MASTER'S PROJECT: MICRO EXPRESSION CLASSIFICATION

Except where reference is made to the work of others, the work described in this project is my own or was done in collaboration with my advisory committee. Further, the content of this project is truthful in regard to my own work and the portrayal of others' work. This project does not include proprietary or classified information.

_____

Pururav Chakravarthy Devanaboyina

Certificate of Approval:

_____

Dr. Michael J. Reale,
Assistant Professor
Department of Computer Science

_____

Dr. Scott Spetka,
Professor
Department of Computer Science

_____

Dr. Jorge Novillo,
Professor
Department of Computer Science

MASTER'S PROJECT: MICRO EXPRESSION CLASSIFICATION

Pururav Chakravarthy Devanaboyina

A Project

Submitted to the
Graduate Faculty of the
State University of New York Polytechnic Institute
in Fulfillment of the
Requirements for the
Degree of

Master of Science

Utica, New York
May 2021

MASTER'S PROJECT: MICRO EXPRESSION CLASSIFICATION

Pururav Chakravarthy Devanaboyina

_____

Signature of Author

_____

Date of Graduation

Micro-Expression Classification

Pururav Chakravarthy Devanaboyina

***Abstract*—**
As micro expressions occur in a fraction of a second in the face it is very hard for the human eye to catch them. Micro expressions tell the person's true emotional state. They are generally given by the person when the person is trying to hide his true emotions unconsciously and they usually cannot be faked by the person. The detection of these spontaneous expressions which happens in a matter of split seconds is very hard for the machine or humans. Micro expression detection can have various applications in the field of national security, psychology, neuroscience, surveillance, interrogation, and other related fields. The aim of this project is to train a machine learning model which uses Deep learning and Transfer learning to identify micro expressions in videos. We tried many different approaches which include using dense optical flow of the images and a CNN-LSTM model. We performed experiments on the CASME II as the dataset for classifying the micro expressions.

Keywords—Micro expressions; Deep Learning, Transfer Learning.

I. INTRODUCTION

Micro expressions are very spontaneous and happen in very brief seconds (0.5-1 seconds). Micro expressions give the true intent or emotion of the person. Because the micro expressions are given subconsciously and occur without the knowledge of the person they cannot be faked. They can be used to study the emotional state of the person. For example, if a person is asked to watch the movie and give a rating for it, the micro expressions on the person's face can be studied and this can tell the actual reaction of the viewer. In the same way the micro expressions can be used in the process of interrogation where people usually tend to hide their emotions. Also, in the field of psychology the micro expressions provide us with the data about how the person is reacting to the treatment provided. People giving testimonies and interview are found to be giving micro expressions which tell their true intent which is usually the opposite of what they are trying to portray. Teachers can also use these micro expression detection systems to read the students emotion in online classes and adjust the course or teaching methods to create a much suitable environment for learning.

Micro expressions where first discovered in 1966 by scientists Isaacs and Haggard [10] while they were scanning films/movies of psychotherapy for the non-verbal communication indications between the therapist and the patients. These were later researched heavily by Dr. Paul Ekman [9] who also popularized the word MICROEXPRESSION. It is believed that the micro expressions are generated by the Amygdala which is also known as the emotion center of the brain. The amygdala controls the frontal cortex for brief seconds and display an expression in response to stimuli. These expressions are called micro expressions.

Since the detection of these micro expressions by a human eye is very challenging and sometimes impossible, the need for developing an automatic micro-expression detection is very high.

.

II. RELATED WORKS

The automatic micro expression detection system is a challenging field of study which has attracted many researchers and scientists. In the recent years deep learning has be proven method to detect facial expressions and micro expressions. This section discusses and reviews some state-of-the-art techniques used in detecting micro expressions.

A) Hand-Designed Methods:
These methods were designed over 10 years ago. Wu et al. [2] used Gabor filters to extract the features and used a Support Vector Machine on the extracted features to identify micro-expressions. Polikovsky et al. [3] used sub regions of the face and found the correlation of each region using 3D Gradient histograms. These histograms would then tell the corresponding micro expression.

B) Deep Learning methods:
These methods were designed more recently and are proven to be better methods at detecting micro expressions. Park et al. [4] used Active Appearance Model which is a geometric based model to extract 70 facial landmarks from the frame sequence. By using the landmarks motion vectors were estimated to give different micro expressions in the videos. The most

recent trend to detect the micro expressions is the Convolutional Neural Network (CNN). Takalkar et al. [5] used different data augmented techniques to develop synthetic images of the frames. These images are then fed to deep convolutional neural networks to identify micro expressions. Mayya et al [6] used the original video sequences to generate temporal interpolation. The estimated output was then fed to a deep CNN for the micro expression detection. Wang et al. [7] have used transfer learning and a residual network on the micro attention units or action units which to identify micro expressions. Our approach was inspired by [7] where we used transfer learning on the images.

Although some of these techniques have some major drawbacks like losing important interpolation data on the faces and not learning spatial-temporal relation etc., these methods were significant in identifying the micro expressions. As the micro expression videos are too short and the frame are similar to each other i.e., the temporal redundancy is high, it is difficult to work on the datasets.

