

Configuration Manual

MSc Research Project MSc in Data Analytics, MSCDAD_JAN21A_I

> Sachin Pralhad Langute Student ID: 19234635

School of Computing National College of Ireland

Supervisor:

Prof. Jorge Basilio



National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sachin Pralhad Langute
Student ID:	19234635
Programme:	MSC in Data Analytics MSCDAD_JAN21A_I Year:2021-22
Module:	Research Project
Supervisor:	Prof. Jorge Basilio
Due Date:	
Project Title:	Facial Emotion Recognition using Deep Convolutional Neural Network
Word Count:	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Date:31st January 2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple	
copies)	
Attach a Moodle submission receipt of the online project	
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both	
for your own reference and in case a project is lost or mislaid. It is not	
sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sachin Pralhad Langute Student ID: 19234635

1 Introduction

In this configuration manual, the author has given end-to-end details of all the process that was equipped in the study for the facial expression recognition system. Based on these aspects, the results were predicted. This manual also highlights the technical study which shows all the Python libraries which were used to study the working of the CNN model which was used for predictive analysis. The aim of this configuration manual is to make it easy for the reader to understand how things were processed from start to end.

2 Design Specification

It is a documented artifact explaining every single detail of the product, the use or the requirement of the product, the elements that make the product more important, and the technicalities associated with the product. This document must incorporate all the précised work equipped in this study including the challenges faced during the research.

2.1 Machine Hardware Configuration

Following hardware combinations were used to finish the project.

Windows edition -

Windows 10 Home Single Language

© Microsoft Corporation. All rights reserved.

System -

Processor:	Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz
Installed memory (RAM):	8.00 GB (7.69 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

2.2 Machine Software Configuration

The specifications of the software utilized for the system were as follows:

Software	Configurations
Operating System	Windows 10 (64-bit Operating system)
IDE	Spyder (Anaconda Navigator)
Scripting Language	Python
Scripting Language version	Python 3.7

2.2.1 Anaconda Installation

For the machine learning models, the most recent version of anaconda was installed. Its setup procedure involved the below steps: -

Go to the official anaconda website and select the link for windows installation.

← → C (e docs anacorda com/anacor	nde/instal/kindows/ 💷 🏠 🎯 🕨 🛩 縃 😘
	JMENTATION
 Home 	=
Anaconda Enterprise 5	Installing on Windows
Anaconda Enterprise 4	1 Develoard the Anarcoscia installer
 Anaconda Distribution 	
Installation	 RECOMMENDED: <u>Verify data integrify with SEG-256</u>. For more information on hashes, see <u>What about cryotographic hash verification?</u>
Installing on Windows	3. Double click the Installer to launch.
Installing on macOS	① Note
Installing on Linux	To prevent permission errors, do not launch the installer from the Eurorites folder.
Installing on Linux POWER	
Installing in silent mode	① Note
Installing for multiple users	If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users understall Anacords and to install it for your user only and to again.
Verifying your installation	insianeu iur an usets, uninsian vulacumua anu reinsian ii no yuur user uniy anu u y again.
Anaconda installer file hashes	4. Click Next.
Updating from older versions	5. Read the licensing terms and click "I Agree".
Uninstalling Anaconda	6. Select an install for "Just Me" unless you're installing for all users (which requires Windows Administrator privileges) and click Next.
User guide	7 Solvert a distribution folder to install Anacourta and click the Next button. See EAO
es e	A stelled a separation route to result Principlical and clark the reck upsion. See PELA

Download the Anaconda installer file and once downloaded, install the software on the machine.

