Introduction

Humans perform the task of detecting faces and interpreting facial expressions under a variety of settings on a daily basis, and we do so without effort. A person's face may reveal their identity, age, gender, and emotional condition. Face processing research is based on the idea that computers can extract information about a user's identity, status, and purpose from photographs and then react appropriately, i.e. by monitoring a person's facial expression. Nonverbal communication is done through facial expressions. They are the most common way for people to communicate social information. Face detection, facial feature extraction, and categorization into expressions are the three fundamental phases in automated facial expression analysis.

In this documentation we are going to mainly work on detecting person's smile and then deciding that, is it a positive smile or fake smile ? via using machine learning models and then finalize the best model for predicting the more accurate results.

Ideology

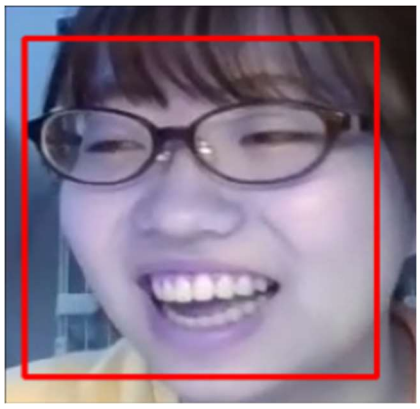
Our main intension is to extract out feature from the given image so we can detect the smile of person and further analyze it. So, we initially we will detect the face in the image and crop the desired part. From that part of image, we'll apply the wavelength transform so that we pixels can differentiate more conveniently. The transform images into vectors and feed the different classification models and train them. We will check the accuracy of the models and selected the best suitable model.

We are also going prepare another code where we will use convolution neural network for processing the image. In this segment we first classify the whether the is person is smiling or not, then we try to detect whether is it a fake smile or a genuine smile.

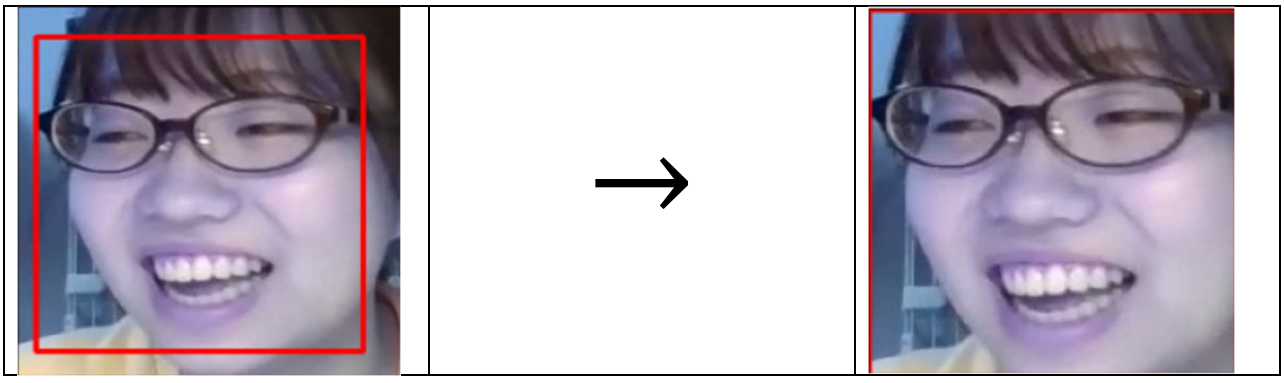
Methodology and Code flow

Firstly, what we will do is that visualize the any sample image for example we taken this image shown below

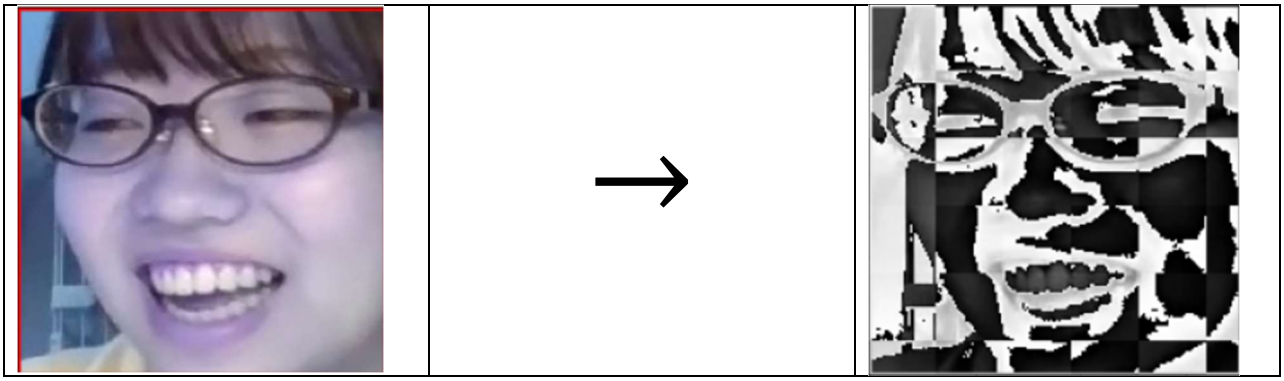
Now in this image we will detect the face using face cascade, if face exists and visible in the image then we draw rectangle over, otherwise we will discard the image. Like if we detect the face in above image then it will be like