III. DATASET CHARACTERISTICS

Data Understanding:

The dataset used for the micro expression detection is the CASME II dataset. The data set was obtained from the official website [1] Institute of Psychology, China by signing a license agreement. The dataset consists of video sequences and cropped image sequences from the videos. The dataset was improved upon the previous CASME dataset with recording in 200fps and higher spatial resolution (280 X 340 pixels) around the face area. For this project, the cropped images where used. Seven different expressions were given in the dataset which include Surprise, Happiness, Disgust, Fear, Sadness, Repression and Others.

Neutral frames were placed before and after the frames which have the micro expression to ease the detection by different algorithms. The labelling of the micro expressions was based on the Facial Action Coding System investigator guide. The video sequences were recorded under proper illumination and with reduced highlighted features on the subject's face. The dataset is unequally divided because some of the facial expressions were difficult to stimulate in lab environments. for e.g., the micro expression sadness had only 7 samples, but the happiness has 32 samples and disgust had 60 samples. The baseline performance on the CASME II for the 5-class classification was 63.41%. [1]

Twenty-six subjects were chosen for the recording of this dataset. Many different motion picture films were played to them and their expressions were recorded. Each sequence is marked with the 3 labels which are onset, apex, offset frames. The onset and offset frames are the starting and ending frames of the sequence and the apex frame is where the micro expression is at a peak.

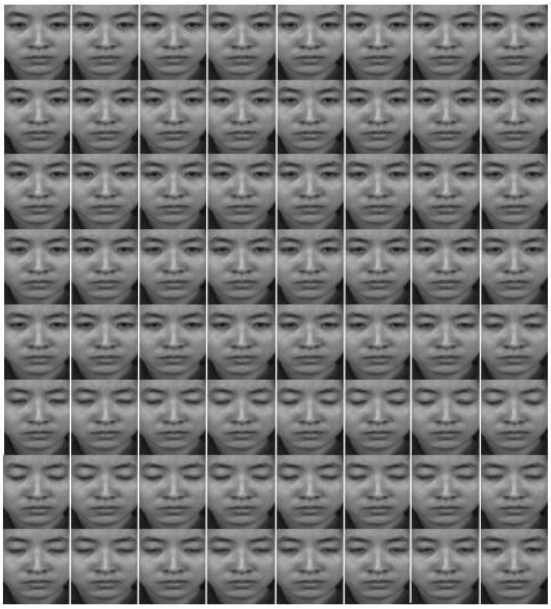

Figure 1: Sample sequence from CASME II [11]

Fig 1 is a sample sequence of micro expression which consists of 64 images.

Expression	Samples
Happiness	32
Sadness	4
Fear	2
Disgust	63
Surprise	28
Repression	27
Others	99

Table 1: No of samples for each Expression in CASME II

IV. METHODS

As this dataset deals with long image sequences with irregular frame numbers and many neutral frames placed before and after the actual frames which

consists of the micro expression, many different methods were employed to get the anticipated result.

The image sequences of each expressions were fed into a Convolutional Neural Network and then fed to a Long Short-Term Memory layer to train the model.

The CNN is used to extract the spatial features of the image sequences and these features were fed as an input to the stand LSTM layer with the tanh as an activation function. Later the architecture of the CNN-LSTM architecture was improved by changing the parameters of the Convolution layers and the dense layers.

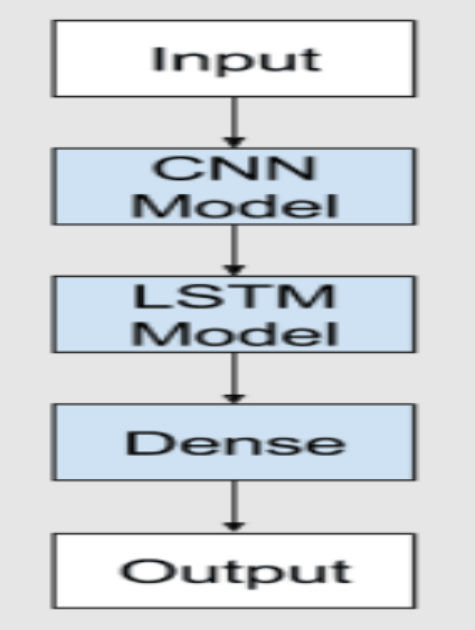

Figure 2: Overview of CNN-LSTM Model

The CNN layers learn the spatio-temporal features of the image sequence and then after the feature extraction these are sent as an input to the LSTM layer (Fig 2) to identify the micro-expressions with the help of ground truth data.

Because of the unequal number of image sequences for each expression, we had to make some changes like dropping some expressions labelled 'Others', 'Disgust', 'Fear', 'Sadness'. In some approaches only the expressions labelled 'Others' and 'Fear' were removed. Also, all the approaches except method 1 were repeated with 7, 5 and 3 micro expressions.