Welcome to Anaconda3 2019.10 G4-bit) Setup. Lip will gude you through the installation of Anaconda 2019.10 (64-bit). Lip will gude you through the installation of Anaconda 2019.10 (64-bit). Lip ecomended that you doe all other applications before starting Setup. This will make it possible to update rejeant system files without having to reboot your computer. Decision 1000 Decision 2019.10 (64-bit) Setup Decision 2019.10 (64-bit). Decision 2019.10 (64-bit). Decision 2019.10 (64-bit). Decision 2019.10 (64-bit). Press Page Down to see the rest of the agreement. Anaconda End User License Agreement Copyright 2015, Anaconda, Inc. All rights reserved under the 3-clause BSD License: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anacon	O Anaconda3 2019.1	0 (64-bit) Setup		-		<	
Next> Cancel Anaconda3 2019.10 (64-bit) Setup — Anaconda3 2019.10 (64-bit) License Agreement Please review the license terms before installing Anaconda3 2019.10 (64-bit). Please review the license terms before installing Anaconda3 2019.10 (64-bit). Press Page Down to see the rest of the agreement.	ANACOND	A. Welcome (64-bit) S Setup will guid 2019, 10 (64- It is recommer before startin relevant syste computer. Click Next to c	e to Anaconda Setup le you through the instal it). nded that you dose all of g Setup. This will make it m files without having to continue.	3 2019 llation of Ar ther applica possible to o reboot yo	a. 10 naconda3 stions o update ur		
Anaconda3 2019.10 (64-bit) Setup			Nex	t> [Cancel		
Icense Agreement Please review the license terms before installing Anaconda3 2019.10 (64-bit). Press Page Down to see the rest of the agreement. Anaconda End User License Agreement Copyright 2015, Anaconda, Inc. All rights reserved under the 3-clause BSD License: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2019.10 (64-bit). Anaconda, Inc. Affection and User I source and binary forms, with or without modification, are permitted provided that the following conditions are met: V Anaconda, Inc. If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2019.10 (64-bit). Anaconda, Inc. Anaconda, Inc.	Anaconda3 2019.10 (64-	bit) Setup			-		×
Press Page Down to see the rest of the agreement. Anaconda End User License Agreement Copyright 2015, Anaconda, Inc. All rights reserved under the 3-clause BSD License: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 20 19. 10 (64-bit). Anaconda, Inc.		Please review th 2019.10 (64-bit)	ment ne license terms b).	efore in	stalling A	naconda3	I
Anaconda End User License Agreement Anaconda End User License Agreement Copyright 2015, Anaconda, Inc. All rights reserved under the 3-clause BSD License: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2019. 10 (64-bit). Anaconda, Inc. Anaconda, Inc.	Press Page Down to see th	e rest of the agree	ment.				
All rights reserved under the 3-clause BSD License: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 20 19. 10 (64-bit). Anaconda, Inc. < Back I Agree Cancel	Anaconda End User Licens	e Agreement					^
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 20 19. 10 (64-bit). Anaconda, Inc. Back I Agree Cancel 	All rights reserved under th	he 3-clause BSD Lic	ense:				
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2019.10 (64-bit). Anaconda, Inc. < Back I Agree Cancel	Redistribution and use in s permitted provided that th	ource and binary for e following condition	orms, with or with ons are met:	nout mod	dification,	are	~
Anaconda, Inc < Back I Agree Cancel	If you accept the terms of agreement to install Anaco	the agreement, clic nda3 2019.10 (64-	k I Agree to cont bit).	inue. Yo	ou must a	ccept the	
< Back I Agree Cancel	Anaconda, Inc. ————						
			< Back	I Ag	ree	Can	cel

Click on all the positive terms mentioned in the installation setup.

A HUGONDA	Select Install	ation Type			
ANACONDA	Please select 1 Anaconda3 20	the type of installat)19.10 (64-bit).	tion you would like	e to perforr	n fo
Install for:					
() Just Me (recommended	0				
O All Users (requires adm	in privileges)				
naconda, Inc. ————					

Anaconda3 2019.10 (64	-bit) Setup		-		\times
	Installing Please wait wi	hile Anaconda3 201	9.10 (64-bit) is b	eing installe	ed.
Show details					
Anaconda, Inc. ———					
		< Back	Next >	Cano	el

The setup starts installing the application.



Create a new environment and give it a name.