As this rectangular part is the only desired part for us so well crop it and save the desired location



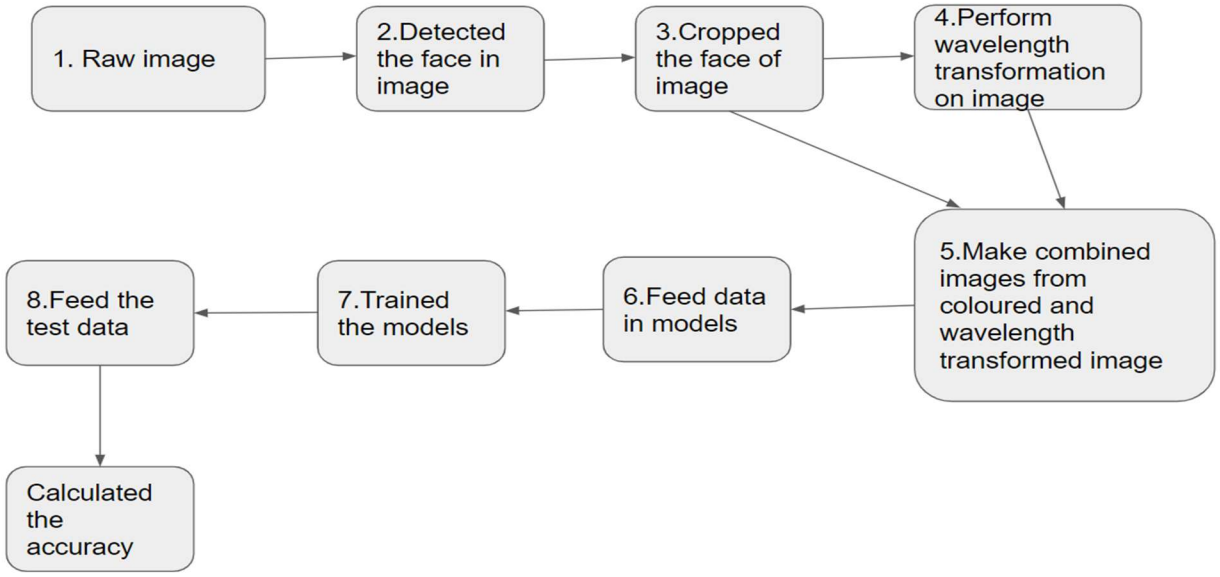
And then applied wavelength transformation on the image



After that we applied all the described this on the all the training images and produce ***X*** matrices such that every entry is containing the wavelength transformed image the and the coloured image, and ***tar*** has corresponding results of the that image.

After that we have feed that data into the different classification models with suitable parameters and then predicted the results.