Method 1: Padding each sequence with the highest number of frames

In this approach, the highest number of image frames were taken as a limit for padding each frame so that we get equal frames for all the image sequence. The highest number of frames was found to be 141. All image sequences were padding with the last frame of the sequence so that all the sequences have the frame number of 141. 20 subjects were chosen for the training dataset and 6 subjects for the testing dataset. Two TensorFlow datasets was built using the image sequences and the one hot encoded labels. The image sequences were passed in a TensorFlow input pipeline to build the training and testing datasets.

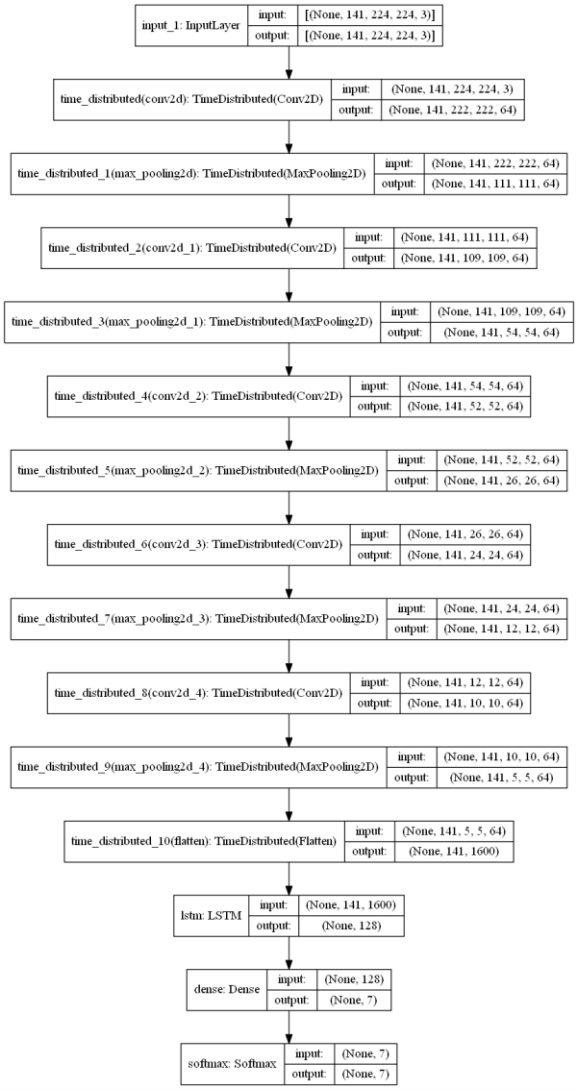

Figure 3: CNN-LSTM architecture

The above model did not achieve the expected output as the accuracy and loss kept on repeating after a few epochs. Also, this model did not perform well on the testing dataset as it only got an accuracy of 20%. This model(Fig 3) had an input shape (141,224,224,3) and the dense layer with 7 as an output.

As the model failed to get the expected results, it was then rerun using an improved version of the model (fig 4) using more dense layers and dropout layers.

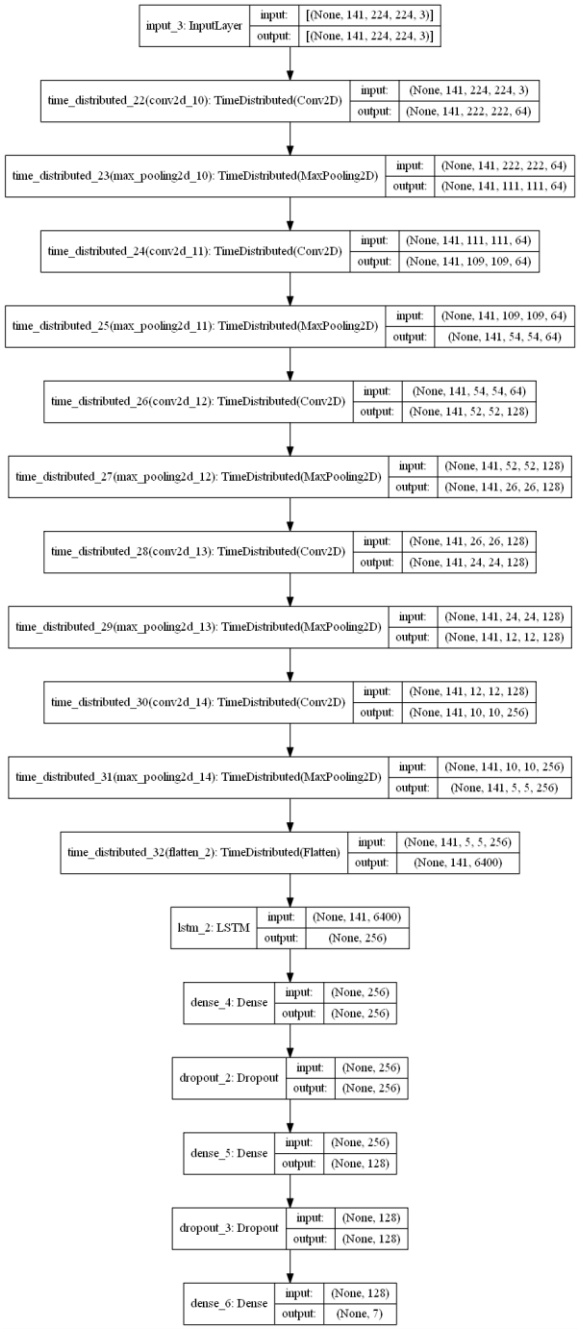

Figure 4: New CNN-LSTM model architecture

The new model also failed to achieve the result as it repeated the loss and accuracy after some epochs. So, the method of the padding to get and equal frames was dropped.