2.3 Libraries used for model implementation

```
import json
import zipfile
import os
import pandas as pd
from matplotlib import pyplot
from math import sqrt
import numpy as np
import scipy.misc
from IPython.display import display
from keras.utils import np_utils
from keras.preprocessing.image import ImageDataGenerator
from keras.utils.vis_utils import plot_model
from keras.models import Model, Sequential
from keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from tensorflow.keras.optimizers import Adam, SGD
from keras.regularizers import 11, 12
from matplotlib import pyplot as plt
from sklearn.metrics import confusion_matrix
import itertools
from pylab import rcParams
```

Model 3: VGG16

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import skimage.io
import keras.backend as K
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten, Dropout,BatchNormalization ,Activation
from tensorflow.keras.models import Model, Sequential
from keras.applications.nasnet import NASNetLarge
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlyStopping
from tensorflow.keras.optimizers import Adam
```

3 Dataset:

≡	kaggle	Q Search	A
+	Create	Dataset	Î
Ø	Home		
Φ	Competitions	Facial Expression Recognition Challenge Dataset	
	Datasets		- 1
<>	Code	Debanga Raj Neog • updated 2 years ago (Version 4)	
	Discussions	Data Code (25) Discussion Activity Metadata Download (778 MB) New Notebook :	
ଚ	Courses		
\sim	More	E Usability 5.3 Viags arts and entertainment	
Ê	Your Work	Description	
•	RECENTLY VIEWED	(c) https://www.kagqle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data	
(1)	Facial Emotion Dete View Active Events	 The data consists of 48×48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on 	

Data Explorer

777.8 MB

- icml_face_data.csv
 icml_face_data.csv
- test.csv
 - example_submission.csv
 - face_embs.npy
 - fer2013.tar
 - metadata_processed.csv

<	test.cs	v (1 files)
test	.csv	
601	3 MB	

Summary

- 6 files
- 8 columns

The data comprises grayscale pictures of faces 48x48 pixels in size. The faces have been already selected in such a way that highlights the center of the image with equal size for each image. There is a need for feature mapping for each image such that every image is categorized into one of the seven Ekman's emotions classes which include the facial expressions with a labeled numerical class (0=Angry, 1=Disgust, 2=Fear,3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

The dataset used for the FER system has two CSV files namely, train.csv and test.csv. The training dataset includes two-column data having the pixel value of the input image and the emotion associated with the pixel. There is also a column named "feeling" which depicts the mapped emotion class. The emotion column comprises the string value of the emotion which is enclosed in a quotation mark.

There is a single column in the testing dataset file with an emotion column enclosed with a string value for a particular emotion and the job is to predict the emotion depicted in that string value.

The training dataset includes 28,709 instances whereas, the leaderboard's public test set has 3,589 samples.

4 Data Preprocessing:

In Data preprocessing the author has converted the images from grayscale to RGB and also amplified the pixels of each image.

Preprocessing