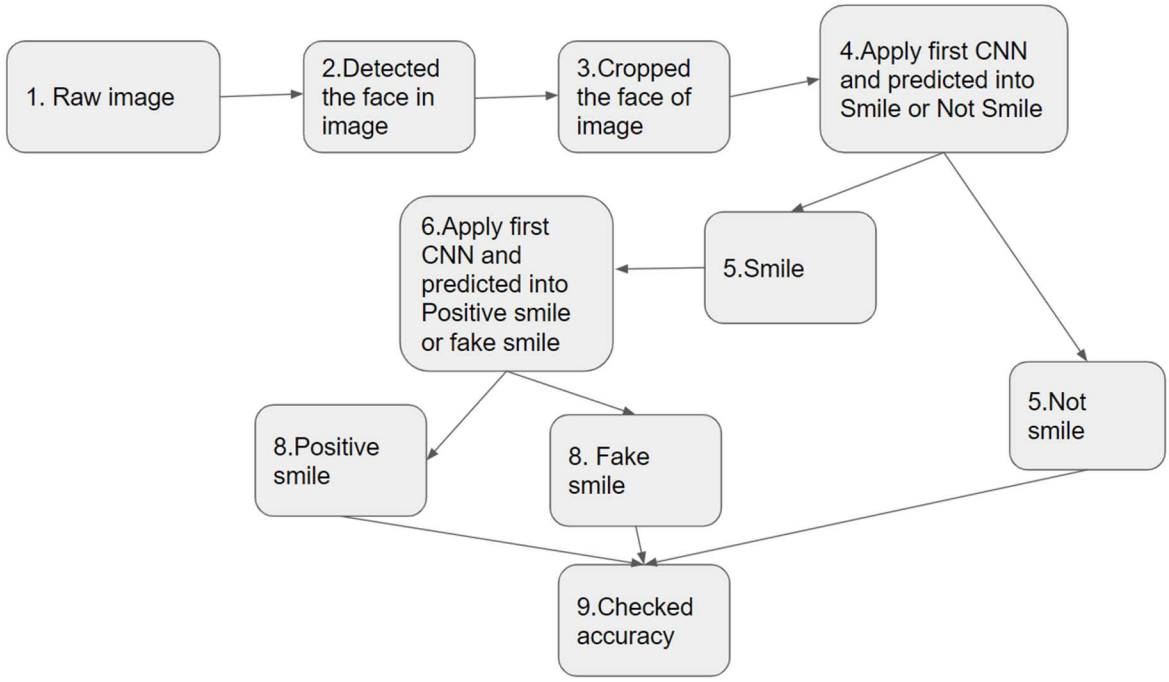
Flow chart of code



Convolution neural network methodology and code flow

In this also we initially detected the face and then cropped the faces from images. Here we use two CNN networks, in first network from the cropped training (given) set we have made two folder one has not smile images and other one smile images (both fake and real smile images). Here we use binary classification, and then feed the test data and classify into smile and not smile. In the second neural network, here also made folders of training data which contains fake and positive smile images and feed the output of first neural network(which are predicted as smile). Here also using binary classification classify into positive smile and fake smile. Finally, from the actual results given in the test data, check the accuracy to our predicted results.

Code flow chart



Results

Here to predicting the image emotion we are using 6 models

Model 1 - Support vector classifier with “rbf” kernel

Model 2 - Support vector classifier with “linear” kernel

Model 3 – Random Forest classifier

Model 4 – Logistic regression

Model 5 – Decision tree classifier

Model 6 – Convolution neural network

For model 1, Support vector classifier with "rbf" kernel we have

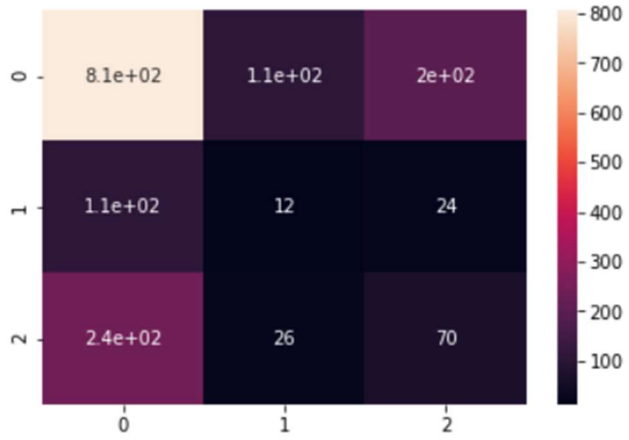
Classification report during validation

	precision	recall	f1-score	support
0	0.77	0.93	0.84	825
1	0.16	0.06	0.08	120
2	0.61	0.38	0.47	253
accuracy			0.72	1198
macro avg	0.51	0.46	0.46	1198
weighted avg	0.67	0.72	0.69	1198

Classification report of testing

	precision	recall	f1-score	support
0	0.70	0.72	0.71	1113
1	0.08	0.08	0.08	142
2	0.24	0.21	0.22	338
accuracy			0.56	1593
macro avg	0.34	0.34	0.34	1593
weighted avg	0.55	0.56	0.55	1593