Method 2: Taking 6 frames around the apex frames

As the apex frame was where the micro-expression peaked in the whole sequence, 6 frames were chosen around the apex frames as an input. This made sure the repeating frames in the sequence were greatly reduced. So, this time the input shape was (6,224,224,7). After obtaining these frames the sequences were used to build TensorFlow input pipelines. The new training dataset was fed to the old CNN-LSTM (Fig 3) architecture. The model's input shape was changed so that it accepted the new input shape of (6,224,224,7)

This also did not yield results as expected as the accuracy kept on repeating after few epochs. So, the training dataset was rerun on the new CNN-LSTM (Fig 4). The models input shaped was tweaked to accept input batch of 6 instead of 141, but the results were the same as before.

When this method was repeated after dropping unequal labelled expressions with input shapes (6,224,224,5) and (6,224,224,3) the results were the same. The training accuracy kept on repeating and did not perform well on the testing dataset.

Method 3: No Padding frames

In this method, the frames were not padding with and any frames and were given as an input to the old CNN-LSTM (Fig 3) by making some changed in building the training and testing TensorFlow datasets. The training and testing image sequences were passed as a ragged tensor to the TensorFlow input data pipelines.

This ensured that the inputs of any number of frames could be passed as an input to the CNN-LSTM model.

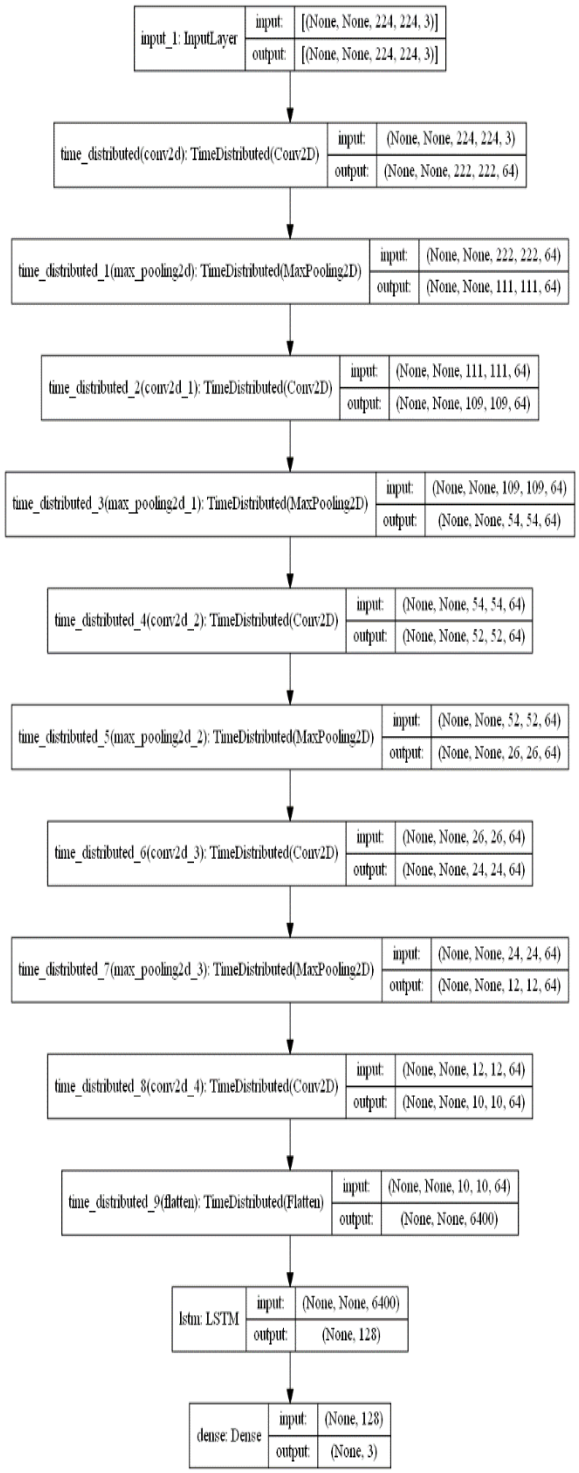

Figure 5: Old CNN-LSTM model with None Input shape

Although the model was changed to accept inputs of different batches, the model remained stuck and kept predicting the same class for all the inputs.