```
In [24]: def gray_to_rgb(im):
                    converts images from single channel images to 3 channels
                    w, h = im.shape
                    ret = np.empty((w, h, 3), dtype=np.uint8)
ret[:, :, 2] = ret[:, :, 1] = ret[:, :, 0] = im
                    return ret
                 def convert_to_image(pixels, mode="save", t="gray"):
                    ....
                    convert the input pixels from the single string row to 48*48 array with real pixel values
                    when mode = "save" it keeps the images in flat array shape, otherwise it converts it to 48*48
when t (for type) = "gray, it keeps the pixels single channel, otherwise it converts it to 3 channels
                   if type(pixels) == str:
                    pixels = np.array([int(i) for i in pixels.split()])
if mode == "show":
                      if t == "gray":
                           return pixels.reshape(48,48)
                  return gray_to_rgb(pixels.reshape(48,48))
else:
                      else:
                           return pixels
                 data["pixels"] = data["pixels"].apply(lambda x : convert_to_image(x, mode="show", t="gray"))
               Gata[ pixels ] = Gata[ pixels ].apply(lamOda X : Convert_to_image(x, mode= snow , t= gray ))
from sklearn.model_selection import train_test_split
#split the data to train, test, and validation
X_train, X_test, y_train, y_test = train_test_split(data["pixels"], data["emotion"], test_size=0.2, random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=1)
                X_train = np.array(list(X_train[:]), dtype=np.float)
X_val = np.array(list(X_val[:]), dtype=np.float)
X_test = np.array(list(X_test[:]), dtype=np.float)
               y_train = np.array(list(y_train[:]), dtype=np.float)
y_val = np.array(list(y_val[:]), dtype=np.float)
y_test = np.array(list(y_test[:]), dtype=np.float)
               X_train = X_train.reshape(X_train.shape[0], 48, 48, 1)
X_val = X_val.reshape(X_val.shape[0], 48, 48, 1)
X_test = X_test.reshape(X_test.shape[0], 48, 48, 1)
In [25]: num_train = X_train.shape[0]
               num_val = X_val.shape[0]
num_test = X_test.shape[0]
```

5 Data Augmentation:

The first strategy is data augmentation, which was utilized in all of our models as a preprocessing step. The author used Keras' "ImageDataGenerator" for data augmentation. It generates 32 augmented images from a single image by rotating, flipping, and employing other predefined techniques.

Data Augmentation



6 Design specification:

The author has used 3 models namely:

- 1. Convolutional Neural Network with 5 convolutional layers
- 2. Convolutional Neural Network with 3 convolutional layers
- 3. VGG16 model

Out of all the three models, VGG performs the best with an accuracy of 85.75% and a loss of 1.73. The CNN model with 5 convolutional layers has the lowest accuracy of 25.65% and loss of 1.78 and on the other hand, the CNN model with 3 convolutional layers performs the second-best with an accuracy of 57.27% and loss of 1.89.

6.1 Convolutional Neural Network with 5 convolutional layers:

Model 1: 5-Layered CNN 1

```
conv5 model = Sequential()
conv5_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu',name = 'conv1'))
conv5_model.add(BatchNormalization())
conv5_model.add(MaxPooling2D(pool_size=(2, 2)))
conv5 model.add(Dropout(0.25))
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv2'))
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv4'))
conv5_model.add(BatchNormalization())
conv5_model.add(MaxPooling2D(pool_size=(2, 2)))
conv5_model.add(Dropout(0.25))
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv5'))
conv5 model.add(BatchNormalization()
conv5_model.add(Conv20(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv6'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv7'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv8'))
conv5_model.add(BatchNormalization())
conv5_model.add(MaxPooling2D(pool_size=(2, 2)))
conv5_model.add(Dropout(0.25))
conv5_model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv9'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(128, kernel_size(3, 3), padding="same", activation='relu', name = 'conv10'))
conv5_model.add(Conv2D(128, kernel_size(3, 3), padding="same", activation='relu', name = 'conv10'))
Conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(BatchNormalization())
conv5_model.add(Dropout(0.25))
conv5_model.add(Conv2D(256, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv13'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(256, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv14'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(256, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv16'))
conv5_model.add(BatchNormalization())
conv5_model.add(Conv2D(256, kernel_size=(3, 3), padding="same", activation='relu', name = 'conv17'))
conv5_model.add(BatchNormalization())
conv5_model.add(MaxPooling2D(pool_size=(2, 2)))
conv5_model.add(Dropout(0.25))
conv5_model.add(Flatten())
conv5_model.add(Dense(num_classes, activation='softmax'))
print(conv5_model.summary())
Model: "sequential"
```

Result Evaluation for CNN5:

```
loss = conv5_model.evaluate_generator(test_flow, steps=len(X_test) / batch_size)
print("Test Loss " + str(loss[0]))
print("Test Acc: " + str(loss[1]))
C:\Users\langu\anaconda3\lib\site-packages\keras\engine\training.py:2006: UserWarning:
and will be removed in a future version. Please use `Model.evaluate`, which supports g
warnings.warn('`Model.evaluate_generator` is deprecated and '
Test Loss 1.7888416051864624
Test Acc: 0.2565308213233948
```

6.2 Convolutional Neural Network with 3 convolutional layers:

Model 2: CNN with less convulational layers.

