Postgraduate Certificate in Software Design with Artificial Intelligence

Advanced Machine Learning and Neural Networks - (AL_KSAIG_9_1)

Minor Exercise 2

Student ID: A00267948

Student Name: Daniel Foth

GIT: https://github.com/DanielsHappyWorks/aml-computer-vision

Brief Description:

Using both OpenCV cascade classifier and DLIB, create a classifier which will be able to determine whether a subject is happy or sad based on facial expression.

- Any classifier used so far can be used
- Dataset may need to be normalised and transformed

Contents

Introduction 3
Data 3
Models 3
NN Results 3
SVC Results 3
Linear SVC Results 3
Naïve Bays Results 3
Conclusion 4

Introduction

This project aims to create a classifier which will be able to determine whether a subject is happy or sad based on facial expression. The data will be collected manually from the internet and ran through multiple Machine Learning Algorithms to see if an accurate result can be achieved.

Data

The data set contains 25 images categorised into:

1. Happy
2. Sad
3. Neutral

The category for each image is specified before the image name. The format is <emotion>_<image name>.<image type>
The images that make up the data has been taken from https://unsplash.com/. These can be found using the image name.

Models

4 models were created to categorise the face expression into happy, sad or neutral.
All the models are fed in the x and y position for each of the 68 landmarks predicted by dlib.

NN Results

Accuracy: 0.625
Confusion Matrix:

	0	1	2
0	4	0	0
1	0	1	2
2	1	0	0

SVC Results

Accuracy: 0.375
Confusion Matrix:

	0	1	2
0	1	1	2
1	1	1	1
2	0	0	1

Linear SVC Results

Accuracy: 0.5
Confusion Matrix:

	0	1	2
0	3	1	0
1	0	1	2
2	1	0	0

Naïve Bays Results

Accuracy: 0.5
Confusion Matrix:

	0	1	2
0	3	1	0
1	2	1	0
2	1	0	0

Conclusion

None of the algorithms performed very well in categorising the images. There could be a multitude of issues that could cause this.
They include:

1. ***Not enough data*** – the dataset consist of only 25 images which isn't enough. This is very limiting and any variation in random number generation can flip the accuracy a lot. For example, if I changed the shuffle seed, I could have a neural net that was 70% accurate or 0% accurate.
2. ***Incorrect features*** – The features may not inform the algorithm very well. A lot of the values vary based on where the image is on the photo which could cause issues. In the next iteration I would consider using distances between points instead of the x and y values as that would probably influence the learning better.
3. ***Unoptimized algorithms*** – Its very hard to tweak algorithms that are inconsistent due to limited data. This could mean that the algorithms aren't as effective as they could be.
4. ***Predict likely hood*** – Instead of predicting the category, it could be worth while trying to predict the likelihood of a specific emotion. A lot of the images can be misleading even when categorising an image manually. These kinds of errors could influence the algorithms incorrectly therefore giving an inaccurate result.

Overall through changes in randomness the most stable algorithm has been Naïve Bays. It held up over 50% accuracy most of the time but there is no way to be sure that this is even accurate based on the issues listed above.