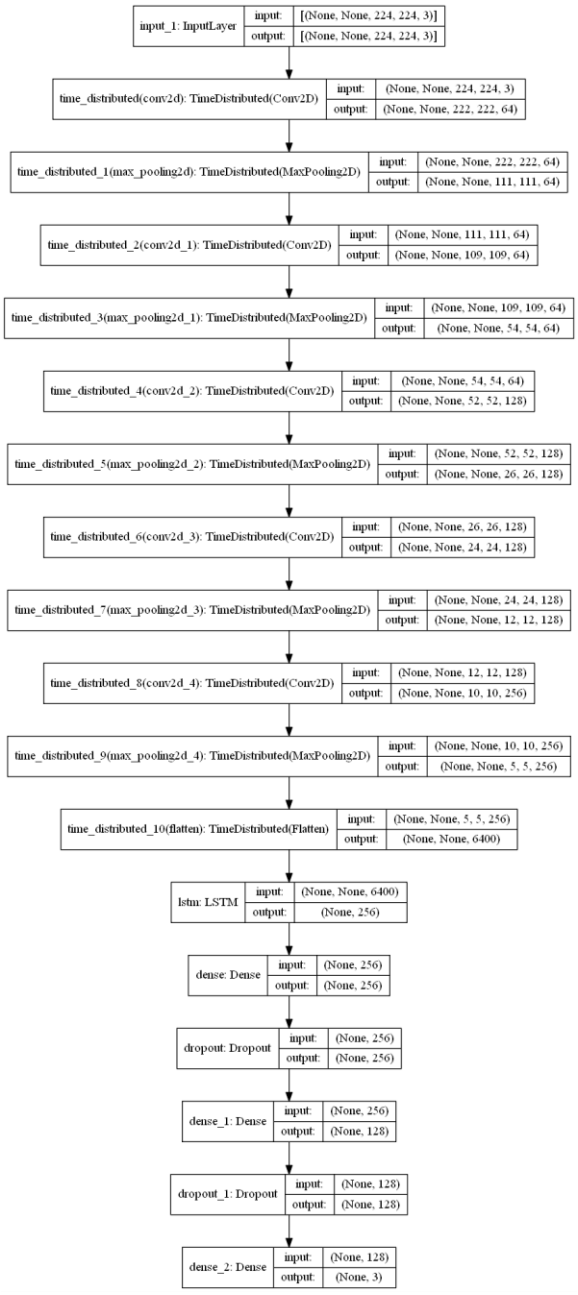

Figure 6: New CNN-LSTM with no input batches

So, the inputs were rerun in the new CNN-LSTM model by changing it to accept any input batches. This

also did not perform well as the accuracy was stuck after few epochs.

Method 4: Dense Optical Flow of image sequence custom CNN model

For the image sequences of the expressions Fear, Sadness, Disgust and Others were removed from the dataset as they had way less sequences than compared to the Happiness, Surprise, Repression emotions. After reducing the dataset, the Dense Optical Flow of the sequences were used as the input.

Optical Flow is a 2D vector pattern which shows the displacement of the moving points from the first frame to the second frame.

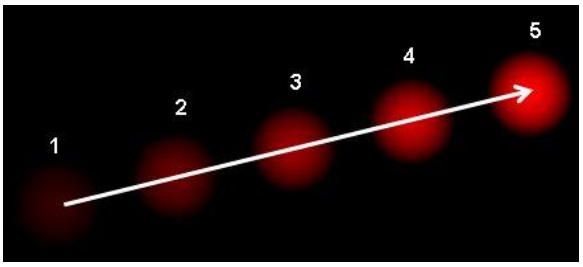

Figure 7: Optical Flow image [8]

The above image shows a ball moving in 5 frames and the vector shows the displacement of the ball.

Dense optical flow gives the flow vectors of the entire frame relative to the first frame, which covers all the pixels in the frame. So, Dense optical flow images of the sequence were generated using the OpenCV library.

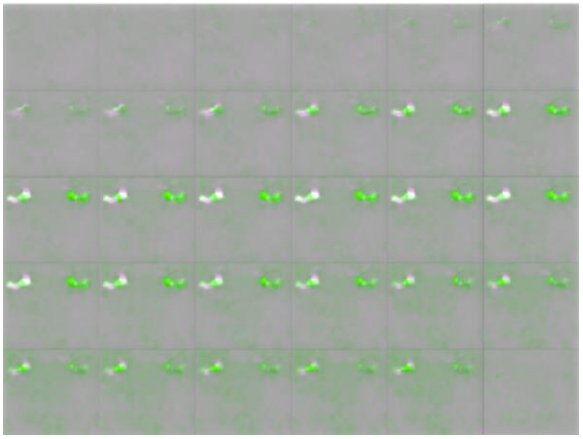

Figure 8: Dense Optical Flow image sequence

Figure 8 shows the dense optical flow images of the original image sequence of an expression. We can see clearly that there is some movement in the sequence. This movement is where the micro expression occurred. These images were then used to build a TensorFlow input pipeline like in the method 2. As, there was no padding done the frame number from each sequence were different.

After obtaining the dense optical flow images and building the input pipeline, the same CNN-LSTM models from Fig 5 and Fig 6 were used. Again, the results were not as expected.

Method 5: Dense Optical Flow of Apex Image using Transfer Learning

In this method the dense optical flow of the apex frame was computed using the first frame and the apex frame. This process was repeated to all the image sequences. First all the 7 expressions were used for this method. The second time the 'others' and 'disgust' were removed. For the third time the expressions 'Others', 'Disgust', 'Fear' and 'Sadness' were excluding.