Confusion matrix of test data



Test Accuracy (model 1) = 0.5574387947269304

For model 2, Support vector classifier with “linear” kernel we have

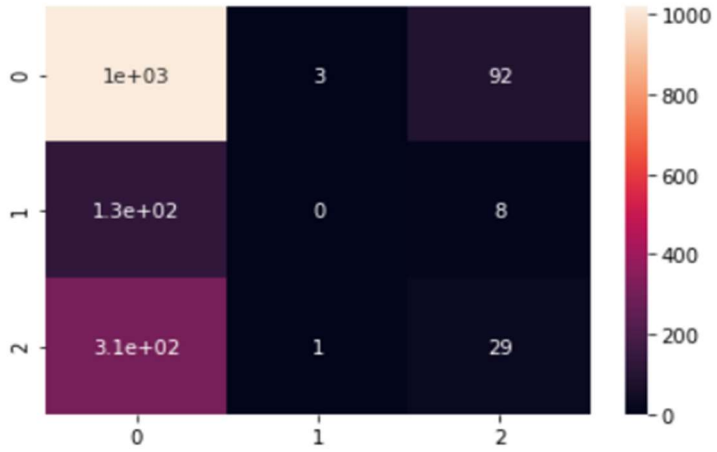
Classification report of validation

	precision	recall	f1-score	support
0	0.72	0.99	0.83	825
1	0.00	0.00	0.00	120
2	0.75	0.17	0.28	253
accuracy			0.72	1198
macro avg	0.49	0.39	0.37	1198
weighted avg	0.65	0.72	0.63	1198

Classification report of testing

	precision	recall	f1-score	support
0	0.70	0.91	0.79	1113
1	0.00	0.00	0.00	142
2	0.22	0.09	0.12	338
accuracy			0.66	1593
macro avg	0.31	0.33	0.31	1593
weighted avg	0.53	0.66	0.58	1593

Confusion matrix



Test Accuracy = 0.6572504708097928

For model 3, Random Forest classifier

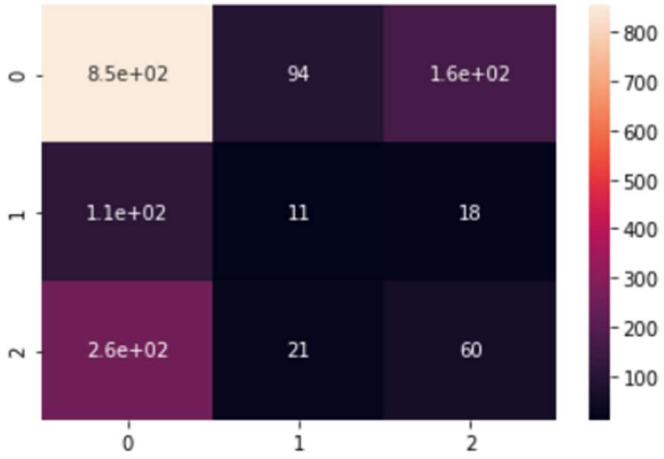
Classification report of validation

	precision	recall	f1-score	support
0	0.71	1.00	0.83	825
1	0.00	0.00	0.00	120
2	0.76	0.10	0.17	253
accuracy			0.71	1198
macro avg	0.49	0.37	0.33	1198
weighted avg	0.65	0.71	0.61	1198

Classification report of testing

	precision	recall	f1-score	support
0	0.70	0.77	0.73	1113
1	0.09	0.08	0.08	142
2	0.25	0.18	0.21	338
accuracy			0.58	1593
macro avg	0.34	0.34	0.34	1593
weighted avg	0.55	0.58	0.56	1593

Confusion matrix



test accuracy = 0.5806654111738857

Model 4, Logistic regression

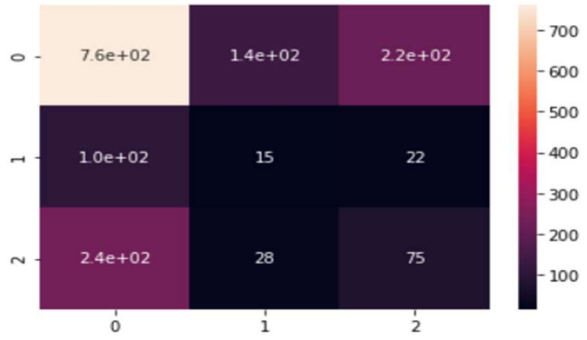
Classification report of validation

	precision	recall	f1-score	support
0	0.77	0.80	0.79	825
1	0.15	0.15	0.15	120
2	0.44	0.38	0.41	253
accuracy			0.65	1198
macro avg	0.45	0.45	0.45	1198
weighted avg	0.64	0.65	0.64	1198

Classification report of testing

	precision	recall	f1-score	support
0	0.69	0.68	0.69	1113
1	0.08	0.11	0.09	142
2	0.24	0.22	0.23	338
accuracy			0.53	1593
macro avg	0.34	0.34	0.34	1593
weighted avg	0.54	0.53	0.54	1593

