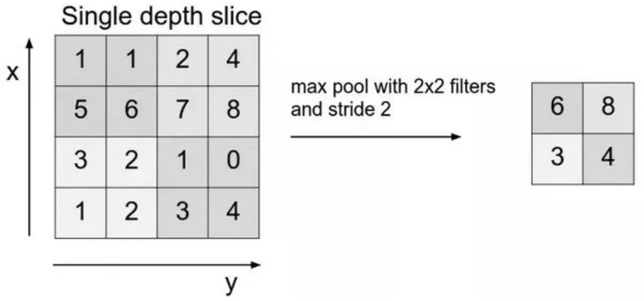
Stride

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

Padding

Sometimes filter does not fit perfectly fit the input image. We have two options:

Pad the picture with zeros (zero-padding) so that it fits Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 1.6: example of pooling layer

Relu

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max((0, x)$.

Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists uses ReLU since performance wise ReLU is better than other two.

Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

1. Max Pooling
2. Average Pooling
3. Sum Pooling

Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

Fully connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.

Chapter 2

CH2 - Review

2.1 Preprocessing

2.1.1 Data sets

A collection of instances is a dataset and when working with machine learning methods we typically need a few datasets for different purposes.

Training Dataset: A dataset that we feed into our machine learning algorithm to train our model.

Testing Dataset: Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset. **Brief about used Datesets.**

FER13

FER2013 is a large, publicly available Facial Expression Detection dataset consisting of 35,887 face crops. The dataset is challenging as the depicted faces vary significantly in terms of person age, face pose, and other factors, reflecting realistic conditions.The dataset is split into training, validation, and test sets with 28,709, 3,589, and 3,589 samples, respectively.fer13 contains seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).Basic expression labels are provided for all samples. All images are grayscale

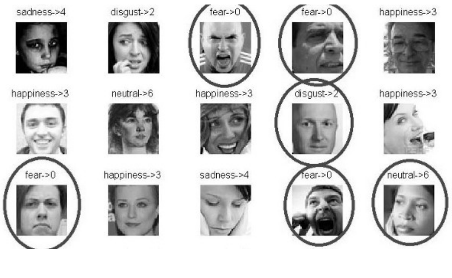


Figure 2.1: example of neural networks

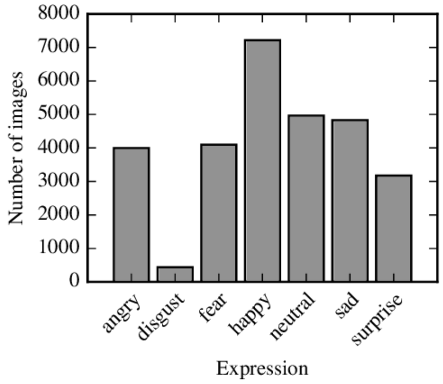


Figure 2.2: example of neural networks

and have a resolution of 48 by 48 pixels. The human accuracy on this dataset is around 65.5%.

This makes the classification harder because the model have to generalize well and be robust to incorrect data. The best accuracy results obtained on this dataset, as far as I know, is 75.2% described in this paper:[1][Facial Expression Recognition using Convolutional Neural Networks: State of the Art, Pramerdorfer & al. 2016]

CK+

Extended cohen kandle dataset is a small dataset ,publicly available Facial Expression Detection dataset Participants are 18 to 50 years of age. 69% female, 81% Euro-American 13% Afro-Americn and 6% other groups .

Image sequances for frontal views and 30-degree views were digitalized into either 640*940 or 640*480 pixel arrays with 8-bit gray scale or 124 color value. contains eight categories neutral, sadness, surprise, happiness, fear, anger, contempt and disgust.

2.1.2 Face detection

hello face detection

2.1.3 Image enhancements

2.1.4 Landmark extraction

2.1.5 HOG

2.1.6 Data augmentation

Data Augmentation?

As we mentioned before we face a problem of imblanced data set while using fer2013 dataset ,so we check this technique as an over sampling method.and we will give a breif about it.

what is Data Augmentation? Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples.

Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image.

Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

The intent is to expand the training dataset with new, plausible examples. This means, variations of the training set images that are likely to be seen by the model. For example, a horizontal flip of a picture of a cat may make sense, because the photo could have been taken from the left or right. A vertical flip of the photo of a cat does not make sense and would probably not be appropriate given that the model is very unlikely to see a photo of an upside down cat.

As such, it is clear that the choice of the specific data augmentation techniques used for a training dataset must be chosen carefully and within the context of the training dataset and knowledge of the problem domain. In addition, it can be useful to experiment with data augmentation methods in isolation and in concert to see if they result in a measurable improvement to model performance, perhaps with a small prototype dataset, model, and training run.

Modern deep learning algorithms, such as the convolutional neural network, or CNN, can learn features that are invariant to their location in the image. Nevertheless, augmentation can further aid in this transform invariant approach to learning and can aid the model in learning features that are also invariant to transforms such as left-to-right to top-to-bottom ordering, light levels in photographs, and more.

Image data augmentation is typically only applied to the training dataset, and not to the validation or test dataset. This is different from data preparation such as image resizing and pixel scaling; they must be performed consistently across all datasets that interact with the model.