Data augmentation techniques like random vertical and horizontal flip and random crop was applied to reduce the overfitting while training the model and increase the robustness of the model.

The Optical flow images were only generated using two dimensions in the x and y planes. So, the generated images only had the 2 channels. To overcome, this problem a third channel was added to the image which consisted of the Euclidian norm of the elements along the two dimensions concatenated with the dense optical flow image. This added the third channel to the images instead of repeated any one of the channels.

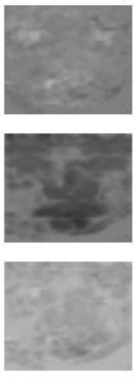

Figure 9: Sample of 3 Channels of Apex frame after applying the Dense Optical Flow

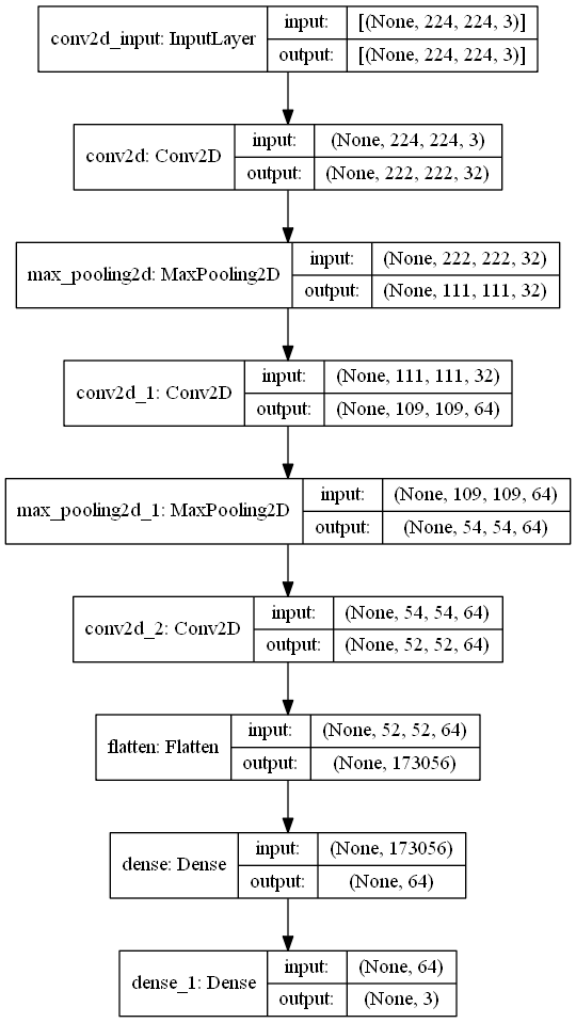

Figure 10: Custom CNN model

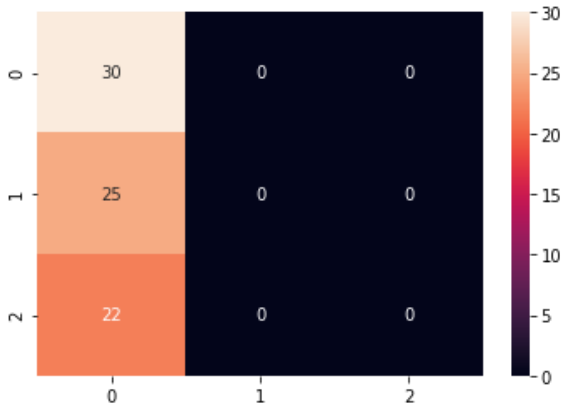

Figure 11:Confusion matrix on the training dataset

The confusion matric in Fig 10 shows that the CNN model only predicts a single class out of the three classes. As a result, the accuracy was very low at 38%.

Method 6: Transfer Learning on Resnet50

In this method Transfer Learning was used on the dense optical flow image of the apex frames.

Transfer Learning is a method in Machine Learning utilizing the knowledge gained by solving one problem and using it to solve another problem. For example, classification tasks on custom datasets can be solved by utilizing the knowledge gained by using pre-trained models on the well-known ImageNet dataset.

Model: "model"

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
max_pooling2d (MaxPooling2D)	(None, 3, 3, 2048)	0
global_average_pooling2d (Gl	(None, 2048)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 128)	262272
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 3)	195

Figure 12: Transfer Learning Model using Resnet50

The apex frames of the dense optical flow images were used. Data augmentation techniques of vertical and horizontal flip were used. After this step the images were passed into the ResNet50 [12] pre-trained model. The ImageNet dataset was used as the weights for the ResNet50 model. After the ResNet50 model trained on the apex frames by using the knowledge of the ImageNet dataset, the output was passed through some Pooling Layers and Dense layers with Dropouts. The final output was passed through a Dense layer with a SoftMax activation.

When the transfer learning model was used for classifying the 3 labels ‘happiness’, ‘surprise’ and

'repression' we got a testing accuracy of 60% but this was not consistent as the accuracy differed with each run.