Confusion matrix



Test accuracy = 0.5342121782799749

Model 5, Decision tree classifier

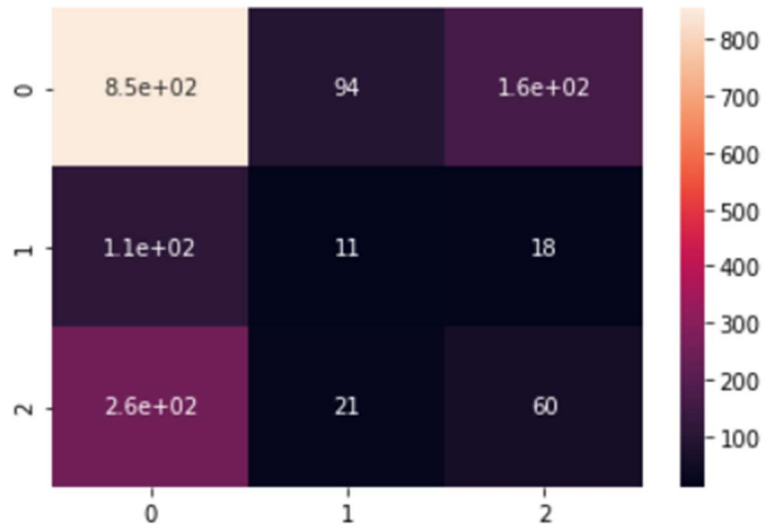
Classification report of validation

	precision	recall	f1-score	support
0	0.72	0.68	0.70	825
1	0.13	0.15	0.14	120
2	0.30	0.33	0.31	253
accuracy			0.55	1198
macro avg	0.38	0.39	0.38	1198
weighted avg	0.57	0.55	0.56	1198

Classification report of testing

	precision	recall	f1-score	support
0	0.70	0.66	0.68	1113
1	0.09	0.11	0.10	142
2	0.23	0.24	0.23	338
accuracy			0.52	1593
macro avg	0.34	0.34	0.34	1593
weighted avg	0.54	0.52	0.53	1593

Confusion matrix



Test accuracy = 0.5241682360326428

Model 6, CNN

First CNN (for detecting smile and not smile)

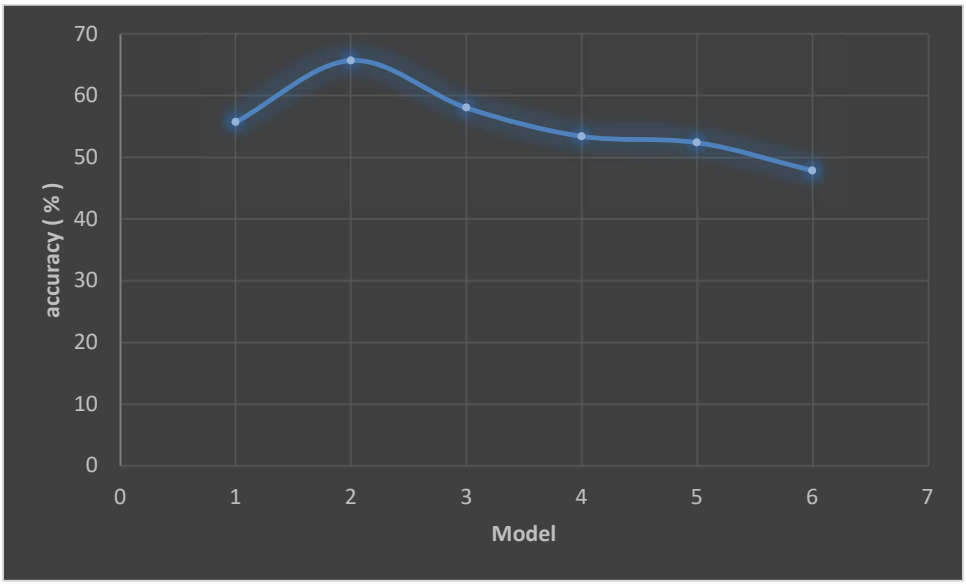
loss: 0.5530
accuracy: 0.7966
validation loss: 0.5277
validation accuracy: 0.7675

Second CNN (for detecting positive smile and fake smile)

loss: 0.6075
accuracy: 0.7108
validation loss: 0.6319
validation accuracy: 0.6159

overall accuracy = 0.4788265077998362

Accuracy analysis



Model 2, **Support vector classifier with "linear" kernel** gives highest accuracy of 65.72 %.

Conclusion

Here we used various classification models and CNN network to classify the images in different categories. But the accuracy that we have got is not good enough. To furthermore increase the accuracy the models like we can also detect the eyes also in the images as eyes are important feature to predicting the expression of face.