```
#Let's try smaller Conv model
model_conv = Sequential()
model_conv.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model_conv.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model_conv.add(MaxPooling2D(pool_size=(2, 2)))
model conv.add(Dropout(0.25))
model_conv.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model_conv.add(MaxPooling2D(pool_size=(2, 2)))
model_conv.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model_conv.add(MaxPooling2D(pool_size=(2, 2)))
model_conv.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model_conv.add(MaxPooling2D(pool_size=(2, 2)))
model_conv.add(Dropout(0.25))
model_conv.add(Flatten())
model_conv.add(Dense(1024, activation='relu'))
model_conv.add(Dropout(0.5))
model_conv.add(Dense(7, activation='softmax'))
```

Result:

```
loss = model_conv.evaluate_generator(test_flow, steps=len(X_test) / batch_size)
print("Test Loss " + str(loss[0]))
print("Test Acc: " + str(loss[1]))
```

Test Loss 1.189395785331726 Test Acc: 0.5727969408035278

6.3 VGG model:

Building Model

```
model=Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(4096,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1024,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7,activation='softmax'))
```

Model Summary
for layer in base_model.layers:
 layer.trainable=False
model.summary()

Model: "sequential_10"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 1, 1, 512)	14714688
flatten_2 (Flatten)	(None, 512)	0
dropout_8 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 4096)	2101248
dropout_9 (Dropout)	(None, 4096)	0
dense_12 (Dense)	(None, 1024)	4195328
dropout_10 (Dropout)	(None, 1024)	0
dense_13 (Dense)	(None, 7)	7175

Total params: 21,018,439 Trainable params: 6,303,751

Non-trainable params: 14,714,688

Result:

uc: 0.7324 - val_f1_score: 0.0343

nistory=model.fit(train_dataset,validation_data=valid_dataset,epochs = 5,verbose = 1,callbacks=[lrd,mcp,es])

Fooch 1/5 - auc: 0.6741 - f1_score: 0.0259 - val_loss: 1.7264 - val_accuracy: 0.8581 - val_precision: 0.6194 - val_recall: 0.0167 - val_a uc: 0.7160 - val_f1_score: 0.0323 Epoch 2/5 - auc: 0.6938 - f1_score: 0.0390 - val_loss: 1.7014 - val_accuracy: 0.8580 - val_precision: 0.6957 - val_recall: 0.0111 - val_a uc: 0.7267 - val_f1_score: 0.0219 Epoch 3/5 - auc: 0.7000 - f1_score: 0.0426 - val_loss: 1.6890 - val_accuracy: 0.8591 - val_precision: 0.6840 - val_recall: 0.0253 - val_a uc: 0.7316 - val_f1_score: 0.0480 Epoch 4/5 359/359 [=== uc: 0.7304 - val_f1_score: 0.0063 Epoch 5/5

References

Facial Expression Recognition Challenge Dataset | Kaggle

Mishra S., Prasada G.R.B., Kumar R.K., Sanyal G. (2017) Emotion Recognition Through Facial Gestures - A Deep Learning Approach. In: Ghosh A., Pal R., Prasath R. (eds) Mining Intelligence and Knowledge Exploration. MIKE 2017. Lecture Notes in Computer Science, vol 10682. Springer, Cham. <u>https://doi.org/10.1007/978-3-319-71928-3_2</u>

Mehta, D.; Siddiqui, M.F.H.; Javaid, A.Y. Facial Emotion Recognition: A Survey and Real-World User Experiences in Mixed Reality. Sensors 2018, 18, 416. <u>https://doi.org/10.3390/s18020416</u>

Castellano G., Kessous L., Caridakis G. (2008) Emotion Recognition through Multiple Modalities: Face, Body Gesture, Speech. In: Peter C., Beale R. (eds) Affect and Emotion in Human-Computer Interaction. Lecture Notes in Computer Science, vol 4868. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-85099-1_8