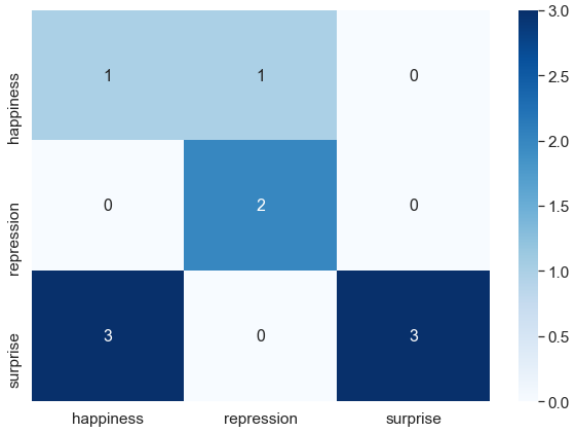

Figure 13:Confusion matrix of Resnet50 model on the testing dataset with 3 classes.

After repeating the transfer learning model with the expressions labelled 'happiness', 'surprise', 'repression', 'sadness', 'disgust' we got a testing accuracy of 51%.

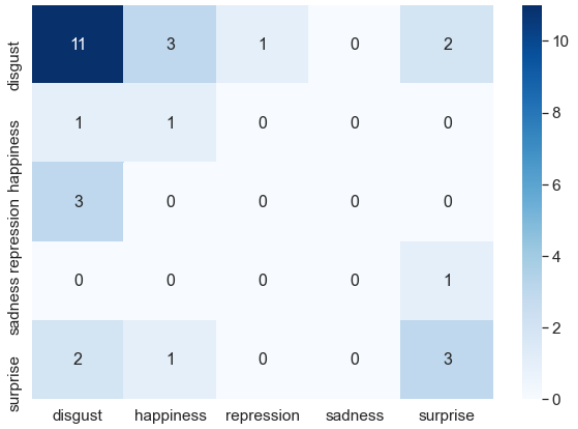

Figure 14:Confusion matrix of Resnet50 model on testing dataset with 5 classes

Method 7: Transfer Learning on Resnet50 using Leave-one-subject-out

In this method we obtained all the apex frames returned after applying the dense optical flow by the reference of the first frame. After obtaining the frames of all the subjects we ran each subject in the transfer learning model (Fig 12) where one subject was used for testing and the rest for training. Each subject was run separately for the training and testing with an epoch value of 100. This was done so that the GPU memory does not retain the previous values trained on. The training size increased as we trained each subject separately. Using this method ensured the GPU memory was reallocated each time a different subject was passed for training and testing. All the predictions obtained after running all the training and testing sets the results obtained were averaged and compared to the true ground values for the accuracy matrix.

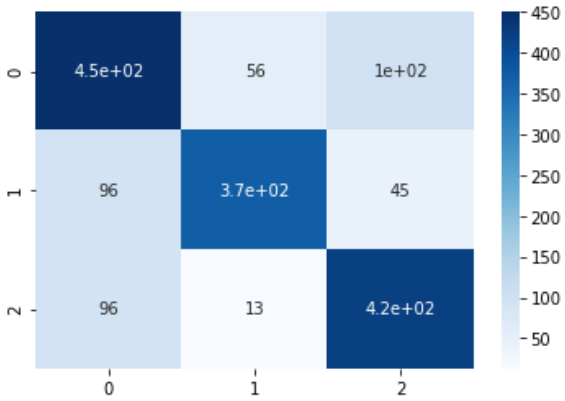

Figure 15:Confusion matrix on training 26 subjects.

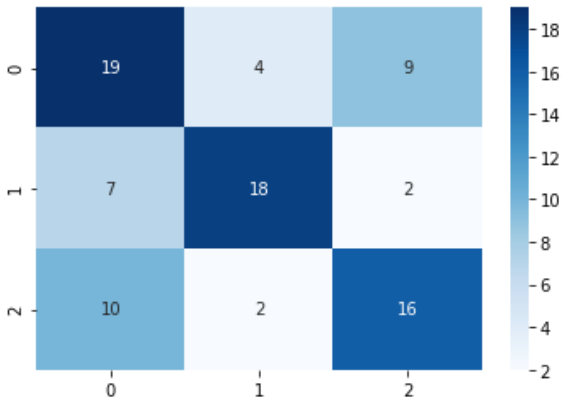

Figure 16: Confusion matrix of testing 26 subjects.

The training accuracy obtained by this method was 75% and the testing accuracy was 60.9%.

V. Results and Comparison

As different methods were employed in training on the dataset, we will discuss the results from the CNN and the Resnet50 model. The CNN model failed to predict properly as it was only predicting a single class or in other words it was choosing the class which had more training images and

predicting the same class for every image. So, a more powerful model was needed which already learned on a huge dataset like the ImageNet. Resnet50 was chosen for this. The Resnet50 was trained on the original ImageNet dataset and the knowledge was used to train again on the dense optical flow images of CASME II apex frames.