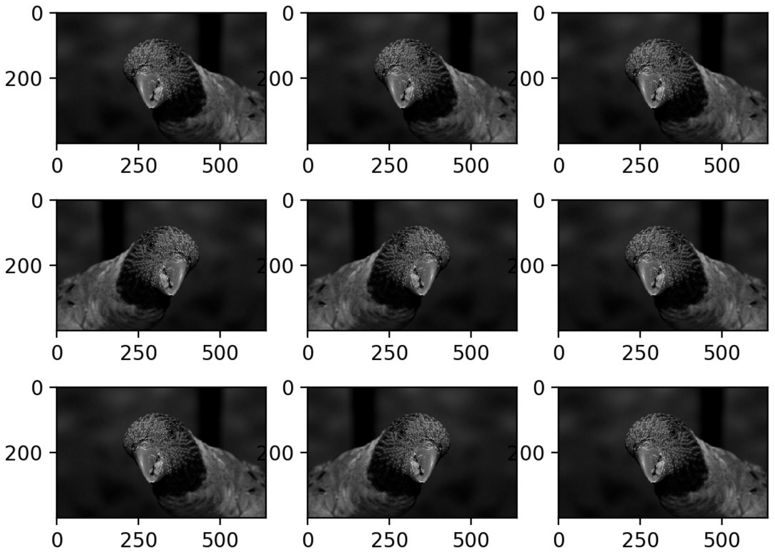


Figure 2.3: example of Data Augmentation "Horizontal flip"

Results

2.1.7 Under sampling

Breif it's a technique we can use when the data distribution is imbalanced, first we define what an "imbalance" is, data is called imbalanced if the output class values aren't evenly represented, for example let's say we study data with output feature having 2 possible values, either 0 or 1, the value 0 happens 80% of the time while the value 1 is represented by only 20% of the data, if we start training on this data right away we can expect the trained model to be biased towards the value 0 which appears to be the most common outcome, so even if he value 0 is not ha frequent, the model will assume so because that's what the data say.

the solution for this is to balance the data by removing some instances from the large class(s) until they are close enough to the smaller classes, this way we can decrease the bias of our model towards the large classes, this technique is called **under sampling**.

under sampling should solve the problem with bias in the trained model, but at he cost of decreasing the size of the data set when you drop large portion of it, hence decreasing what he model learns, so it can decrease the accuracy as well, if the biggest problem for your model is bias then under sampling can solve it, however if te biggest problem is in learning itself then under sampling shall make it worse.

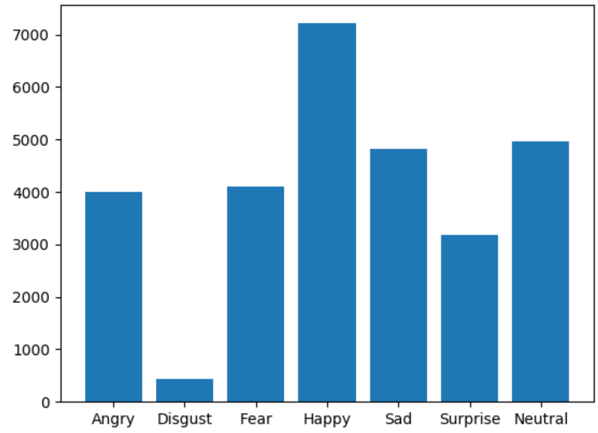


Figure 2.4: the unbalanced dataset, classes have large difference in their sizes

How we used it in our project one of the data sets we worked was imbalanced which made the trained model more biased towards negative emotions (which was more presented in the data set), so we tried to apply under sampling to solve this problem.

results unfortunately this made things even worse with our case since the difference between the size of smallest class and other classes was significant(see Figure 2.4) so it results in a large decrease in data set size so a large decrease in learning.

table 2.1 shows the how the prediction accuracy of model changes during training epochs, as the table shows, by more training the training accuracy increases, while testing accuracy actually decreases, so this technique couldn't help solve the overfitting problem we had with the model before it which stopped at about 50% testing accuracy before overfitting, it even caused a decrease in accuracy, so this technique wasn't a success in his case.

Table 2.1: this table holds the progress of learning for model after under sampling for data set

Number of Epochs	Training accuracy	Testing accuracy	Error
10	39.33 %	37.32 %	1.69
20	49.08 %	36.80 %	1.73
30	53.01 %	35.64 %	1.83

2.2 model

Chapter 3

CH3 - Proposed System

3.1 System Architecture

3.2 Preprocessing

3.3 Model

3.4 Library

3.5 Summary

Chapter 4

CH4 - Results

Chapter 5

CH5 - Conclusion

Chapter 6

References

1. Facial Expression Recognition using Convolutional Neural Networks: State of the Art

2. Y. Tang, "Deep Learning using Support Vector Machines," in International Conference on Machine Learning (ICML) Workshops, 2013

3. B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, "Hierarchical committee of deep convolutional neural networks for robust facial expression recognition," Journal on Multimodal User Interfaces, pp. 1–17, 2016

4. Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in ACM International Conference on Multimodal Interaction (MMI), 2015, pp. 435–442

5. A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going Deeper in Facial Expression Recognition using Deep Neural Networks," CoRR, vol. 1511, 2015.

6. Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, "Learning Social Relation Traits from Face Images," in Proc. IEEE Int. Conference on Computer Vision (ICCV), 2015, pp. 3631–3639

7. B.-K. Kim, S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee, "Fusing Aligned and Non-Aligned Face Information for Automatic Affect Recognition in the Wild: A Deep Learning Approach," in IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 48–57.