Figure 17 Categorical accuracy of the transfer learning model on training dataset for 3 classes

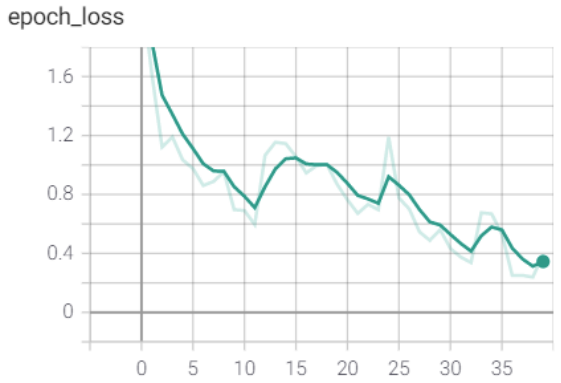

Figure 18: Epoch loss of the transfer learning model on training dataset for 3 class

By looking at the images in Fig 17 and Fig 18, it can be concluded that the transfer learning model had kept the training loss at minimum while steadily increasing the training accuracy with each epoch but also the testing loss was starting to increase (Fig 20).

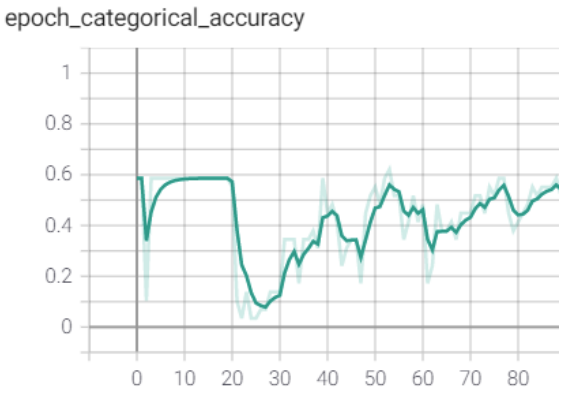

Figure 19: accuracy on testing dataset for 3 classes

Figure 20:Epoch loss on the testing dataset

The benchmark accuracy [1] on the CASME II was 63.41% on 5 classes. The accuracy of the transfer learning model with dense optical flow of the apex frames was not consistent when using the original train/test spilt as the samples in testing were low. So, we used the leave one subject out method in the last experiment. This method got us a testing accuracy of 60.9%. By looking at the confusion matrix in fig 15 and fig 16, we can say the model did a reasonable job at predicting the training and testing dataset than other methods employed before.

VI. Conclusion and Future Work

We tried different approaches on classifying the CASME II and the best result was found when we chose 3 classes and used the pre-trained ResNet50 Model for transfer learning with the multiprocessing module of python. The project can be improved on the other classes by performing some data augmentations on the expressions which were left out. As these expressions (disgust, fear, sadness) had a smaller number of image sequences because they were difficult to reproduce under laboratory conditions. By

using the existing video sequences new data can be generated by flipping the whole image sequence. Also, the transfer learning method could be used on the original sequences by taking a time series input in the transfer learning model. We would also like to increaser our dataset size by adding another benchmark micro expression dataset SMIC [13].

VII. References

[1] Yan, W.J., Li, X., Wang, S.J., Zhao, G., Liu, Y.J., Chen, Y.H. and Fu, X., 2014. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. PloS one, 9(1), p.e86041.

[2] Q. Wu, X. Shen, and X. Fu, "The machine knows what you are hiding: an automatic micro-expression recognition system," in Affective Computing and Intelligent Interaction. Springer, 2011, pp. 152–162.

[3] S. Polikovsky, Y. Kameda, and Y. Ohta, "Facial micro-expressions recognition using high speed camera and 3d-gradient descriptor," 2009.

[4] Salvatore J. Stolfo, Wei Fan, Wenke Lee and Andreas L. Prodromidis; "Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project"; 0-7695-0490-6/99, 1999 IEEE

[5] M. Takalkar and M. Xu, "Image based facial micro-expression recognition using deep learning on small datasets," International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017.

[6] M. Veena, M. P. Radhika, and M. P. M. M., "Combining temporal interpolation and dcnn for faster recognition of micro-expressions in video sequences," International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016.

[7] Su-Jing, L. Bing-Jun, L. Yong-Jin, Y. Wen-Jing, O. Xinyu, H. Xiaohua, X. Feng, and F. Xiaolan, "Micro-expression recognition with small sample size by transferring long-term convolutional neural network," Neurocomputing, 2018.

[8] The images of optical Flow, https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html

[9] Ekman, P. (2002). Micro expression Training Tool (METT). San Francisco, CA:University California.

[10] Haggard, E., and Isaacs, K. (1966). "Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy," in Methods of Research in Psychotherapy, eds L. A. Gottschalk and A. H. Auerbach (New York, NY:Appleton-Century-Crofts), 154–165.

[11] Zhi, Ruicong, et al. "Combining 3D convolutional neural networks with transfer learning by supervised pre-training for facial micro-expression recognition." IEICE Transactions on Information and Systems 102.5 (2019): 1054-1064.

[12] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

[13] X. Li, T. Pfister, X. Huang, G. Zhao and M. Pietikäinen, "A Spontaneous Micro-expression Database: Inducement, collection and baseline," *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, pp. 1-6, doi: 10.1109/FG.2013.6553717.